2009 10th International Conference on Document Analysis and Recognition

Text Detection and Localization in Complex Scene Images using Constrained AdaBoost Algorithm

Shehzad Muhammad Hanif, Lionel Prevost
Institut des Systèmes Intelligents et de Robotique, CNRS, UMR 7222
UPMC Université Paris 06, F-75005, Paris, France
shehzad.muhammad@isir.fr, lionel.prevost@upmc.fr

Abstract

We have proposed a complete system for text detection and localization in gray scale scene images. A boosting framework integrating feature and weak classifier selection based on computational complexity is proposed to construct efficient text detectors. The proposed scheme uses a small set of heterogeneous features which are spatially combined to build a large set of features. A neural network based localizer learns necessary rules for localization. The evaluation is done on the challenging ICDAR 2003 robust reading and text locating database. The results are encouraging and our system can localize text of various font sizes and styles in complex background.

1. Introduction

With an exponential increase in the production of images and video sequences, the automated procedures for the image understanding and interpretation are of great interest in present days. The textual information present in these images and video sequences is quite useful and can be used for application like indexation [12], assistive technologies for blind and visually impaired persons [3] etc. Due to a large variability of text in appearance (font style, size, surface), complex background and occlusions make it very difficult to extract textual information from images. Majority of the previous works on text detection and localization are based on hypothesis generation and validation paradigm [2] [6] [17]. In this framework, potential text candidate regions are generated using some image processing methods and verified by a validation scheme. Moreover, these works can also be categorized by the features used in these two steps: gradient features based, color based and texture features based [12]. Gradient features based approaches are known for their rapidness while texture and color based approaches are more noise tolerant. The techniques for hypotheses generation vary to a large extent. Some use simple image processing methods [2] [17] while others employ elegant cascade detector [3] [11]. Text localization schemes found in literature (e.g. [6] [17]) are generally based on simple heuristics based on geometric and spatial characteristics of text regions.

Our proposed text detector is based on the cascade of boosted ensemble. One of the novel contributions of this paper is the consideration of feature complexity in AdaBoost feature selection algorithm [7]. In case of heterogeneous feature set, the integration of feature complexity in feature selection algorithm helps in reducing the overall complexity of strong classifier and also the computational load which is an important consideration in real time applications. We have investigated in detail the effect of text and non-text examples in cascade training. Another contribution of this paper is an effective method for text localization. we have employed a sophisticated neural network to learn the localization rules automatically. Our approach is different from the those who employ hand made rules. These hand crafted rules are naive and hence, the choice of learning such rules automatically is a far better option. The complete system is shown in figure 1.

The rest of the paper is organized as follows: the complete description of proposed text detector and localizer can be found in section 2 and 3 respectively. Experimental setup and results on ICDAR 2003 robust reading and text locating database [10] are presented in section 4. Finally, conclusions and future works are presented.

2. Text detector

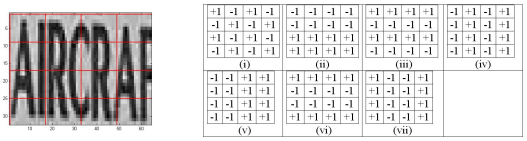
2.1. Features extraction

The features extracted in this work for text detection are same as used in our previous work [8] but this time, they are computed on rectangular text segments and with different normalization scheme. In our approach, various over-

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.172

Figure 1. Detection/Localization system

lapping text segments are first extracted from an image containing text lines or words. Each text segment is of NxM (32x16) pixels or its integer multiples. K (=9) different scales are considered in our experimentation and depends on the maximum font size that we want to detect. Each text segment contains at least 2 or more characters as we are not interested in detecting single character. Three different types of features (Mean Difference Feature (MDF), Standard Deviation (SD) and Histogram of oriented Gradient(HoG)) are extracted from a text segment on block level by placing a grid of 4x4 blocks on a text segment of NxM pixels or on its integer multiples, making a total of 16 blocks per text segment (see figure 2a). The HoG features have been successfully employed for object detection tasks (e.g. pedestrian detection [4]). In our approach, a HoG histogram of 8 bins is computed for each block and is normalized in a manner so that its sum equals to one. The MDF is a vector of 7 components defined as the weighted mean of each text segment (see figure 2b). Different weightings extract average horizontal, average vertical and average diagonal gray level changes in the text segment. The SD feature is a vector of 16 dimensions and represents the standard deviation of each block. Finally, we get a feature set of 39 features (7 MDF, 16 SD and 16 HoG). The dimension of this feature vector is 151. These features can be computed in a fast manner using integral image [15] and integral histogram [4].



(a) (b)

Figure 2. (a) Grid placement on text segment (b) MDF weight scheme

2.2. Weak classifier

In boosting framework, the choice of weak classifier is not trivial. We have chosen Likelihood Ratio Test (LRT) as weak classifier based on our previous experience [8]. Likelihood-ratio test is a statistical test for taking a decision between two hypotheses. In our design, probability density functions for text and non-text classes are supposed to be gaussian and a single gaussian is used to model each class using maximum likelihood criterion. Mathematically, likelihood-ratio test has the following form:

$$g(x) = -2ln\frac{p(x|T)}{p(x|\neg T)}$$

where $p(x|T)$ and $p(x|\neg T)$ are probability densities of text and non-text classes respectively.

It has been well established that feature combination leads to good results. We have used the same fact and have constructed a large feature space using single and combination of two or more features of the same or different types. We have 39 features in total and a combination of two features gives a total of 741 features set. The dimensions of feature vector vary from 1 (MDF or SD) to 16 (HoG+HoG). One weak classifier defined earlier can be associated with each feature set in feature space. Thus, a total of 780 weak classifiers can be obtained using single and combination of two features. The AdaBoost algorithm search sequentially one weak classifier from the given 780 weak classifiers at each iteration and combines all selected weak classifiers into an accurate classifier. Mathematically, the j^{th} weak classifier based on log-likelihood ratio test is given by:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j g_j(x) < p_j \theta_j \\ -1 & \text{otherwise} \end{cases}$$

where θ_j is the threshold and p_j is the parity.

2.3. AdaBoost algorithm and feature complexity

The feature selection process in classical AdaBoost algorithm [7] is based on the weighted classification error done by each weak classifier. All selected hypotheses based on the features are combined to construct a **strong classifier**. If the weak classifiers (and associated features) have same complexity then weighted classification error is the correct choice in feature selection process. If applied to a heterogeneous situation (where all the weak classifiers are not of same complexity), due to its greedy nature, classical AdaBoost algorithm will select most complex features in early iterations as they give low weighted error.

We propose a modified AdaBoost algorithm (called CAdaBoost) which takes into account the complexity of the weak classifiers at feature selection step. The algorithm is described in table 1. The feature selection is based on a function F of complexity (c_j) and weighted error (ϵ_j) of the associated weak classifier. The feature which minimizes this function, will be selected. We define it as:

$$F(\epsilon_j, c_j) = \epsilon_j * c_j$$

In the AdaBoost framework, we define the complexity as a function of computational cost that can be denoted as

Given $(X,Y) = (x_1,y_1),...,(x_N,y_N)$ with $x \in \Re$ and $y_i \in \{-1,+1\}$. Let h_j be the weak hypotheses based on f_j (feature vectors) for $j = 1, ..., M$ with complexity factor $(c_1, c_2, ..., c_M) \in (0, 1)$

Initialize $\omega_1(i) = \frac{1}{N}, i = 1, ..., N$
For $t = 1, ..., T$

1. Train weak hypotheses using distribution ω_t
   $(h_1, h_2, ..., h_M) = WeakLearn(X, Y, \omega_t)$
2. Select best weak hypothesis h_t and error ϵ_t
   $(h_t, \epsilon_t) \leftarrow \arg\min_j F(\epsilon_j, c_j)$
   where ϵ_j is the weighted error of classifier h_j computed as
   $\epsilon_j = Pr_{i\sim\omega_t}[h_j(x_i) \neq y_i] = \sum_{i:h_j(x_i)\neq y_i} \omega_t(i)$
3. Choose $\alpha_t = \frac{1}{2}(\frac{1-\epsilon_t}{\epsilon_t})$
4. Update: $\omega_{t+1}(i) = \frac{\omega_t(i)exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
   where Z_t is a normalization factor chosen so that ω_{t+1} is a distribution

Output the final hypothesis: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
where sign(...) returns the sign of the real valued output of strong classifier and serves as the label for x

Table 1. Complexity AdaBoost (CAdaBoost)

$G(t(f), t(h))$. The $t(.)$ is the time function and G relates the time of a feature vector f and the associated weak classifier h. The complexity function G is defined as:

$$c_j = G(t(f_j), t(h_j)) = t(f_j) + t(h_j)$$

For feature pairs, the complexity factor is defined as:

$$c(f_1 + f_2) = c(f_1) + c(f_2)$$

It is possible to estimate the computational cost of a certain feature $t(f_j)$ and associated weak classifier $t(h_j)$ on the training database. However, in our experiments, we have used arbitrary (empirically selected) values to define complexity and our choice is found to be effective (see result section 4). In [5], the authors also suggested empirical values to define complexity for a set of features and proposed a modified version of AdaBoost. In another work [14], McCane and Novins have modeled the trade-off between the cost of execution of a cascade classifier and the required false positive rate. Our proposed method is quite different from the above two related works.

2.4. Attentional cascade

The main benefit of attentional cascade, originally proposed by Viola and Jones [15], is its speed and generalization ability. In the first few stages, with the help of very few tests, a lot of *non-object* instances can easily be thrown while dedicating more tests and time to *object* instances in the later stages. At each stage, a strong classifier is trained by using AdaBoost algorithm and the threshold of strong classifier is adjusted in such a way so as to pass all object instances while rejecting a certain percentage of non-object instances. The cascade uses bootstrap mechanism to collect non-object instances to train a new stage. Each stage is trained to achieve a certain goal defined in terms of : 1) maximum allowable false alarm rate (FA_{max}), 2) minimum allowable detection rate (DT_{min}), 3) maximum number of weak classifiers per stage (N) and 4) maximum number of stages (Q). The overall false alarm rate of cascade is F_o and it is the product of false alarm rates of individual stages.

3. Text Localizer

The objective of text localizer is to output one text rectangle per text word present in the image as ICDAR database [10] is annotated at word level. The output of text detector are rectangles of varying sizes covering text regions and some non-text regions. Here, we assume that text is quasi-horizontal. We start by merging all detections. In other words, we take a union of all detection rectangles and a binary image is constructed from this union by placing 1's in the area of merged rectangles and 0's otherwise. Next, the connected components are extracted from the binary image. Each connected component represents a text region (can contain several lines of text) or a false alarm. These connected components are projected to gray level image and an edge map using Canny filter is computed for each connected component. Morphological operators generate connected components containing large characters or words or even some text strings where characters are small and words are touching each other. These connected components must be verified in order to reduce false alarms and to generate valid text regions containing large characters or words.

The validation problem is a bi-class classification problem where connected components containing text should be separated from non-text ones. Initially, we trained a classifier in AdaBoost framework but the results were not encouraging due to a small number of features per connected component. Later, we decide to use a Multilayer Perceptron (MLP) for validation scheme. The MLP has been successfully employed for various classification and regression problems [1] and generalizes well if designed carefully. We have extracted a total of 15 features from each connected component. These features are:

- **Edge features**: edge count per connected component and per bounding box, area, line edge count (average edge count in vertical projection), width and height
- **Gradient features**: gradient mean, gradient standard deviation and gradient maximum
- **Haralick texture features [9]**: contrast, homogeneity, dissimilarity, energy, entropy and correlation

In training, all features are normalized between -1 and 1 and are fed to a 2 layer MLP classifier with 15 inputs, N_H hidden cells and 2 outputs. The optimal number of hidden cells is found through hit and trial on training database. The cost function is the mean squared error and an adaptive learning step is used in the training. Finally, all verified connected components for a text word are clustered together to construct a single rectangle around the text word using three simple rules based on components vicinity. Given two neighboring rectangles (x_i, y_i, w_i, h_i) and (x_j, y_j, w_j, h_j), the vicinity based rules can be written as:

- $abs(x_i - x_j) < 1.5 \max(w_i, w_j)$
- $abs(y_i - y_j) < 0.3 \max(h_i, h_j)$
- $\min(h_i/h_j, h_j/h_i) > 0.5$

The first rule ensures that the connected components are neighbors in horizontal direction. The second rule emphasizes that the connected components belongs to same text line and third one ensures that they have similar height.

4. Experimental results

The training and test databases are taken from ICDAR 2003 robust reading and text locating database [10] which contains 250 images each. This database has been used for training competition entires on robust reading and text localization held in 2005 (see [13] for competition results). The text in this database varies greatly in fonts, their sizes, writing styles and appearance. In all experiments, size of text segment varies from 32x16 pixels to 288x144 pixels in 9 steps. We collected 2142 text segments and 5000 non-text segments to train different detectors. The test database contains 2301 text segments and more than 7 million non-text segments obtained from 250 test images. The computation time given in various cases is the average time to process an image of 640x480 pixels on an AMD Athlon machine at 2.2GHz and 1Gb memory.

Single detectors: In the first experiment, we compare the performance of single strong classifier trained by using classical AdaBoost and modified AdaBoost (CAdaBoost) algorithms. The number of boosting rounds is 300 in each case. We have observed that the HoG and HoG pairs are generally selected in the start due to low weighted error but they are computationally expensive. The complexity factors can be defined in a manner to restrict HoG features to be selected in start and allowing MDF and SD features to be selected. Three sets of complexity factors are used in our experimentation and are listed in table 2a. The results of single detectors on test database are summarized in table 2b. The performance is comparable but the computation time is reduced by one-third. Although, we have chosen the complexity factors arbitrarily but we have observed that it is the relative difference and not the absolute value that plays the major role.

	Feature Type		
Complexity Factors	MDF	SD	HoG
C1	0.1	0.2	0.5
C2	0.1	0.12	0.25
C3	0.1	0.12	0.15

(a) Complexity factors

Boosting Algorithm	Complexity Factors	Detection Rate	False Alarm Rate	Time (seconds)
AdaBoost	-	0.87918	0.00198	10.1
CAdaBoost	C1	0.86745	0.00638	6.63
	C2	0.86266	0.00469	6.70
	C3	0.88787	0.00227	7.07

(b) Performance of single detectors

Table 2. Single detectors

Cascade detectors: In our design, the cascade parameters used are: $FA_{max} = 0.5$, $DT_{min} = 0.99$, $N = 500$, $Q = 10$ and $F_o = (0.5)^Q = 0.00098$. We have used 9000 images which do not contain any text instance for bootstrapping purpose. One third (1/3) of the train database is used as validation database which is employed to tune the threshold of strong boosted detector at each stage. We have trained 3 realizations of each cascade and the average detection rate, average false alarm rate and their standard deviations on test database are summarized in the table 3. We can see that the average false alarm rate is relatively high in CAdaBoost cases (cascades D3 and D4) but the standard deviation is low. This means that cascades trained with CAdaBoost are stable as compared to its counter parts (cascades D1 and D2) and generalize well. The effect of increasing text segments in training results in increased detection rate but at the same time the false alarm rate also increases. This is due to the fact that now the training database contains very hard text segments which are difficult to be separated from non-text segments. The benefit of speed is obvious.

Localizer: The connected components are collected by applying the text detector at each scale and position and some preprocessing steps as defined in section 3. The training database contains 2376 text and 15301 non-text connected components. The test database contains 2030 and

Boosting Algorithm		AdaBoost		CAdaBoost	
Cascade Realizations		D1	D2	D3	D4
Text Segments/Stage		1472	2500	1472	2500
Non-Text Segments/Stage		3000	5000	3000	5000
Number of Stages		9	7	9	7
Number of weak classifiers		700	796	1058	1290
Detection Rate	μ	0.81095	0.88396	0.78792	0.90539
	σ	0.01372	0.00959	0.00230	0.00802
False Alarm Rate	μ	0.00078	0.00193	0.00095	0.00342
	σ	0.00015	0.00049	0.00003	0.00042
Time (seconds)		1.28	1.84	1.55	2.34

Table 3. Cascade detectors

Metric		Normal Cases (184 images)	Difficult Cases (36 images)	All (250 images)
Area	Precision	0.52	0.39	0.56
	Recall	0.75	0.53	0.64
ICDAR	Precision	0.28	0.13	0.25
	Recall	0.41	0.22	0.35
Wolf	Precision	0.31	0.18	0.30
	Recall	0.55	0.37	0.49

Table 4. Average recall and precision

17171 connected components containing text and non-text elements respectively. In our experimentation, the number of hidden cells (N_H) varies from 1 to 20. A total of 25000 iterations have been done and a cross validation database (20% of training database) is used for stopping. We have trained 3 realizations of each neural network and results are the average of the three realizations. The optimal network is the one with 5 neurons in hidden layer. The evaluation on test database gives 96.2% as detection rate and a false alarm rate of 7.6%.

Complete System: Finally, we combine proposed detector and localizer (see figure 1) and results are measured in standard precision and recall terms. For evaluation, we have separated the ICDAR *TestTrial* database into three parts: 1-) Images containing single characters or font size more that 144 pixels (recall that these are the detector limitations), 2-) Images containing printed characters and regular fonts (normal cases) 3-) Images containing text of irregular font and having very low contrast (difficult cases). There are 29 images in first category, 184 in second and 36 in third one. The results using three difference metrics defined in literature [16], on last two categories and on complete database are summarized in table 4. In Wolf method, tp and tr are 0.4 and 0.8 respectively. Some of the detection images are shown in figure 3. The first column shows *normal* cases while second column contains *difficult* ones.

5. Conclusions and perspectives

In this paper, we have presented a complete system for text detection and localization. The evaluation of the system on the difficult ICDAR database shows that it is capable of detecting and locating text of different sizes, styles and types present in natural scenes. The integration of complexity in feature selection algorithm like AdaBoost helps producing efficient detectors. A neural network based localizer learns automatically all the rules based on connected components geometry and spatial relationships. In future, we will be evaluating our system on a large database of images. We are exploring more features that are helpful in text detection and we are also working on automatic determination of complexity factors used in CAdaBoost algorithm.

Figure 3. Test images with detections

References

[1] C. M. Bishop. *Neural Network for Pattern Recognition*. Oxford University Press, NY, USA, 1995.

[2] D. Chen, J. M. Odobez, and J. P. Thiran. A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods. *Image Communication*, 19(3):205–217, 2004.

[3] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *CVPR*, 2:366–373, 2004.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 1:886– 893, 2005.

[5] P. Dollar, Z. Tu, H. Tao, and S. Belongie. Feature mining for image classification. *CVPR*, pages 1–8, 2007.

[6] N. Ezaki, M. Bulacu, and L. Schomaker. Text detection from natural scene images : Towards a system for visually impaired persons. *ICPR*, 2:683–686, 2004.

[7] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *ICML*, pages 148–156, 1996.

[8] S. M. Hanif, L. Prevost, and P. A. Negri. A cascade detector for text detection in natural scene images. *ICPR*, pages 1–4, 2008.

[9] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textual features for image classification. *Systems, Man, and Cybernetics*, SMC-3(6):1973, 1973.

[10] ICDAR. Icdar 2003 robust reading and locating database. *http://algoval.essex.ac.uk/icdar/RobustReading.html*, 2003.

[11] M. Lalonde and L. Gagnon. Key-text spotting in documentary videos using adaboost. *Symposium on Electronic Imaging, Science and Technology*, 38:1N–1–1N–8, 2006.

[12] J. Liang, D. Doermann, and L. Huiping. Camera-based analysis of text and documents : A survey. *IJDAR*, 7:84–104, 2005.

[13] S. M. Lucas. Icdar 2005 text locating competition results. *ICDAR*, 1:80–84, 2005.

[14] B. McCane and K. Novins. On training cascade face detectors. *Image and Vision Computing New Zealand*, pages 239–244, 2003.

[15] P. Viola and M. J. Jones. Robust real-time face detection. *Computer Vision*, 57(2):137–154, 2004.

[16] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *IJDAR*, 8(4):280–296, 2006.

[17] V. Wu, R. Manmatha, and E. M. Risemann. Text finder: An automatic system to detect and recognize text in images. *PAMI*, 21(11):1224–1228, 1999.