2009 10th International Conference on Document Analysis and Recognition

A New Block Partitioned Feature for Text Verification

Xiufei Wang[1], Lei Huang[1,2], Changping Liu[1,2]
[1]*Institute of Automation, Chinese Academy of Science*
[2]*HanWang Technology Co., Ltd., China*
E-mail: xiufei.wang@ia.ac.cn, lei.huang@mail.ia.ac.cn, changping.liu@mail.ia.ac.cn

Abstract

In this paper, a new feature for text verification is proposed. The difficulties for the selection of features for text verification (FTV) are first discussed, followed by two principles for the FTV: the FTV should minimize the influence of backgrounds, and it should also be expressive enough for all the texts varied in structures prominently. In this paper, we exploit different block partition methods and introduce two widely used features: the gray scale contrast (GSC) feature to eliminate the background difference, and the edge orient histogram (EOH) feature to distinguish the structure of texts from that of non-texts. A texture classifier can be got by SVM training of pre-labeled data. The candidate text lines can be verified by this classifier. Experimental results show that our feature performs well.

1. Introduction

With the development of internet and digital media technique, a quantity of multimedia comes forth, which leads to an urgent demand for content based browsing and retrieving system. Text in images and videos always carries rich useful information, which can help the computer to understand the content of images and videos. So text location is very important for the fields of automatic annotation, indexing and parsing of images and videos.

A variety of approaches for text location have been proposed during the past decades [1, 2]. In the early stages of text location research area, the methods and algorithms are relatively simple. Very few and simple information, like edges, corners, connected components and etc., are used to locate texts in images and videos [3]. These methods perform fast, but would inevitably introduce a large number of false detections, especially when the texts are located in complex background. More recently, many researchers have tried to apply the theory of pattern classification to text location research. That is, a texture classifier is first got by some machine learning method such as SVM, MLP, Adaboost and etc. based on some special selected features, then the classifier is used to classify the sub-regions of the input images into texts and non-texts [5~8]. For the process of text classification is time consuming, a two-stage text location frame has been proposed. In this frame, candidate texts are first got by some heuristic text information, and then a texture classifier is used to eliminate false alarms. The flow chart of this frame is shown in Fig. 1.

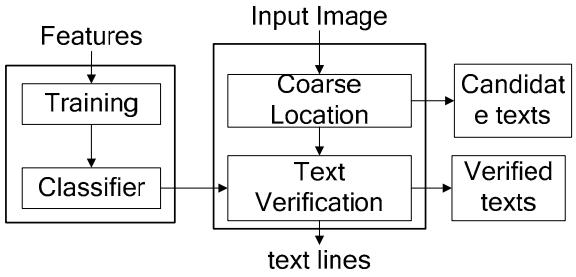


Fig. 1 Flow chart of the two-stage text location frame

As the text verification step is adopted in this text location frame to eliminate false detections, there comes another problem: how to choose an effective texture feature? Many researchers have proposed different texture features for text verification (FTV), such as the gray scale feature by Kim, edge-map feature, derivations and etc.. Well, some of these features do show good performance, but that depends on the database and the text backgrounds. The selection of these features is more based on the experimental tests. That is to say, no one answers the basic question: what on earth is text?

In this paper, we would draw our attention mainly to the task of text verification, and attempt to solve the basic problem of the selection of FTV. We exploit different block partition methods and introduce two

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.61

widely used features: the gray scale contrast (GSC) feature to eliminate the background difference, and the edge orient histogram (EOH) feature to distinguish the structure of texts from that of the non-texts. The Experimental results show that our feature performs better than the traditional texture features.

The rest of this paper is organized as follows: section 2 elaborates the difference of FTV and features for text recognition (FTR); the proposed features are shown in section 3; section 4 are related experiments results and analysis; finally we draw our conclusions in section 5.

2. Features for text verification

The selection of FTV is a challenging task. It is quite different from the selection of FTR, mainly in the following aspects: FTV aim to distinguish between texts and non-texts, while FTR try to distinguish texts from texts; more usually, texts for recognition have already been segmented from the background, so FTR are less influenced by noises. But texts to be localized are usually embedded in complex backgrounds, so FTV should be robust to noise. Briefly speaking, the FTR should answer the question: which word this text is, while FTV should answer the question: whether this is text or not.

Some researchers have tried to apply some traditional texture features, such as edge map, SIFT [9] and etc., to the text verification area. But most of them don't perform well. Some researchers have also proposed some different FTV, such as the gray scale feature by Kim [6], constant gradient variance feature by Chen [5] and etc. [8]. It seems that these features perform well in the artificial database, while actually they don't answer the basic question of the selection of FTV.

As we have mentioned before, the difficulties for the selection of FTV are mainly based on two factors: the influence of complex backgrounds and the variance of the different structures of texts. Therefore, to choose an effective FTV, we must solve the following two problems:

1) The selected FTV should minimize the influence of background as far as possible;

2) The selected FTV should be expressive enough for all the texts varied in structures.

Based on the two points below, we propose a new FTV. The elaboration of this feature is shown in the following section.

3. Proposed features

When looking into the features we mentioned below carefully, it's not difficult to find that these features are mainly point-based, that is, the features are extracted based on the specified pixel point of the texts, or some kind of transformation of these points such as FFT, DCT and etc.. As different texts under different backgrounds show quite different in style, color and structure, these point-based features can hardly represent the features of texts in common.

Compared with the point-based feature, we propose a new definition of region-based FTV. Although the specified points in different texts show different, they do have something in common in some specified regions. We find some characters of texts in common by analysis and statistics:

1) All the texts are formed by edges in different directions;

2) These edges are distributed regularly in some local regions.

Based on the two points below, we partition the text block into eight parts as shown in Fig. 2.

Fig. 2 Text partition sketch map

The validation and superiority of our text partition method is showed in section 5.3, in which different block portioned methods are compared with our method.

Then for each sub-region, we introduce the following text features:

3.1. Gray scale contrast feature (GSC)

Texts are usually embedded in complex background. Even the same text may show different under different backgrounds, which increases the complexity of the text verification task. To eliminate the influence of text background, we introduce the gray scale contrast feature, including the mean contrast R_m and variance contrast R_v, which are defined as:

$$Rm_k = \frac{Lm_k}{Gm} \tag{1}$$

$$Rv_k = \frac{Lv_k}{Gv} \tag{2}$$

where Lm_k and Lv_k refer to the local mean value and the local variance value of sub-region k. Gm and Gv denote the global mean value and global variance value of the total text region respectively.

3.2. Edge orientation histogram feature (EOH)

Texts are formed by regularly distributed edges in different directions. To express this special edge structures in the texts, the edge orientation histogram (EOH) feature is introduced as another feature in our work. EOH is a useful and effective texture feature. In this paper, the EOH is extracted by the following steps:

1) Detect the edges of the original image by Sobel mask. Get horizontal edge map $E_X(p)$ and vertical edge map $E_Y(p)$;

1	2	1
0	0	0
-1	-2	-1

(a) Sobel Mask X

1	0	-1
2	0	-2
1	0	-1

(b) Sobel Mask Y

Fig. 3 Sobel Mask

2) Calculate the edge orientation map $\hat{\theta}(p)$ by:

$$\hat{\theta}(p) = \arctan\frac{E_Y(p)}{E_X(p)} \tag{3}$$

3) Map $\hat{\theta}(p)$ to region $[0,2\pi]$, and get $\theta(p)$;

4) Quantize $\theta(p)$ into eight parts with a gap of $\pi/4$. Get the histogram of each direction.

Then for each sub-region of the text block, the GSC and EOH features are extracted. As the text block is partitioned into eight parts, the dimension of the feature in our paper is $(2+8)\times 8 = 80$.

4. Text verification

As our work is mainly focused on the text verification, we do not pay much attention to the extraction of candidate texts. In this paper, we assume that the candidate texts are all bound texts. To get the texture classifier, we introduce the support vector machines (SVM) for the training. In the verification step, we use the weighted region method as Chen proposed in [5].

4.1. The training of text classifier

In this paper, SVM [10] is used to train the text classifier. SVM is a machine learning method proposed by V. Vapnik, and has been widely used in classification problems for its good performance. The core idea of SVM is to find an optimal separating hyper-plane, one which maximizes the margin between the two classes. For the non-linear situation, the kernel functions are used to map the original data into higher dimension, in which a linear separating hyper-plane can be found.

In short, given m labeled training samples: $(x_1, y_1), \cdots, (x_m, y_m)$ where $y_i = \pm 1$ is the label of the input data, the trained SVM text classifier is shown as

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{4}$$

where α and b are parameters got by SVM training, and (x_i, y_i) is the selected support vector by SVM.

4.2. Text verification method

In the text verification step, we adopt the weighted method proposed by Chen [5] as the verification scheme in this paper. A slide window is used to scan the candidate text with some slide steps. For the scanned sub-regions, the FTV proposed in Section 3 is extracted and imported to the SVM text classifier in Section 4.1. Then a set of verification scores can be obtained by the scanning. The confidence of the whole candidate text is defined as:

$$Conf(R) = \mathrm{sgn}\left(\sum_{i=1}^{l} f(x_i^R) \times \frac{1}{\sqrt{2\pi}\sigma_0} e^{-d_i^2/2\sigma_0^2}\right) \tag{5}$$

Where d_i is the distance from the geometric center of the ith sliding window to the geometric center of the candidate text R. σ_0 is a scale factor depending on the text line length, and $f(x_i^R)$ is the classification score calculated by equation (4).

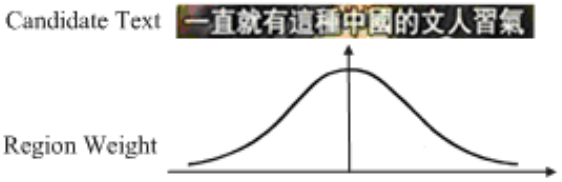


Fig. 4. Text verification

5. Experiments and analysis

To prove the superiority of our FTV, we do groups of experiments. The details of our experiment data and results are shown in the following parts. All the experiments in this paper are done on the computer with a CPU of Pentium IV 2.8GHZ.

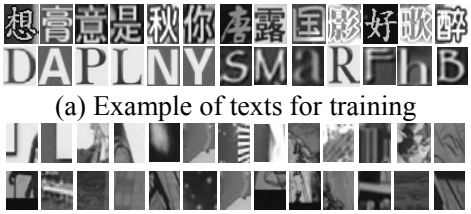
5.1. Dataset

We grabbed 2147 video frames with texts from 76 videos, including movies, TVs, MTVs and etc.. The size of the frames varies from 352×240 to 720×576, and the height of text lines from 12 to 35 pixels.

5.1.1. Training data. In most of the text verification methods, the training data for the texture classifier is mainly based on the labeled text lines, that is, the text lines in the images are labeled instead of the single text, and then a slide window is used to extract the texts as positive training data [5~8]. This method is not so accurate for that in actual situations the margins between texts may be also extracted as texts, which would certainly influence the training results.

To be precise, the training data used in this paper is based on the labeled texts. We label exactly every text bound to make sure that no non-text noises would be taken in.

We select 300 images from the database and choose 1000 texts from the text lines and 2000 non-texts from the non-text regions as training data. Some of the texts and non-texts training data are shown in Fig. 5.

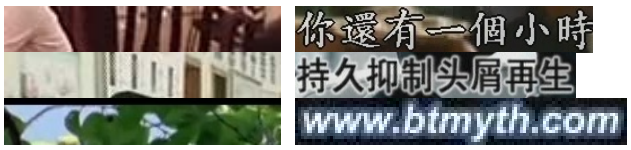


(a) Example of texts for training

(b) Example of non-texts for training

Fig. 5 Training samples

5.1.2. Testing data. We extract 1000 text lines and 3000 non-text lines as testing data. Our aim is to test the performance of the texture classifier got by training of our proposed FTV. Some examples of the testing data are shown in Fig. 6.



(a) Non-texts (b) Texts

Fig. 6 Testing samples

5.2. Performance evaluation

We consider a texture classifier for text verification as a good one if it satisfies the following conditions:

1) For the texts, the classifier can correctly verify it as texts;

2) For the non-texts, the text classifier can lower the false classifications.

Therefore, to evaluate the performance of texture classifiers, we define two measurements: correctly classification rate (CCR) and false classification rate (FCR), defined as:

$$CCR = \frac{N_{ct}}{N_{text}} \times 100\% \quad (6)$$

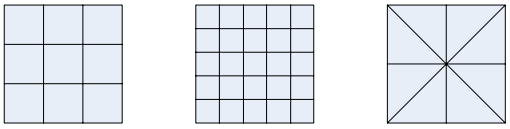
$$FCR = \frac{N_{fnt}}{N_{non-text}} \times 100\% \quad (7)$$

where N_{ct} and N_{text} denote the number of correctly classified texts (these texts are correctly classified as texts) and the total number of texts in the database, and N_{fnt} and $N_{non-text}$ mean the number of false classified non-texts (these non-texts are wrongly classified as texts) and the total number of non-texts in the database. Generally speaking, a text classifier with a high CCR and a low FCR can be regarded as a good one.

5.3. Comparison experiments

We did several groups of experiments to testify the superiority of our features. All the experiments are implemented by Visual Studio 2003 .Net.

In this paper, the text block is partitioned into eight parts as showed in Fig. 2, which is more rational and suitable for it fits people's writing habits well. To prove the superiority of our block partition method, we adopt some other schemes (3×3 and 5×5) and do comparison experiments with our method. For each block partition method showed below, the same FTVs and text verification method proposed in the paper are adopted. Experiment results are shown in Tab. 1.



(a) 3×3 (b) 5×5 (c) our method

Fig. 7 Different block partition methods used in the experiments

Table 1. Comparison experiments of different block partitioned methods

	3×3	5×5	Our method
Dimension	90	250	80
CCR	85.6%	90.0%	96.5%
FCR	16.8%	21.2%	1.4%

As we have discussed in Section 2, the FTV should be expressive enough for all the texts varied in structures. So the size of text sub-block is very important. If the size is too large, the extracted features may be robust to the noises but would also lose the specified character of texts. If the size is too small, what we get is the detail of every text, and it would be difficult to refine the features that all the texts share in common. It can be seen from the result that neither

3×3 nor 5×5 block partition methods don't perform quite well, while our method shows good performance.

To solve the FTV selection problems we discussed in Section 2, we introduce two widely used features: the gray scale contrast (GSC) feature to eliminate the background difference, and the edge orient histogram (EOH) feature to distinguish the structure of texts from that of the non-texts. In order to test the influence of the two features, we extract the two related features respectively and do some comparison experiments. The results are shown in Tab. 2.

Table 2. Comparison experiments of different part of the proposed text features

	GSC	EOH	GSC+EOH
Dimension	16	64	80
CCR	63.1%	64.3%	96.5%
FCR	2.9%	6.6%	1.4%

It can be seen from the results that it doesn't perform well when GSC and EOH are used separately, while the mergence of the two features shows good performance in CCR and FCR. This proves the discussion in Section 2.

We also do some comparison experiments of our features with some other FTV. In this paper, the gray scale feature (GS) [6] and the edge map feature (EM) [5] are adopted as comparison features. The experiment results are shown in Table 3.

From the results, we can see clearly that our feature performs better than the other two in both CCR and FCR. This is mainly because both the GS feature and the EM feature are point-based features. As we discussed in Section 3, the expression ability of point-based features is lower than the region-based features. The experiments fits the discussions well.

Table 3. Comparison experiments of the proposed features and some other FTV

	GS	EM	Our feature
Dimension	57	512	80
CCR	86.2%	78.1%	96.5%
FCR	5.1%	5.1%	1.4%

6. Conclusion and future work

In this paper, we propose a new FTV for text verification. The difficulties for the selection of FTV mainly come from the influence of complex backgrounds and the variance of different text structures. To overcome these two obstacles, we first partition the text block into eight parts by the spelling habits and the distribution of text edges, then two widely used texture features are introduced: the GSC feature to eliminate the influence of background difference, and the EOH feature to distinguish the texts from non-texts. Experiment results demonstrate the impressive performance of our feature.

To extract the texts information in images and videos, we also need to segment the verified texts from the complex background and recognize them, which is also a main task for our future work.

Acknowledgements

The work of this paper is supported by the National High Tech. 863 Programs of China under grant NO. 2007AA01Z174.

References

[1] R. Lienhart, "Video OCR : A Survey and Practitioner's Guide[R]", *Intel Corporation, Microprocessor Research Labs*, Santa Clara, California, USA, 2003.

[2] K. Jung, K.I. Kim, A.K. Jain, "Text information extraction in images and video: a survey", *IEEE Trans. on Pattern Recognition,* 37 (2004) 977-997.

[3] X.S. Hua, X.R. Chen, L.W. Yin, H.J. Zhang, "Automatic Location of Text in Video Frames", *Proceedings of the 2001 ACM workshops on Multimedia: multimedia information retrieval*, pp.24-27, 2001.

[4] R. Lienhart, A. Wernicke, "Localizing and Segmenting Text in Images and Videos", *IEEE Transaction on Circuits and System for Video Technology*, Vol.12, NO.4, April 2002, pp.256-268

[5] D. Chen, J.M. Odobez, H. Bourlard, "Text Detection And Recognition In Images And Video frames", *IEEE Transaction on Pattern Recognition*, 37 (2004) 595-608.

[6] K.I. Kim, K. Jung, J. H. Kim, "Texture-Based Approach for Text Detection in Images Using Support Vector machines and Continuously Adaptive Mean Shift Algorithm", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2003.

[7] L. Chuang, X. Q. Ding, Y.S. Wu, "An Algorithm for Text Location in Images Based on Histogram Features and AdaBoost", *Journal of Image and Graphics,* Vol.11, NO.3, Mar., 2006.

[8] Q.X. Ye, Q.M. Huang, W. Gao, D.B. Zhao, "Fast and robust text detection in images and video frames", *Images and Vision Computing* 23 (2005) 565-576.

[9] Lowe, David G. "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision* 2 (1999): 1150–1157

[10] Christopher JC Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". *Data Mining and Knowledge Discovery*, Vol. 2, No. 2. (1998), pp. 121-167