Extraction of characters on signboards in natural scene images by stump classifiers

Minoru Maruyama and Takuma Yamaguchi
Dept. Information Engineering, Shinshu University
4-17-1 Wakasato, Nagano 380-8553, Japan
maruyama@cs.shinshu-u.ac.jp s07t213@shinshu-u.ac.jp

Abstract

We present a method to detect characters on signboards in natural scene images. For many applications, both classifier with small computational cost and the efficient feature set, which gives rise to accurate recognition are required. Texture based features are often used for target detection. It has been also shown that the shape of the intensity distribution is often useful for character extraction. The intensity distribution in the character regions is often different from the unimodal distribution. We measure the discrepancy between the observed region and the normal distribution by skewness and kurtosis. We use these statistics along with the texture based features. Character regions in a natural scene image are detected by using the linear combination of stump classifiers, each of which sees only one component of multidimensional feature vector. Selection of a feature component for each stump and determination of coefficients of linear combination are carried out by AdaBoost. We experimentally show the effectiveness of the proposed method.

1. Introduction

With the increasing availability of digital imaging devices, new fields for character recognition is expanding. Using the devices, for examples a digital camera attached to a cellular phone, we can capture document images in natural scenes such as documents on signboards. Fast and reliable techniques to recognize such documents give us means to access to the further information and are useful for various kinds of applications. There has been research work on text location in scene images and video frames [6, 3]. To extract characters in natural scene images, it is commonly done to shift a search window over the input image and to categorize with some classifiers. The large range of possible variations within a class of target (i.e. characters in natural scenes) makes the recognition problem difficult. For accurate recognition, selection of image features is very important. Moreover, for wide range of applications, computational cost (i.e. speed) is also important. It is desirable that the total extraction process is performed in real-time. This requires fast classification method[4].

In this article, we treat a problem of character extraction on signboards in natural scene images. In Fig.1,Fig. 2, example scene images and image regions in search windows are shown. In our previous work [11], we proposed a character extraction method based on cascade of two types of classifiers : histogram based classifier and RBF (SVM). The histogram based classifier works due to the properties of the intensity distribution over character regions in a natural scene. In the previous method, after rejecting apparent non-character regions by the classifier, final decision was made by non-linear SVM. Although the performance of the method was fairly well, it was not satisfactory with respect to accuracy and computational cost. In the present article, we try to improve the performance.

To improve the extraction accuracy, we examine features based on edges and their gradients (Haar wavelet and HOG (Histogram of Oriented Gradient)), statistics of intensity distribution (skewness and kurtosis). After evaluating the performance of SVMs, which are trained by using these features separately, we consider to use the combination of these features.

For target detection in images, SVM has been used successfully[7]. However, given several different types of features, it is not straightforward to apply SVM, because scale adjustment among features should be carried out. In addition, although SVM with non-linear kernel can perform very well, its computational cost is usually high. To avoid these difficulties, we use a set of stump classifiers with boosting algorithm[8]. We examine the performance of proposed method by experimental evaluation.

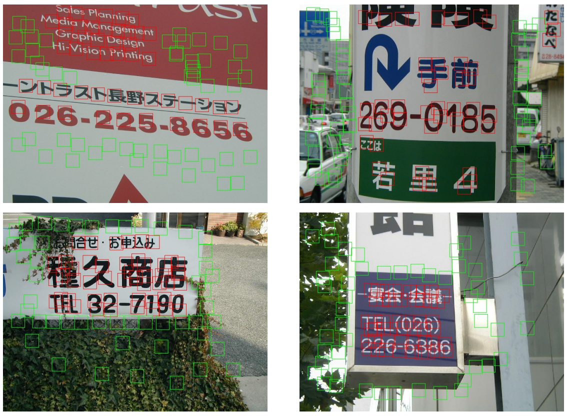

Figure 1. Example images of characters in signboards

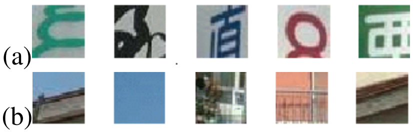

Figure 2. Examples of target areas for classification : (a) character regions, (b) non-character regions

2. Features for character extraction on signboards

2.1. Edge based features

To improve accuracy of classifiers for character extraction, selecting efficient set of features is required. For target detection (e.g. pedestrian detection) edge and gradient based features such as Haar wavelet [9], HOG (Histogram of Oriented Gradient)[1] have been successfully used. We use these features for character extraction. In our method, we use sparse Haar wavelet. For each search window in a gray scale image, Haar wavelet representation is obtained. Then, among the total coefficients, only 5% with greatest absolute values are unchanged. The other coefficients are replaced with zeros to make sparse representation.

We also examine HOG descriptors. HOG is a 1-D histogram of edge orientations over the pixels within a search window. It is expected that the HOG representation has invariance to local geometric and photometric transformations. Let $I(x, y)$ be the intensity at position (x, y) and $I_x,\ I_y$ be the directional derivatives. Edge direction θ and the squared gradient m are obtained as :

$$\theta = \arctan(I_x, I_y),\ \ m = \sqrt{I_x^2 + I_y^2}\ (-\pi < \theta < \pi) \quad (1)$$

If $\theta < 0$, then we let $\theta \leftarrow \pi + \theta$. In our implementation the space of orientation is partitioned into 9 bins ($20\,\mathrm{deg}$ per bin). Weighted voting using the squared gradient $m(x, y)$ is carried out to make a histogram.

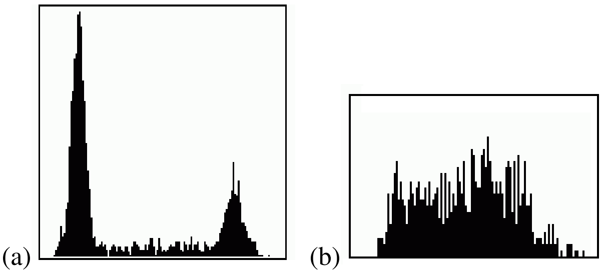

Figure 3. Examples of intensity distribution in search windows : (a) character region, (b) non-character region

2.2. Moment statistics on intensity distribution

As well as edges and their gradients, intensity distribution over the image region carries useful information for character extraction. In this article, our target is to detect characters on signboards in an image. The signboards are usually designed so that people can recognize characters clearly. There should be apparent difference in luminance between the characters and their background. We used this property in the previous work[11]. Typically the intensity distribution over the character region is often multimodal. On the other hand, the distribution over the non-character region is unimodal (see Fig.3). In the previous work, we extracted character candidate by counting the number of peaks of smoothed intensity histogram. However, careful adjustment of parameters for smoothing and threshold is required to make this method work well. The basic idea behind the method was : the shape of the intensity distribution over the character regions are very much different from the symmetric unimodal distribution. In the present work, instead of counting the peaks, discrepancy between the intensity distribution and the normal distribution is measured. For that purpose, skewness and kurtosis are calculated. Let $I_i (i = 1, \cdots, N)$ be the pixel values in the image region, and $\hat{I}_i$ be the normalized pixel values, where

$$\bar{I} = \frac{1}{N}\sum_{i=1}^{N} I_i,\ V = \frac{1}{N}\sum_{i=1}^{N}(I_i - \bar{I})^2,\ \hat{I}_i = \frac{1}{\sqrt{V}}(I_i - \bar{I}) \tag{2}$$

Let M_k be the sample moment of order k.

$$M_k = \frac{1}{N}\sum_{i=1}^{N} \hat{I}_i^k \tag{3}$$

Skewness γ_1 and kurtosis γ_2, which vanish for normal distribution, are given as :

$$\gamma_1 = M_3,\ \gamma_2 = M_4 - 3 \tag{4}$$

We use thse statistics (γ_1, γ_2) for image features for character extraction.

2.3. Performance of SVMs

To evaluate the effectiveness of the features described above, preliminary experiments were carried out. Using Haar wavelet (full and sparse), HOG, and moment statistics (skewness and kurtosis), separately, we trained SVMs with RBF kernel. In the experiment, the size of the search window is 32 × 32 pixels. Total number of training samples is 2250 (1125 positive and 1125 negative). Parameters needed for SVM learning (such as scale parameter of Gaussian kernel) were selected manually. We used SVM^{light}[5] for learning. The performance of the resultant classifiers was examined by using 2250 test samples (1130 positive and 1120 negative samples). In table 1, we show precision (true positive / (true positive + false positive)) and recall (true positive / (true positive + false negative) = true positive / actual positive) values on the test set and number of support vectors of each classifier. Each of the features represents different kind of property of the image region. In the next section, we consider to combine them.

Table 1. Classification performance by SVMs

feature	precision	recall	#SV
Harr	0.856	0.887	1631
Harr 5%	0.898	0.860	1020
HOG	0.825	0.895	967
moment	0.825	0.964	853

3. Ensemble of stump classifiers

When different types of features are given, in order to train a single classifier (such as a SVM) which uses all of the features, careful scale adjustment is required. Usually it is not an easy task. In this article, instead of finding a single very powerful classifier, we train many classifiers each of which relies on small subset of features. For that purpose, we use boosting (AdaBoost[2, 8]). Boosting is well known as an effective method of producing a very accurate classifier by combining mediocre classifiers (base learners)[10]. For simplicity and speed (i.e. computational cost) decision stump is exploited as a base learner. Let $\mathbf{x} = (x_1, \cdots, x_d)^T$ be a multidimensional feature vector. A stump classifier $h(\mathbf{x})$ is given as :

$$h(\mathbf{x}) = \text{sgn}[x_i - \theta] \tag{5}$$

where, sgn$[\cdot]$ is :

$$\text{sgn}[u] = \begin{cases} 1 & u \geq 0 \\ -1 & u < 0 \end{cases}$$

A stump classifier h only sees single component x_i among the feature vector. To evaluate $h(\mathbf{x})$, the computation needed is just a comparison of a feature element x_i with the threshold θ. The final decision is made by combining these stump classifiers. The final classifier $f(\mathbf{x})$ is given as :

$$f(\mathbf{x}) = \text{sgn}[\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})] = \text{sgn}[\sum_{t=1}^{T} \alpha_t \text{sgn}[x_{i_t} - \theta_t]] \tag{6}$$

Coefficients $\{\alpha_t\}$, thresholds $\{\theta_t\}$ and feature selection $\{i_t\}$ are determined by learning from examples. Suppose that the training samples $\{(\mathbf{x}_\ell, y_\ell)\}$ $(\ell = 1, \cdots, L)$ are given, where $\mathbf{x}_\ell \in \mathbf{R}^d$ is a feature vector, and $y_\ell \in \{+1, -1\}$ is a class label. The learning algorithm (AdaBoost) to learn coefficients of linear combination α_t is as follows:

1. Initialize the weight for each sample $D_1(\ell) \leftarrow \frac{1}{L}$
2. `For` $t \leftarrow$ `1 to L`
    - (a) Train base learner (stump classifier) using $\{D_t(\ell)\}$
    - (b) $\varepsilon_t \leftarrow \sum_{\ell=1}^{L} D_t(\ell) y_\ell h_t(\mathbf{x}_\ell)$
    - (c) $\alpha_t \leftarrow \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$
    - (d) $Z_t \leftarrow \sum_{\ell=1}^{L} D_t(\ell)\exp(-\alpha_t y_\ell h_t(\mathbf{x}_\ell))$
    - (e) $D_{t+1} \leftarrow \frac{1}{Z_t} D_t(\ell)\exp(-\alpha_t y_\ell h_t(\mathbf{x}_\ell))$

To implement the above boosting algorithm, a learning algorithm that gives base learner h_t is needed. The algorithm should determine the feature component, on which the stump classifier operates, and an appropriate threshold. In our implementation a brute force method is used. We select threshold value θ_t from the components of training samples $\{(x_{\ell 1},\cdots,x_{\ell d})\}_{\ell=1}^{L}$. Suppose that a feature component $x_{\ell i}$ is picked from the training data. It also determines a stump classifier

$$h^{(\ell,i)}(\mathbf{x}) = \mathrm{sgn}[x_i - x_{\ell i}]$$

For each pair (ℓ, i), the following weighted error is calculated

$$\varepsilon_t(\ell,i) = \sum_{k=1}^{L} D_t(k) y_k h^{(\ell,i)}(\mathbf{x}_k) \qquad (7)$$

If $\varepsilon_t(\ell,i) < \frac{1}{2}$, then we let

$$h^{(\ell,i)}(\mathbf{x}) \leftarrow -h^{(\ell,i)}(\mathbf{x}) \quad \varepsilon_t(\ell,i) \leftarrow 1 - \varepsilon_t(\ell,i) \qquad (8)$$

A base learner at step t is given

$$h_t(\mathbf{x}) = h^{(\ell^*,i^*)}, \ \ (\ell^*,i^*) = \arg\min_{(\ell,i)} \varepsilon_t(\ell,i) \qquad (9)$$

The implementation detail is as follows. Suppose that for each dimension i the training samples $\{x_{\ell i}\}$ are sorted as $x_{\ell_1 i} \leq x_{\ell_2 i} \leq \cdots x_{\ell_L i}$ and then divided into groups so that elements in each group have same value : $S_p = \{\ell' | x_{\ell' i} = v_p\}$. If we let $\theta < v_1$, then the weighted error is given by

$$\varepsilon^0(i) = \sum_{y_\ell = -1} D_t(\ell)$$

We define $\varepsilon^m(i)$ as the weighted error when we let $\theta \leftarrow v_m$

$$\varepsilon^m(i) = \sum_{\{\ell | y_\ell = -1, x_{\ell i} \geq v_m\}} D_t(\ell) - \sum_{\{\ell | y_\ell = 1, x_{\ell i} < v_m\}} D_t(\ell)$$

$\varepsilon^m(i)$ can be calculated incrementally as :

$$\varepsilon^m(i) = \varepsilon^{m-1}(i) + \sum_{\ell \in S_m} y_\ell D_t(\ell) \qquad (10)$$

Since the weighted error of the reversed classifier sgn$[x_i - v_m]$ is $1 - \varepsilon^m(i)$, we can evaluate the effect of the pair (i, v_m) by $|\frac{1}{2} - \varepsilon^m(i)|$. $\min_{(\ell,i)} \varepsilon_t(\ell,i)$ is calculated by examining $\{\varepsilon^m(i)\}$. With this method we can easily find the best base learner which satisies (9). Apparently this brute force method can run fast especially for discrete features and sparse features.

4. Experimental Results

4.1. Stump classifiers based on single feature type

We examined the effectiveness of the classifiers based on stump classifiers obtained by learning method described above. In the experiment, we used image samples described in 2.3. We first trained the stump classifiers based on the single type of feature. In Fig 4 experimental results (precision and recall values with respect to the number of steps T, which is almost identical to the number of decision stumps) are presented. As the figure shows, for Haar wavelet, the performance of the resultant classifier is almost comparable or even better than the corresponding RBF-SVM (see Table 1). For other feature types, the performance of the RBF-SVMs is slightly better than the stump-based classifier. On the other hand, judging from the number of support vectors and the kernel type (Gaussian), the computational cost of the proposed method is much less than the corresponding SVM.

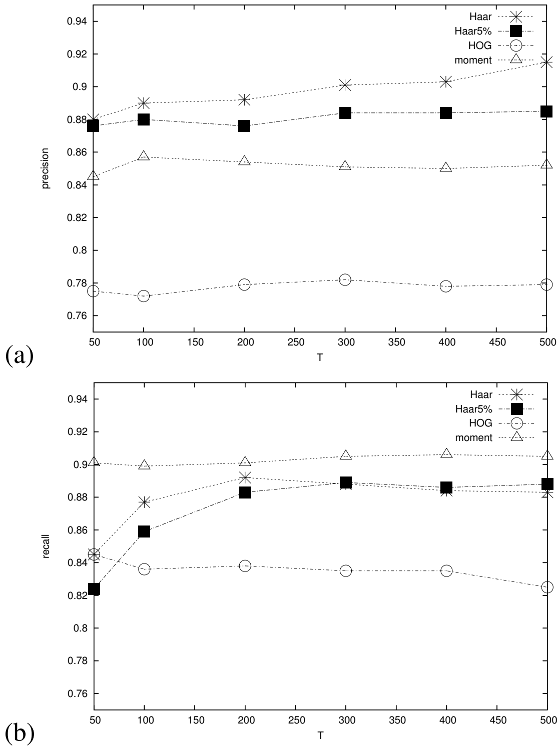

Figure 4. Classification performance by the stump classifiers with a single feature type : (a) precison, (b) recall

4.2. Stump classifiers based on combination of features

We next examined the performance of classifiers using the combination of different types of features. We examined the following combination : HOG+Haar5%, HOG+moment, Haar5%+moment, all the features. The resultant precision and recall values with respect to the value of T, were shown in Fig.5.

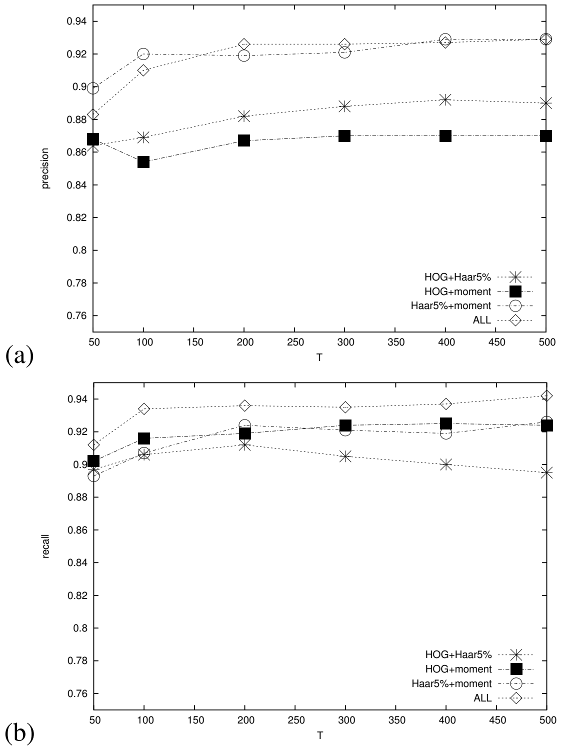

Figure 5. Classification performance by the stump classifiers with combination of feature types : (a) precison, (b) recall

As the figure shows, with the increase of feature types, classification accuracy is improved. Apparently, the performance of sparse Haar and HOG based classifiers was improved by adding the moment features. Comparing sparse Haar (Haar 5%) (see Fig.4), sparse Haar + HOG, and sparse Haar + moment, it is suggested that combining different kinds of information (i.e. edge + intensity distribution) is more effective than combining similar kinds of information (edge + orientation distribution). Among all the classifiers treated in this article, combined stump classifiers based on all the features (Haar 5%, HOG, moment) performed best. As the experimental results show, even if the number of decision stumps is fairly small (around 200), the performance of the best classifier was very good.

5. Conclusions

In this article, we have proposed a method for designing classifier for extracting characters on signboards in natural scene images. For wide range of applications, both classification accuracy and computational cost (speed) are important. To discriminate character regions from non-character regions, robust and efficient set of features is needed. In our work, we examined effectiveness of moment statistics of intensity distribution along with frequently used edge based features : Haar wavelet and HOG. To combine all of these features, we use boosting technique. For simplicity and computational cost, we have used stump classifier, which relies on a single component in the multidimensional feature vector as a base learner. Experimental results show the combination of stump classifiers based on various kinds of features can perform very well.

References

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

[2] Y. Freund and R. Shapire. A decision theoretic generalization of on-line learning and application to boosting. *J. of Computer and System Sciences*, 55(1):119–139, 1997.

[3] J. Gao, J. Yang, Y. Zhang, and A. Waibel. Text detection and translation from natural scenes. Technical Report CMU-CS-01-139, School of Computer Science, Carnegie Mellon University, 2001.

[4] B. Heisele, T. Serre, S. Prentice, and T. Poggio. Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognition*, 36(9):2007–2017, 2003.

[5] T. Joachims. Making large-scale svm learning practical. In *Advances in kernel methods*, chapter 11. MIT Press, 1999.

[6] H. Li and D. Doermann. Automatic identification of text in digital video key frames. *Proc. ICPR'98*, pages 129–132, 1998.

[7] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International J. of Computer Vision*, 38(1):15–33, 2000.

[8] R. Shapire. The boosting approach to machine learning : An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, and B. Yu, editors, *Nonlinear estimation and classification*. Springer, 2003.

[9] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. Wavelets for computer graphics: A primer, part 1. *IEEE Computer Graphics and Applications*, 15(3):76–84, 1995.

[10] P. Viola and M. Jones. Rapid object detection using a boosted casdade of simple features. *Proc. CVPR*, pages 511–518, 2001.

[11] T. Yamaguchi and M. Maruyama. Character extraction from natural scene images by hierarchical classifiers. *Proc. ICPR2004*, 2:687–690, 2004.