Feature Representations for Scene Text Character Recognition: A Comparative Study

Chucai Yi[1], Xiaodong Yang[2], Yingli Tian[1,2]
[1]Dept. of Computer Science, The Graduate Center, City University of New York, USA
[2]Dept. of Electrical Engineering, The City College, City University of New York, USA
{cyi@gc.cuny.edu, xyang02@ccny.cuny.edu, ytian@ccny.cuny.edu}

Abstract— **Recognizing text character from natural scene images is a challenging problem due to background interferences and multiple character patterns. Scene Text Character (STC) recognition, which generally includes feature representation to model character structure and multi-class classification to predict label and score of character class, mostly plays a significant role in word-level text recognition. The contribution of this paper is a complete performance evaluation of image-based STC recognition, by comparing different sampling methods, feature descriptors, dictionary sizes, coding and pooling schemes, and SVM kernels. We systematically analyze the impact of each option in the feature representation and classification. The evaluation results on two datasets CHARS74K and ICDAR2003 demonstrate that Histogram of Oriented Gradient (HOG) descriptor, soft-assignment coding, max pooling, and Chi-Square Support Vector Machines (SVM) obtain the best performance among local sampling based feature representations. To improve STC recognition, we apply global sampling feature representation. We generate Global HOG (GHOG) by computing HOG descriptor from global sampling. GHOG enables better character structure modeling and obtains better performance than local sampling based feature representations. The GHOG also outperforms existing methods in the two benchmark datasets.**

Keywords—scene text character recognition, performance evaluation, text feature representation, feature descriptors, Global HOG, dictionary of visual words, coding-pooling

I. INTRODUCTION

Text characters and strings in natural scene provide straightforward and unambiguous information on special target and ambient environment. Text extraction from scene images plays a significant role in many image/video-based applications, such as context retrieval, assistant navigation [20], aid reading [23] and scene understanding. However, scene text extraction is a challenging problem due to cluttered background and a great variety of text patterns, fonts, and scales. In camera-captured scene images, text characters and strings normally have a low frequency of occurrence and small occupied areas and appear in multiple fonts, sizes, colors, and orientations.

In general, scene text extraction methods [25] can be divided into two main processes: detection and recognition. Text detection is to localize the image regions containing text information, and filter out most non-text background outliers [7, 21, 22]. Text recognition is to generate readable text codes in the form of words and phrases from the detected text regions. This paper focuses on text character recognition.

Most off-the-shelf Optical Character Recognition (OCR) systems are designed to work on scanned document images with relatively clean background and uniform text patterns, and they could not obtain good recognition performance on text regions of scene images. Text recognition is implemented by STC segmentation and STC recognition. STC segmentation partitions a detected text region into multiple image patches, each of which contains only one text character (see Fig. 1). STC recognition is a multi-class classification within pre-defined sample space, predicting recognized code from extracted features of a character patch. In previous text recognition algorithms, STC segmentation and recognition were processed in three ways [3]. First, character-like properties were defined to dissect text regions into candidate patches, where STC recognition was then applied to. Second, text regions were densely searched for text component with high confidence score of one character class, and the confidence score was obtained from STC recognition. Third, lexical analysis is applied to directly infer the whole words from confused STC recognition within text regions. Above methods show that STC recognition would always play a significant role in text word recognition. Therefore, accuracy improvement of STC recognition will result in better performance of word/phrase recognition in text regions.

A variety of feature representations for STC recognition were proposed. In [19], Gabor filter responses were employed to extract features of character appearance. Then word recognition was performed by combining character recognition with language, similarity and lexicon model. In [14], similarity expert was built from SIFT descriptors to compute the character similarity, and integer program based on character similarity was applied for word recognition. In [17], HOG descriptors were densely extracted and cascaded as feature representations of character patches, and normalized cross correlation analysis of character similarity was used for STC recognition. In [18], Random Ferns algorithm was adopted for character detection, and pictorial structures with lexicon model were employed for word configuration and recognition. In [11], HOG feature was extracted for character recognition conditional random field was adopted to combine character detection and word-level lexicon analysis. In [4], local features of character patches were extracted by an unsupervised learning method related to a variant of K-means clustering, and spatially pooled by cascading sub-patch features. In [12], feature extraction for STC recognition was generated from Maximally Stable Extremal Regions (MSER), which is split into 8 levels by MSER boundary orientations. In [26], STC recognition for Chinese, Japanese and Korean characters was performed by Scale Invariant Feature Transform (SIFT) feature matching to template character patches, in which a voting and geometric verification algorithm was designed to remove false positive matches. However, most previous algorithms considered STC recognition as a small component of the whole framework of scene text information extraction. They focused more on lexicon-based word configuration and recognition, without complete quantitative analysis of image-based feature representation. However, most word-level processing depends on the results of character recognition, e.g., prediction score of character classifier. In this paper, we present performance evaluations on STC recognition under a general framework of object recognition, which consists of two processes: feature representation and multi-class classification.

Most of the existing feature representations for STC recognition are generated from local sampling. In this paper, we apply global sampling to STC recognition and obtain better performance compared with local-sampling based feature representations. To obtain feature representation from local sampling, we detect key-points, compute local descriptors, build dictionary of visual words, and perform feature coding and pooling to obtain a histogram of visual words, i.e., bag-of-words. To obtain feature representation from global sampling, we compute descriptor directly from the whole character patch without processing key points, dictionary, coding or pooling. In both ways, each character patch is mapped into a feature vector, which is regarded as an input point into training and prediction process of multi-class classification. To learn a robust character classifier, we adopt the state-of-the-art SVM learning model. Fig. 1 depicts the flowchart of feature representation from local and global sampling. These methods will be described in detail in Section III. According to STC recognition performance, we evaluate the local sampling based feature representation in Section IV and global sampling based feature representation in Section V. In addition, Section II introduces the public datasets in our experiments and Section VI will analyze our evaluation results and make a conclusion.

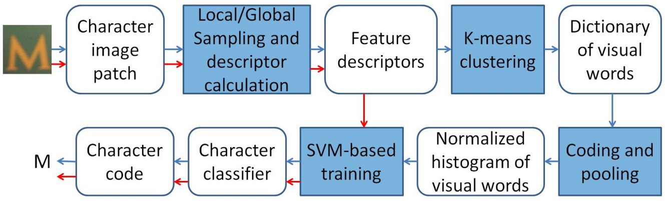

Figure 1. Flowchart of STC recognition framework in performance evaluation, where the shadowed boxes denote the processing steps with multiple options and the white boxes denote the outputs of the processing steps. The blue arrows denote the processing of local sampling based feature representation, and the red arrows denote the processing of global sampling based feature representation.

II. Scene Text Character Datasets

Many datasets have been released for the research work on scene text detection and recognition. They are composed of natural scene images with text information in complex background. The image regions containing text are provided as ground truth labels. However, most datasets only label word-level ground truth regions, without dissecting them into image patches of single characters. To evaluate STC recognition, we employ two public datasets composed of image patches of single characters, CHARS74K [6] and ICDAR2003 [10].

CHARS74K dataset consists of three types of text characters, image-based characters cropped from natural scene image, hand-written characters, and computer-generated font characters. The first type is compatible with our STC recognition task. It contains 62 character classes, i.e., digits *0~9*, English letters in upper case *A~Z*, and lower case *a~z*. The 62 character classes have nearly balanced numbers of character patches.

ICDAR2003 dataset is prepared for the Robust Reading Competition of scene text detection and recognition. It has 509 scene images and 2268 word-level text regions. The text regions are partitioned into 11615 image patches of characters. There are 6185 training patches and 5430 testing patches. They cover all the 62 character classes, but the numbers of character patches between different classes are imbalanced.

III. Feature Representations from Local Sampling

A. Local Sampling and Global Sampling

Local features are extracted from character patches to describe appearance and structure of STCs from all 62 classes. In our work, dense sampling is performed to detect key-points for STC because it covers more complete character structure, than sparse interest point detectors that are widely used in image matching. Given a character patch, we first resize it into a square patch whose width equals to height, and then extend the side length into the nearest power of 2 (e.g., 128× 128, 256×256). Next, in an $L \times L$ character patch, sub-patch in $(L/2) \times (L/2)$ is generated as feature window to extract feature descriptor, sliding from top-left to horizontal and vertical directions. The center of a sub-patch is regarded as a key-point, and the stride of two neighboring key-points is $L/8$ in both directions. Since L is a power of 2, we obtain $[(L/2)/(L/8)+1] \times [(L/2)/(L/8)+1] = 25$ key points from a character patch. Each key-point regards the sub-patch as its support region and generates a feature descriptor $x \in R^M$ where M denotes the dimensions. In this process, key point locations and sub-patch sizes are determined by the character patch size L, so this local sampling method can be adaptive to scale changes of character patches.

In global sampling, the whole character patch is used as a feature window to extract features. It skips key point detection, coding and pooling process to reduce information loss.

B. Feature Descriptors

In global sampling, only Histogram of Oriented Gradient (HOG) descriptor [5] is extracted as character structure features. In local sampling, besides HOG, we adopt 5 other state-of-the-art feature descriptors which have been extensively used in the general visual recognitions, including SIFT [9], Speed Up Robust Features (SURF) [1], DAISY [15], Binary Robust Independent Elementary Features (BRIEF) [2], and Oriented Fast and Rotated BRIEF (ORB) [13]. Previous work has demonstrated their effectiveness on object, texture, and scene recognitions.

In global (or local) sampling HOG, block size is set to be half of patch (or sub-patch) size, and block stride is set to be half of block size. Each block contains 4×4 cells, and the bin number of gradient orientations is 9. The other 5 descriptors are implemented by default parameters in public available source code and OpenCV2.4. We tried to tune the parameters, but did not obtain any apparent improvement in CHARS74K and ICDAR2003 datasets.

C. Dictionary Sizes in Local Sampling

Feature descriptors are extracted from sampled key-points of character patches, and we apply K-means clustering to build dictionary $D \in R^{M\times K}$, where M denotes the dimension of feature descriptors and K denotes the number of visual words. We use d_j to denote the j-th visual word. The dense sampling extracts 25 key-points from each character patch, so we generate total of 23250 feature descriptors from 930 training patches in CHARS74K and 154625 feature descriptors from 6185 training patches in ICDAR2003. To evaluate the impact of dictionary size on STC recognition, each type of feature descriptor at each dataset generates 5 dictionaries in different sizes. According to the number of feature descriptors from the datasets, we set the dictionary sizes to be 500, 1000, 2000, 3000, and 5000 respectively.

D. Coding and Pooling Schemes in Local Sampling

A number of feature descriptors $\{x_i \mid 1 \le i \le N\}$ are extracted from a character patch where N denotes the total number of descriptors. They are mapped into a histogram of visual words by coding and pooling [8]. The coding process is used to map each feature descriptor x_i into a histogram of visual words c_i based on the dictionary D. The state-of-the-art coding schemes include Hard Assignment (HARD), Soft Assignment (SOFT), and Sparse Coding [24] (SC) as (1) in top-down order. The parameter β and γ in SOFT and SC are used to control softness and sparseness respectively. The pooling is employed to aggregate coded features c_i into the final bag-of-words feature representation p . The popular pooling schemes include Average Pooling (AVE) and Max Pooling (MAX) as (2) in top-down order.

$$c_{ij} = \begin{cases} 1 & if\ j = \underset{j=1,\dots,n}{\operatorname{argmin}} \|x_i - d_j\|_2^2 \\ 0 & otherwise \end{cases}$$

$$c_{ij} = \frac{\exp\,(\beta\|x_i - d_j\|_2^2)}{\sum_{k=1}^{N} \exp\,(\beta\|x_i - d_k\|_2^2)} \qquad (1)$$

$$c_i = \underset{c}{\operatorname{argmin}} \|x_i - Dc\|_2^2 + \gamma\|c\|_1$$

$$p_j = (1/N)\sum_{i=1}^{N} c_{ij} \qquad (2)$$

$$p_j = \max_i c_{ij}$$

IV. Performance Evaluations of Feature Rrepresentations from Dense Local Sampling

The options in Table I are used to evaluate STC recognition, which is measured by the average accuracy rate, i.e., the ratio of correctly predicted STCs in the testing set. For fair comparisons, in Chars74K, we refer to the training/testing splits in [27] to perform Chars74K-15 evaluation. In ICDAR2003, we use training patches of its own [28] to learn character classifier without adding any other training samples, and then evaluate the classifier in its testing patches.

TABLE I. OPTIONS OF FEATURE REPRESENTATION AND SVM MODEL

Sampling	Local Sampling; Global Sampling
Descriptors	HOG; SIFT; SURF; BRIEF; ORB; DAISY
Dictionary sizes	500; 1000; 2000; 3000; 5000
Coding-Pooling schemes	HARD-AVE;SOFT-AVE;SOFT-MAX;SC-MAX
SVM kernels	Linear; Chi-Square χ^2

A. Evaluating Feature Descriptors

Fig. 2 demonstrates performance evaluations of the 6 types of feature descriptors under local sampling. Under HARD-AVE scheme, SURF obtains the best performance and HOG obtains the close second best. Under SOFT-AVE and SOFT-MAX schemes, HOG obtains the best performance.

BRIEF and ORB could obtain good performance in recognizing texture-rich object. The simple binary tests between pixels in a local support region in BRIEF and ORB is not well adapted to character recognitions because binary tests from uniform intensity regions (frequent in character patches) are not able to provide sufficient discriminative information. Fig. 2 also shows that SOFT-MAX scheme obtains better performance than SOFT-AVE and HARD-AVE schemes. We will describe detailed evaluations in Section IV.C.

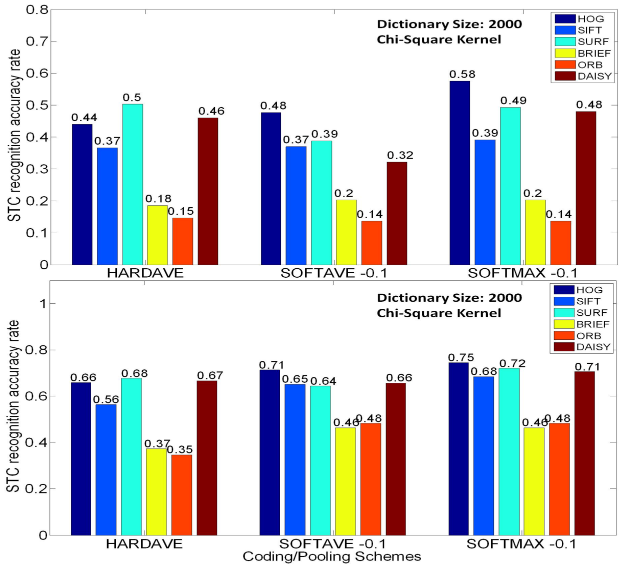

Figure 2. Performance evaluations of 6 types of feature descriptors under three coding/pooling schemes, dictionary size 2000 and Chi-square SVM kernel. The value -0.1 in horizontal axis denotes $\boldsymbol{\beta}$ value in (1). The top figure shows results from CHARS74K and the bottom figure shows results from ICDAR2003.

B. Evaluating Dictionary Sizes

The evaluation results in Fig. 3 show the relationship between STC recognition rate and the dictionary size. When the dictionary size is less than 2000, the performance is increased along the dictionary size change. However, growth will saturate when the dictionary size reaches a certain level. Then the performance will keep approximately consistently or slightly decease. Compared with general object recognition in multiple scales and view angles in scene image, STC in image patches have relatively stable structure since the scale has been normalized with the STC patch size and the view angle does not largely change STC appearance. Thus this amount of visual words is sufficient to represent local features extracted from STC, and growth saturation of STC recognition performance on dictionary size is reached more rapidly.

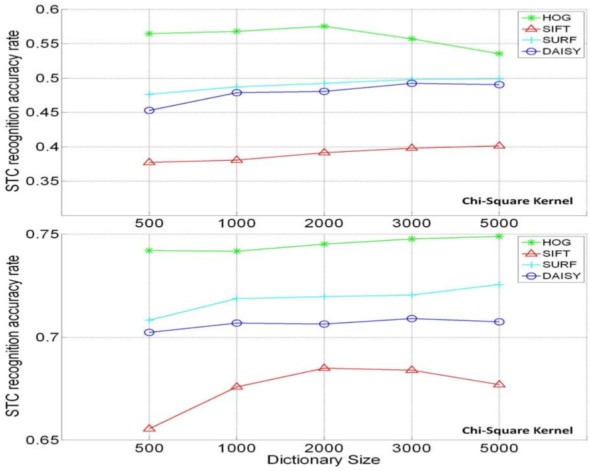

Figure 3. Performance evaluations of 5 dictionary sizes under four feature descriptors and Chi-Square SVM kernel. Top figure results from CHARS74K and bottom figure results from ICDAR2003.

C. Evaluating Coding-Pooling Schemes

Fig. 4 depicts the performance evaluations of different coding and pooling schemes in Table I.

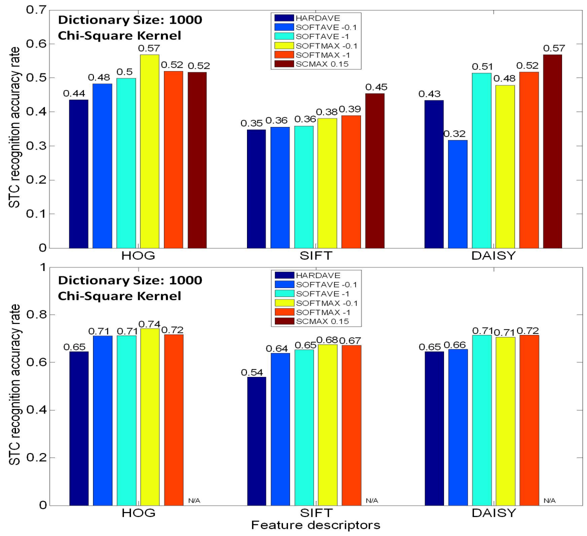

Figure 4. Evaluating 6 coding/pooling schemes, under three feature descriptors, dictionary size 1000, and Chi-Square SVM kernel. Top figure is from CHARS74K and bottom figure is from ICDAR2003.

In SOFT, we obtain two groups of results by setting the parameter β in (1) as -0.1 and -1 respectively. In SC, we set the parameter γ in (1) as 0.15. Currently, SC is not applied to ICDAR2003 because of its high computational cost in coding optimization. In coding schemes, SOFT and SC are comparable, and both obtain better performance than HARD. As shown in (1), this is probably because the extreme sparseness of codes generated by HARD (only one coefficient per code is non-zero) might be ill-suited to character images, and SOFT and SC loose the constraint to alleviate information lose. In pooling schemes, MAX always obtains better performance than AVE, because maximum value usually contains the most significant information and its statistical properties make it well adapted to sparse representations.

D. Evaluating Classification Model

Besides the design of STC feature representation, the choice of classification model plays an important role in STC recognition. The feature vector of a character patch, which is a histogram of visual words, is regarded as an observation point in classification model. Currently, all the experimental results of STC recognition are obtained from SVM learning models with linear kernel and χ^2 kernel [16]. In future work, other learning models such as random forest and naive Bayes nearest neighbor will be evaluated on STC recognition.

V. Performance Evaluations of Feature Representations Based on Global Sampling

A. Global HOG

Above experiments are based on dense sampling of key-points in character patch. Feature descriptor is computed from each key point and then normalized and pooled through coding-pooling schemes. In this section, we propose a new feature representation for STC recognition based on global sampling. We extract GHOG features directly from the whole character patch. Compared to local sampling, the GHOG based global sampling has the following advantages: (1) there is no coding so no information loss (2) spatial structure is preserved when concatenating descriptors of grids in order. The evaluation results show that GHOG obtains accurate rate up to 0.62 at CHARS74K and 0.76 at ICDAR2003, which are better than the highest results (0.58 at CHARS74K and 0.75 at ICDAR2003) in local sampling. In addition, GHOG outperforms most existing methods as shown in Table II.

B. Incomplete Patch and Preprocessed Patch

STC recognition is usually based on the resulting character patches from text region detection and STC segmentation. However, the two steps cannot ensure complete character patches. We evaluate the performance of GHOG on these incomplete (truncated) character patches and illustrate the results in Fig. 5. It shows that more complete structure obtains better recognition performance, and the top and bottom parts of character patch generate more discriminative structure features than the left and right parts.

Moreover, we apply Gaussian and Laplacian filters to preprocess character patches before extracting GHOG features. Gaussian filter removes background noise, while Laplacian filter emphasizes the boundary that contains much information on character structure. The evaluation results (see Fig. 5) show that Gaussian smooth improves the recognition performance slightly, but Laplacian lowers the performance because the noise negatively influences HOG descriptors.

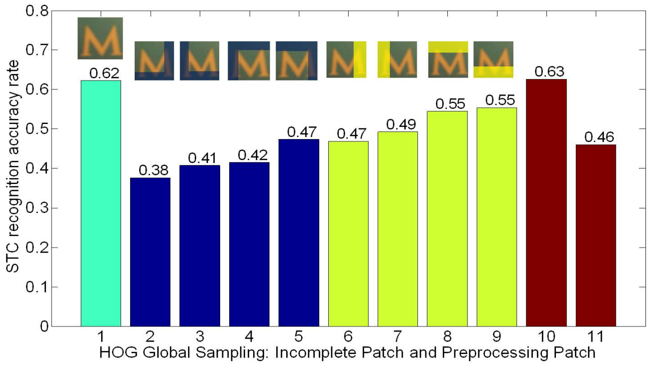

Figure 5. Performance evaluations based on GHOG global sampling. The left first bar denotes the accurate rate of global HOG in complete and original patch. Blue bars and yellow bars denote accuracy rate of incomplete patches as their top examples. The first red bar denotes accuracy rate (0.63) of preprocessed patches by Gaussian and the second red bar denotes that of Laplacian filters (0.46).

TABLE II. COMPARATIONS BETWEEN OUR BEST RESULTS AND EXISTING METHODS OF STC RECOGNITION OVER THE BENCHMARK DATASETS.

	CHARS74K-15	ICDAR2003CH
Global HOG+SVM	**0.62**	**0.76**
Local HOG+SVM	**0.58**	**0.75**
Geometrical blur + NN[6]	0.47	/
Geometrical blur + SVM[6]	0.53	/
Shape context + NN[6]	0.34	/
Shape context + SVM[6]	0.35	/
Multiple kernel learning [6]	0.55	/
ABBYY [6]	0.31	/
Coates method [4]	/	0.82
HOG+NN [18]	0.58	0.52
SYNTH+FERNS [18]	0.47	0.52
NATIVE+FERNS [18]	0.54	0.64

NN: Nearest neighbor classification; SYNTH: synchronic patch for training; NATIVE: native scene image patch for training.

VI. CONCLUSIONS

We have evaluated STC recognition performance of several types of feature representations. The evaluation results demonstrate that GHOG achieves better performance on STC recognition than other feature descriptors. Also our results confirm that soft-assignment performs better than hard-assignment and maximum pooling performs better than average pooling over STC recognition. We employ GHOG to achieve significant improvement over local feature representations and existing state-of-the-art methods.

In future work, we will design more robust and effective feature representations to improve STC recognition performance, and will extend the scene text recognition from character into word level recognition.

ACKNOWLEDGEMENT

This work was supported in part by NSF grant IIS-0957016, EFRI-1137172 and FHWA grant DTFH61-12-H-00002.

REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, L. Gool, "SURF: Speeded Up Robust Features, " *CVIU*, 2008.

[2] M.Calonder, V.Lepetit, M.Ozuysal, T.Trzcinski, C.Strecha and P.Fua, "BRIEF: Computing a Local Binary Descriptor Very Fast," *TPAMI*, 2012.

[3] R. Casey. "A survey of methods and strategies in character segmentation," *IEEE Transactions on PAMI*, 1996.

[4] A.Coates,B.Carpenter,C.Case,S.Satheesh,B.Suresh,T.Wang,D.Wu,A.Ng, "Text detection and character recognition in scene images with unsupervised feature learning," *ICDAR* 2011

[5] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *CVPR*, 2005

[6] T. De-Campos, , B. Babu,, and M. Varma, "Character recognition in natural images," *VISAPP*, 2009

[7] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," *CVPR*, 2010.

[8] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," *ICCV* 2011

[9] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV,* 2004.

[10] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R.Young, "ICDAR 2003 Robust Reading competitions," *ICDAR*, 2003

[11] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," In CVPR 2011.

[12] L.Neumann, J. Matas, "A method for text localization and detection," *ACCV* 2010

[13] E. Rublee, V. Rahbaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *ICCV*, 2011.

[14] D. Smith, J. Feild, and E. Learned-Miller, "Enforcing Similarity Constraints with Integer Programming," *CVPR,* 2011.

[15] E. Tola, A. Fossati, C. Strecha, and P. Fua, "large occlusion completion using normal maps," *ACCV,* 2010.

[16] A. Vedaldi, and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. on PATMI*, 2011.

[17] K. Wang, and S. Belongie, "Word spotting in the wild. ,"*ECCV*, 2010.

[18] K. Wang, B. Bbenko, and S. Belongie, "End-to-End scene text recognition.," *ICCV*, 2011.

[19] J. Weinman, E. Learned-Miller, and A. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," *IEEE Transactions on PAMI*, 2009.

[20] X. Yang, Y. Tian, C. Yi, and A. Arditi, "Context-based Indoor Object Detection as an Aid to Blind Persons Accessing Unfamiliar Environments," *ACM Multimedia,* 2010.

[21] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting Texts of Arbitrary Orientations in Natural Images," *CVPR*, 2012.

[22] C. Yi, and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE TIP*, 2011.

[23] C. Yi and Y. Tian. Assistive Text Reading from Complex Background for Blind Persons. In *ICDAR Workshop on CBDAR*, Springer LNCS, 2011

[24] K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," *CVPR*, 2009.

[25] J. Zhang, and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress," *DAS*, 2008

[26] Q. Zheng, K. Chen, Y. Zhou, G. Cong, H. Guan, "Text localization and recognition in complex scenes using local features," *ACCV* 2010

[27] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

[28] http://algoval.essex.ac.uk/icdar/Datasets.html.