LETTER

Learning Co-occurrence of Local Spatial Strokes for Robust Character Recognition*

Song GAO[†], ***Student Member***, **Chunheng WANG**[†a)], ***Member***, **Baihua XIAO**[†], **Cunzhao SHI**[†], **Wen ZHOU**[†], ***and*** **Zhong ZHANG**[†], ***Nonmembers***

SUMMARY In this paper, we propose a representation method based on local spatial strokes for scene character recognition. High-level semantic information, namely co-occurrence of several strokes is incorporated by learning a sparse dictionary, which can further restrain noise brought by single stroke detectors. The encouraging results outperform state-of-the-art algorithms.

key words: *robust character recognition, local spatial stroke, co-occurrence, sparse dictionary*

1. Introduction

Text information contained in scene images is very helpful for image understanding. A robust scene-text-extraction system could be used in many areas, like image retrieval, intelligent transportation, robot vision, etc. Text detection and text recognition are performed sequentially to extract scene text information. In the text detection stage, regions containing scene texts are localized from entire images. Afterwards, text recognition block performs scene character recognition based on cropped text blocks. Language models are often incorporated to generate whole-word recognition results. We focus on scene text recognition, especially scene character recognition in this paper.

In general, scene text recognition methods can be divided into two categories: traditional Optical Character Recognition (OCR) based methods and object recognition based methods. Relying on the highly developed OCR techniques, OCR based methods [1], [2] focus on binarizing scene text blocks before feeding binarized texts into the off-the-shelf OCR engines. However, traditional OCR techniques are designed for clean scanned documents while binarization of scene text blocks is very difficult due to low resolutions, different illumination conditions and complex backgrounds as shown in Fig. 1 (a). Object recognition based methods [3]–[5] skip the binarization step and regard each kind of scene character as a special object. These methods usually extract features from one image patch which is considered as containing only one character and then feed the features into various classifiers to obtain a class label. Moreover, language models are often incorporated to get whole-word recognition results. Wang et al. [4] choose random ferns as the classifier and use pictorial structures to model lexicons. Mishra et al. [5] utilize multi-class SVMs to recognize scene characters and CRF is used to take all detection results into consideration to form the final words. Recently, object recognition based methods are inspiring more and more enthusiasm from the computer vision community for their effectiveness and robustness.

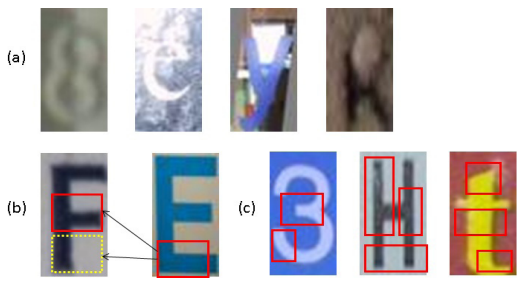

Fig. 1 Our motivation: (a) scene characters are not easy to binarize due to low resolutions, different illumination conditions and complex backgrounds; (b) discriminative part of 'E' (in red rectangles) appearing in another location of 'F'; (c) co-occurrence of several local spatial strokes which may be discriminative (in red rectangles).

Object bank is firstly proposed in [6] to generate high-level image representation for scene categorization. It uses maximal output of base detectors sliding at multi-scales on one whole image as features for second-layer classifiers. When different strokes are regarded as different objects, we can build a stroke bank for character recognition. If a stroke bank is applied to scene character recognition directly, multi-scales sliding window search means every stroke detector in the bank should be applied to classify different sizes of windows in global image range. That may not be necessary as strokes are highly spatially related. Actually, global search may result in classification confusion. It is because one part of a character may appear in a different location of another character, such as 'E' and 'F' like Fig. 1 (b). Besides, the original stroke bank is only able to model appearance of single strokes. However, character strokes are highly correlated which means co-occurrence of several strokes may be more discriminative for classification as shown in Fig. 1 (c).

In this paper, we proposes to build a stroke bank for scene character recognition. All character training images

Manuscript received December 2, 2013.
Manuscript revised March 7, 2014.

[†]The authors are with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China.

*This work is supported by the National Natural Science Foundation of China under Grant No.61172103, No.60933010, State 863 Project under Grant No.2012AA041312.

a) E-mail: chunheng.wang@ia.ac.cn (Corresponding author)

DOI: 10.1587/transinf.E97.D.1937

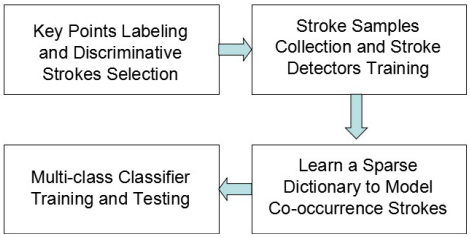

Fig. 2 Overview of the proposed method.

are labeled manually in order to collect stroke training samples for stroke detectors. To overcome the two drawbacks of object bank mentioned above, we mainly make two contributions: (1) the response regions for one stroke detector are limited to positions of positive training samples, which alleviates computation burden and retains discrimination power at the same time; (2) rather than training second-layer classifiers directly on stroke detectors' outputs, a sparse dictionary is further learned to model co-occurrence of several strokes. Then reconstruction coefficients are used as final feature vectors. Experiments on two public datasets ICDAR2003 and CHARS74K prove the effectiveness of our method and the results outperform state-of-the-art algorithms.

The paper is organized as follows. In Sect. 2, overview of the proposed method is given. Section 3 details every stage of the system. Experiments are given in Sect. 4 and conclusions are drawn in Sect. 5.

2. Framework

The proposed method consists of four parts: (1) labeling key points for character training images and choosing discriminative strokes for every character; (2) collecting stroke training samples and training stroke detectors; (3) learning a sparse dictionary to model co-occurrence of several strokes; (4) multi-class classifier training and testing. The overall framework is given in Fig. 2.

3. Proposed Method

3.1 Key Points Labeling and Discriminative Strokes Selection

To collect training samples for stroke detectors in the next section, we propose to label key points for every training images of all character categories. Then once a desired stroke is selected on one character training image, the same strokes can be extracted from other training images of the same character category automatically based on labeled key points. The procedure is as follows.

Before key points' labeling, all training images are scaled to size $H*W$ using bilinear normalization. Key points are designed for every kind of scene character as shown in Fig. 3.

When selecting key points for class c_i (i $\in \{1, 2, 3, \ldots, N_c\}$, N_c is the number of character categories),

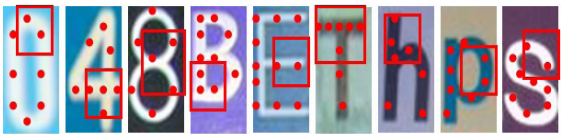

Fig. 3 Labeled characters (red points) and selected stroke structures (red rectangles).

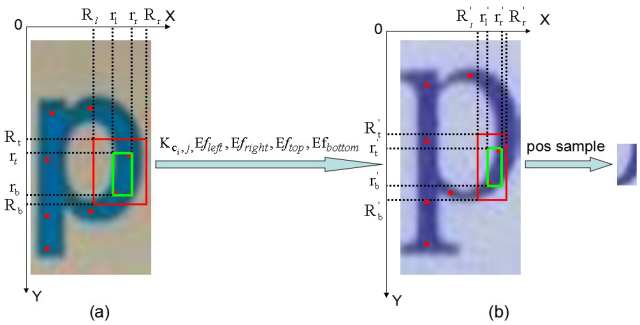

Fig. 4 Collecting positive samples for a stroke detector: (a) recording key points and extending factors for selected stroke on a training image; (b) collecting positive strokes from other training images of the same category based on recorded key points and extending factors. Green rectangles are for key points bounding and the red rectangle are positive stroke patches.

labeled key points should cover the main structures of characters as shown in Fig. 3. Then based on these densely labeled key points, we can select discriminative strokes from character c_i as desired. The numbers of labeled key points for 0-9, A-Z and a-z are 8, 6, 8, 9, 8, 10, 8, 7, 7, 8, 10, 10, 7, 7, 9, 9, 9, 11, 9, 6, 9, 6, 9, 7, 8, 8, 10, 10, 9, 8, 7, 7, 9, 9, 7, 9, 9, 8, 7, 8, 8, 8, 8, 7, 4, 6, 9, 5, 7, 8, 8, 8, 6, 7, 9, 9, 8, 7, 9, 9, 8 and 9 respectively. For any training images from any character categories, key points are labeled manually.

For every character c_i, we choose N_{stroke,c_i} discriminative strokes, namely selecting N_{stroke,c_i} rectangles on one training image belonging to c_i like Fig. 3. The picked training image can be a random one, because after strokes' selection, the same strokes will be extracted from all training images of c_i in the next section. When selecting the N_{stroke,c_i} rectangles bounding certain strokes, discriminative strokes that can tell class c_i from other classes should be chosen. For example, if one stroke only belongs to c_i, this stroke should be selected. To help extracting the same strokes from other training samples of c_i, the following information needs to be recorded for every selected stroke structure $Stroke_{c_i,j}$ as shown in Fig. 4 (a) (i $\in \{1, 2, 3, \ldots, N_c\}$, j $\in \{1, 2, 3, \ldots, N_{stroke,c_i}\}$):

(1) key points falling into the selected rectangle R (assume coordinates of left, right, top and bottom boundary are R_l, R_r, R_t and R_b respectively) denoted by $K_{c_i,j}$;

(2) calculate the minimal rectangle r bounding the above key points (assume coordinates of left, right, top and bottom boundary are r_l, r_r, r_t and r_b respectively; then width is $r_{width} = r_r - r_l + 1$ and height is $r_{height} = r_b - r_t + 1$.);

(3) compute and record extending factors of left, right, top and bottom as: $Ef_{left} = (r_l - R_l + 1)/r_{width}$, $Ef_{right} =$

$(R_r - r_r + 1)/r_{width}$, $Ef_{top} = (r_t - R_t + 1)/r_{height}$ and $Ef_{bottom} = (R_b - r_b + 1)/r_{height}$.

3.2 Stroke Samples Collection and Stroke Detectors Training

Based on the labeled key points and reserved information of selected strokes, stroke samples are collected. Then stroke detectors are trained using these extracted samples.

When collecting a positive training sample for detector $Stroke_{c_i,j}$ on a training image from character c_i, we firstly locate the minimal rectangles r' (coordinates of left, right, top and bottom boundaries are r'_l, r'_r, r'_t and r'_b; then width is $r'_{width} = r'_r - r'_l + 1$ and height is $r'_{height} = r'_b - r'_t + 1$) containing key points $K_{c_i,j}$ and then calculate coordinates of the positive stroke sample as follows: $R'_l = r'_l + 1 - Ef_{left} * r'_{width}$, $R'_r = r'_r - 1 + Ef_{right} * r'_{width}$, $R'_t = r'_t + 1 - Ef_{top} * r'_{height}$ and $R'_b = r'_b - 1 + Ef_{bottom} * r'_{height}$.

When this positive stroke sample is extracted as in Fig. 4 (b), $N_{neg/pos}$ negative patches are extracted from random training images of random remaining character categories with the same patch coordinates (R'_l, R'_r, R'_t, R'_b). It should be noted that positions of all positive samples need to be reserved for every stroke detector $Stroke_{c_i,j}$ forming a set denoted by $Area_{c_i,j}$. Every member in $Area_{c_i,j}$ is a positive sample's coordinates denoted as (R'_l, R'_r, R'_t, R'_b).

All positive and negative stroke samples are scaled to $h_s * w_s$. Based on these samples, stroke detectors are trained. HOG feature [7] calculate histogram of oriented gradients to represent objects' shapes and has been widely used by the computer vision community for its efficiency and effectiveness. Here, HOG [7] is extracted for every stroke samples, and linear SVM [8] is used as the detector style for its simplicity. Totally we get $N_s = \sum_{i=1}^{N_c} N_{stroke,c_i}$ stroke detectors in the stroke bank, in which each member corresponds to a response region set $Area_{c_i,j}$.

3.3 Learning a Sparse Dictionary to Model Co-occurrence Strokes

For every training image of a character, each detector in the stroke bank is applied to classify its corresponding response regions $Area_{c_i,j}$. The maximal value outputed by the corresponding linear SVM is denoted as $Out_{c_i,j}$. Then confidence vector for this training image is as:

$$f = (Out_{c_1,1}, \ldots, Out_{c_1,N_{stroke,c_1}}, \ldots, Out_{c_{N_c},1}, \ldots, Out_{c_{N_c},N_{stroke,c_{N_c}}}) \quad (1)$$

Restricting detectors' classification areas to positions of positive stroke samples can alleviate computation burden. Besides, it also retains more discrimination power. It's because when one part of a character appears in a different location of another character (like 'E' and 'F' in Fig. 1 (b)), global search may result in classification confusion. Similar conditions may happen between 'L' and 'E', 'X' and 'Y', 'V' and 'W' and so on.

If confidence vectors f are directly fed into multi-class linear SVMs [8] for training, we are only able to model single stroke appearing in one position while failing to model co-occurrence of several structures in different locations. Intuitively, modeling co-occurrence of different stroke structures can introduce high-level semantic information and may restrain noise brought by single stroke detectors. So it's appealing to find an appropriate way to model co-occurrence strokes.

To model co-occurrence of several strokes, we propose to learn a sparse dictionary $D = [d_1, d_2, \ldots, d_{N_D}] \in R^{N_s \times N_D}$ (N_s is the number of stroke detectors and N_D is the dictionary size) based on the confidence vectors f using elastic net [9] as in [10]. The learned dictionary D should be sparse. It means most entries of $d_i \in R^{N_s}$ are zeros. The non-zero entries often correspond to co-occurrence of some strokes, which will be demonstrated in the experiment section.

Given a set of training images represented as confidence vectors $F = \{f_1, f_2, f_3, \ldots, f_i, \ldots, f_N\}$ ($f_i \in R^{N_s}$ and N is the number of training images), sparse reconstruction coefficient w_i is learned simultaneously when learning sparse dictionary D as in Eq. (2). Then given an image represented as confidence vector f, sparse coefficient w is computed so that f can be reconstructed from D like Eq. (3).

$$\min_{D \in C, W \in R} \sum_{i=1}^{N} \left(\frac{1}{2} \|f_i - Dw_i\|_2^2 + \lambda \|w_i\|_1\right) \quad (2)$$

$$\min_{w \in R^{N_D}} \frac{1}{2} \|f - Dw\|_2^2 + \lambda \|w\|_1 \quad (3)$$

where $W = [w_1, w_2, \ldots, w_N] \in R^{N_D \times N}$, λ is a regularization parameter and C is the convex set which D belongs to. The convex set C can be constructed as follows:

$$C = \{D \in R^{N_s \times N_D}, s.t. \forall i, \|d_i\|_1 + \frac{\gamma}{2} \|d_i\|_2^2 \leq 1\} \quad (4)$$

The sparsity requirement of dictionary D is achieved by enforcing l_1-norm and l_2-norm on convex set C. This is called the elastic-net [9]. In the two optimization problems, Eq. (3) is convex and Eq. (2) is convex with respect to each of the two variables D_s and W when the other one is fixed. We use SPASM toolbox [11] to solve the two optimization problems.

3.4 Multi-class Classifier Training and Testing

Coefficients W computed from character training samples in Eq. (2) are used to train multi-class linear SVMs [8]. The regularization parameter is set to the best by cross validation on the training set.

In the test stage, given a test image (scaled to $H * W$ already), all stroke detectors are used to classify their corresponding response regions to obtain confidence vector f in Eq. (1). Afterwards, w are calculated as in Eq. (3). Finally, w is fed into pre-trained multi-class SVMs to obtain a class label. The procedure is shown in Fig. 5. Note that test images don't need to be manually labeled.

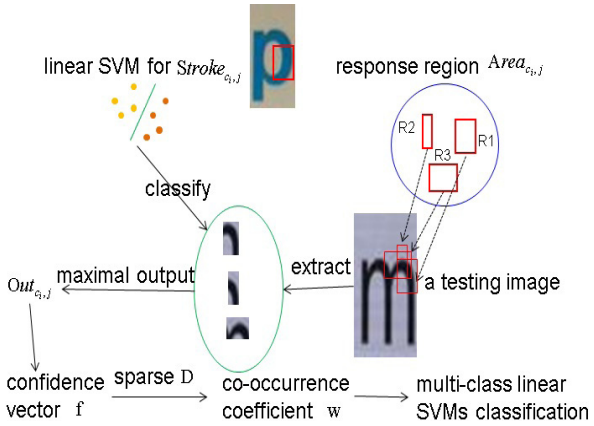

Fig. 5 Illustration of how to classify one testing image based on co-occurrence strokes.

4. Experiments

4.1 Dataset and Settings

We employ two public scene text character datasets: ICDAR2003 [12] and CHARS74K [13]. Both of these two datasets contain 62 character classes, namely digits 0-9, upper English letters A-Z and lower English letters a-z. ICDAR2003 dataset contains 6185 training patches and 5430 testing patches cropped from 509 scene images. It is originally designed for Robust Reading Competition of scene text detection and recognition so the contained samples can cover different conditions in natural scenes, such as heavy occlusions, different illuminations and complex backgrounds. Character samples from ICDAR2003 testing set are shown in Fig. 6. Similarly, CHARS74K has totally 12503 scene character images cropped from various natural scenes and these samples are not split into training and testing datasets. Previous algorithms [4], [13]–[16] of scene character recognition usually report their results on these two datasets so it's reasonable for us to test the proposed method on these two datasets. When performing CHARS74K-15 evaluation, we split training and testing set as in [16].

All of the image patches are normalized to $W = 32, H = 64$. Positive and negative samples for stroke detectors are scaled to $h_s = 16, w_s = 16$. HOG features are extracted with bin number 9, cell size of 4 pixels and block size of 2*2 cells. N_{stroke,c_i} is set uniformly for all characters from 3 to 15 with step 3. $N_{neg/pos}$ is set to be 2. Referring to [10], parameters of sparse dictionary learning are set empirically. N_D is set to be 400 and 600 for ICDAR2003 and CHARS74K respectively. λ in Eq. (2) and Eq. (3) is set to be 0.1 and γ in Eq. (4) is set to be 0.3.

4.2 Learning Co-occurrence of Local Spatial Strokes

When high level semantic information, namely co-occurrence of different strokes, is incorporated, the learned

Fig. 6 Scene character samples from ICDAR2003 testing set.

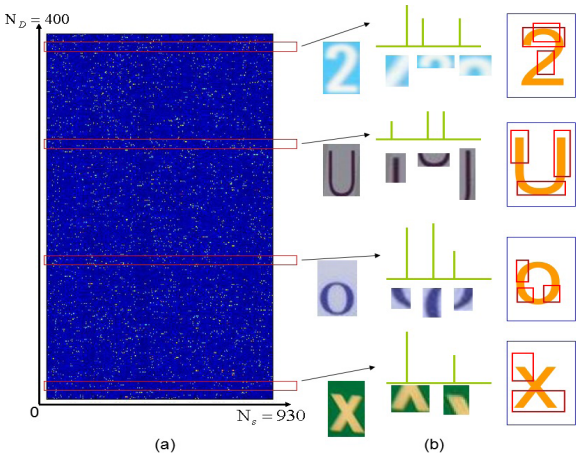

Fig. 7 Learned sparse dictionary with size of 400 for ICDAR2003 dataset. In the dictionary, red points indicate large magnitudes while blue color represents small magnitudes. (a) illustration of D^T; (b) large non-zero entries of rows often correspond to co-occurrence of several strokes from the same character.

sparse dictionary from ICDAR2003 training set with size of 400 is shown in Fig. 7. From Fig. 7, we can see that dictionary D is very sparse. Given a row of D^T as shown in Fig. 7 (a), elements with large magnitudes correspond to the co-occurrence strokes which usually belonging to the same character category as shown in Fig. 7 (b). On the contrary, the deep blue color with zeros entries correspond to the irrelevant strokes.

Comparison between modeling co-occurrence of character structures and training SVMs directly on confidence vectors is shown in Fig. 8. From Fig. 8, we can see that more stroke detectors may result in better performance in a range before coming to a peak point. Incorporating co-occurrence strokes always works better than only modeling single strokes' appearance.

4.3 Comparison with Other Algorithms

Recent published methods on scene character recognition mainly focus their attention on feature representation. For example, [13] compares different feature representations for scene character recognition. Tian et al. [14] propose to use co-occurrence histogram of oriented gradients (Co-HOG) to

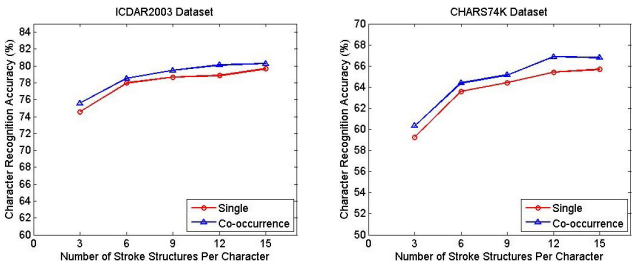

Fig. 8 Superiority of co-occurrence strokes over single stroke: left figure is for ICDAR2003 dataset and right figure is for CHARS74K dataset.

Table 1 Character recognition results on ICDAR2003 and CHARS74K dataset (%).

Algorithm	ICDAR2003	CHARS74K-15
ABBYY [13], [14]	26.6	31.1
HOG+NN [4]	51.5	58
SYNTH+FERNS [4]	52	47
NATIVE+FERNS [4]	64	54
MSER [18]	67	-
Global HOG [16]	76	62
Co-HOG [14]	79.4	-
Coates' method [15]	81.7	-
Geometrical blur+SVM [13]	-	53
Multiple Kernel Learning [13]	-	55
HOG Columns [17]	-	66.5
Our method (single)	**79.5**	**65.7**
Our method (co-occurrence)	**80.3**	**66.8**

recognize scene characters. Coates el al. [15] introduce unsupervised feature learning for robust character recognition and report promising recognition results. Yi et al. [16] generate Global HOG (GHOG) by computing HOG descriptor from global sampling for scene character recognition. Newell and Griffin [17] propose two extensions of the HOG descriptor to include features at multiple scales and demonstrate superiority of these new features over HOG for robust character recognition.

When N_{stroke,c_i} is set to be 15 and 12 for ICDAR2003 and CHARS74K datasets respectively, results of our method outperform state-of-the-art algorithms as in Table 1. Especially for ICDAR2003 dataset, we only use training samples from ICDAR2003 training set while other methods [14], [15] often introduce other training samples to avoid overfitting. Referring to [13] and [14], recognition results of commercial OCR software ABBYY FineReader are also reported to demonstrate the superiority of object recognition methods over traditional OCR techniques for scene character classification.

It should be noted that, by incorporating co-occurrence strokes rather than simply using single stroke detectors' responses, 0.8 percent and 1.1 percent improvement are realized upon ICDAR2003 dataset and CHARS74K dataset respectively.

5. Conclusion

This paper propose a high-level feature representation based on local spatial strokes for scene character recognition. We model co-occurrence of different strokes in different locations by learning a sparse dictionary and using reconstruction coefficients as final features. The results outperform state-of-the-art algorithms. If combined with sliding window technique and some language models, the proposed method can be extended to perform word-level recognition. That will be our future work.

References

[1] X. Chen and A.L. Yuille, "Detecting and reading text in natural scenes," CVPR, pp.366–373, 2004.
[2] L. Neumann and J. Matas, "Real-time scene text localization and recognition," CVPR, pp.3538–3545, 2012.
[3] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," CVPR, pp.2961–2968, 2013.
[4] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," ICCV, pp.1457–1464, 2011.
[5] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," CVPR, pp.2687–2694, 2012.
[6] L.J. Li, H. Su, L. Fei-Fei, and E.P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," NIPS, pp.1378–1386, 2010.
[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," CVPR, pp.886–893, 2005.
[8] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "Liblinear: A library for large linear classification," J. Machine Learning Research, vol.9, pp.1871–1874, 2008.
[9] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," J. Royal Statistical Society: Series B (Statistical Methodology), vol.67, no.2, pp.301–320, 2005.
[10] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," ICCV, pp.1331–1338, 2011.
[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," J. Machine Learning Research, vol.11, pp.19–60, 2010.
[12] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," ICDAR, pp.682–687, 2003.
[13] T.E.D. Campos, B.R. Babu, and M. Varma, "Character recognition in natural images," Computer Vision Theory and Applications, pp.273–280, 2009.
[14] S. Tian, S. Lu, B. Su, and C.L. Tan, "Scene text recognition using co-occurrence of histogram of oriented gradients," ICDAR, pp.912–916, 2013.
[15] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D.J. Wu, and A.Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," ICDAR, pp.440–445, 2011.
[16] C. Yi, X. Yang, and Y. Tian, "Feature representation for scene text character recognition: A camparative study," ICDAR, pp.907–911, 2013.
[17] A.J. Newell and L.D. Griffin, "Multiscale histogram of oriented gradient descriptors for robust character recognition," ICDAR, pp.1085–1089, 2011.
[18] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," ACCV, pp.770–783, 2010.