Could Scene Context be Beneficial for Scene Text Detection?

Anna Zhu[a,*], Renwu Gao[b], Seiichi Uchida[b]

[a] *Huazhong University of Science and Technology, State Key Lab for Multispectral Information Processing Technology , School of automation, Wuhan, China*
[b] *Kyushu University, Human Interface Laboratory, Information Science and Electrical Engineering, Fukuoka, Japan*

Abstract

Scene text detection and scene segmentation are meaningful tasks in the computer vision field. Could the semantic scene segmentation assist scene text detection in any degree? For example, can we expect the probability of a region being text is low if its surrounding segment, i.e., its context, is labeled as sky? In this paper, we have a positive answer by constructing a scene context-based text detection model. In this model, we use texton features and a fully-connected conditional random field (CRF) to estimate pixel-level scene class's probability to be considered as image's context feature. Meanwhile, maximally stable extremal regions (MSERs) are extracted, integrated and extended as image patches of character candidates. Then, each image patch is fed to a simple two-layer convolutional neural network (CNN) to automatically extract its character feature. The averaged context feature of the corresponding patch is considered as the patch's context feature. The character feature and context feature are fused as the input into a support vector machine for text/non-text determination. Finally, as a post-processing, neighboring text regions are grouped hierarchically. The performance evaluation on ICDAR2013 and SVT databases, as well as a preliminary evaluation on a patch-level database, proves that the scene context can improve the performance of scene text detection. Moreover, the compara-

*Corresponding author: Tel.: +86-186-2770-8137;
Email address: annakkk@live.com (Anna Zhu)

tive study with state-of-the-art methods shows the top-level performance of our method.

Keywords: Scene text detection, Fully connected CRF, Convolutional neural network, Character feature, Context feature.

---

1. Introduction

Recently, scene text reading captures much attention in the computer vision field [1, 2]. It refers to an attempt to recognize text from camera-captured images and contains two parts: scene text detection and scene text recognition. The technologies of this area can be applied to various applications, such as language translation system, visual-based navigation, and content-based web image search. Accurate detection of text in natural scene images is an essential step and primary work for successfully recognizing scene text. Therefore, we focus our attention on scene text detection in this paper.

Scene text detection meets great challenge [3, 4]. The difficulty comes from the huge variations of scene text. Specifically, scene text (1) varies in size, color, font, and style; (2) has no clear layout; (3) tends to have complex background, and (4) undergoes non-uniform lighting, partial occlusion, blur, rotation, perspective distortion, and low resolution.

To tackle with those difficulties, we propose a framework to detect scene text by means of scene context information. The idea arises from the observation that text appear frequently in certain scene context and rarely in others. For example, they are always embedded in signboard, cars, books and other surfaces, but rarely appear in rivers, the sky, trees or grass. If this idea is valid, a better detection performance can be expected by using the framework shown in Fig. 1.

In this framework, we use not only the character feature but also context feature for discriminating a connected component (CC) into text or non-text. In this situation, even if the character feature suggests that a non-text CC is a character or a part of a character, it may still be discriminated into non-text, when the context feature suggests its surrounding segment is “sky”. Conversely,

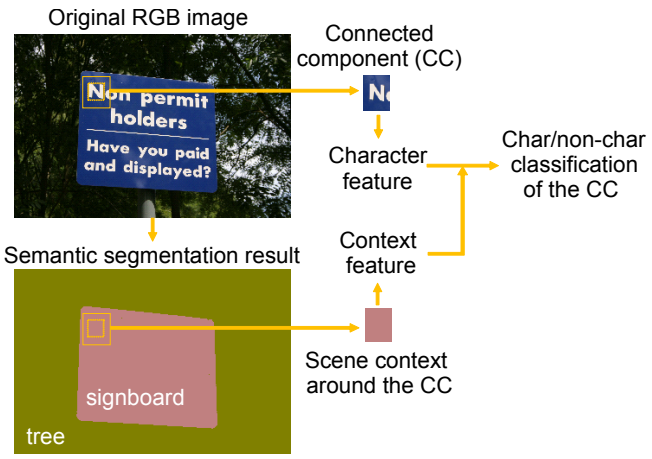


Figure 1: Our hypothesis: confuse context information for classification.

even if the character feature suggests that a text CC is non-text, it can be detected as text, when the context feature suggests its surrounding segment is "signboard".

Scene context information represents the scene class attribution of pixels in the images, which can be extracted through semantic scene segmentation. Though semantic scene segmentation has been another difficult task, recent researches [5, 6] show very promising results. In our approach, we employ Philipp's work [7], in which images are segmented and semantic labeled in pixel-level by TextonBoost and then refined by a fully-connected conditional random field (CRF). From the segmentation result, we obtain K-dimensional feature vector of each pixel, namely the scene context, where K is the number of scene classes (such as sky and signboard) and the kth element of the vector is a probability that the pixel lies on the kth scene class. The averaged scene context probabilities of a CC region is considered as this CC's context feature. The context feature is then combined with the character feature generated by a two-layer convolutional neural network (CNN) [8] and fed to a support vector machine (SVM) classifier for text/non-text classification.

The contribution of our work is threefold. The first contribution is to propose a new idea of utilizing scene context for scene text detection. In traditional

scene text detection researches, they focus only on "text-ness" at each region in a scene image. Specifically, they assume several heuristics about texts, such as, high contrast, high spatial frequency, dense edges, uniform stroke width, and then use them for detection. Even though these heuristics work reasonably, they cannot avoid false alarms and false negatives. False alarms will be detected around text-like shapes in scene. For example, car tires can be detected as "o" because it satisfies most of the above heuristics. False negatives will happen on texts with irregular appearance. For example, texts printed in a fancy font may not have uniform stroke width and texts on complex background may not have enough contrast. Some recent methods are free from those heuristics by using machine learning techniques, but they learn text-ness only and thus still have similar false alarms and false negatives. The introduction of scene text will be an essential solution for this problem.

The second contribution is to emphasis the usefulness of semantic segmentation for scene text detection. Semantic segmentation is a rather new technology and developed mainly for scene object recognition and scene understanding. To our best knowledge, semantic segmentation has never been used for scene text detection, i.e., there is no class "text" in the task of semantic scene segmentation. This may come from that most semantic segmentation methods assume some smoothness and prevent detecting very fine and thin structures, i.e., texts. We will see that semantic segmentation is still very useful for scene text detection by the help of CC-based text detection strategy, even though semantic segmentation has no enough resolution to detect individual text by itself.

The third contribution of our work is that, to prove our hypothesis, four variable controlled comparison experiments, which control the variable whether to use context feature and how to train CNN, are performed to provide adequate and valid proof for test. Besides, a patch-level database is given in this work. In this database, all the patch images are manually labeled and cropped from text regions in non-overlapping images of the ICDAR 2011 and ICDAR 2013's training databases. The results on this patch-level database and ICDAR 2013 database achieve convincible performance.

The following section gives an overview of our pipeline. We review related work in two directions in Sect. 2. Sect. 3 presents our proposed method in detail. In Sect. 4, we give the experimental results which include the details of databases and the experimental setup. Finally, Sect. 5 gives a summarization and conclusion of this paper.

2. Related work

In this section, we summarize some related previous work. Since our work refers to scene segmentation but focus on scene text detection, we introduce the related works on scene segmentation briefly and text detection methods emphatically.

2.1. Scene segmentation methods

Scene image segmentation aims to label every pixel in the image with several predetermined classes, thus concurrently perform recognition and segmentation of multiple classes. The segmentation techniques can be classified as follows: graph-based approaches, region-based approaches, boundary detection approaches, perceptual organization approaches, multi-class image segmentation, and hybrid approaches.

The graph-based image segmentation approach defines the boundaries between regions by measuring the dissimilarity between the neighboring pixels by a graph, where the node representing each pixel and the weights denoting the dissimilarity between pixels [9]. Region-based techniques make use of common patterns in intensity values within a cluster of neighboring pixels and group regions according to their anatomical or functional roles [10]. The boundary detection approach refers to a contour in the image plane that represents dissimilar pixels between the neighboring segments [11]. Perceptual organization refers to a basic capability of the human visual system to obtain relevant groupings and structures from an image without having prior knowledge of the image's contents [12]. Multi-class image segmentation uses one of a number of classes (e.g.,

road, sky and water, etc) for labeling every pixel in an image. Many state-of-the-art methods first over-segment the image into superpixels (or small coherent regions) and then classify each region [13, 14]. Hybrid techniques combine the above segmentation approaches [15] to take advantages of them. Our scene segmentation method belongs to hybrid method, because it combines multi-class image segmentation methods and graph-based methods together.

2.2. Scene text detection methods

Scene text detection involves extracting regions that contain text and filtering out regions without text in the images. It is a typical classification problem. We classify the text detection methods into three categories: global-classification methods, local-classification methods, and hybrid methods.

The global-classification methods split images to region-based patches and attempt to classify these patches into text/non-text by using feature, such as Histogram of Oriented Gradients (HOG) [16], Local Binary Pattern (LBP) [17], and learns classifiers [18, 19] on these features to discriminate text and non-text. Those patches are selected by scanning the whole image with sliding windows. To resist different sizes of text, multi-scale strategy are typically used. After classification, the patches containing text are merged into text blocks. Chen et al. [20] trained a cascade of 4 strong AdaBoost classifiers containing 79 informative features to classify the regions. Wang et al. [21] classified all the patches extracted from the sliding window by using a random ferns classifier trained on HOG features. Neumann et al. [22] used sliding windows for stroke detection by convolving the gradient field with a set of oriented bar filters. Mishra et al. [23] exploited bottom-up cues that classify individual character regions by a HOG-based SVM classifier and top-down cues that prune false detected regions with a CRF model. Wang et al. [8] normalized all the extracted regional patches to input into a CNN classifier and showed that a deep feature-based CNN is effective for character patch classification. These kinds of methods treat text detection as object detection tasks. They can capture the inter-relationships of text, by which text can be detected even if there are noises. However, they often

use exhausted searching strategy, which results in the significant computational cost.

The second category filters out most non-text pixels initially and uses distinct features based on CC level to extract the candidate characters on the rest pixels. It groups spatial pixels to CCs conditionally. Either CC level features or extended region-based features are inputted into trained classifier or heuristic rules to discriminate text components from non-text components. Candidate characters are grouped into words based on their relationship like location, text line, distance, stroke width and color features. Epstein et al. [24] proposed the Stroke Width Transform (SWT) algorithm which computed per pixel the width of the most likely stroke containing the pixel to obtain text CCs. Many modified SWT methods [25, 26] were proposed with the same idea. The MSER method [27] and ER method [28] extracted some extremal regions as candidate characters. These kinds of methods showed its outstanding effectiveness and high efficiency on text detection area. Yin et al. [29] exploited a MSER tree to extract character candidates and learned the distance weights and clustering threshold to cluster the character candidates into text candidates. Further, they proposed a multi-orientation text detection system [30] to construct text candidates through three sequential coarse-to-fine grouping steps with adaptive clustering and distance metric learning. Sun et al. [31] extracted color-enhanced CERs as character candidates in six component-trees which were built from the gray scale, hue and saturation channel images in a perception-based illumination invariant color space, respectively. Because all of the features extracted from these methods are normalized in process, it is convenient in dealing with text of variant scales and orientations. Besides, the detected result can be used directly for text recognition.

The third category combines the advantages of the two above categories in order to achieve higher robustness while keeping lower computational cost. In Pan's work [32], a region-based method first estimated the text existing confidence, and then non-text components are filtered out by CRF. Huang et al. [33] used MSERs by incorporating with a text saliency map which is generated by a

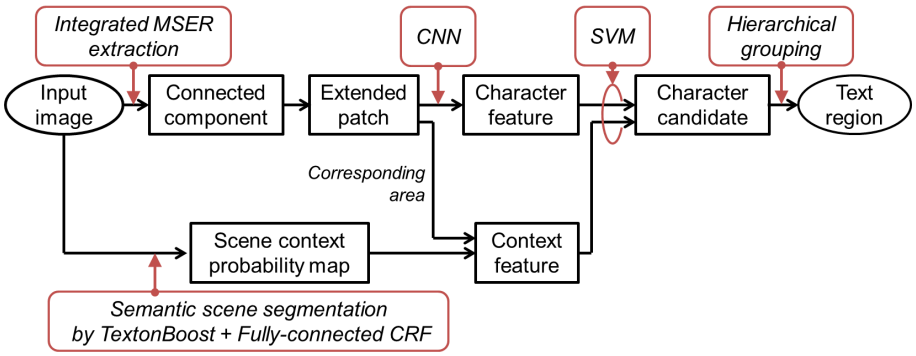


Figure 2: The flowchart of the proposed method.

strong CNN classifier to efficiently prune the trees of Extremal Regions. These hybrid methods first roughly select candidate text regions with one category and then use other strategies to confirm the confidence of text regions and filter out non-text regions.

3. The proposed method

3.1. Overview of our approach

A flowchart of our text detection framework is presented in Fig. 2. Our method belongs to local-classification category as it extracts MSERs in images.

To extract the context feature of each pixel, a fully connected CRF model that is defined on the complete set of pixels in an image is used as a tool to update pixel-level context probabilities, which are initialized by TextonBoost. It is a global-classification process, but only context feature of pixels who lay in the extended rectangle regions of MSERs are utilized.

Four sets of parameters are used to get the MSER maps. We can get four different MSER maps by processing on the original gray-scale image, then integrate them to one map. The same process is performed on the 0-255 reversed gray-scale image and finally there are two integrated MSERs maps without multiple repeating structures. All the integrated MSERs are converted to bounding box regions and then extened to larger size based on the bounding box's width

and height. They are normalized to the same size and fed to CNN for character feature extraction.

For text/non-text classification, the CNN classifier is selected because of its excellent performance on object classification [34] and character classification [35, 36]. There are two schemes to combine context feature. One is using context features and image patches to train CNN concatenated with a SVM classifier together. The parameters of CNN and SVM fine tuning sequentially. The other is to train a SVM classifier with character features which is generated by a pre-trained CNN from image patches and add the corresponding context features. The parameters in CNN and SVM are trained separately. These two schemes correspond to scheme (2) and (3) in the experiment section. And both schemes illustrate that scene context features can affect the text detection result to some degree.

After classification, a hierarchical grouping process is proposed to retrieve missing characters and connects character to words. This provides the possibility for improving detection results on the basis of the former stage. In the following sections, each part will be described in detail.

3.2. Context feature extraction

In this section, the context feature of each pixel is computed by TextonBoost and refined by a fully-connected CRF. It is a 14-dimensional vector which presents the probability of pixel belonging to K=14 semantic scene labels. The 14 semantic scene labels are sky, trees, grass, rivers, houses, books, cars, airplanes, signboards, human faces, human clothes, walls, roads and text streamers. We statistically compute a histogram of all the objects appeared in scene text images of ICDAR 2011 and ICDAR 2013 training databases. Scene classes with high frequency of appearance in the databases and high probability of containing text are selected. Finally we obtain the 14 classes.

3.2.1. Preliminary context feature generation by TextonBoost

The initial context feature, namely the probabilities, is derived from TextonBoost [13, 37]. The features are obtained by convolution the input image with a 17-dimensional filter bank suggested by Shotton et al. [13]. And then, follow the work from Ladick et al. [37] by adding color, HOG, and pixel location features. 12 objects categories' training images are obtained from the MSRC-23 database [13] and the training images for wall and text streamer are either downloaded from Google images or ICDAR211 database. The output of TextonBoost is considered as the unary potentials of Fully-connected CRF.

3.2.2. Updating pixel-level context probability using fully-connected random field

Basic CRF models are composed of unary potentials on individual pixels or image patches, and binary potentials on neighboring pixels or patches [14, 38, 39]. The resulting adjacency CRF structure is limited in its ability to model long-range connections within images and generally results in excessive smoothing of object boundaries. To avoid the above problem, we use a fully-connected CRF. It establishes pairwise potentials on all pixels, by which it can greatly refine segmentation and labeling. Here, we briefly introduce the fully-connected CRF model.

Consider a random field $\mathbf{Y}$ defined over a set of variables $\{Y_1, \cdots, Y_N\}$. The domain of each variable is a set of labels $L = \{l_1, \cdots, l_K\}$, K is the number of classes. Consider also a random field $\mathbf{X}$ defined over variables $\{X_1, \cdots, X_N\}$. In our setting, $\mathbf{X}$ ranges over possible input image's size N and $\mathbf{Y}$ ranges over possible pixel-level image labeling. X_j is the observation vector of pixel j and Y_j is the label assigned to pixel j.

A CRF $(\mathbf{X}; \mathbf{Y})$ is characterized by a Gibbs distribution, as in Eq. 1:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \exp\left(- \sum_{c \in C_G} \phi_c \left(\mathbf{Y}_c|\mathbf{X}\right) \right), \tag{1}$$

where each clique c in a set of cliques C_G in G induces a potential ϕ_c. $G = (V, E)$

is a graph on $\mathbf{Y}$. The Gibbs energy of a labeling $y \in L^N$ is:

$$E(y|\mathbf{X}) = \sum_{c \in C_G} \phi_c\left(y_c|\mathbf{X}\right). \tag{2}$$

The maximum a posteriori (MAP) labeling of the random field is $y^* = \text{argmax}_{y \in L^N} P(y|\mathbf{X})$. For notational convenience we will omit the conditioning in the rest of the paper and use $\psi_c(y_c)$ to denote $\phi_c(y_c|\mathbf{X})$.

In the fully-connected CRF model, G is the complete graph on $\mathbf{Y}$ and C_G is the set of all unary and pairwise cliques. The corresponding Gibbs energy is presented as Eq. 3:

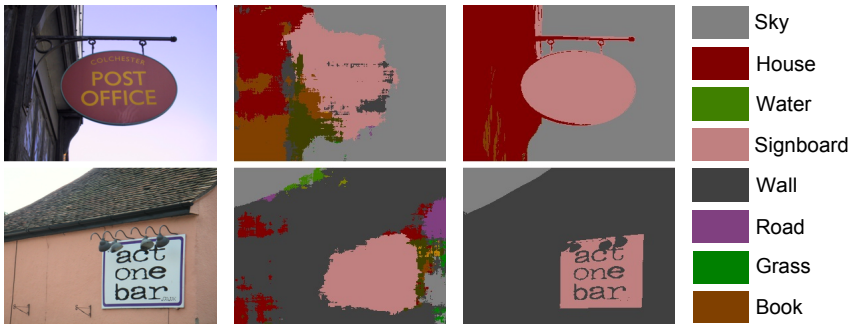
$$E(y) = \sum_i \psi_u(y_i) + \sum_{i<j} \psi_p\left(y_i, y_j\right), \tag{3}$$

where i and j range from 1 to N. $\psi_u(y_i)$ is the unary potential obtained from TextonBoost. The pairwise edge potentials $\psi_p(y_i, y_j)$ are defined by a linear combination of Gaussian kernels in Eq. 4:

$$\begin{gathered}\psi_p(y_i, y_j) = \mu(y_i, y_j) \underbrace{\Sigma_{m=1}^{K} \omega^m k^m(f_i, f_j)}_{k(f_i, f_j)}, \\ k(f_i, f_j) = \omega^1 \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + \omega^2 \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}},\end{gathered} \tag{4}$$

where vectors f_i and f_j are feature vectors defined in terms of color vectors I_i and I_j and position vectors p_i and p_j for pixel i and j, ω^m are linear combination weights, and μ is a label compatibility function. The appearance kernel is inspired by the observation that nearby pixels with similar color are likely to be in the same class. The degrees of nearness and similarity are controlled by learned parameters θ_α and θ_β. The smoothness kernel which can remove small isolated regions.

A mean field approximation [40] is applied to the CRF distribution for its high efficiency. Some of the segmentation results are shown in Fig. 3(c). Com-



(a) (b) (c)

Figure 3: Scene context feature extraction. (a) The original images. (b) Segment results after TextonBoost. (c) Segment results after fully connected CRF.

pared to the coarse segmentation result of TextonBoost in Fig. 3(b), we can see that some initially incorrect segmented pixels are changed to correct categories. To utilize the segmentation result yielded by the fully-connected CRF, we need to update the context probability of the pixels whose category attribution changes. We denote the original context probability of a pixel as $\mathrm{P} = \{p_1, p_2, \cdots, p_K\}$, K is the total object categories. If its category attribution changed to a different label l_u after fully-connected CRF, we updated the context probability of p_{l_u} to 0.8, and set other $p_{l_v \in L, l_v \neq l_u}$ to $0.2 \times \frac{p_{l_v \in L, l_v \neq l_u}}{1 - p_{l_u}}$. This process assigned a larger probability to the updated object labeling, meanwhile keeping the sum of each category's probabilities to 1.

3.3. Text/Non-text classification

In this stage, we confuse the context features and character features to classify text and non-text. Our method belongs to the local-classification method. The MSER-based text detection methods [41, 42, 43] are state-of-the-art methods for their outstanding effectiveness and high efficiency. It defines an extremal region as a CC whose pixels have intensity contrast against its boundary pixels in the image. We applied various sets of parameters of MSER to extract as many components as possible, and then integrated them into two maps to overcome the overlapping problem. The classification process focuses on the two maps. All the CCs on the two maps are extended to region-based patches. A

well trained CNN classifier with two layers is used to classify the patches to text or non-text. For the character adjoining problem, the width and height ratio is a judgement. Therefore, a coarse segmentation is implemented for adjacent joint character regions. The total classification scores of all patches in the corresponding adjacent joint character region will determine whether it is text or not.

3.3.1. Maximally stable extremal regions (MSERs) integration

In the work[44], Matas defined the extremal region as a connected component of an image whose pixels have either higher or lower intensity than its outer boundary pixels. The Maximally stable extremal region is an extremal region whose size remains virtually unchanged over a range of intensity levels. Consider an extremal region Q_t, and the branch of the tree rooted at $T(Q_t) = \{Q_t, Q_{t+1}, \cdots, Q_{t+\Delta}\}$. The instability of Q_t is defined as $\nu(Q_t) = |Q_{t+\Delta} - Q_t|/|Q_t|$. $|Q|$ denotes the number of pixels in region Q. Q_t is a maximally stable extremal region if its instability is more lower than its parent Q_{t-1} and child Q_{t+1}. Extracting MSERs in an image involves five parameters that control its performance. Their definition and meaning are listed below:

(1) Δ; It compares $(size_{t+\Delta}\text{-}size_t)/size_t$,

(2) maxArea; It prunes the area bigger than maxArea,

(3) minArea; It prunes the area smaller than minArea,

(4) maxVariation; It prunes the area to have a similar size to its children,

(5) minDiversity; It traces back to cut off MSERs with diversity less than set value.

Table 1: Four sets of parameters in MSER.

PARAM	Δ	maxVariation	minDiversity
1	2	0.5	0.5
2	5	0.5	0.5
3	5	0.1	0.9
4	5	0.5	0.5

Algorithm 1 Integrate MSER map.

Input: MSER maps I = $\{I_1, \cdots, I_n\}$ generated by the n parameter sets and original RGB image. $I(x, y)$represents the value in coordinate point (x, y).
Initialize: Transform I to 0-1 map.
Output: Integrated map M.

```
I_temp(x, y) = I_1(x, y);
for i: 2∼n do
    I_temp(x, y) = I_temp(x, y) + I_i(x, y);
    Label CCs of I_temp(x, y) = 1 (denote as C_1) and get each CC's height
  H and width W; Label CCs of I_temp(x, y) = 2 (denote as C_2);
    for each c ∈ C_2
        Search for the adjacent two CCs c_i and c_j in C_1;
        if W_i + W_j > min(H_i, H_j), and I_temp(x, y) =1
            I_temp(x, y) = 0;
        else
            Compute the adjacent pixels' average RGB vector V in c, c_i, c_j;
            if max(|V_c - V_{c_i}|, |V_c - V_{c_j}|, |V_{c_j} - V_{c_i}|)>10, and I_temp(x, y) =1
                I_temp(x, y) = 0;
            end if
        end if
    end for
    Remove CCs in C_1 with area size less than 10;
    if I_temp(x, y) =2
        I_temp(x, y) =1;
    end
end for
M = I_temp;
```

Different parameters result in different MSER regions. For different images, the same parameters applied to MSER may generate diffident results. The characters may adjoin the background or removed. It is difficult to set fixed or adaptive parameters for obtaining the perfect performance on all the images.

To robustly extract MSERs in all the images, four different combinations of parameters are set. Among them, the minArea is fixed to 10 and maxArea = 1/3 $\times$ image size. The four sets of the other three parameters are listed in Tab. 1. If all the MSERs from the four sets are classified in the classification step, a high

(a) (b) (c)

Figure 4: Integrated MSER map for the four parameter sets. (a) The examples of scene text images. (b) Four MSER maps generated by the four parameter sets. (c) Integrated MSER map.

recall will be achieved. However, it costs more time for computation since there are many repeating MSERs or similar MSERs of the same text in the images. To avoid this problem, we integrate all the MSERs. The details of the integration steps are given in Algorithm 1. The same process is applied to 0-255 reversed gray-scale images. Finally, we obtain two maps with non-overlapping CCs. The adjoining characters of some CCs will separate to individual characters after this process as shown in Fig. 4. This method successfully avoids the repeating problem as mentioned in [31]. It can also extract as many text CCs as possible and also preserve them efficiently.

3.3.2. Patch-based character/non-character classification

Scene text detection in natural image is a high level visual task, which is difficult to be solved completely by a set of low-level operations or manually designed features. Therefore, we exploit convolutional neural networks (CNN) to extract character feature in our work, since a deep network is capable of leaning meaningful high-level features and semantic representations for visual recognition through a hierarchical architecture with multiple-layers of feature convolutions. The deep structure of the CNN allows it to refine feature representation and abstract sematic meaning gradually. Applications of applying CNN have achieved great success on digit and hand-written character recognition [45, 46].

A simple two-layer CNN structure connected with a SVM classifier in the last layer is adapted as shown in Fig. 5. The input of this classification system

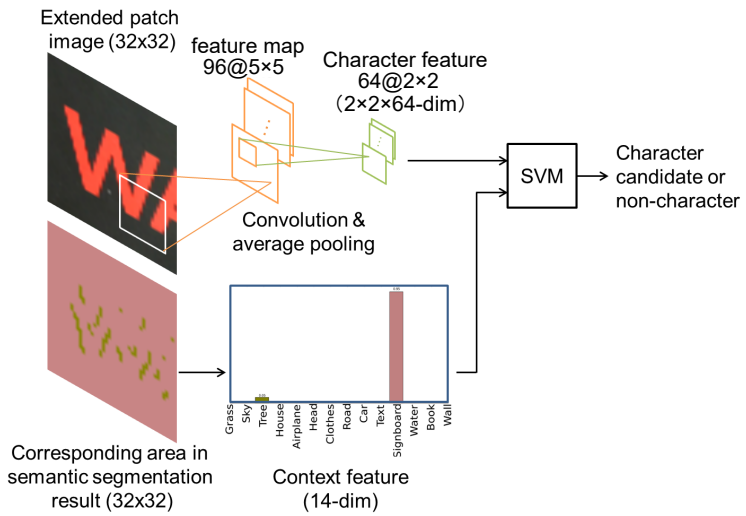


Figure 5: The classification system.

is the region-based patch of CCs on the integrated MSER map and an averaged context feature in this region. The output is a vector of two dimensions representing the score of text and non-text.

Since the input of the CNN is region-based patches, we need to transfer the CCs of the integrated MSER map to patches. For the CCs obtained in the former step, connections may exist, as shown in Fig. 6. The height and width ratio R is used to discriminate whether the CCs are connected or not. If $R < T_R$ (in our work, we set the threshold T_R to 1.8), the CCs are non-connected. Otherwise, they are connected CCs. For a non-connected CC region, we directly turn it to a patch which is the region of bounding box. However, to extract more context information of it, we extend the patch region to larger size by adding $1/3 \times$ CC region's height to the boundaries. For connected CCs, many characters are adjoined to words or strings. It is difficult to estimate the size of each character. So, accurate segmentation meets great challenge. In our method, we coarsely segment the characters based on the CC region's height and extract many overlapped regions with two strides: $0.6 \times$ CC's height and $0.7 \times$ CC's height. The same extension is processed as the individual CCs. The average

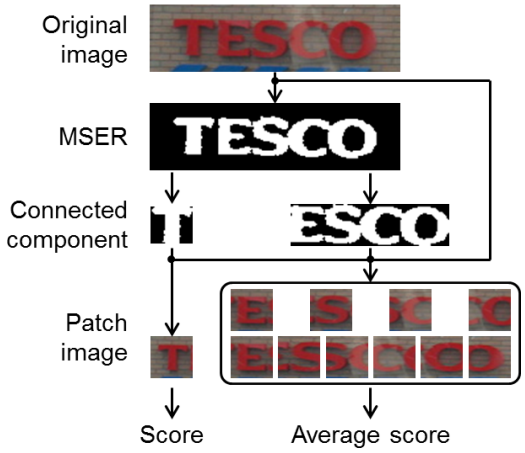


Figure 6: Patch extension for individual CCs and connected CCs

text score of all patches in the connected region is considered as its score.

In this classification system, the input patches of the CNN are normalized to 32 × 32. The settings of the CNN architecture used in our work is displayed in Tab. 2. 96 and 64 nodes are designed in the first and second layers. We use average pooling and rectified activation. The stride sizes of convolution and pooling are both set to 1. Therefore, we can get 5 × 5 and 2 × 2 size feature maps in the first and second layers of the CNN respectively. In the first layer of the CNN , we train the filters in an unsupervised learning algorithm [8]. In a given set of 32 × 32 training images, we randomly select 8 patches with size 8 × 8. Then, perform contrast normalized and whitened zero-phase component analysis (ZCA) [47] of all these patches to form 64-dimension input vectors of each patch. A K-means clustering method is used to learn a set of low-level filters D$\in R^{64\times n}$. Here we set n = 96 which is equal to the first layer's node number. For a normalized and whitened 8×8 patch x, we compute its first layer responses by performing inner product with the filter bank followed by a scalar activation function with response = max{0, $|D^T x|$-α}, here $\alpha = 0.5$ is a hyperparameter. In the second layer, 14-dimensional averaged context feature vector is confused with the projected 2 × 2 × 64-dimensional character feature vector and cascaded to an SVM classifier. The network is discriminatively trained by backpropagating

Table 2: The parameters used in CNN architecture.

Layers	Kernel size	Pooling size	Nodes
Layer-1	8	5	96
Layer-2	2	2	64

the L_2-SVM classification error. Only the parameters in second layer of CNN are updated in the fine-tuning.

After classification, the CCs are labeled with probability of text. We filter out some of the non-text CCs by setting a threshold $T = 0$ to the output score of the CNN.

3.4. Hierarchical grouping process

After classification, the CCs classified to text are defined as text components (TCs). However, there still are false negatives and false positives. To tackle this problem, we use a hierarchical grouping method to retrieve the false negatives and remove the false positives.

First, we define four types of the candidate TCs: left TC, right TC, middle TC and isolated TC. A right or a left TC has no adjacent right or left TC respectively, a middle TC has both adjacent left and right TCs, and an isolated TC has no adjacent TCs. Since the similar geometric properties, color appearance and context information are shared by the text CCs in the same word or text line, they can be used to measure the adjacent relationship and discriminate the CC's type. For a certain candidate text CC, we search its left and right regions from near to far with the distance of two times of its height to find candidate text CCs. If they satisfy condition (1) or condition (2) listed below, they are adjacent TCs.

Condition (1): $|C_i - C_j| < T_{c1}$; $\min(T_i - T_j, B_i - B_j) < \min(H_i, H_j)$.
Condition (2): $|C_i - C_j| < T_{c2}$; $|SC_i - SC_j| < 1$; $A_i/A_j > T_s$ and $A_i/A_j < 1/T_s$; $(T_i - T_j \parallel B_i - B_j) < \min(H_i, H_j)/2$.

Here, T and B represent the top coordinate and bottom coordinate respectively. H is the height of TC. A is the area size, C is the averaged RGB color

Figure 7: Example of some grouping results. (a) shows the classification results before grouping. (b) shows the grouping results by connecting adjacent characters' centers with red lines. (c) The final text detecting results.

vector, and SC is the averaged context feature. In all of our framework, we set T_{c1}=20, T_{c2}=40 and T_s=0.3.

For a middle TC, if its right and left adjacent TCs' inter-character distance difference is lager than one third of the three TCs' middle height value, there might be missed characters or string breaks. We search the CCs on the integrated MSER map in this region and discriminate whether to retrieve the CCs by the following rules.

(1) If the CC satisfies $H_X > \max(H_i, H_j)/3$, $H_X < \max(H_i, H_j)\times 1.5$, $X_T > \min(T_i, T_j)$ and $X_B < \max(B_i, B_j)$, retrieve it, otherwise, turn to (2).

(2) Count C_X of this CC limited in the region $\max(T_i, T_j)$, $\min(B_i, B_j)$, X_L and X_R. If $|C_X - C_i| < T_{c1}$ or $|C_X - C_j| < T_{c1}$, retrieve this CC.

Here, i and j are neighbored TCs of X. X is the CC between i and j which locates on the integrated MSER map. X_T , X_B, X_L and X_R represent the top, bottom, left and right boundary coordinate of X.

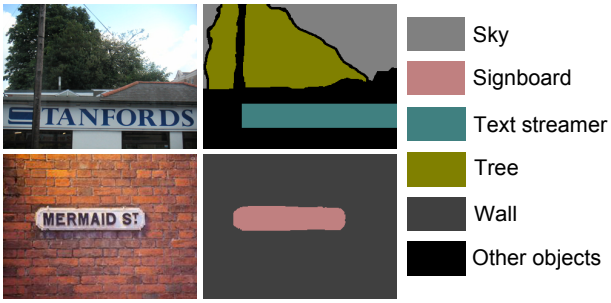


Figure 8: Pixel level labeling of wall and text steamer images.

After that, we connect all the adjacent TCs to obtain the detection result. For isolated CCs, we compute their parallel edge pair ratio [48]. If the edge pair ratio is larger than 0.5, they are considered as text. Otherwise, we classify them as background and remove them. Some grouping results and final detection results are shown in Fig. 7.

4. Experiments and results

In this section, we introduce the two collected databases for scene segmentation and scene text detection. The evaluation and analysis on scene text detection are presented.

4.1. Data collocation

4.1.1. Database for scene segmentation

In our method, 14 scene categories are used for segmentation and labeling. The images of 12 scene categories are extracted from the MSRC-23 database [13]. Each categories contains 30 images with the corresponding ground truth labels. Besides that, the two additional object images (wall and text streamer) are downloaded from Google Images or selected from the ICDAR 2011 database [49]. Each ground truth image is fully annotated at pixel-level, with careful labeling around complex boundaries as shown in Fig. 8. Each object category contains 30 images for both training and testing.

4.1.2. Database for scene text detection

To train the parameters of CNN and SVM, we collected 440 images from the ICDAR 2011 and ICDAR 2013 text location training databases. We manually labeled the character regions with the bounding boxes in each image. By extending them to larger size based on the process in Section 3.3.2, we obtained 8317 text patches. For background patches, we performed an MSER algorithm to the databases and removed the CCs belonging to text region. The non-text CCs were extended to patches and cropped the same way as text patches. In total, we got 15440 background patches. The process of collection was shown in Fig. 9. They compose the patch-based database and were used to evaluate four variable controlled classification methods. Meanwhile, we store the context features of each patch region in a matrix. All of the patches with the corresponding context feature matrices are randomly divided to two sets: 15000 training patches and 8757 test patches. The training patches contain 6000 text patches and 9000 background patches. The rest of the parts belong to the test patches. To compare our proposed method with other scene text detection methods, the public ICDAR 2013 database and SVT database [21] are used for evaluation.

4.2. Scene context feature extraction performance with fully-connected CRF

To evaluate the performance of TextonBoost and the fully-connected CRF, we randomly selected 6 images in each object category from the scene segmentation database for test. The others were split into training images. The unary potentials and parameters of the CRF model were learned on the training set and the training time cost was about 5 hours on an Intel i7 processor clocked at 12.0GHz. It yielded the global score of 82.5% and average score of 74.3%, where the global score denotes the overall percentage of correctly classified image pixels and the average score is the unweighted average of per-category classification accuracy [13, 50]. The trained parameters are used to extract the context features in the scene text detection databases. The examples of segmentation results in scene segmentation test database and scene text detection databases are given in Fig. 10(a) and Fig. 10(b). However, it may fail for some scene im-

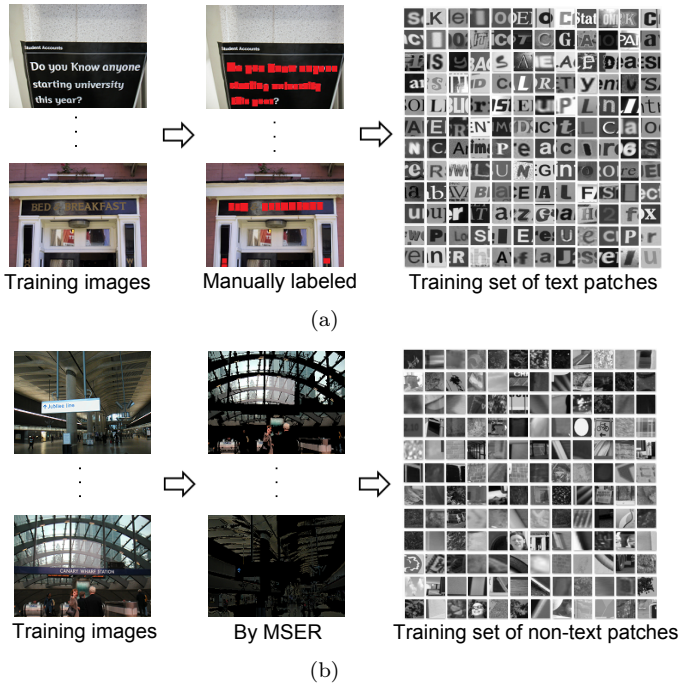


Figure 9: Patch extraction. (a) Extracted text patches from manually labeled text regions. (b) Extracted background patches from non-text regions generated by MSER.

ages as shown in Fig. 10(c). Some text regions can not be segmented and some objects are incorrect classified to other objects. For example, in the first and third images of Fig. 10(c), the signboard on the wall and the text streamer on the airplane are not correctly segmented. The background in the second image are classified to a book and the airplane in the forth image is classified to a car. Since the training images in scene segmentation databases are limited, it is not so robust for segmenting all types of object categories. Because some backgrounds are very complicated, it is difficult to define them to certain categories. In the scene text detection database, there are some objects undefined, such as a microwave, sculpture, mouse cursor, *etc.*, these objects will be classified to other objects in the 14 object categories.

Figure 10: Scene segmentation performance of some example images. (a) Results in scene segmentation database. (b) Results from scene text detection database. (c) Some incorrect segmented images.

4.3. Effect of context feature for text detection

The segmentation results are used to update the context features, namely the probabilities of the pixels. Then, they are averaged as the input context feature of the region to be classified. Three evaluations are performed in our experiment: evaluation on patch-based classification, evaluation on public ICDAR 2013 database and SVT database.

4.3.1. Results of patch-based classification by CNN

Four variable controlled experiments are implemented on patch-based database. We conclude them to four schemes as below:

(1) Train CNN+SVM without context features;

(2) Train CNN+SVM with context features;

(3) Train SVM with pre-trained CNN features and context features;

(4) Train SVM with pre-trained CNN features and averaged equal fake context features (All are 1/14).

The input of Scheme (1) is only patch-level images. Scheme (2) is the frame of our work. In training process, the The pre-trained CNN features in Scheme (3) and (4) means that the parameters in the CNN are not trained with the SVM but trained in advance with the patch-level training images. Then, after training, we combine the output of the CNN features and corresponding context features to train the SVM classifier. To figure out whether the context features and the CNN's structure affect the classification result, we add fake context features with all the values as 1/14 in Scheme (4).

Since the patches in the database belong to text or background, it is a two-class classification problem for the patch-based images. We define an accuracy that equals to the correctly classified patch number $SUM(P_{correct-classified})$ divides by all of the patch number $SUM(P)$, namely accuracy $= \frac{SUM(P_{correct-classified})}{SUM(P)}$, to measure the classification performance. Then, we give the results of each schemes in Tab. 3. The training accuracy and test accuracy is the accuracy for the training patches and test patches respectively. In Fig. 11, the misclassified patches generated by our proposed method and non-context feature method in

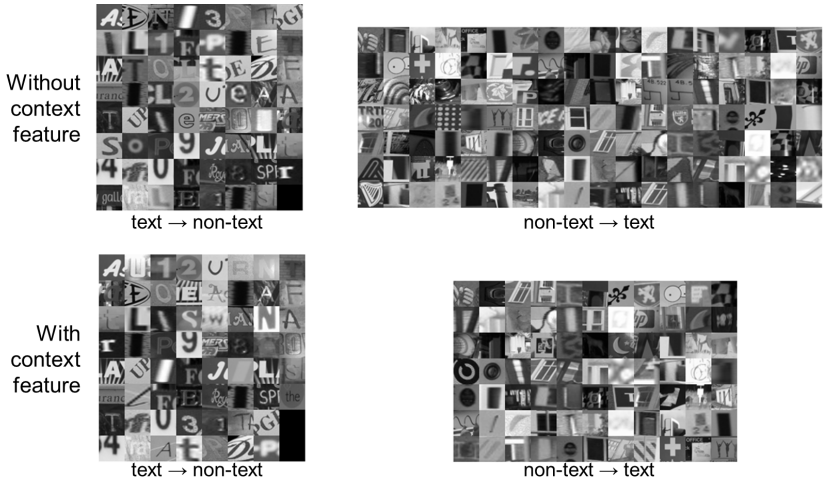
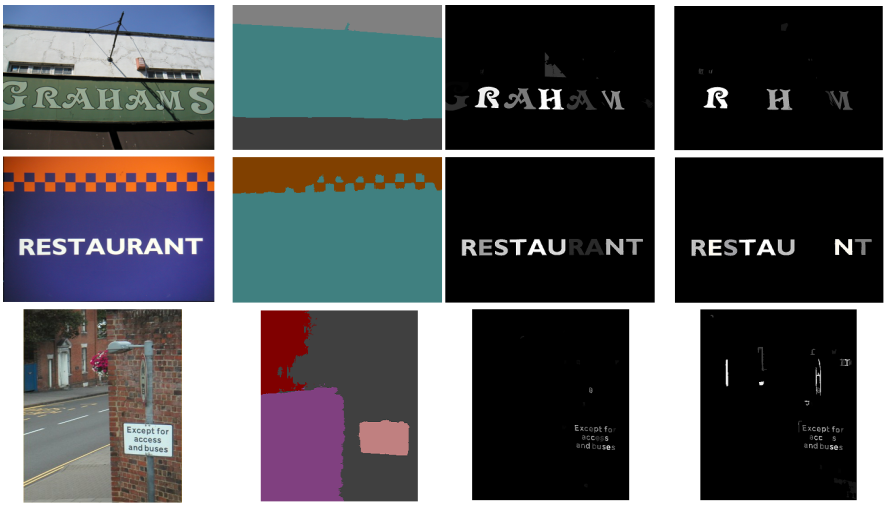


Figure 11: Misclassified text patches and background patches.

Table 3: The patch-based classification accuracy on four schemes.

Scheme number	1	2	3	4
Scene context	No	Yes	Yes	Equal-fake
Train CNN with scene context	no	no	yes	yes
Train accuracy	99.99%	99.77%	99.99%	99.99%
Test accuracy	97.84%	98.44%	97.88%	97.77%

Scheme (1) are given. Less background patches are misclassified in our method, thus, with the context information, it can remove more manually labeled background regions, which can increase the precision in text detection task. We can see that, our proposed method, namely Scheme (2), achieves the best performance on test set and the pre-trained CNN features combining context feature method ranks second. They all performed better than the method without context features. The patch-based experimental result proves that context features can be of benefit for scene text detection.



(a) (b) (c)

Figure 12: The output results after text and non-text classification by non-context and with-context methods. The different gray-scale value represents the text verification degree. (a) The original images. (b) Classification results by with-context method. (c) Classification results by non-context method.

4.3.2. Comparing the without-context method to our framework on scene text detection

After the patch-based test, we evaluate the Scheme (1) and (2) on the ICDAR 2013 test database and SVT test database. Scheme (1) is the method without using context features and Scheme (2) is the proposed method and achieves the best performance on the patch-based database. The experiments were implemented on the gray-scale channel and multi-channels that contain a and b channels of Lab color space. We follow the same evaluation scheme of the ICDAR-2013 competition which makes use of the framework proposed by Wolf and Jolion [51], and classify the matching to one-to-one, one-to-many and many-to-one matches.

Tab. 4 displays the comparison results on ICDAR 2013 database. From the comparison, we found that the recall of the non-context method is lower than the with-context method, but the precision is higher when evaluating on gray-scale images. Since the recall measures how many true text regions are

Table 4: Comparison Results of Scheme (1)and Scheme (2) on ICDAR 2013 Database.

Method name	Precision	Recall	F-score
Scheme 1 (gray)	88.35%	64.38%	74.49%
Scheme 2 (gray)	88.26%	66.12%	75.60%
Scheme 1 (gray+a+b)	85.76%	71.05%	77.72%
Scheme 2 (gray+a+b)	85.80%	74.34%	79.66%

Table 5: Comparison Results of Scheme (1)and Scheme (2) on SVT Database.

Method name	Precision	Recall	F-score
Scheme 1 (gray)	38.21%	29.21%	33.11%
Scheme 2 (gray)	42.33%	28.60%	34.14%
Scheme 1 (gray+a+b)	36.19%	35.09%	35.63%
Scheme 2 (gray+a+b)	40.55%	34.27%	37.15%

detected, the result illustrates that the with-context method can detect more text regions. The precision measures that the detected text region divide by all detected regions, so we can infer that there are more false detected regions in the with-context method. It may result from the scene segment results. Some undefined background regions are segmented to text streamers, cars or books on which the text appear with high probability. This might be a reason of the low precision. The segmentation results of the text regions are almost correct, so it results in high recall when adding context features. The comparison results on SVT database is shown in Tab. 5. Images in SVT database contains a lot of buildings, street and billboard. Scenes classed are simpler than that in ICDAR database. But text of SVT database are mostly in low resolution, that results in incorrect segmented scene. Thus, with the context, our method could exclude more background and get higher precision. Meanwhile, some text are classified to background with the incorrect scene context that decreases the recall. In

Fig. 12, a comparison result of the two methods is displayed. The intensity of the detected CCs represent the text score of the classifier's output. In the images of first and second lines, when the context are text streamers, the real text components can be detected. And the CCs in the wall of the third image are classified to non-text.

4.3.3. Comparison with state-of-the-art methods

The performances of the proposed algorithm as well as other methods on the ICDAR 2013 database and SVT database are shown in Tab. 6 and Tab. 7 respectively. We use the combined channels, gray-scale, a and b channels for comparison. The average running time of our method on MATLAB platform is

Table 6: Comparison of different methods on ICDAR 2013 database.

Method name	Precision	Recall	F-score
Xu's Method [52]	85.0%	78.0%	81.0%
Our method	85.80%	74.34%	79.66%
Liu's Method [53]	85.39%	68.72%	76.15%
USTB TexStar	88.47%	66.45%	75.89%
TextSpotter	87.51%	64.84%	74.49%
CASIA NLPR	78.89%	68.24%	73.18%
Baseline	60.76%	34.74%	44.21%

about 263 seconds by using a standard PC with 12 GHz Intel processor. Since we do not use a GPU to accelerate the CNN, the time cost is high. Otherwise, it could speed up by more than 20 times. SVT database should be used for cropped lexicon-driven word recognition or full image lexicon-driven word detection and recognition [21]. In our method, we do not use any lexicon system for text detection. The low resolution of images results in large scale of text candidates missing. Consequently, it causes low recall on SVT database by using our method. However, comparing to other state-of-the-art text detection algorithms, our method is comparable. This proves the achievement by applying

Table 7: Comparison of different methods on SVT database.

Method name	Precision	Recall	F-score
SnooperText [54]	36.0%	54.0%	43.0%
Our method	40.55%	34.27%	37.15%
Neumann[28]	19.0%	33.0%	26.0%
TESSFRONE+HOG	15.0%	15.0%	1.0%
TESSFRONT	4.0%	18.0%	6.0%

context information to scene text detection task. In other words, if we combine those methods with scene context, we can expect to enhance the performance further.

5. Conclusion

In this paper, we had the hypothesis that scene context could be of benefit for scene text detection. Meanwhile, we proposed a novel approach to prove it. It combined character feature generated by CNN with context feature extraction by TextonBoost and fully-connected CRF. The experiment results on the patch-based database, the ICDAR 2013 database and SVT database proved our hypothesis and also illustrated that the context based approach could perform better than some state-of-the-art methods in scene text detection.

Our work contained two fields of computer vision: scene understanding and scene text detection. We tried to build the connection between them to find the benefits. The results are encouraging. However, there are still two problems to be solved: more accurate scene segmentation and non-connected character detection. They correspond to improve precision and recall in text detection task respectively. Therefore, we will focus on the two aspects in our future work to find more benefits of them.

References

[1] L.-J. Li, R. Socher, L. Fei-Fei, Towards total scene understanding: Classification, annotation and segmentation in an automatic framework, in: Proceedings of the CVPR, 2009, pp. 2036–2043.

[2] A. Geiger, M. Lauer, C. Wojek, C. Stiller, R. Urtasun, 3D traffic scene understanding from movable platforms, IEEE Trans. PAMI 36 (5) (2014) 1012–1025.

[3] Q. Ye, D. Doermann, Text detection and recognition in imagery: A survey, IEEE Trans. PAMI 37 (7) (2014) 1480–1500.

[4] H. Zhang, K. Zhao, Y.-Z. Song, J. Guo, Text extraction from natural scene image: A survey, Neurocomputing 122 (2013) 310–323.

[5] T. Scharwächter, M. Enzweiler, U. Franke, S. Roth, Efficient multi-cue scene segmentation, Pattern Recognit. 8142 (2013) 435–445.

[6] C. Liu, D. Wang, J. Zhu, B. Zhang, Learning a contextual multi-thread model for movie/tv scene segmentation, IEEE Trans. Multimedia 15 (4) (2013) 884–897.

[7] P. Krähenbühl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, arXiv preprint arXiv:1210.5644.

[8] T. Wang, D. J. Wu, A. Coates, A. Y. Ng, End-to-end text recognition with convolutional neural networks, in: Proceedings of the ICPR, 2012, pp. 3304–3308.

[9] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2) (2004) 167–181.

[10] C. Pantofaru, C. Schmid, M. Hebert, Object recognition by integrating multiple image segmentations, in: Proceedings of the ECCV, 2008, pp. 481–494.

[11] D. Hoiem, A. N. Stein, A. Efros, M. Hebert, et al., Recovering occlusion boundaries from a single image, in: Proceedings of the CVPR, 2007, pp. 1–8.

[12] C. Cheng, A. Koschan, C.-H. Chen, D. L. Page, M. Abidi, et al., Outdoor scene image segmentation based on background recognition and perceptual organization, IEEE Trans. Image Process. 21 (3) (2012) 1007–1019.

[13] J. Shotton, J. Winn, C. Rother, A. Criminisi, Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, Int. J. Comput. Vis. 81 (1) (2009) 2–23.

[14] S. Gould, J. Rodgers, D. Cohen, G. Elidan, D. Koller, Multi-class segmentation with relative location prior, Int. J. Comput. Vis. 80 (3) (2008) 300–316.

[15] A. Levin, Y. Weiss, Learning to combine bottom-up and top-down segmentation, Int. J. Comput. Vis. 81 (1) (2009) 105–118.

[16] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the CVPR, 2005, pp. 886–893.

[17] D. Huang, C. Shan, M. Ardabilian, Y. Wang, L. Chen, Local binary patterns and its application to facial image analysis: a survey, IEEE Trans. Syst. Man Cybern. 41 (6) (2011) 765–781.

[18] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, C. Koch, Adaboost for text detection in natural scene, in: Proceedings of the ICDAR, 2011, pp. 429–434.

[19] 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 Trans. PAMI 25 (12) (2003) 1631–1639.

[20] X. Chen, A. L. Yuille, Detecting and reading text in natural scenes, in: Proceedings of the CVPR, 2004, pp. 366–373.

[21] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: Proceedings of the ICCV, 2011, pp. 1457–1464.

[22] L. Neumann, J. Matas, Scene text localization and recognition with oriented stroke detection, in: Proceedings of the ICCV, 2013, pp. 97–104.

[23] A. Mishra, K. Alahari, C. Jawahar, Top-down and bottom-up cues for scene text recognition, in: Proceedings of the CVPR, 2012, pp. 2687–2694.

[24] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proceedings of the CVPR, 2010, pp. 2963–2970.

[25] A. Mosleh, N. Bouguila, A. B. Hamza, Image text detection using a bandlet-based edge detector and stroke width transform., in: Proceedings of the BMVC, 2012, pp. 1–12.

[26] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, B. Girod, Robust text detection in natural images with edge-enhanced maximally stable extremal regions, in: Proceedings of the ICIP, 2011, pp. 2609–2612.

[27] L. Neumann, J. Matas, A method for text localization and recognition in real-world images, in: Proceedings of the ACCV, 2011, pp. 770–783.

[28] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proceedings of the CVPR, 2012, pp. 3538–3545.

[29] X.-C. Yin, X. Yin, K. Huang, H.-W. Hao, Robust text detection in natural scene images, IEEE Trans. PAMI 36 (5) (2014) 970–983.

[30] X.-C. Yin, W.-Y. Pei, J. Zhang, H.-W. Hao, Multi-orientation scene text detection with adaptive clustering, IEEE Trans. PAMI 37 (9) (2015) 1930–1937.

[31] L. Sun, Q. Huo, W. Jia, K. Chen, A robust approach for text detection from natural scene images, Pattern Recognit. 48 (9) (2015) 2906–2920.

[32] Y.-F. Pan, X. Hou, C.-L. Liu, A hybrid approach to detect and localize texts in natural scene images, IEEE Trans. Image Process. 20 (3) (2011) 800–813.

[33] W. Huang, Y. Qiao, X. Tang, Robust scene text detection with convolution neural network induced mser trees, in: Proceedings of the ECCV, 2014, pp. 497–511.

[34] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, <arXiv preprint arXiv:1502.01852>.

[35] A. Bissacco, M. Cummins, Y. Netzer, H. Neven, Photoocr: Reading text in uncontrolled conditions, in: Proceedings of the ICCV, 2013, pp. 785–792.

[36] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, Int. J. Comput. Vis. (2014) 1–20.

[37] L. Ladicky, C. Russell, P. Kohli, P. H. Torr, Associative hierarchical random fields, IEEE Trans. PAMI 36 (6) (2014) 1056–1077.

[38] B. Fulkerson, A. Vedaldi, S. Soatto, Class segmentation and object localization with superpixel neighborhoods, in: Proceedings of the ICCV, 2009, pp. 670–677.

[39] B. Triggs, J. J. Verbeek, Scene segmentation with crfs learned from partially labeled images, in: Proceedings of the NIPS, 2007, pp. 1553–1560.

[40] T. Hotta, Mean-field approximation, in: Nanoscale Phase Separation and Colossal Magnetoresistance, 2003, pp. 157–167.

[41] X. Yin, X.-C. Yin, H.-W. Hao, K. Iqbal, Effective text localization in natural scene images with mser, geometry-based grouping and adaboost, in: Proceedings of the ICPR, 2012, pp. 725–728.

[42] H. I. Koo, D. H. Kim, Scene text detection via connected component clustering and nontext filtering, IEEE Trans. Image Process. 22 (6) (2013) 2296–2305.

[43] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, Scene text detection using graph model built upon maximally stable extremal regions, Pattern Recognit. Lett. 34 (2) (2013) 107–116.

[44] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vis. Comput. 22 (10) (2004) 761–767.

[45] K. Chellapilla, S. Puri, P. Simard, High performance convolutional neural networks for document processing, in: Proceedings of the International Workshop on FHR, 2006.

[46] P. Sermanet, S. Chintala, Y. LeCun, Convolutional neural networks applied to house numbers digit classification, in: Proceedings of the ICPR, 2012, pp. 3288–3291.

[47] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (4) (2000) 411–430.

[48] A. Zhu, G. Wang, Y. Dong, Detecting natural scenes text via auto image partition, two-stage grouping and two-layer classification, Pattern Recognit. Lett. 67 (2015) 153–162.

[49] A. Shahab, F. Shafait, A. Dengel, Icdar 2011 robust reading competition challenge 2: Reading text in scene images, in: Proceedings of the ICDAR, 2011, pp. 1491–1496.

[50] P. Kohli, P. H. Torr, et al., Robust higher order potentials for enforcing label consistency, Int. J. Comput. Vis. 82 (3) (2009) 302–324.

[51] C. Wolf, J.-M. Jolion, Object count/area graphs for the evaluation of object detection and segmentation algorithms, Int. J. Doc. Anal. Recognit. 8 (4) (2006) 280–296.

[52] H. Xu, F. Su, Robust seed localization and growing with deep convolutional features for scene text detection, in: Proceedings of the ACM on ICMR, 2015, pp. 387–394.

[53] J. Liu, H. Su, Y. Yi, W. Hu, Robust text detection via multi-degree of sharpening and blurring, Signal Processing.

[54] R. Minetto, N. Thome, M. Cord, N. J. Leite, J. Stolfi, Snoopertext: A text detection system for automatic indexing of urban scenes, Comput. Vis. and Image Understanding 122 (2014) 92–104.

Anna Zhu received the B.S. degree from Huazhong University of Science and Technology, of China in 2011. She is currently pursuing Ph.D in School of Automation, Huazhong University of Science and Technology. She visited Human Interface Laboratory, Kyushu University of Japan in 2015. Her current research interests include text detection, image processing, and machine learning.

Renwu Gao received his M.S. degree from Nanjing university of science and techniques in 2012, and PhD degree from Kyushu university in Japan in 2015. His current interest researches include text detection, image processing and pattern recognition.

Seiichi Uchida received B.E. and M.E. and Dr. Eng. degrees from Kyushu University in 1990, 1992 and 1999, respectively. From 1992 to 1996, he joined SECOM Co., Ltd., Japan. Currently, he is a professor at Kyushu University. His research interests include pattern recognition and image processing. Dr. Uchida is a member of IEEE and IPSJ.