Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees

Weilin Huang[1,2], Yu Qiao[1], and Xiaoou Tang[2,1]

[1] Shenzhen Key Lab of Comp. Vis and Pat. Rec.,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2] Department of Information Engineering,
The Chinese University of Hong Kong, China

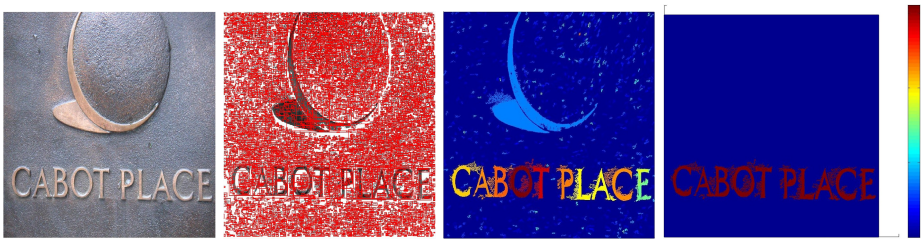
Abstract. Maximally Stable Extremal Regions (MSERs) have achieved great success in scene text detection. However, this low-level pixel operation inherently limits its capability for handling complex text information efficiently (e. g. connections between text or background components), leading to the difficulty in distinguishing texts from background components. In this paper, we propose a novel framework to tackle this problem by leveraging the high capability of convolutional neural network (CNN). In contrast to recent methods using a set of low-level heuristic features, the CNN network is capable of learning high-level features to robustly identify text components from text-like outliers (e.g. bikes, windows, or leaves). Our approach takes advantages of both MSERs and sliding-window based methods. The MSERs operator dramatically reduces the number of windows scanned and enhances detection of the low-quality texts. While the sliding-window with CNN is applied to correctly separate the connections of multiple characters in components. The proposed system achieved strong robustness against a number of extreme text variations and serious real-world problems. It was evaluated on the ICDAR 2011 benchmark dataset, and achieved over 78% in F-measure, which is significantly higher than previous methods.

Keywords: Maximally Stable Extremal Regions (MSERs), convolutional neural network (CNN), text-like outliers, sliding-window.

1 Introduction

With the rapid evolvement and popularization of high-performance mobile and wearable devices in recent years, scene text detection and localization have gained increasing attention due to its wide variety of potential applications. Although recent progresses in computer vision and machine learning have substantially improved its performance, scene text detection is still an open problem. The challenge comes from extreme diversity of text patterns and highly complicated background information. For example, texts appeared in a natural image can be in a very small size or in a low contrast against the background color, and even regular texts can be distorted significantly by strong lightings, occlusion, or blurring. Furthermore, a large amount of noise and text-like outliers, such as

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 497–511, 2014.



(a) (b) (c) (d)

Fig. 1. The text detection pipeline of our method. The input image is shown in (a). We first apply the MSERs operator on the input image to generate a number of text component candidates (b). We then apply the CNN classifier to generate a component confidence map (c). The components with positive confident scores are applied for constructing text-lines, which are scored by the mean values the components included. The final detection results are generated by a simple thresholding on (d).

windows, leaves, and bricks, can be included in the image background, and often cause many false alarms in detection.

There are mainly two groups of methods for scene text detection in the literature, sliding-window based and connected component based methods. The sliding-window based methods detect text information by sliding a sub-window in multiple scales through all locations of an image [11,3,9,28,29,18,1]. Text and non-text information is then distinguished by a trained classifier, which often uses manually designed low-level features extracted from the window, such as SIFT and Histogram of Oriented Gradients (HoG) [6]. The main challenge lies in the design of local features to handle the large variance of texts, and computational demand for scanning a large amount of windows, which may increase to N^2 for an image with N pixels. Hand crafted features like SIFT and HoG are effective to describe image content information, but these features are not optimized for text detection.

The connected component based methods achieved the state-of-the-art performance in scene text detection. They first separate text and non-text pixels by running a fast low-level filter and then group the text pixels with similar properties (e. g. intensity, stroke width, or color) to construct component candidates [23,24,22,34,7,31,10,32,2]. Stroke width transform (SWT) [7,31,10] and Maximally Stable Extremal Regions (MSERs) [16,23,24,22,34] are two representative low-level filters applied for scene text detection with great success. The main advantages of these methods are the computational efficiency by detecting text components in an one pass computation in complexity of $O(N)$, and providing effective pixel segmentations, which greatly facilitate the subsequent recognition task.

It has been shown that MSERs based methods have high capability for detecting most text components in an images [24]. However, they also generate a large number of non-text components at the same time, leading to high ambiguity

between text and non-text in MSERs components. Robustly separating them has been a key issue for improving the performance of MSERs based methods. Efforts have been devoted to handling this problem, but most of current methods for MSERs pruning focus on developing low-level features, such as heuristic characteristics or geometric properties, to filter out non-text components [23,24,22,34]. These low-level features are not robust or discriminative enough to distinguish true texts from text-like outliers, which often have similar heuristic or geometric properties with true texts. Besides, the MESRs methods are based on pixel level operations, and hence are highly sensitive to noise or single pixel corruption. This may lead to incorrect component connections, such as a single component includes multiple characters, which significantly affect the performance of text-line construction in the subsequent step.

In order to tackle these inherent problems, this paper aims to develop a robust text detection system by embedding the high-capability deep learning method into the MSERs model, and taking the advantages of both MESRs and sliding-window methods. The main contributions of the paper are:

1. We apply deep convolutional neural network to learn high-level features from the MSREs components. This high-capability classifier correctly distinguishes texts from a large amount of non-text components, and shows high discriminant ability and strong robustness against complicated background components (see Fig. 1 and 2), and therefore greatly improves capability of the MSERs based methods.
2. We incorporate the CNN classifier with sliding-window model and non-maximal suppression (NMS) to handle the multiple characters connection problem of the MSREs, and also recover missing characters, as shown in Fig. 3. Our method provides better character candidates than previous MSERs methods. This improvement is a crucial technique for bottom-up scheme to construct text-lines.
3. Our system have the advantages of both MSERs and sliding window methods. Comparing to traditional sliding-window methods, our method not only reduces the number of search window, but also enhances the detection of low contrast texts by using MSREs, as shown in Fig. 2 (a) and (b).
4. Our method achieves state-of-the-art results on the most cited ICDAR 2011 benchmark with over 78% in F-measure, which improves current results with a large margin.

The rest of paper is organized as follow. Section 2 describes all details of the proposed system. Experimental verifications are produced on Section 3, followed by the conclusions in Section 4.

2 Our System

The proposed text detection system includes three main steps, as shown in Fig. 1. Text components are first generated by applying the MSERs detector on the input image. Then, each MSERs component is assigned a confident value by

using a trained CNN classifier. Finally, the text components with high confident scores are employed for constructing the final text-lines. Besides, we also propose a novel approach to enhance character separation by applying siding window with the CNN classifier for error-connected MSERs components. Details of the system are presented bellow.

2.1 MSERs Component Generation

MSERs define an extremal region as a connected component of an image whose pixels have intensity contrast against its boundary pixels [16,25]. The intensity contrast is measured by increasing intensity values, and controls the region areas. A low contrast value would generate a large number of low-level regions, which are separated by small intensity difference between pixels. When the contrast value increases, a low-level region can be accumulated with current level pixels or merged with other lower level regions to construct a higher level region. Therefore, an extremal region tree can be constructed when it reaches the largest contrast (e.g. 255 in a gray-scale image). An extremal region is defined as a maximally stable extremal region (MSER) if its variation is lower than both its parent and child [16,25]. Therefore, a MSER can be considered as a special extremal region whose size remains unchanged over a range of thresholds.

The MSERs has been one of the most widely-used region detectors. For text region detection, it can be assumed that pixel intensity or color within a single text letter is uniform, while the intensity contrast between text and background regions typically exists. Each individual character can be detected as a extremal region or a MSER. Two promising properties make the MSER detector effective in text detection. First, the MSERs detector is computationally fast and can be computed in linear time of the number of pixels in a image [25]. Second, it is a powerful detector with high capability for detecting low quality texts, such as low contrast, low resolution and blurring. With this capability, MSER is able to detect most scene texts in a natural images, leading to high recall on the detection. However, the capability of the MSER is penalized by the increasing number of false detections. It would substantially increase the difficulty to identity true texts from a large number of non-text false alarms, which is one of the main challenge for current MSERs based methods. Previous work often balance the two factors by using a MSERs threshold, which can be changed from 1 to 255 for a gray-scale image.

For text detection system, our goal is to detect as many text components as possible in this step. Because it is difficult or impossible to recover the missed characters in the subsequent processes. The MSERs threshold is set to its lowest value 1 which makes it possible to capture most challenging cases, as shown in the top row of Fig. 2. As shown, although they are a number of error detections, the true text characters in highly difficult cases are also correctly detected. This makes it possible to construct a robust system to correctly detect those challenge texts and result in a high recall. But at the same time, it needs a powerful classifier to identify those low-quality texts from a large number of non-text

components. A high capability classifier based on deep convolutional neural network is present bellow.

2.2 Deep Convolutional Neural Network

Deep learning has been applied to a number of challenging tasks in computer vision with breakthrough improvements achieved in last several years. It has been shown that deep network is capable of leaning meaningful high-level features and semantic representations for visual recognition through a hierarchal architecture with multiple-layers feature convolutions. The deep structure of the CNN allows it to refine feature representation and abstract sematic meaning gradually. The traditional CNN network has achieved great success on digit and hand-written character recognition [12,13]. 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. In contrast to previous works, which often use a set of heuristic features to distinguish text and non-text components [23,24,22,34,31,10], we take the advantages of deep learning and adapt

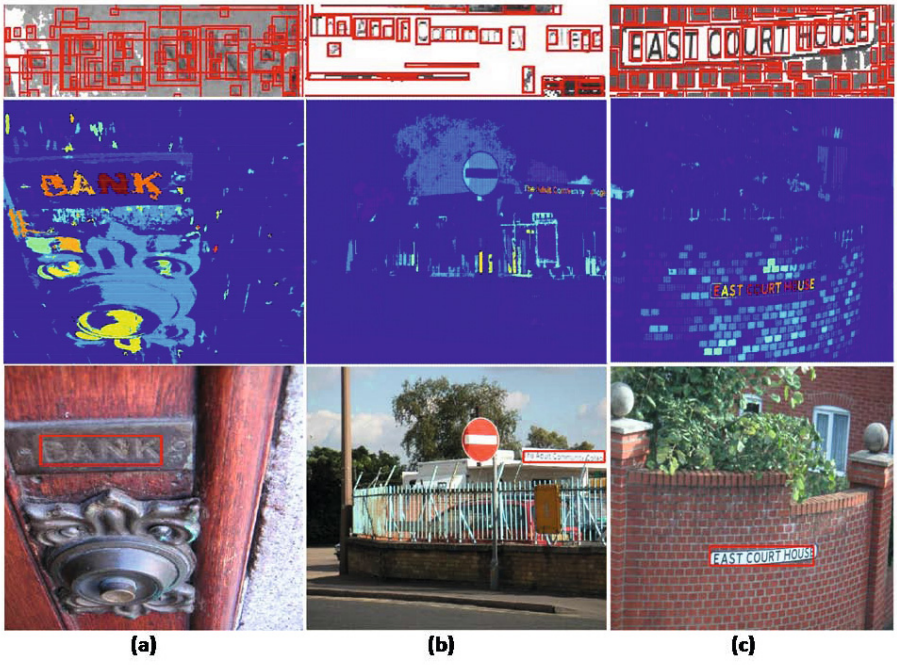


Fig. 2. The performance of the MSERs detector and the CNN classifier for low contrast (a) and low quality texts (b), and text-like outliers (c). The top row are the MESRs detections on text areas. The middle row are the confident maps generated by the CNN classifier. The pixels are displayed by their higher confident scores if they belong to multiple components. The bottom row are the detection results.

a deep convolutional neural network to robustly classify the generated MESRs components.

The structure of the CNN text classifier is similar to that applied in [29,4]. The network includes two convolutional layers, each of which has a convolution and an average pooling steps. The number of filters for the two layers are 64 and 96, respectively. The input patch is with fixed size of 32×32. Similar to [29,4,5,19], the first layer is trained in an unsupervised way by using a variant of K-means described in [4] to learn a set of filters $D \in \mathbb{R}^{k \times n_1}$ from a set of 8×8 patches randomly extracted from training patches, k is the dimension of the convolution kernel, and is 64 for the kernel size with 8×8, $n_1 = 64$ is the number of the filters in the first layer in our system. The responses (r) of the first layer is computed as [29],

$$r = \max\{0, |D^T \mathbf{x}| - \theta\} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{64}$ is an input vector for an 8×8 convolutional kernel, and $\theta = 0.5$. The resulted first layer response maps are with size of $25 \times 25 \times 64$. Then average pooling with window size of 5×5 is applied to the response maps to get reduced maps with the size of $5 \times 5 \times 64$. The second layer is stacked upon the first layer. The sub-window patch for computing response and average pooling is 4×4 and 2×2, respectively. The final output of the two layers is a 96 dimension feature vector, which is input to a SVM to generate the final confident score of the MSERs component. The parameters in the second layer are fully connected and are trained by back-propagating the SVM classification error.

Given a MESR component, we applied the trained CNN classifier to decide whether it is a text component by assigning a confident score to it. In our experiments, we discarded the MSERs components which include very small numbers of pixels (e.g. less than 0.01% of the total pixel number in an image), and keep all other components as input to the CNN classifier. For each retained MSER component, we computed the aspect ratio of its boundary box. If the width of the box is larger than its height, we directly resized the image component into the size of 32×32; otherwise, we extracted a square patch with the same center of the boundary box and with the side length equal to the height of box, and then resized it into 32×32. This alignment scheme makes the input patches consistent with the synthetic training samples used in our experiments, which were originally generated by Wang *et. al.* [28,29]. Two examples for both cases are shown in Fig. 3. The final confident maps for three challenge images are shown in the middle row of the Fig. 2. As shown, our CNN classifier generally assigns higher scores to text components, even for those MSERs with very low quality of the text characters (see Fig. 2 (a) and (b)), and at the same time, classify the text-like outliers (such as the masks and bricks in Fig. 2 (b) and (c)) as low confident scores, demonstrating strong robustness and highly discriminative capability for filtering the non-text components.

The performance of the SWT methods highly depend on the edge detector, which is often not feasible in many challenge cases. Compared to the SWT based methods, MSER operator is capable of detecting more true text components,

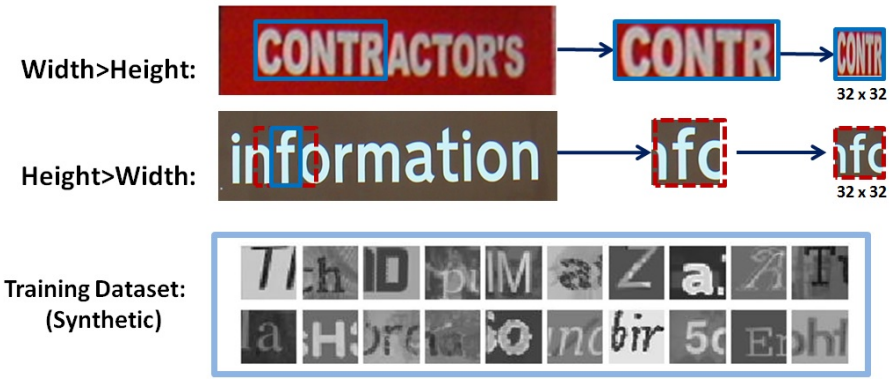


Fig. 3. MESRs component alignment with size of 32×32 and synthetic training samples

often leading to a higher recall. But at the same time, the MESRs methods also generate a larger number of non-text components. It means that the high performance of the MSERs methods heavily depend on a powerful component classifier. Thus we designed the CNN network by leveraging its high learning capability to improve the performance of the MSERs methods. Besides, in our system, the MESRs method and the CNN classifier are strongly compensated to each other. Comparing to general siding-window methods, the MSER operator provides two promising properties. It not only reduces the number of searching windows dramatically by two orders of magnitude, but also provides a significant enhancement on low quality texts, which are difficult to be detected correctly by a general sliding-window method. In our experiment, the average number of the MSERs components in an image input to the CNN classifier is 516 in the ICDAR 2011 database.

2.3 Component Splitting with MSERs Tree

As pointed out in the literature, most connected component based methods suffer from inherent limitations of low-level filters, which easily cause error connections between multiple characters or with background components in some difficult cases, such as low quality or seriously affected texts [24,2,10]. In order to tackle this problem, we proposed a high-level approach by incorporating CNN scores and MSERs tree structure with a sliding window model.

We define an error-connected component as a MSER component including multiple text characters. As mentioned, implementation of the sliding-window model is computationally expensive. We show that only a small number of the MSERs components are considered as the error-connected components by selecting them using the output CNN scores and the structure of MSERs tree. It can be observed that an error-connected component generally has three remarkable characteristics.

- First, it often has a high aspect ratio where the width of the boundary box is much longer than its height (e.g. the example in the top row of Fig. 3).
- Second, differing from other non-text components, such as long horizontal lines or bars which are generally scored with negative confident values by our CNN classifier, the error-connected component actually includes some text information (multiple characters), but it is not very strong, because our CNN classifier is trained on single-character components.
- Third, although the components in high-level of the MSERs trees often include multiple text characters, such as the components in the roots of the tree. Most of these components can be further separated correctly by their children components, which often have higher confident scores than their parents. Thus, we do not consider these components as the error-connected components.

Therefore, the error-connected components are defined as the components having high aspect ratios (e.g. $width/height > 2$ in our experiments) and positive CNN scores, (1) but cannot be further separated in their MSERs trees; or (2) all components in their children sets do not have a higher CNN score than them. The first situation includes the texts having multiple characters truly connected, which cannot be separated by most low-level filters. The components in the second situation often include error separations (resulted in the low or negative confident scores for all their children components), which are caused by some challenging cases.

To present the proposed splitting scheme, we selected an error-connected MSER component sample, as shown in Fig 4. The component has high aspect ratio and positive CNN scores, which is higher than the scores of its children. We applied a sliding window with our CNN classifier to scan through the component, which returns an one dimension continuous confident scores. Finally, non-maximal suppression (NMS) method [20] was applied to the continuous scores to estimate the locations for each single characters. The details of the component splitting method are described in Algorithm 1.

As shown in Fig. 4, the proposed high-level component splitting method effectively handles the component connection problem of the MSERs methods, and often generates better component candidates with high confident values for subsequent text-line construction and recognition. Note that, by integrating MSERs tree structure and CNN confident map for carefully choosing the error-connected components, only a small number of components are selected for splitting. While each component is scanned just once by a sliding window with a single scale (32×32) and the size of component is often small. Therefore, the increase of the computational cost for the proposed splitting method is relatively trivial. With powerful CNN classifier and efficient splitting scheme, a large number of non-text components have been identified correctly and only a small number of text components (with positive confident scores in our experiments) are retained to construct the final text-lines.

Fig. 4. Error-connected component slitting by sliding window model with the CNN classifier

Algorithm 1. Error-connected Component Splitting

Require: Selected error-connected components, MSERs Tree Structure and CNN confident map.

Ensure: Revised MSERs Tree and CNN confident map.

Given N error-connected components
for $k = 1 \rightarrow N$ **do**
  Get the confident score of the current component, W_k
  Normalize the size of component into $32 \times X$
  Use sliding window (32×32) with CNN to compute confident scores S_k
  Apply NMS [20] for estimating the peak values (P_k) in S_k as,

$$P_k(x) = \begin{cases} S_k(x) & \text{if } S_k(x) \geqslant S_k(x + \Delta x), \Delta x < \Theta \\ 0 & otherwise \end{cases} \tag{2}$$

  Generate new components at location x, where $P_k(x) > 0$, $\{P_k^1, P_k^2, \ldots, P_k^m\}$
  if $\max(P_k^1, P_k^2, \ldots, P_k^s) > W_k$ **then**
    Replace children set of current component with new generated ones
    Update confident map with new scores and locations, as shown in Fig 4
  end if
end for
return Revised MSERs tree and new CNN confident map

The text-line construction is now simple and straightforward. Similar to previous work in [7,10], we first grouped two neighboring components into a pair if they have similar geometric and heuristic properties, such as similar intensities, colors, heights, and aspect ratios. Then, the pairs including a same component

and having similar orientations were merged sequentially to construct the final text-lines. The process is ended when no pairs can be merged further. Finally, text-lines were broken into separate words by computing the horizontal distances between consecutive characters. This is different from Yin *et al.*'s method [34], which is difficult to sperate text and non-text MSERs components discriminatively by using heuristic features, and a large number of non-text components are retained to construct the text-lines, leading to a large number of false alarms included in the resulted text-lines (e.g. as indicated in [34], only 9% of the final text-lines are true texts.). It often requires a further computationally costly post-processing to filter out the false alarms by using sophisticated machine learning algorithms [34]. In contrast, our system discards the false alarms by simply thresholding the average confident scores of the text-lines.

3 Experiments and Results

We evaluated the proposed method on two widely cited benchmarks for scene text detection, the ICDAR 2005 [15,14] and the ICDAR 2011 [26] Robust Reading Competition databases.

3.1 Datasets and Evaluation Method

The ICDAR 2005 dataset includes 509 color images having sizes varied from 307×93 to 1280×960. 258 images are included in the training set, while 251 images are used for test. There are 229 training images and 255 testing ones for the ICDAR 2011 dataset. The detection performance were evaluated in the word level in both datasets, which include totally 1114 and 1189 words in their test sets, respectively.

For evaluation, we followed the ICDAR 2011 competition evaluation protocol, which was proposed by Wolf et al. [30]. This evaluation method presents object level precision and recall based on constraints on detection quality. It evaluates both quantity and quality of rectangle matches through all images in the database, and considers not only one-to-one matching, but also one-to-many and many-to-one matchings. The quality of detection or matching is controlled by two parameters which penalizes more on parts matching than larger detection. Specifically, the evaluation is computed by *Precision*, *Recall*, and *F-measure* which are defined bellow,

$$Precision = \frac{\sum_i^N \sum_j^{|D^i|} M_D(D_j^i, G^i)}{\sum_i^N |D^i|} \tag{3}$$

$$Recall = \frac{\sum_i^N \sum_j^{|G^i|} M_G(G_j^i, D^i)}{\sum_i^N |G^i|} \tag{4}$$

$$F_measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

where N is the total number of images in a dataset. $|D^i|$ and $|G^i|$ are the number of detection and ground true rectangles in the i-th image. $M_D(D^i_j, G^i)$ and $M_G(G^i_j, D^i)$ are the matching scores for detection rectangle D_j and ground true rectangle G_j. Their values are set to 1 for one-to-one matching, 0.8 for one-to-many matching and 0 for no matching. Two rectangles are considered as matched when their overlapping ratio is higher than a defined threshold, which controls the quality of the matching.

3.2 Experiments and Results

The CNN classifier was trained by using 15000 toy samples generated by Wang et al. [29]. There are 5000 positive and 10000 negative samples in the training dataset, and all samples are resized into 32×32. Some examples are shown in Fig. 3. The training data on the two datasets were not applied for training in our experiments, which shows strong generalization power of the proposed system. In our experiments, the MSERs operator was run twice on each image, corresponding to both black-to-white and white-to-black texts. Each MSER component was classified by the trained CNN classifier and only the component with positive confident score was used for text-line construction. A text-line with an average component score lower than 1.2 was considered as a fail detection. The final detected boundary boxes for each image are the non-overlap combination of the boxes from both sides. The full evaluation results on the two databases are presented in Table 1 and 2, along with the detection results on several challenging images displayed in Fig. 5.

The proposed method achieved excellent performance on both databases and the improvements are significant in terms of *Precision*, *Recall*, and $F_{measure}$. In the most recent ICDAR 2011 dataset, our method improved the most closed performance with $2 \sim 3\%$ in all three terms and reached the $F_{measure}$ score over 78%. Note that the evaluation scheme of the SFT-TCD method [10] did not follow the standard protocol of the ICDAR 2011. It was evaluated based on each single image and the mean values of all images in the dataset were reported. The improvements by our method mainly gain from two facts. On the one hand, the powerful MSERs detector is able to detect most true texts, which resulted in a high *Recall*. One the other hand, high capability of the CNN classifier with high-level splitting scheme robustly identify true text components from non-text ones, leading to a large improvement on *Precision*.

Fig. 5 shows the successful detection results on a number of challenging cases, which indicate that our system is highly robust to large variations in texts including small font size, low contrast, and blurring. The images also show that our system is robust against strong lighting and highly noise background effects. Fig. 6 shows two failure cases in our experiments. For the left one, our method missed a number of true text-lines. It is mainly caused by the strong masks covering the texts, which significantly break the low-level structure of texts. The text components in the right image do not include strong text information and are easily confused with its background.

Fig. 5. Successful text detection results with extreme variances and significant affects

Fig. 6. Failure cases

Table 1. Experimental results on the ICDAR 2005 dataset

Method	Year	*Precision*	*Recall*	$F-measure$
Our method	–	**0.84**	**0.67**	**0.75**
SFT-TCD [10]	2013	0.81	0.74	0.72
Yao *et al.* [31]	2012	0.69	0.66	0.67
Chen *et al.* [2]	2012	0.73	0.60	0.66
Epshtein *et al.* [7]	2010	0.73	0.60	0.66
Yi and Tian [33]	2013	0.71	0.62	0.63
Neumann and Matas [23]	2011	0.65	0.64	0.63
Zhang and Kasturi [35]	2010	0.73	0.62	–
Yi and Tian [32]	2011	0.71	0.62	0.62
Becker *et al.* [14]	2005	0.62	0.67	0.62
Minetto *et al.* [17]	2010	0.63	0.61	0.61
Chen and Yuille [3]	2004	0.60	0.60	0.58

Table 2. Experimental results on the ICDAR 2011 dataset

Method	Year	*Precision*	*Recall*	$F-measure$
Our method	–	**0.88**	**0.71**	**0.78**
Yin *et al.* [34]	2014	0.86	0.68	0.76
Neumann and Matas [21]	2013	0.85	0.68	0.75
SFT-TCD [10]	2013	0.82	0.75	0.73
Neumann and Matas [22]	2013	0.79	0.66	0.72
Shi *et al.* [27]	2013	0.83	0.63	0.72
Neumann and Matas [24]	2012	0.73	0.65	0.69
González *et al.* [8]	2012	0.73	0.56	0.63
Yi and Tian [32]	2011	0.67	0.58	0.62
Neumann and Matas [23]	2011	0.69	0.53	0.60

4 Conclusions

We have presented a robust system for scene text detection and localization in natural images. Our main contribution lies in effectively leveraging the high capacity of the deep learning model to tackle two main problems of current MSERs methods for text detection, and enable our system with strong robustness and highly discriminative capability to distinguish texts from a large amount of non-text components. A sliding window model was intergraded with the CNN classifier to further improve text character detection on challenging images. Our method has achieved the state-of-the-art performance on two benchmark datasets, convincingly verifying the efficiency of the proposed method.

Acknowledgments. This work is supported by National Natural Science Foundation of China (913201 01), Shenzhen Basic Research Program (JCYJ20120903092050890, JCYJ2012061 7114614438, JCYJ20130402113127496), 100 Talents Programme of Chinese Acad emy of Sciences, and Guangdong Innovative Research Team Program (No. 201001 D0104648280). Yu Qiao is the corresponding author.

References

1. Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: Photoocr: reading text in uncontralled conditions (2013), ICCV
2. Chen, H., Tsai, S., Schronth, G., Chen, D., Grzeszczuk, R., Girod, B.: Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In: ICIP (2012)
3. Chen, X., Yuille, A.: Detecting and reading text in natural scenes. In: CVPR (2004)
4. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D.J., Ng, A.Y.: Text detection and character recognition in scene images with unsupervised feature learning. In: ICDAR (2011)
5. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: AISTATS (2011)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
7. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR (2010)
8. González, A., Bergasa, L., Yebes, J., Bronte, S.: Text location in complex images. In: ICPR (2012)
9. Hanif, S., Prevost, L.: Text detection and localization in complex scene images using constrained adaboost algorithm. In: ICDAR (2009)
10. Huang, W., Lin, Z., Yang, J., Wang, J.: Text localization in natural images using stroke feature transform and text covariance descriptors. In: ICCV (2013)
11. Kim, K., Jung, K., Kim, J.: Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. IEEE Trans. Pattern Analysis and Machine Intelligence 25, 1631–1639 (2003)
12. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W.: Handwritten digit recognition with a back-propagation network. In: NIPS (1989)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324 (1998)
14. Lucas, S.: Icdar 2005 text locating competition results. In: ICDAR (2005)
15. Lucas, S., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: ICDAR (2003)
16. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal region. In: BMVC (2002)
17. Minetto, R., Thome, N., Cord, M., Fabrizio, J., Marcotegui, B.: Snoopertext: A multiresolution system for text detection in complex visual scenes. In: ICIP (2010)
18. Mishra, A., Alahari, K., Jawahar, C.V.: Top-down and bottom-up cues for scene text recognition. In: CVPR (2012)
19. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning
20. Neubeck, A., Gool, L.: Efficient non-maximum suppression. In: ICPR (2006)

21. Neumann, L., Matas, J.: On combining multiple segmentations in scene text recognition. In: ICDAR (2013)
22. Neumann, L., Matas, J.: Scene text localization and recognition with oriented stroke detection. In: ICCV (2013)
23. Neumann, L., Matas, K.: Text localization in real-world images using eficiently pruned exhaustive search. In: ICDAR (2011)
24. Neumann, L., Matas, K.: Real-time scene text localization and recognition. In: CVPR (2012)
25. Nistér, D., Stewénius, H.: Linear time maximally stable extremal regions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 183–196. Springer, Heidelberg (2008)
26. Shahab, A., Shafait, F., Dengel, A.: Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In: ICDAR (2011)
27. Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S.: Scene text detection using graph model built upon maximally stable extremal regions. Pattern Recognition 34, 107–116 (2013)
28. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: ICCV (2011)
29. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural network. In: ICPR (2012)
30. Wolf, C., Jolion, J.-M.: Object count/area graphs for the evaluation of object detection and segmentation algorithms. International Journal on Document Analysis and Recognition 8, 280–296 (2006)
31. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: CVPR (2012)
32. Yi, C., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans. Image Processing 20, 2594–2605 (2011)
33. Yi, C., Tian, Y.: Text extraction from scene images by character appearance and structure modeling. Computer Vision and Image Understanding 117, 182–194 (2013)
34. Yin, X.C., Yin, X., Huang, K., Hao, H.W.: Robust text detection in natural scene images. IEEE Trans. Pattern Analysis and Machine Intelligence (to appear)
35. Zhang, J., Kasturi, R.: Character energy and link energybased text extraction in scene images. In: ACCV (2010)