2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Scene Text Recognition with CNN Classifier and WFST-based Word Labeling

Xinhao Liu, Takahito Kawanishi, Xiaomeng Wu and Kunio Kashino

NTT Corporation, Kanagawa, Japan 243-0198
Emails: {liu.xinhao, kawanishi.takahito, wu.xiaomeng, kashino.kunio}@lab.ntt.co.jp

***Abstract*—Natural scene text recognition has proved to be challenging due to the unconstrained wild conditions. In this paper, to solve this problem we propose a method which first detects and recognizes characters by utilizing the high performance Convolutional Neural Network (CNN). Then for post-processing, inspired by its success in speech recognition, we employ the efficient and flexible Weight Finite State Transducer (WFST) based word labeling model for incorporation with a lexicon or high order language model. In the experiments we show that the proposed approach can correctly and robustly recognize the text in the scene images and the results for serveral public datasets (ICDAR 2003, SVT and IIIT5K) show comparable or superior performance to the state-of-the-art algorithms.**

I. Introduction

Optical Character Recognition (OCR) techniques have been developed and widely used for years. However, these conventional OCR systems are limited to well formatted printed documents and perform poorly for the natural scene text which also contains a lot of semantic value. Text in a natural scene is more difficult due to the unconstrained conditions such as complex backgrounds, nonuniform text sizes and distances, illumination changes and so on. Thus how to handle these problems and recognize the text from the natural scenes has attracted a lot of attention both from the academic and industry side in recent years.

According to [1], an end-to-end text recognition system can be divided into two phases: text localization and word recognition. In this paper we focus on the word recognition task which consists of identifying characters and recognizing them as text format from a given image patch. The challenges for this task come from that the character detection and recognition module needs to be accurate for the un-segmented image and word labeling module needs to be efficient and scalable with a lexicon or high order language model.

To solve the above problems, we propose using the CNN-based character recognition model integrated with a WFST based word labeling approach for efficient and accurate text recognition. The discriminative features learned from CNN models have been proved to be very successful for image classification tasks. In this work we build an accurate character recognition model based on the and Fractional Max Pooling (FMP) [2] structure and Network in Network (NIN) [3] module. Then for post-processing, inspired by the success of the Weight Finite State Transducer (WFST) in the speech recognition [4], we develop a WFST based word labeling approach which encodes a mapping from the CNN labels to desired lexicon word. The benefit of the WFST based representation is that it is not only efficiently handle the ambiguous set of CNN labels, but also very flexible to incorporate with high order lexicon and language models. In summary, our contribution is that a robust word recognition approach based on CNN and WFST classification models is proposed. In the experiments, we show that the proposed approach performs comparably or better for character and word recognition tasks on several public datasets.

The rest of the paper is organized as follows: related works for character and word recognition are described in Section II, the proposed approach is described in Sections III, The experimental setting and results are shown in detail in Section IV. Conclusions and future directions are given in Section V.

II. Related work

A. Character recognition

The character recognition problem has been studied extensively for decades. For the natural scene text, the methods based on feature descriptors has been widely used recently. As studied in [5], [6], classical HOG based features outperforms better that other features. In [7] multi-scale features called *strokelets* are learned to describe the structure of the characters and in [8] a low-dimensional attribute method is proposed to encode the characters. Despite being improved, the discriminative power of these feature representation is still limited. Coates et al. [9] and Wang et al. [10] proposed using the unsupervised learned Convolutional Neural Network (CNN) features to detect and recognize the texts. After that, the CNN with different network structures [11], [12], [13] has been proposed for character recognition, which greatly improve the accuracy.

B. Word recognition

For character-to-word recognition, a general strategy is to combine the classifier and language model to yield optimal score for the recognition. Existing work try to detect and find possible characters for a particular word using graph based structures driven by a lexicon [14], [15] or high order language priors [16]. But the character or word structure could be easily violated due to the complexity of natural scene. In [17], [8] the authors tried to use subspace methods or learned mid-level features to represent the whole word image

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

from a global aspect and then recognize the word image as a retrieval process. However, the discrimination power of the representation is still not enough. In [11], [13] the optimal score is derived from a lexicon-specific character segmentation with an exhaustive search of the most confident location and size of each character. The disadvantage of these algorithms is that it is constrained by the pre-defined lexicon and the exhaustive search of optimal score makes the system complex and inefficient.

III. Proposed approach

The pipeline of the proposed algorithm for word recognition is shown in Fig. 1. A CNN based character classifier is firstly trained offline. The characters are detected and classified with a dense sliding window manner. Then we explore the relations of the detected candidate characters by a WFST based representation method. The details of each step are described in the following sections.

A. CNN model for Character Classification

Let $C \in \{1, 2, ...62\}$ denotes a character set which has 62 classes (26 upper case, 26 lower case letters and 10 digits). An character classifier is to identify which category c a given image patch x belongs. Motivated by the success of CNN, in this work we also adopt the CNN model to improve the character classification accuracy. Instead of using the unsupervised learned features as Wang et al. [10], we train the recognition model in a supervised way. Many techniques and network structures have been proposed to improve the classification accuracy, in our experiments we empirically found the *Fractional Max-Pooling* (FMP) [2] structure gives better performance.

In FMP, the classic fixed sized pooling region is replaced by a random mixtures of $1 \times 1, 1 \times 2, 2 \times 1,$ or 2×2 regions to increase the randomness of the pooling process and prevent over-fitting. Moreover, each time a different random sequence of pooling regions is generated and therefore an FMP network can also be considered as an ensemble of multiple similar networks. To improve the performance we employ this model for character recognition and the detailed structure is shown in Fig. 2. A basic FMP block contains two convolutional layers and one FMP layer. The convolutional layers are with kernel size 2×2 and 1×1. The 1×1 convolutional layer is similar as Network in Network (NIN) [3] module which can be considered as a micro-network to increase the non-linearity of the model. N_ℓ is the number of output feature channels for the ℓ-th block. We set $N_\ell = 32\ell$ and decay factor $\alpha = \sqrt[3]{2}$ as suggested in [2]. For the overall network architecture, a total of $\ell = 12$ blocks are used and followed by another 2 convolutional layers with kernel size 2×2 and 1×1 before the final prediction layer. The final output layer is 62-way softmax layer which each element indicates the score for each category. Dropout [18] is also used to prevent over-fitting.

B. WFST-based Word Labeling

1) CNN Classifier Label Inference: Once the character classifier is trained, the next step is to recognize the cropped image as a whole word. Many algorithms try to find the optimal break point for each character from the word image [11], [13]. However, due to the complex background of the scene image, it is very difficult of segment the word image into characters correctly and efficiently. To avoid incorrect segmentation, we employ the standard sliding window manner to detect characters and then mapped the candidate labels to the desired results with the WFST model.

Given a cropped word image, we first resize it to a fixed height $h = 32$ pixels while keeping its aspect ratio. Then we evaluate the classifier response for each $h \times h$ window in a sliding window manner with a step size $sz = 2$ pixels. From this step we can obtain a $62 \times T$ score matrix $\mathbf{M}$, where T is the number of sliding windows which is determined by the image width, sliding window width and the step size sz. To make the CNN score matrix case insensitive, we set the score for a character at the larger value between its upper and lower case. To eliminate the repeated detections, we perform Non Maximum Suppression (NMS) as introduced in [19], [10] over $\mathbf{M}$ to select the columns where a character is most likely to be present. For each column of $\mathbf{M}$, a response score $R(t)$ is computed as the difference between the best and second best scores in that column.Then the NMS response score $\tilde{R}(t)$ can be calculated as:

$$\tilde{R}(t) = \begin{cases} R(t) & \text{if} \quad R(t) \geqq R(t'), \forall t' \, |t - t'| < \delta \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where δ is a width parameter, which we set at 5 pixels in this work. The columns with response zero are regarded as non-text. The most probable labeling of the word image is derived as the characters with the maximum response score according to $\tilde{R}(t)$. However, the CNN classifier labels at this point are usually error-prone due to the unsegmented characters and should be further corrected using a language model or a lexicon.

2) Word Labeling with WFST-based Model: The CNN classifier labels inevitably include errors, and high order language models or lexicons are usually utilized as constraints to eliminate these errors. The lexicon is a list of words which is assumed to have the ground truth word. In [5], a graph based cost function and in [10] a lexicon-driven segmentation based score function are utilized to find the best matched words. In this work, to model the ambiguity among the detected candidate set of CNN labels, we employ the WFST based representation model which has been proved successful for many tasks such as speech recognition [4], statistical machine translation and OCR post-processing [20], [21]. Unlike the existing algorithms, the proposed approach can take both the advantages of deep neural network and WFST. A WFST can be seen as a finite directed graph with nodes representing states and arcs representing transitions Each transition is labeled with an input symbol and a output symbol from the character set $\{C \cup \epsilon\}$ and a transition weight, where ϵ represents a blank

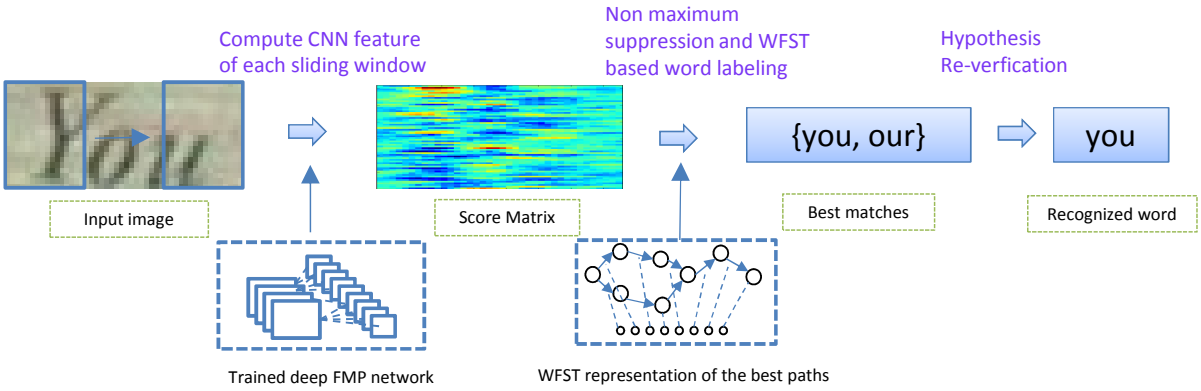

Fig. 1. Pipeline of the proposed algorithm for word recognition.

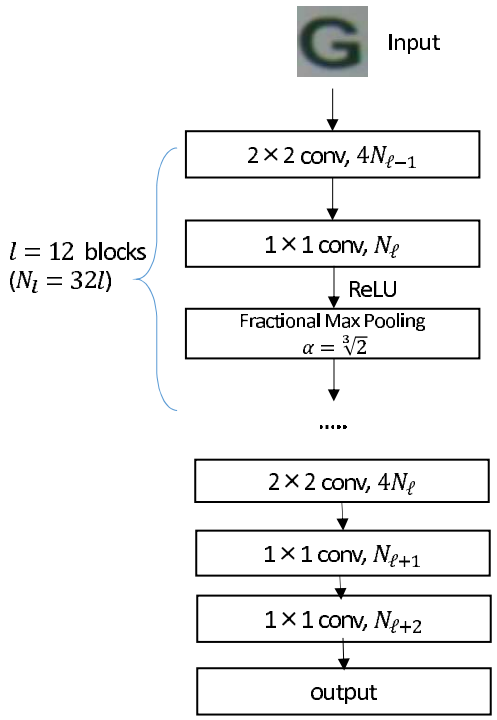

Fig. 2. Network stucture for natural scene character classification.

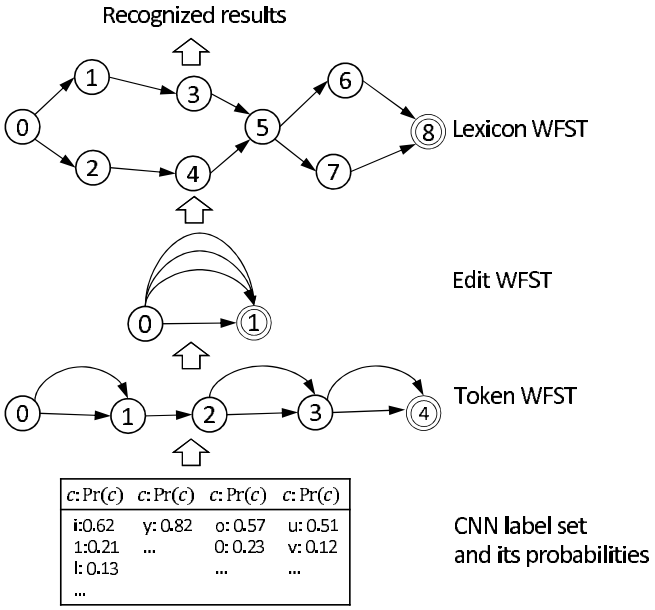

Fig. 3. The CNN induced WFST word labeling processing

or empty symbol. The sequence of non-maximum suppressed CNN labels and their probabilities are utilized to build the *token* WFST T. In this work, instead of the complete set of hypotheses, we use top-3 candidate labels for each frame. The transitions of a *edit* WFST E represent three basic edit operations: substitution, insertion and deletion. Given two symbols c_1, c_2 and the empty symbol ϵ, substitution, insertion and deletion are transductions of type c_1/c_2, ϵ/c_2 and c_1/ϵ respectively. The cost of the operations are set to be 1. To identify the correct set of strings, we can simple use a list of words to build a *lexicon* WFST or more complex sentences and grammars to represent the language model. Thus, the composition of these three WFSTs

$$H = T \circ L \circ G \tag{2}$$

gives an efficient search graph H that maps the ambiguous classifier labels to a sequence of words restricted to lexicon word, where $\circ$ denotes the composition operator of WFST [4]. An illustration of the labeling process is shown in Fig. 3.

For the task of cropped word image recognition only lexicon WFST are used, but the proposed approach can be easily extended for incorporation with a higher order language model to recognize text lines etc.

3) Hypothesis Word Re-verification: For the lexicon words which has same cost with the CNN labels, we also perform a second round verification operation following the method in [22]. Given the recognition score matrix $\mathbf{M}$, the score for the hypothesis word $W = \{c_1, c_2, ..., c_N\}$ with length N can be

derived as:

$$S(W, \mathbf{M}) = \frac{1}{N} \sum_{i=1}^{N} \max_{\Delta \in B} \mathbf{M}(c_i, p_i + \Delta), \quad (3)$$

where $\mathbf{M}(c_i, p_i + \Delta)$ is the recognition score for the character c_i centered at position $p_i + \Delta$. We assume that each character is equally distributed within the image width, so the expected position p_i of each character c_i can be easily obtained on the basis of the width of the score matrix and the word length. $B = [-\delta, \delta]$ is a set of neighbors around p_i and we set $\delta = 5$ in this work. The word with the best score is recognized as the final result.

IV. Experimental results

A. Datasets

There are several public available dataset for scene text detection and recognition. For training the same data is used as in [13] which includes 107k samples collected from the ICDAR 2003, 2005, 2013 training sets, KAIST [23] and Chars74k [24] dataset. We use real-time data augmentation in which for each iteration before the forward pass, the training samples are processed with 1) random affine transforms and 2) random photometric distortion to the intensity of the image as in [2]. This does not increase the increase the actual training data size and training time, but gives more training samples.

We evaluate the proposed algorithm by using word images from the ICDAR 2003, SVT-WORD and IIIT5K word [16] datasets. For ICDAR 2003 test set, same as existing works, words with fewer than two characters or non-alphanumeric are ignored, leaving 860 word images. The SVT-WORD test set contains 647 word images cropped from the Google Street Views. Many of them are low-resolution, curved, noisy or blurred images. In IIIT5K, 5000 word images collected from Internet are split into 2000 and 3000 for the purpose of training and testing. The recognizer is tested with a different sized lexicon. The lexicon is built with the ground truth with a number of random distractors. The lexicon with all words in the ICDAR 2003 test set forms the ICDAR03-full dataset and 50 random selected words [14] forms the ICDAR03-50 dataset. SVT-WORD dataset provides a lexicon of 50 words [5]. For IIIT5K dataset, we use the small and medium sized lexicons which contains 50 and 1000 words.

B. Character Recognition

For the ICDAR 2003 test set there are 5430 characters and for the SVT dataset, only the test set is annotated with a character level bounding box by [25] and it contains 3796 test samples. The experimental results in Table I show the character classification performance of different algorithms. The results with 1 test and average of 12 tests are reported. We can see that the proposed algorithm outperforms existing methods both for case-sensitive and case-insensitive including methods based on CNN models such as Maxout network [13]. Although tests with model averaging increase the computational cost, the performance can also be improved.

TABLE I
Character classification accuracy (%) on the ICDAR 2003 and SVT dataset

	Case sensitive		Case insensitive	
Method	ICDAR03	SVT	ICDAR03	SVT
T. Wang [10]	83.9	-	-	-
Tian [6]	79.4	75.4	83.6	80.6
Alsharif [12]	86.0	-	89.8	-
Jaderberg [13]	86.8	-	91.0	80.3
Liu [22]	89.5	75.4	93.1	84.4
Proposed (1 test)	90.5	78.0	93.9	83.7
Proposed (12 tests)	**92.2**	**81.1**	**95.6**	**86.7**

C. Word Recognition

We compare the proposed algorithm with recent proposed algorithms which are based on feature representation [7], [8], CNN [10], [13], [12], graph structure [14], [15] and so on. Table II show the cropped word recognition results for different algorithms. The number "50" or "1000" in the bracket indicates the number of the words in the pre-defined lexicon and "full" means all words from ICDAR03 test set are included in the lexicon. Since the FMP increases the randomness of the network, the proposed algorithm is reported with the mean and the standard deviation of 10 runs. For each run the posterior probabilities of each class are obtained with an average of 12 tests. We can see that the proposed algorithm outperforms all the existing methods on the tested datasets except [26] and algorithms trained with large scale dataset [27] which set strong baselines for the word recognition task. In [26], bidirectional RNN with CNN features are utilized to learn the context dependent relations among the sliding windows. It is proved to be very effective for word sequence labeling and recognition. In [27] N-gram word recognition model are trained with 9 million word images. The large scale training also boost the performance. The advantage of proposed algorithm is that it is only based on unary character classifier and solving the ambiguous sequence of characters by the WFST word representation. The WFST model is efficient to handle all the uncertainties and it does not require a training process. Moreover, it is also very flexible for incorporating with higher language models. Note that the proposed approach performs well on IIIT5K dataset even thought we do not use its training samples, which indicates the adaptivity and generalization ability of the proposed approach.

In terms of speed, testing a Intel Xeon E5-2650 2.6 GHz $\times$ 2 machine, without model averaging the proposed algorithm currently runs at about 0.25 seconds per sample on average with GPU on IIIT5K test set (1000 sized lexicon) and 0.83 seconds with CPU only. It generally compares favorably with others. It is reported that for recognizing a cropped word image the computation time is 0.77 seconds for [8] and 1.4

Fig. 4. Examples from the SVT-WORD dataset: correctly recognized images.

seconds for [11]. We believe that greater efficiency can be better reflected when more complex language model or large lexicon has to be taken into account.

Fig. 4 shows some correctly recoginzed examples by the proposed algorithm from SVT-WORD dataset and Fig. 5 shows some failures. Fig. 5 (a) are the word images with slanted or curved texts. For this kind of images, the failure comes from the sliding window based detection which can not find correct location of the characters. If preprocessing steps such as correcting the distortions or inferring the text region are applied, these images are not difficult to recognize. Fig. 5 (b) are images which are low resolution images, fancy font styles and so on. Some of them are even difficult for humans to read. A effective learning based solution for this kind of failure is to increase the training set size to include more samples as [27], or one can develop image enhancement algorithms to improve the image quality.

(a)

(b)

Fig. 5. Examples from the SVT-WORD dataset: incorrectly recognized ones.

V. Conclusion

In this paper, we proposed an efficient and accurate algorithm for natural scene cropped word recognition which integrated the CNN and WFST classification models. Experimental results for character and word recognition on different datasets showed that the performance of the proposed algorithm is comparable or superior to the state-of-the-art. The reasons that our method outperforms existing works are that: 1) The features learned from the FMP structure are more robust to background variations of the natural scene images. 2) Unlike the graph structure based method [14], [15] which may easily fail for some complex cases, the WFST based representation can efficiently handle the ambiguous set of labels, and maps them to the lexicon or language. In future work, we will continue to improve the recognition system by enhancing the image quality through pre-processing to deal with the failure cased in Fig. 5 or incorporating a sequence classifier such as RNN [29] or structural classification [30] for post-processing.

References

[1] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–366.

[2] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[3] M. Lin, Q. Chen, and Yan, "Network in network," in *International Conference on Learning Representation*, 2014.

[4] M. Mohri, F. Pereira, and M. Riley, "Weighted Finite-State Transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[5] K. Wang and S. Belongie, "Word spotting in the wild," in *European Conference on Computer Vision*. Springer, 2010, pp. 591–604.

[6] S. Tian, S. Lu, B. Su, and C. L. Tan, "Scene text recognition using Co-occurrence of Histogram of Oriented Gradients," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 912–916.

[7] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 4042–4049.

[8] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 12, pp. 2552–2566, 2014.

[9] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 440–445.

[10] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3304–3308.

[11] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 785–792.

[12] O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid HMM maxout models," in *International Conference on Learning Rep-*

TABLE II
CROPPED WORD RECOGNITION ACCURACY (%) ON THE ICDAR 2003, SVT-WORD AND IIIT5K DATASETS. THE NUMBERS IN THE BRACKET ARE LEXICON SIZE. THE PROPOSED ALGORITHM IS REPORTED WITH THE MEAN AND STD. DEV. OF 10 RUNS.

Method	ICDAR03 (50)	ICDAR03 (full)	SVT-WORD (50)	IIIT5K (50)	IIIT5K (1000)
Wang et al.[14]	76.0	62.0	57.0	-	-
Mishar et al. [16]	81.8	67.8	73.2	-	-
Wang & Wu [10]	90.0	84.0	70.0	-	-
Shi et al. [15]	87.4	79.3	73.5	-	-
Yao et al. [7]	88.5	80.3	75.9	80.2	69.3
Su et al. [28]	0.92	0.82	0.83	-	-
Almazan et al. [8]	-	-	87.0	88.6	75.6
Alsharif et al. [12]	93.1	88.6	74.3	-	-
Gordo et al. [17]	-	-	90.7	93.3	86.6
Jaderberg et al. [13]	96.2	91.5	86.1	-	-
He et al. [26]	**97.0**	**93.8**	**93.5**	94.0	**91.5**
Proposed	96.8±0.31	92.2±0.54	92.5±0.37	**94.1±0.20**	84.7±0.28
	Training with large additional dataset				
PhotoOCR [11]	-	-	90.4	-	-
Jaderberg et al. [27]	**98.7**	**98.6**	**95.4**	**97.1**	**92.7**

resentation, 2014.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *European conference on computer vision*. Springer, 2014, pp. 512–528.

[14] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1457–1464.

[15] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2961–2968.

[16] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *BMVC 2012-23rd British Machine Vision Conference*. BMVA, 2012.

[17] A. Gordo, "Supervised mid-level features for word image representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2956–2964.

[18] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Pattern Recognition, 18th International Conference on*, vol. 3. IEEE, 2006, pp. 850–855.

[20] R. Llobet, J.-R. Cerdan-Navarro, J.-C. Perez-Cortes, and J. Arlandis, "OCR post-processing using weighted finite-state transducers," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 2021–2024.

[21] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *European Conference on Computer Vision*. Springer, 2012, pp. 752–765.

[22] X. Liu, T. Kawanishi, X. Wu, and K. Kashino, "Scene text recognition with high performance cnn classifier and efficient word inference," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 1322–1326.

[23] (2011) Kaist Scene Text Database. http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database.

[24] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, February 2009.

[25] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2687–2694.

[26] P. He, W. Huang, Y. Qiao, C. L. Chen, and X. Tang, "Reading scene text in deep convolutional sequences," in *The 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

[27] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[28] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 35–48.

[29] A. Graves *et al.*, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[30] Y. Kubo, S. Watanabe, T. Hori, and A. Nakamura, "Structural classification methods based on Weighted Finite-State Transducers for automatic speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 8, pp. 2240–2251, 2012.