

Accurate Scene Text Recognition Based on Recurrent Neural Network

Bolan Su^(✉) and Shijian Lu

Institute for Infocomm Research, 1 Fusionopolis Way,
21-01 Connexis, Singapore, Singapore
{subl,slu}@i2r.a-star.edu.sg

Abstract. Scene text recognition is a useful but very challenging task due to uncontrolled condition of text in natural scenes. This paper presents a novel approach to recognize text in scene images. In the proposed technique, a word image is first converted into a sequential column vectors based on Histogram of Oriented Gradient (HOG). The Recurrent Neural Network (RNN) is then adapted to classify the sequential feature vectors into the corresponding word. Compared with most of the existing methods that follow a bottom-up approach to form words by grouping the recognized characters, our proposed method is able to recognize the whole word images without character-level segmentation and recognition. Experiments on a number of publicly available datasets show that the proposed method outperforms the state-of-the-art techniques significantly. In addition, the recognition results on publicly available datasets provide a good benchmark for the future research in this area.

1 Introduction

Reading text in scenes is a very challenging task in Computer Vision, which has been drawing increasing research interest in recent years. This is partially due to the rapid development of wearable and mobile devices such as smart phones, digital cameras, and the latest google glass, where scene text recognition is a key module to a wide range of practical and useful applications.

Traditional Optical Character Recognition (OCR) systems usually assume that the document text has well defined text fonts, size, layout, etc. and scanned under well-controlled lighting. They often fail to recognize camera-captured texts in scenes, which could have little constraints in terms of text fonts, environmental lighting, image background, etc., as illustrated in Fig. 1.

Intensive research efforts have been observed in this area in recent years and a number of good scene text recognition systems have been proposed. One approach is to combine text segmentation with existing OCR engines, where text pixels are first segmented from the image background and then fed to OCR engines for recognition. Several systems have been reported that exploit Markov Random Field [5], Nonlinear color enhancement [6] and Inverse Rendering [7] to extract the character regions. However, the text segmentation process by itself is a very challenging task that is prone to different types of segmentation errors.

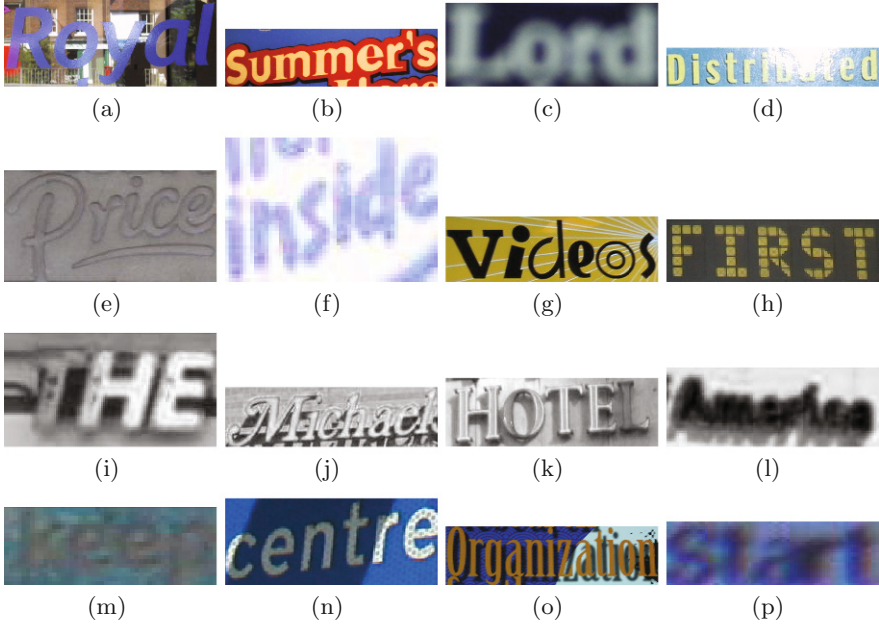


Fig. 1. Word image examples taken from the recent Public Datasets [1–4]. All the words in the images are correctly recognized by our proposed method.

Furthermore, the OCR engines may fail to recognize the segmented texts due to special text fonts and perspective distortion, as the OCR engines are usually trained on characters with fronto-parallel view and normal fonts.

A number of scene text recognition techniques [4, 8–11] have been reported in recent years that tend to train their own scene character classifiers. Most of these methods follow a bottom-up approach to group the recognized characters into a word based on the context information. The grouping process to form a word is usually defined as finding the best word alignment that fits the set of detected characters. Lexicon and n-gram language model are also incorporated as a top-down clue to recover some common errors, such as spelling and ambiguities [6, 7, 10]. On the other hand, these new techniques also require robust and accurate character-level detection and recognition, otherwise the word alignment will lead to incorrect result due to the error accumulation from lower levels to higher levels.

In this paper, we propose a novel scene text recognition technique that treats a word image as an unsegmented sequence and does not require character-level detection, segmentation, and recognition. Figure 2 shows the overall flowchart of our proposed word recognition system. First, a word image is converted into a sequence of column feature, where each column feature is generated by concatenating the HOG features extracted from the corresponding image patches in the same column of the input image. A multi layer recurrent neural network (RNN) with bidirectional Long Short-Term Memory (LSTM) [12] is then trained for

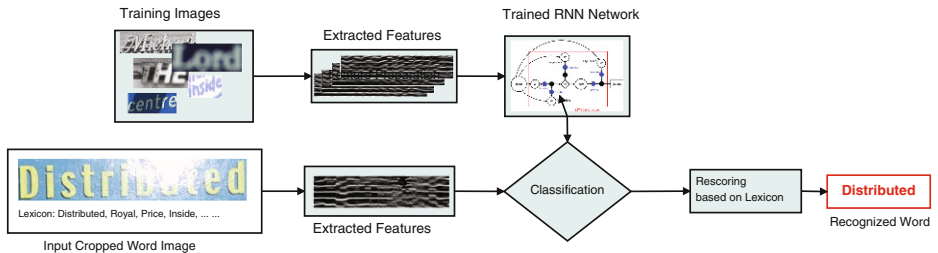


Fig. 2. The overall flowchart of our proposed scene word recognition system.

labelling the sequential data. Finally, the connectionist temporal classification (CTC) [13] technique is exploited to find out the best match of a list of lexicon words based on the RNN output of the sequential feature using.

We apply our proposed technique on cropped word image recognition with a lexicon. The cropped word denotes a word image that is cropped along the word bounding box within the original image as illustrated in Fig. 1. Such word regions can be located by a text detector or with user assistance in different real-world scenarios. The lexicon refers to a list of possible words associated with the word image, which can be viewed as a form of contextual information. It is obvious that the search space can be significantly narrowed down in real world applications. For example, the lexicon can be nearby signboard names collected using Google when recognizing signboards at one location [4]. In other cases, the lexicon can be food names, sporter names, product lists, etc., depending on different scenarios.

The proposed scene text recognition technique has a number of novel contributions. First, we describe an effective way of converting a word image into a sequential signal where techniques used in relevant areas, such as speech processing and handwriting recognition, can be introduced and applied. Second, we adapt RNN and CTC techniques for recognition of texts in scenes, and designed a segmentation-free scene word recognition system that obtains superior word recognition accuracy. Third, unlike some systems that rely heavily on certain local dataset (which are not available to the public) [8, 14], our system makes use of several publicly available datasets, hence providing a baseline for easier benchmarking of the ensuing scene text recognition techniques.

2 Related Work

2.1 Scene Text Recognition

In general, the word recognition in natural scene consists of two main steps, text detection and text recognition. The first step usually detects possible regions as character candidates, the second step recognizes the detected regions and groups them into word.

Different visual feature detectors and descriptors have been exploited for the detection and recognition of texts in scenes. The Histogram of Oriented Gradient (HOG) feature [15] is widely used in different methods [4, 10, 16]. Recently,

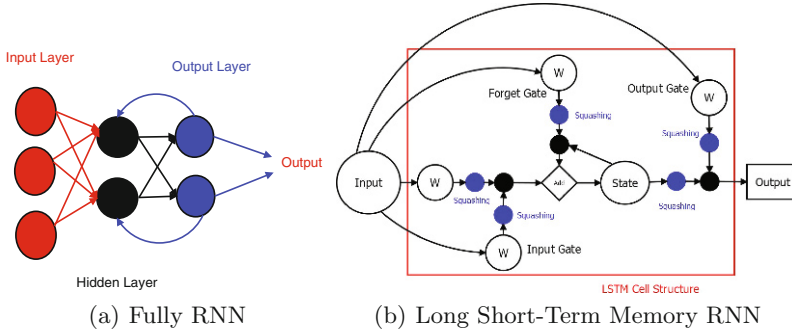


Fig. 3. Illustration of Recurrent Neural Network.

the part-based tree structure [9] is also proposed along with the HOG descriptor to capture the structure information of different characters. In addition, the Co-occurrence HOG feature [17] is developed to represent the spatial relationship of neighbouring pixels. Other features including stroke widths (SW) [18, 19], maximally stable extremal regions (MSER) [20, 21] and weighted direction code histogram (WDCH) [8] have also been applied for character detection and classification. The HOG feature is also exploited in our proposed system for construction of sequential features of word images.

With large amount training data, the systems reported in [8, 14, 22] obtain high recognition accuracy with unsupervised feature learning techniques. Moreover, different approaches are proposed to group the detected characters into word using contextual information, including Weighted Finite-State Transducers [11], Pictorial Structure [4, 16], and Conditional Random Field Model [9, 10]. We propose a novel scene text recognition technique that treat each word image as a whole without requiring character-level segmentation and recognition. The CTC technique [13] is exploited to find the best word alignment for each unsegmented column feature vector and so character-level segmentation and recognition is not needed.

2.2 Recurrent Neural Network

Recurrent Neural Network (RNN) is a special neural network that has been used for handling sequential data, as illustrated in Fig. 3(a). The RNN aims to predict the label of current time stamp with the contextual information of past time stamps. It is a powerful classification model but not widely used in the literature. The major reason is that it often requires a long training process as the error path integral decays exponentially along the sequence [23].

The long short-term memory (LSTM) model [23] was proposed to solve this problem as illustrated in Fig. 3(b). In LSTM, an internal memory structure is used to replace the nodes in the traditional RNN, where the output activation of the network at time t is determined by the input data of the network at time t and

the internal memory stored in the network at time $t - 1$. The learning procedure under LSTM therefore becomes local and constant. Furthermore, a forget gate is added to determine whether to reset the stored memory [24]. This strategy helps the RNN to remember contextual information and withdraw errors during learning. The memory update and output activation procedure of RNN can be formulated in Eq. 1 as follows:

$$S^t = S^{t-1} \cdot f(W_{forget}X^t) + f(W_{in}X^t) \cdot f(W_cX^t) \quad (1a)$$

$$Y^t = f(W_{out}X^t) \cdot g(S^t) \quad (1b)$$

where S^t, X^t, Y^t denote the stored memory, input data, and output activation of the network at time t , respectively. Functions $g(*)$ and $f(*)$ refer to the sigmoid function that squashes the data. W_* denotes the weight parameters of the network. In addition, the first term ($W_{forget}X^t$) is used to control whether to withdraw previous stored memory S^{t-1} .

Bidirectional LSTM [13, 25] is further proposed to predict the current label with past and future contextual information by processing the input sequence in two directions (i.e. from beginning to end and, from end to beginning). It has been applied for handwriting recognition and outperforms the widely-used Hidden Markov Model (HMM). We introduce the bidirectional LSTM technique into scene text recognition domain, and the powerful model obtains superior recognition performance.

3 Feature Preparation

To apply the RNN model, the input word image needs to be first converted into a sequential feature. In speech recognition, the input signal is already a sequential data and feature can be directly extracted frame by frame. Similarly, a word image can be viewed as a sequential array if we take each column of the word image as a frame. This idea has been applied for handwriting recognition [13] and achieved great success.

However, the same procedure cannot be applied on the scene text recognition problem due to two factors. First, the input data of the handwriting recognition task is usually binary or has a clear bi-modal pattern, where the text pixels can be easily segmented from the background. Second, the handwritten text usually has a much smaller stroke width variation compared with texts in scenes, where features extracted from each column of handwriting images contains more meaningful information for classification.

On the other hand, quite a number of visual features in computer vision such as HOG are extracted in image patches, which perform well in text recognition due to their robustness to illumination variation and invariance to the local geometric and photometric transformations. We first convolutionally partition the input image into patches with step size 1, which is described in Algorithm 1 below.

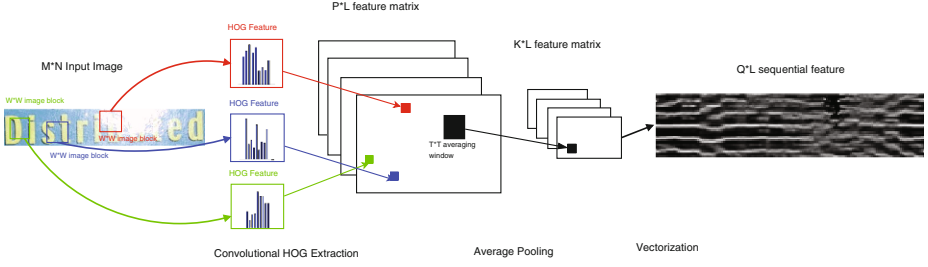


Fig. 4. The overall process of column feature construction. From left to right, HOG features are extracted convolutionally from every image blocks of the input image exhaustively; Average pooling is then performed on each column; Finally, the vectorization process is applied to obtain the sequential feature. It is worth to note that, $P = M - W + 1$, $L = N - W + 1$, $K = P/T$, and $Q = K * H$, where H denotes the length of the HOG feature.

Algorithm 1. Convolutional Image Patch Extraction

```

1: procedure PATCHEXTRACTION( $img, W$ )
2:    $M$  : height of  $img$ 
3:    $N$  : width of  $img$ 
4:    $Patches$  :  $(M-W+1) \times (N-W+1)$  matrix, each entry is a  $W \times W$  matrix.
5:   for  $i = 1:M-W+1$  do
6:     for  $j = 1:N-W+1$  do
7:        $Patches(i,j) = img(i:i+W-1,j:j+W-1)$ 
8:     end for
9:   end for
10:  Return  $Patches$ 
11: end procedure

```

where img denotes the input word image, W denotes the size of one image patch. The HOG feature is then extracted and normalized for each image patch. After that, the HOG features of the image patches at the same column are linked together. An average pooling strategy is applied to incorporate information of neighbouring blocks by averaging the HOG feature vectors within a neighbouring window as defined in Eq. 2.

$$HOG(i, j) = \sum_{p=i-T/2}^{i+T/2} (HOG(p, j))/T \quad (2)$$

where i, j refer to index, HOG denotes the extracted normalized HOG feature vector of corresponding patch, HOG_{avg} denotes the feature vector after averaging pooling, and T denotes the size of neighbouring window for average pooling. A column feature is finally determined by concatenating the averaged HOG feature vectors at the same column.

Figure 4 shows the overall process of column feature vector construction. To ensure that all the column features have the same length, the input word image needs to be normalized to be of the same height M beforehand. Furthermore, the patch size W and neighbouring windows size T can be set empirically, and will be discussed in the experiment section.

4 Recurrent Neural Network Construction

A word image can thus be recognized by classifying the correspondingly converted sequential column feature vector. We use the RNN [23] instead of the traditional neural network or HMM due to its superior characteristics in several aspects. First, unlike the HMM that generates observations based only on the current hidden state, RNN incorporates the context information including the historical states by using the LSTM structure [13] and therefore outperform the HMM greatly. Second, unlike the traditional neural network, the bidirectional LSTM RNN model [13] does not require explicit labelling of every single column vector of the input sequence. This is very important to the scene text recognition because characters in scenes are often connected, broken, or blurred where the explicit labelling is often an infeasible task as illustrated in Fig. 1. Note that RNNLIB [26] is implemented to build and train the multi-layer RNN.

CTC [13] is applied to the output layer of RNN to label the unsegmented data. In our system, a training sample can be viewed as a pair of input column feature and a target word string $(\mathbf{C}, \mathcal{W})$. The objective function of CTC is then defined as follows:

$$\mathcal{O} = - \sum_{(\mathbf{C}, \mathcal{W}) \in \mathcal{S}} \ln p(\mathcal{W}|\mathbf{C}) \quad (3)$$

where \mathcal{S} denotes the whole training set and $p(\mathcal{W}|\mathbf{C})$ denotes the conditional probability of word \mathcal{W} given a sequence of column feature \mathbf{C} . The target is to minimize \mathcal{O} , which is equivalent to maximize the conditional probability $p(\mathcal{W}|\mathbf{C})$.

The output path π of the RNN output activations has the same length of the input sequence \mathbf{C} . Since the neighbouring column feature vectors might represent the same character, some column feature vectors may not represent any labels. An additional ‘blank’ output cell therefore needs to be added into the RNN output layer. In addition, the repeating labels and empty labels also need to be removed to map to the target word \mathcal{W} . For example, $(\text{'-'}, a', a', \text{'-'}, \text{'-'}, b', b', b')$ can be mapped to (a, b) , where ‘-’ denotes the empty label. So the $p(\mathcal{W}|\mathbf{C})$ is defined as follows:

$$p(\mathcal{W}|\mathbf{C}) = \sum_{V(\pi)=\mathcal{W}} p(\pi|\mathbf{C}) \quad (4)$$

where V denotes the operator that translates the output path π to target word \mathcal{W} . It is worth to note that the translation process V is not unique. $p(\pi|\mathbf{C})$ refers

to the conditional probability of output path π given input sequence \mathbf{C} , which is defined as follows:

$$p(\pi|\mathbf{C}) = \prod_{t=1}^L p(\pi_t|\mathbf{C}) = \prod_{t=1}^L y_{\pi_t}^t \quad (5)$$

where L denotes the length of the output path and π_t denotes label of output path π at time t . The term y^t denotes the network output of RNN at time t , which can be interpreted as the probability distribution of the output labels at time t . Therefore $y_{\pi_t}^t$ denotes the probability of π_t at time t .

The CTC forward backward algorithm [13] is then applied to calculate $p(\mathcal{W}|\mathbf{C})$. The RNN network is trained by back-propagating the gradient through the output layer based on the objective function as defined in Eq. 3. Once the RNN is trained, it can be used to convert a sequential feature vector into a probability matrix. In particular, the RNN will produce a $L \times G$ probability matrix \mathbf{Y} given an input sequence of column feature vector, where L denotes the length of the sequence, and G denotes the number of possible output labels. Each entry of \mathbf{Y} can be interpreted as the probability of a label at a time step.

5 Word Scoring with Lexicon

Given a probability matrix \mathbf{Y} and a lexicon \mathcal{L} with a set of possible words, the word recognition can be formulated as searching for the best match word w^* as follows:

$$w^* = \arg \max_{w \in \mathcal{L}} p(w|\mathbf{Y}) = \arg \max_{w \in \mathcal{L}} \sum_{V(\pi)=w} p(\pi|\mathbf{Y}) \quad (6)$$

where $p(w|\mathbf{Y})$ is the conditional probability of word w given \mathbf{Y} . A direct graph can be constructed for the word w so that each node represents a possible label of w . In another word, we need to sum over all the possible paths that can form a word w on the probability matrix \mathbf{Y} to calculate the score of a word w .

A new word w^i can be generated by adding some blank interval into the beginning and ending of w as well as the neighbouring labels of w , where the blank interval denotes the empty label. The length of w^i is $2 * |w| + 1$, where $|w|$ denotes the length of w . A new $|w^i| \times L$ probability matrix \mathfrak{P} can thus be formed, where $|w^i|$ denotes the length of w^i and L denotes the length of the input sequence. $\mathfrak{P}(m, t)$ denotes the probability of label w_m^i at time t , which can be determined by the probability matrix \mathbf{Y} . Each path from $\mathfrak{P}(1, 1)$ to $\mathfrak{P}(|w^i|, L)$ denotes a possible output π of word w , where the probability can be calculated using Eq. 5.

The problem thus changes to the score accumulation along all the possible paths in \mathfrak{P} . It can be solved with the CTC token pass algorithm [13] using dynamic programming. The computational complexity of this algorithm is $O(L \cdot |w^i|)$. Finally, the word with highest score in the lexicon is determined as the recognized word.

6 Experiments and Discussion

6.1 System Details

In the proposed system, all cropped word images are normalized to be of the same height, i.e., $M = 32$. The patch size W , the HOG bin number, and the averaging window size T are set to 8, 8, and 5, respectively. For RNN, the number of input cells is the same as the length of the extracted column feature at 40. The output layer is 64 including 62 characters ([a...z,A...Z,0...9]), one label for special characters ([+,&,\$,...]), and one empty label. 3 hidden layers are used that have 60, 100, and 140 cells, respectively. The system is implemented on Ubuntu 13.10 with 16 GB RAM and Intel 64 bit 3.40 GHz CPU. The training process takes about 1 h on a training set with about 3000 word images. The average time for recognizing a cropped word image is around one second. This is comparable with the state-of-the-art techniques and can be further improved to satisfy the requirement of real-time applications.

6.2 Experiments on ICDAR and SVT Datasets

The proposed method has been tested on three public datasets, including ICDAR 2003¹ dataset, ICDAR 2011² dataset, and Street View Text (SVT)³ dataset. The three datasets consist of 1156 training images and 1110 testing images, 848 training images and 1189 testing images, 257 training images and 647 testing images, respectively.

During the experiments, we add the Char74k character images⁴ [27] to form a bigger training dataset. For the Char74k dataset, only English characters are used that consists of more than ten thousands of character images in total. In particular, 7705 characters images are obtained from natural images and the rest are hand drawn using a tablet PC. The trained RNN is applied to the testing images of the three datasets for word recognition.

We compare our proposed method with eight state-of-the-art techniques, including markov random field method (MRF) [5], inverse rendering method (IR) [7], nonlinear color enhancement method (NESP) [6], pictorial structure method (PLEX) [16], HOG based conditional random field method (HOG+CRF) [10], weighted finite-state transducers method (WFST) [11], part based tree structure method (PBS) [9] and convolutional neural network method (CNN) [14].

To make the comparison fair, we evaluate recognition accuracy on testing data with a lexicon created from all the words in the test set (as denoted by ICDAR03(FULL) and ICDAR11(FULL) in Table 1), as well as with lexicon consisting of 50 random words from the test set (as denoted by ICDAR03(50) and ICDAR11(50) in Table 1). For SVT dataset, a lexicon consisting of about 50 words is provided and directly adopted in our experiments.

¹ <http://algoval.essex.ac.uk/icdar/Datasets.html>.

² <http://robustreading.opendfki.de/wiki/SceneText>.

³ <http://vision.ucsd.edu/~kai/grocr/>.

⁴ <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>.

Table 1. Word recognition accuracy on the ICDAR 03 testing dataset, the ICDAR 11 testing dataset and the SVT dataset.

Datsets	ICDAR03 (Full)	ICDAR03 (50)	ICDAR11 (Full)	ICDAR11 (50)	SVT
MRF [5]	0.67	0.69	-	-	-
IR [7]	0.75	0.77	-	-	-
NESP [6]	0.66	-	0.73	-	
PLEX [16]	0.62	0.76	-	-	0.57
HOG + CRF [10]	-	0.82	-	-	0.73
PBS [9]	0.79	0.87	0.83	0.87	0.74
WFST [11]	0.83	-	0.56	-	0.73
CNN [14]	0.84	0.90	-	-	0.70
Proposed	0.82	0.92	0.83	0.91	0.83

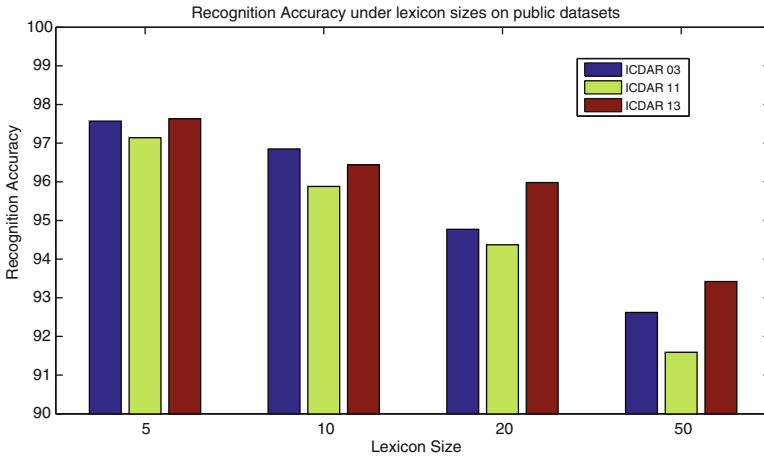
**Fig. 5.** Recognition accuracy of our proposed method on ICDAR 03, ICDAR 11 and ICDAR 13 datasets when the lexicon size is different.

Table 1 shows word recognition accuracy of the proposed technique and the compared techniques. The text segmentation methods (MRF, IR, and NESP) produce lower recognition accuracy than other methods because robust and accurate scene text segmentation by itself is a very challenging task. Our proposed method produces good recognition results on all the testing datasets. Especially in the SVT data set, our proposed method achieves 83% word recognition accuracy, which outperforms the state-of-the-art methods significantly. The deep learning technique using convolutional neural network [14] also produces good performances but it requires a much larger training dataset together with a portion of synthetic data which are not available to the public.

We also tested our proposed method on the ICDAR 2011 and 2013 word recognition dataset for Born-Digital Images (Web and Email). It achieves 92% and 94% recognition accuracies, respectively, which clearly outperforms other

methods as listed on the competition websites (with best recognition accuracy at 82.9%⁵ and 82.21%⁶, respectively). The much higher recognition accuracy also demonstrates the robustness and effectiveness of our proposed word image recognition system.

In addition, the correlation between lexicon size and word recognition accuracy of our proposed method is investigated. Figure 5 shows the experimental results based on the ICDAR03, ICDAR11, and ICDAR13 datasets. In addition, four lexicon sizes are tested that consist of 5, 10, 20, and 50 words, respectively. As illustrated in Fig. 5, the word recognition performance can actually be further improved when some prior knowledge is incorporated and the lexicon size is reduced.

We further apply our proposed method on the recent ICDAR 2013 Rubust Text Reading Competition dataset [3]⁷, where 22 algorithms from 13 different research groups have been submitted and evaluated under the same criteria. The data of ICDAR 2013 is a actually subset of ICDAR 2011 dataset where a small number of duplicated images are removed.

The winning PhotoOCR method [8] makes use of a large multi-layer deep neural network and obtains 83% accuracy on the testing dataset. As a comparison, our proposed method achieved 84% recognition accuracy. Note that our proposed method incorporates a lexicon with around 1000 words, whereas PhotoOCR method does not use lexicon. The lexicon is generated by including all the ground truth words of ICDAR 2013 test dataset. On the other hand, the PhotoOCR method uses a huge amount of training data that consists of more than five million word images (which are not available to the public).

6.3 Discussion

In our proposed system, we assume that the cropped words are more or less horizontal. This is true in most of the real world cases. However, it might fail when texts in scenes are severely curved or suffer from severe perspective distortion as illustrated in Fig. 6. It partially explains the lower recognition accuracy on SVT dataset, which consists of a certain amount of severely perspective distorted word images. In addition, the SVT dataset is generated from the Google Street View and so consists of a large amount of difficult images such as shop names, street names, etc. In addition, the proposed technique could fail when the word image has a low resolution or inaccurate text bounding boxes are detected as illustrated in Fig. 6. We will investigate these two issues in our future study.

With a very limited set of training data, our proposed method produces much higher recognition accuracy than the state-of-the-art methods as illustrated in Table 1. The PhotoOCR reports a better word recognition accuracy (90%) on SVT dataset. The better accuracy is largely due to the usage of a huge amount of training data that is not available to the public. As a comparison, our proposed

⁵ ICDAR 2011: <http://www.cvc.uab.es/icdar2011competition/>.

⁶ ICDAR 2013: <http://dag.cvc.uab.es/icdar2013competition/>.

⁷ <http://dag.cvc.uab.es/icdar2013competition>.

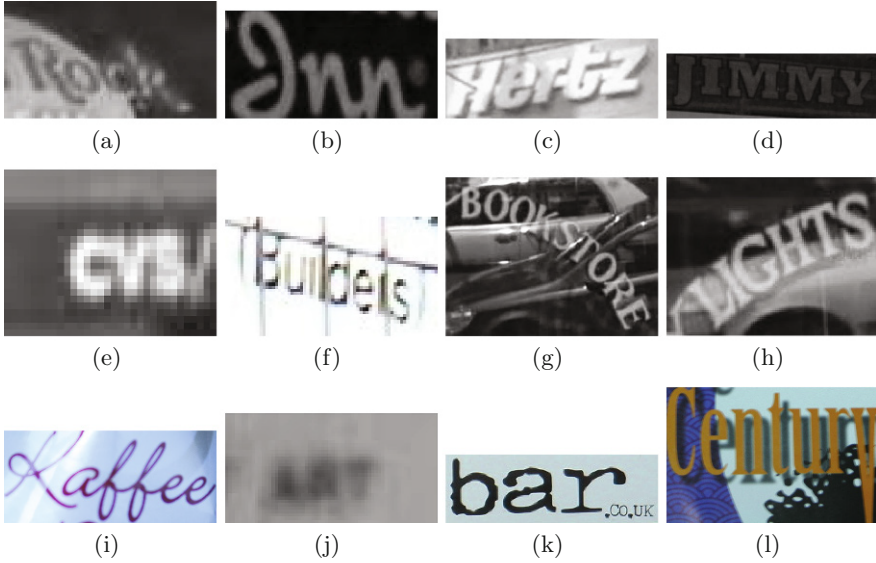


Fig. 6. Examples of falsely recognized word images

method is better in terms of training data size, training time, and computational costs. More importantly, our proposed model is trained by using the publicly available datasets, which provides a benchmarking baseline for the future scene text recognition techniques.

7 Conclusion

Word recognition under unconstrained condition is a difficult task and has attracted increasing research interest in recent years. Many methods have been reported to address this problem. However, there still exists a large gap for computer understanding of texts in natural scene. This paper presents a novel scene text recognition system that makes use of the HOG feature and RNN model.

Compared with state-of-the-art techniques, our proposed method is able to recognize the whole word images without segmentation. It works by integrating two key components. First, it converts a word image into a sequential feature vector and so requires no character-level segmentation and recognition. Second, the RNN is introduced and exploited to classify the sequential column feature vectors into word accurately. Experiments on several public datasets show that the proposed technique obtains superior word recognition accuracy. In addition, the proposed technique is trained and tested over several publicly available datasets which forms a good baseline for future benchmarking of other new scene text recognition techniques.

References

1. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: ICDAR 2003 robust reading competitions. In: 2003 International Conference on Document Analysis and Recognition (ICDAR), pp. 682–687 (2003)
2. Shahab, A., Shafait, F., Dengel, A.: ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In: Proceedings of the 2011 International Conference on Document Analysis and Recognition, ICDAR 2011, pp. 1491–1496 (2011)
3. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Gomez i Bigorda, L., Robles Mestre, S., Mas, J., Fernandez Mota, D., Almazan Almazan, J., de las Heras, L.P.: ICDAR 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1484–1493 (2013)
4. Wang, K., Belongie, S.: Word spotting in the wild. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 591–604. Springer, Heidelberg (2010)
5. Mishra, A., Alahari, K., Jawahar, C.V.: An MRF model for binarization of natural scene text. In: 2011 11th International Conference on Document Analysis and Recognition (ICDAR), pp. 11–16 (2011)
6. Kumar, D., Anil Prasad, M.N., Ramakrishnan, A.G.: Nesp: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images. In: Proceedings of SPIE, vol. 8658 (2013)
7. Zhou, Y., Feild, J., Learned-Miller, E., Wang, R.: Scene text segmentation via inverse rendering. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 457–461 (2013)
8. Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: PhotoOCR: Reading text in uncontrolled conditions. In: 2013 IEEE International Conference on Computer Vision (ICCV) (2013)
9. Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S., Zhang, Z.: Scene text recognition using part-based tree-structured character detection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2961–2968 (2013)
10. Mishra, A., Alahari, K., Jawahar, C.: Top-down and bottom-up cues for scene text recognition. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2687–2694 (2012)
11. Novikova, T., Barinova, O., Kohli, P., Lempitsky, V.: Large-lexicon attribute-consistent text recognition in natural images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 752–765. Springer, Heidelberg (2012)
12. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Netw.* **18**, 602–610 (2005)
13. Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 855–868 (2009)
14. Wang, T., Wu, D., Coates, A., Ng, A.: End-to-end text recognition with convolutional neural networks. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 3304–3308 (2012)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 1, pp. 886–893 (2005)

16. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1457–1464 (2011)
17. Tian, S., Lu, S., Su, B., Tan, C.L.: Scene text recognition using co-occurrence of histogram of oriented gradients. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 912–916 (2013)
18. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2963–2970 (2010)
19. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1083–1090 (2012)
20. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3538–3545 (2012)
21. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: 2013 IEEE International Conference on Computer Vision (ICCV) (2013)
22. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D., Ng, A.: Text detection and character recognition in scene images with unsupervised feature learning. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 440–445 (2011)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
24. Gers, F.A., Schmidhuber, J.A., Cummins, F.A.: Learning to forget: Continual prediction with lstm. *Neural Comput.* **12**, 2451–2471 (2000)
25. Zhang, X., Tan, C.: Segmentation-free keyword spotting for handwritten documents based on heat kernel signature. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 827–831 (2013)
26. Graves, A.: Rnnlib: A recurrent neural network library for sequence learning problems. (<http://sourceforge.net/projects/rnnl/>)
27. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: Proceedings of the International Conference on Computer Vision Theory and Applications (2009)