Reading Scene Text in Deep Convolutional Sequences

Pan He,[*1, 2] **Weilin Huang,**[*1, 2] **Yu Qiao,**[1] **Chen Change Loy,**[2, 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
{pan.he,wl.huang,yu.qiao}@siat.ac.cn, {ccloy,xtang}@ie.cuhk.edu.hk

Abstract

We develop a Deep-Text Recurrent Network (DTRN) that regards scene text reading as a sequence labelling problem. We leverage recent advances of deep convolutional neural networks to generate an ordered high-level sequence from a whole word image, avoiding the difficult character segmentation problem. Then a deep recurrent model, building on long short-term memory (LSTM), is developed to robustly recognize the generated CNN sequences, departing from most existing approaches recognising each character independently. Our model has a number of appealing properties in comparison to existing scene text recognition methods: (i) It can recognise highly ambiguous words by leveraging meaningful context information, allowing it to work reliably without either pre- or post-processing; (ii) the deep CNN feature is robust to various image distortions; (iii) it retains the explicit order information in word image, which is essential to discriminate word strings; (iv) the model does not depend on pre-defined dictionary, and it can process unknown words and arbitrary strings. It achieves impressive results on several benchmarks, advancing the-state-of-the-art substantially.

Text recognition in natural image has received increasing attention in computer vision and machine intelligence, due to its numerous practical applications. This problem includes two sub tasks, namely text detection (Huang, Qiao, and Tang 2014; Yin et al. 2014; He et al. 2015; Zhang et al. 2015; Huang et al. 2013; Neumann and Matas 2013) and text-line/word recognition (Jaderberg, Vedaldi, and Zisserman 2014; Almazán et al. 2014; Jaderberg et al. 2014; Bissacco et al. 2013; Yao et al. 2014). This work focuses on the latter that aims to retrieve a text string from a cropped word image. Though huge efforts have been devoted to this task, reading text in unconstrained environment is still extremely challenging, and remains an open problem, as substantiated in recent literature (Jaderberg et al. 2015b; Almazán et al. 2014). The main difficulty arises from the large diversity of text patterns (e.g. low resolution, low contrast, and blurring), and highly complicated background clutters. Consequently, individual character segmentation or separation is extremely challenging.

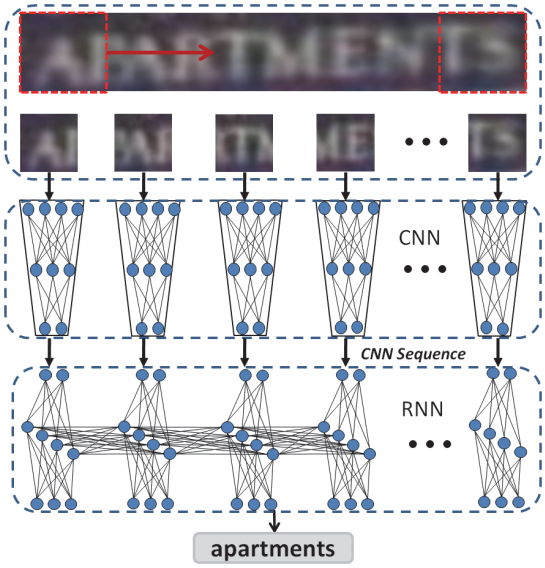


Figure 1: The word image recognition pipeline of the proposed *Deep-Text Recurrent Networks (DTRN)* model.

Most previous studies focus on developing powerful character classifiers, some of which are incorporated with a language model, leading to the state-of-the-art performance (Jaderberg, Vedaldi, and Zisserman 2014; Bissacco et al. 2013; Yao et al. 2014; Lee et al. 2014). These approaches mainly follow the pipeline of conventional OCR techniques by first involving a character-level segmentation, then followed by an isolated character classifier and post-processing for recognition. They also adopt deep neural networks for representation learning, but the recognition is still confined to character-level classification. Thus their performance are severely harmed by the difficulty of character segmentation or separation. Importantly, recognizing each character independently discards meaningful context information of the words, significantly reducing its reliability and robustness.

First, we wish to address the issue of context information learning. The main inspiration for approaching this issue comes from the recent success of recurrent neural networks (RNN) for handwriting recognition (Graves, Liwicki, and Fernandez 2009; Graves and Schmidhuber 2008), speech

[*]Authors contributed equally

recognition (Graves and Jaitly 2014), and language translation (Sutskever, Vinyals, and Le 2014). We found the strong capability of RNN in learning continuous sequential features particularly well-suited for text recognition task to retain the meaningful interdependencies of the continuous text sequence. We note that RNNs have been formulated for recognizing handwritten or documented images (Graves, Liwicki, and Fernandez 2009; Graves and Schmidhuber 2008; Breuel et al. 2013), nevertheless, the background in these tasks is relatively plain, and the raw image feature can be directly input to RNN for recognition, or the text stroke information can be easily extracted or binarized at pixel level, making it possible to manually design a sequential heuristic feature for the input to RNN. In contrast, the scene text image is much more complicated where pixel-level segmentation is extremely difficult, especially for highly ambiguous images (Fig. 1). Thus it is non-trivial to directly apply the sequence labelling models to scene text.

Consequently, the second challenge we need to resolve is the issue of character segmentation. We argue that individual character segmentation is not a 'must' in text recognition. The key is to acquire strong representation from the image, with explicit order information. The strong representation ensures robustness to various distortions and background clutters, whilst the explicit order information is crucial to discriminate a meaningful word. The ordered strong feature sequence computed from the sequential regions of word image allows each frame region to locate the part of a character, which can be stored sequentially by the recurrent model. This makes it possible to recognize the character robustly by using its continuous parts, and thus successfully avoid the character segmentation.

To this end, we develop a deep recurrent model that reads word images in deep convolutional sequences. The new model is referred as Deep-Text Recurrent Network (DTRN), of which the pipeline is shown in Fig. 1. It takes both the advantages of the deep CNN for image representation learning and the RNN model for sequence labelling, with the following appealing properties:

1) Strong and high-level representation without character segmentation – The DTRN generates a convolutional image sequence, which is explicitly ordered by scanning a sliding window through a word image. The CNN sequence captures meaningful high-level representation that is robust to various image distortions. It differs significantly from manually-designed sequential features used by most prior studies based on sequence labelling (Breuel et al. 2013; Graves, Liwicki, and Fernandez 2009; Su and Lu 2014). The sequence is generated without any low-level operation or challenging character segmentation.

2) Exploiting context information In contrast to existing systems (Bissacco et al. 2013; Jaderberg, Vedaldi, and Zisserman 2014; Wang et al. 2012) that read each character independently, we formulate this task as a sequence labelling problem. Specifically, we build our system on the LSTM, so as to capture the interdependencies inherent in the deep sequences. Such consideration allows our system to recognize highly ambiguous words, and work reliably without either pre- or post-processing. In addition, the recurrence allows it to process sequences of various lengths, going beyond traditional neural networks of fixed-length input and output.

3) Process unknown words and arbitrary strings With properly learned deep CNNs and RNNs, our model does not depend on any pre-defined dictionary, unlike exiting studies (Jaderberg et al. 2015b; Jaderberg, Vedaldi, and Zisserman 2014; Wang et al. 2012), and it can process unknown words, and arbitrary strings, including multiple words.

We note that CNN and RNN have been independently exploited in the domain of text recognition. Our main contribution in this study is to develop a unified deep recurrent system that leverages both the advantages of CNN and RNN for the difficult scene text recognition problem, which has been solved based on analyzing character independently. This is the first attempt to show the effectiveness of exploiting convolutional sequence with sequence labeling model for this challenging task. We highlight the considerations required to make this system reliable and discuss the unique advantages offered by it. The proposed DTRN demonstrate promising results on a number of benchmarks, improving recent results of (Jaderberg, Vedaldi, and Zisserman 2014; Almazán et al. 2014) considerably.

Related Work

Previous work mainly focuses on developing a powerful character classifier with manually-designed image features. A HoG feature with random ferns was developed for character classification in (Wang, Babenko, and Belongie 2011). Neumann and Matas proposed new oriented strokes for character detection and classification (Neumann and Matas 2013). Their performance is limited by the low-level features. In (Lee et al. 2014), a mid-level representation of characters was developed by proposing a discriminative feature pooling. Similarly, Yao *et al.* proposed the mid-level Strokelets to describe the parts of characters (Yao et al. 2014).

Recent advances of DNN for image representation encourage the development of more powerful character classifiers, leading to the state-of-the-art performance on this task. The pioneer work was done by LeCun *et al.*, who designed a CNN for isolated handwriting digit recognition (LeCun et al. 1998). A two-layer CNN system was proposed for both character detection and classification in (Wang et al. 2012). PhotoOCR system employs a five-layer DNN for character recognition (Bissacco et al. 2013). Similarly, Jaderberg *et al.* (Jaderberg, Vedaldi, and Zisserman 2014) proposed novel deep features by employing a Maxout CNN model for learning common features, which were subsequently used for a number of different tasks, such as character classification, location optimization and language model learning.

These approaches treat isolated character classification and subsequent word recognition separately. They do not unleash the full potential of word context information in the recognition. They often design complicated optimization algorithm to infer word string by incorporating multiple additional visual cues, or require a number of post-processing steps to refine the results (Jaderberg, Vedaldi, and Zisserman 2014; Bissacco et al. 2013). Our model differs significantly

from them by exploring the recurrence of deep features, allowing it to leverage the underlying context information to directly recognise the whole word image in a deep sequence, without a language model and any kind of post-processing.

There is another group of studies that recognise text strings from the whole word images. Almazan *et al.* (Almazán et al. 2014) proposed a subspace regression method to jointly embed both word image and its string into a common subspace. A powerful CNN model was developed to compute a deep feature from a whole word image in (Jaderberg et al. 2015b). Again, our model differs from these studies in the deep recurrent nature. Our sequential feature includes explicit spatial order information, which is crucial to discriminate the order-sensitive word string. While the other global representation would lost such strict order, leading to poorer discrimination power. Furthermore, the model of (Jaderberg et al. 2015b) is strictly constrained by the predefined dictionary, making it unable to recognise a novel word. By contrast, our model can process an unknown word.

For unconstrained recognition, Jaderberg *et al.* proposed another CNN model, which incorporates a Conditional Random Field (Jaderberg et al. 2015a). This model recognizes word strings in character sequences, allowing it for processing a single unknown word. But the model is highly sensitive to the non-character space, making it difficult to recognize multiple words. Our recurrent model can process arbitrary strings, including multiple words, and thus generalizes better. Our method also relates to (Su and Lu 2014), where a RNN is built upon HOG features. However, its performance is significantly limited by the HOG. While the strong deep CNN feature is crucial to the success of our model.

Our approach is partially motivated by the recent success of deep models for image captioning, where the combination of the CNN and RNN has been applied (Andrej and Li 2015; Donahue et al. 2015; Alsharif and Pineau 2013). They explored the CNN for computing a deep feature from a whole image, followed by a RNN to decode it into a sequence of words. ReNet (Visin et al. 2015) was proposed to directly compute the deep image feature by using four RNN to sweep across the image. Generally, these models do not explicitly store the strict spatial information by using the global image representation. By contrast, our word images include explicit order information of its string, which is a crucial cue to discriminate a word. Our goal here is to derive a set of robust sequential features from the word image, and design an new model that bridges the image representation learning and sequence labelling task.

Deep-Text Recurrent Networks

The pipeline of Deep-Text Recurrent Network (DTRN) is shown in Fig. 1. It starts by encoding a given word image into an ordered sequence with a specially designed CNN. Then a RNN is employed to decode (recognise) the CNN sequence into a word string. The system is end-to-end, i.e. it takes a word image as input and directly outputs the corresponding word string, without any pre- and post-processing steps. Both the input word image and output string can be of varying lengths. This section revisits some important details of CNN and RNN and highlight the considerations that make their combination reliable for scene text recognition.

Formally, we formulate the word image recognition as a sequence labeling problem. We maximize the probability of the correct word strings (S_w), given an input image (I),

$$\hat{\theta} = \arg\max_{\theta} \sum_{(I,S_w)} \log P(S_w|I;\theta), \tag{1}$$

where θ are the parameters of the recurrent system. $(I, S_w) \in \Omega$ is a sample pair from a training set, Ω, where $S_w = \{S_w^1, S_w^2, ..., S_w^K\}$ is the ground truth word string (containing K characters) of the image I. Commonly, the chain rule is applied to model the joint probability over S_w,

$$\log P(S_w|I;\theta) = \sum_{i=1}^{K} \log P(S_w^i|I, S_w^0, ..., S_w^{i-1};\theta) \tag{2}$$

Thus we optimize the sum of the log probabilities over all sample pairs in the training set (Ω) to learn the model parameters. We develop a RNN to model the sequential probabilities $P(S_w^i|I, S_w^0, ..., S_w^{i-1})$, where the variable number of the sequentially conditioned characters can be expressed by an internal state of the RNN in hidden layer, h_t. This internal state is updated when the next sequential input x_t is presented by computing a non-linear function $\mathcal{H}$,

$$h_{t+1} = \mathcal{H}(h_t, x_t) \tag{3}$$

where the non-linear function $\mathcal{H}$ defines exact form of the proposed recurrent system. $X = \{x_1, x_2, x_3, ..., x_T\}$ is the sequential CNN features computed from the word image,

$$\{x_1, x_2, x_3, ..., x_T\} = \varphi(I) \tag{4}$$

Designs of the φ and $\mathcal{H}$ play crucial roles in the proposed system. We develop a CNN model to generate the sequential x_t, and define $\mathcal{H}$ with a long short-term memory (LSTM) architecture (Hochreiter and Schmidhuber 1997).

Sequence Generation with Maxout CNN

The main challenge of obtaining low-level sequential representation from the word images arises from the difficulties of correct segmentation at either pixel or character level. We argue that it is not necessary to perform such low-level feature extraction. On the contrary, it is more natural to describe word strings in sequences where their explicit order information is retained. This information is extremely important to discriminate a word string. Furthermore, the variations between continuous examples in a sequence should encode additional information, which could be useful in making more reliable prediction. By considering these factors, we propose to generate an explicitly ordered deep sequence with a CNN model, by sliding a sub window through the word image.

To this end, we develop a Maxout network (Goodfellow et al. 2013) for computing the deep feature. It has been shown that the Maxout CNN is powerful for character classification (Jaderberg, Vedaldi, and Zisserman 2014; Alsharif and Pineau 2013). The basic pipeline is to compute point-wise maximum through a number of grouped feature maps or channels. Our networks is shown in Fig 2 (a), the

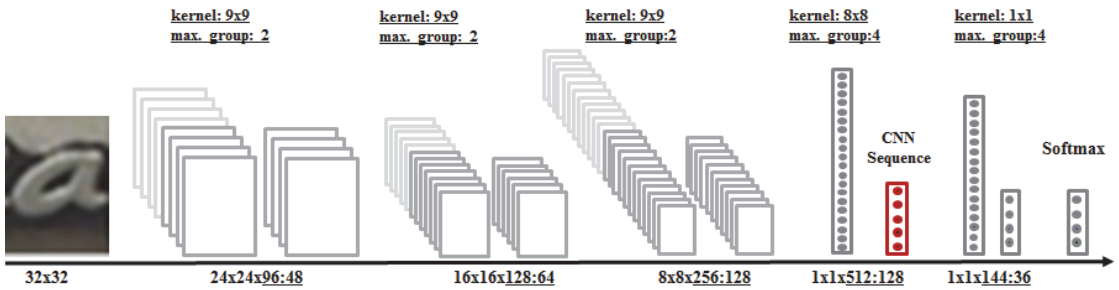


Figure 2: The structures of our maxout CNN model.

input image is of size 32×32, corresponding to the size of sliding-window. It has five convolutional layers, each of which is followed by a two- or four-group Maxout operation, with various numbers of feature maps, i.e. 48, 64, 128,128 and 36, respectively. Similar to the CNN used in (Jaderberg, Vedaldi, and Zisserman 2014), our networks does not involve any pooling operation, and the output of last two convolutional layers are just one pixel. This allows our CNN to convolute the whole word images at once, leading to a significant computational efficiency. For each word image, we resize it into the height of 32, and keep its original aspect ratio unchanged. We apply the learned filters to the resized image, and get a 128D CNN sequence directly from the output of last second convolutional layer. This operation is similar to computing deep feature independently from the sliding-window by moving it densely through the image, but with much computational efficiency. Our Maxout CNN is trained on 36-class case insensitive character images.

Sequence Labeling with RNN

We believe that the interdependencies between the convolutional sequence include meaningful context information which would be greatly helpful to identify an ambitious character. RNN has shown strong capability for learning meaningful structure from an ordered sequence. Another important property of the RNN is that the rate of changes of the internal state can be finely modulated by the recurrent weights, which contributes to its robustness against localised distortions of the input data (Graves, Liwicki, and Fernandez 2009). Thus we propose the use of RNN in our framework to model the generated CNN sequence $\{x_1, x_2, x_3, ..., x_T\}$. The structure of our RNN model is shown in Fig. 3.

The main shortcoming of the standard RNN is the vanishing gradient problem, making it hard to transmit the gradient information consistently over long time (Hochreiter and Schmidhuber 1997). This is a crucial issue in designing a RNN model, and the long short-term memory (LSTM) was proposed specially to address this problem (Hochreiter and Schmidhuber 1997). The LSTM defines a new neuron or cell structure in the hidden layer with three additional multiplicative gates: the *input gate*, *forget gate* and *output gate*. These new cells are referred as memory cells, which allow the LSTM to learn meaningful long-range interdependencies. We skip standard descriptions of the LSTM memory cells and its formulation, by leaving them in the supplementary material.

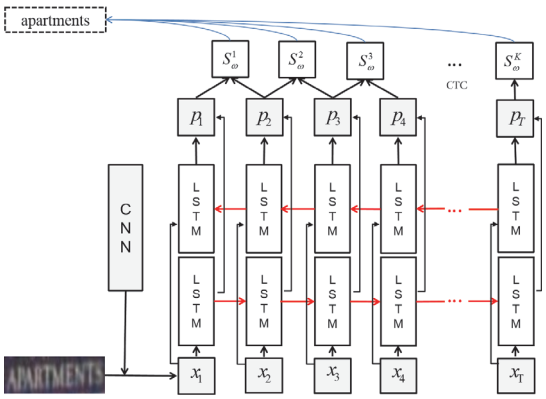


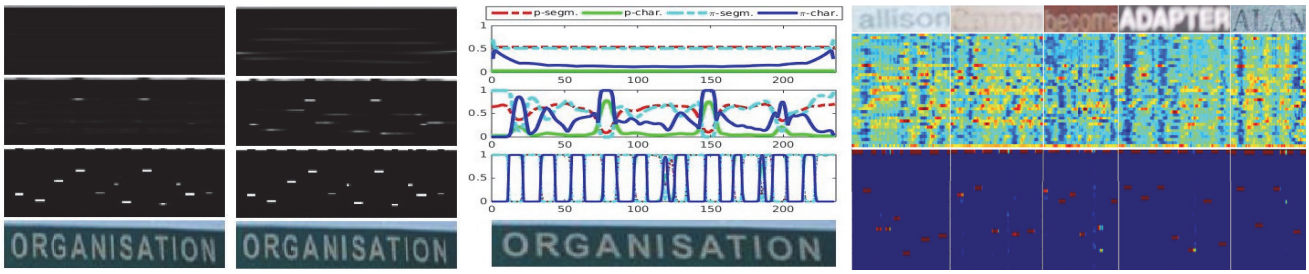
Figure 3: The structure of our recurrent neural networks.

The sequence labelling of varying lengths is processed by recurrently implementing the LSTM memory for each sequential input x_t, such that all LSTMs share the same parameters. The output of the LSTM h_t is fed to the LSTM at next input x_{t+1}. It is also used to compute the current output, which is transformed to the estimated probabilities over all possible characters. It finally generates a sequence of the estimations with the same length of input sequence, $\mathbf{p} = \{p_1, p_2, p_3, ..., p_T\}$.

Due to the unsegmented nature of the word image at the character level, the length of the LSTM outputs (T) is not consistent with the length of a target word string, $|S_w| = K$. This makes it difficult to train our recurrent system directly with the target strings. To this end, we follow the recurrent system developed for the handwriting recognition (Graves, Liwicki, and Fernandez 2009) by applying a connectionist temporal classification (CTC) (Graves and Schmidhuber 2005) to approximately map the LSTM sequential output ($\mathbf{p}$) into its target string as follow,

$$S_w^\star \approx \mathcal{B}(\arg\max_{\pi} P(\pi|\mathbf{p})) \tag{5}$$

where the projection $\mathcal{B}$ removes the repeated labels and the non-character labels (Graves and Schmidhuber 2005). For example, $\mathcal{B}(-gg-o-oo-dd-) = good$. The CTC looks for an approximately optimized path (π) with maximum probability through the LSTMs output sequence, which aligns the different lengths of LSTM sequence and the word string.



(a) the LSTM output (b) the CTC path (c) $\mathbf{p}$ and π (d) DTRN vs DeepFeatures

Figure 4: (a-c)RNNs training process recorded at epoch 0 (row 1), 5 (row 2) and 50 (row 3) with a same word image (row 4). (a) the LSTM output ($\mathbf{p}$); (b) the CTC path (π) mapped from ground truth word string ($\mathcal{B}^{-1}(S_w)$); (c) maximum probabilities of the character and segmentation line with $\mathbf{p}$ and π; (d) output confident maps of the DeepFeatures (middle) and the LSTM layer of the DTRN (bottom).

The CTC is specifically designed for the sequence labelling tasks where it is hard to pre-segment the input sequence to the segments that exactly match a target sequence. In our RNN model, the CTC layer is directly connected to the outputs of LSTMs, and works as the output layer of the whole RNN. It not only allows our model to avoid a number of complicated post-processing (e.g. transforming the LSTM output sequence into a word string), but also makes it possible to be trained in an end-to-end fashion by minimizing an overall loss function over $(X, S_w) \in \Omega$. The loss for each sample pair is computed as sum of the negative log likelihood of the true word string,

$$L(X, S_w) = -\sum_{i=1}^{K} \log P(S_w^i|X) \tag{6}$$

Finally, our RNNs model follows a bidirectional LSTM architecture, as shown in Fig. 3 (b). It has two separate LSTM hidden layers that process the input sequence forward and backward, respectively. Both hidden layers are connected to the same output layers, allowing it to access both past and future information. In several sequence labelling tasks, such as handwriting recognition (Graves, Liwicki, and Fernandez 2009) and phoneme recognition (Graves and Schmidhuber 2005), the bidirectional RNNs have shown stronger capability than the standard RNNs. Our RNNs model is trained with the Forward-Backward Algorithm that jointly optimizes the bidirectional LSTM and CTC. Details are presented in the supplementary material.

Implementation Details

Our CNN model is trained on about 1.8×10^5 character images cropped from the training sets of a number of benchmarks by (Jaderberg, Vedaldi, and Zisserman 2014). We generate the CNN sequence by applying the trained CNN with a sliding-window, followed by a column-wise normalization. Our recurrent model contains a bidirectional LSTM. Each LSTM layer has 128 cell memory blocks. The input layer has 128 neurons (corresponding to 128D CNN sequence), which are fully connected to both hidden layers. The outputs of two hidden layers are concatenated, and then fully connected to the output layer of LSTM with 37 output classes (including the non-character), by using a softmax function. Our RNN model has 273K parameters in total. In our experiments, we found that adding more layers LSTM does not lead to better results in our task. We conjecture that LSTM needs not be deep, given the deep CNN which has provided strong representations.

The recurrent model is trained with steepest descent. The parameters are updated per training sequence by using a learning rate of 10^{-4} and a momentum of 0.9. We perform forward-backward algorithm (Graves et al. 2006) to jointly optimize the LSTM and CTC parameters, where a forward propagation is implemented through whole network, followed by a forward-backward algorithm that aligns the ground truth word strings to the LSTM outputs, $\pi \in \mathcal{B}^{-1}(S_w)$, $\pi, \mathbf{p} \in \mathbb{R}^{37 \times T}$. The loss of E.q.(6) is computed approximately as:

$$L(X, S_w) \approx -\sum_{t=1}^{T} \log P(\pi_t|X) \tag{7}$$

Finally, the approximated error is propagated backward to update the parameters. The RNN is trained on about 3000 word images (all characters of them are included in previously-used 1.8×10^5 character images), taken from the training sets of three benchmarks used bellow. The training process is shown in Fig. 4.

Experiments and Results

The experiments were conducted on three standard benchmarks for cropped word image recognition: the Street View Text (SVT) (Wang, Babenko, and Belongie 2011), ICDAR 2003 ($IC03$) (Lucas et al. 2003) and IIIT 5K-word ($IIIT5K$) (Mishra., Alahari, and Jawahar 2012). The SVT has 647 word images collected from Google Street View of road-side scenes. It provides a lexicon of 50 words per image for recognition (SVT-50). The $IC03$ contains 860 word images cropped from 251 natural images. Lexicons with 50 words per image (IC03-50) and all words of the test set (IC03-FULL) are provided. The $IIIT5K$ is comprised of 5000 cropped word images from both scene and born-digital

images. The dataset is split into subsets of 2000 and 3000 images for training and test. Each image is associated with lexicons of 50 (IIIT5k-50) and 1k words (IIIT5k-1k) for test.

DTRN vs DeepFeatures

The recurrence property of the DTRN makes it distinct against the current deep CNN models, such as DeepFeatures (Jaderberg, Vedaldi, and Zisserman 2014)) and the system of (Wang et al. 2012). The advantage is shown clearly in Fig. 4 (d), where the output maps of the LSTM layer and the Maxout CNN of DeepFeatures are compared. As can be observed, our maps are much clearer than those of the DeepFeatures in a number of highly ambiguous word images. The character probability distribution and segmentation are shown accurately on our maps, indicating the excellent capability of our model for correctly identifying word texts from challenging images. The final word recognition is straightforward by simply applying the $\mathcal{B}$ projection (E.q. 5) on these maps. However, the maps of DeepFeatures are highly confused, making it extremely difficult to infer the correct word strings from their maps. Essentially, the recurrent property of DTRN allows it to identify a character robustly from a number of continuous regions or sided windows, while the DeepFeatures classifies each isolated region independently so that it is confused when a located region just includes a part of the character or multiple characters.

Comparisons with State-of-the-Art

The evaluation is conducted by following the standard protocol, where each word image is associated with a lexicon, and edit distance is computed to find the optimized word. The recognition results by the DTRN are presented in Fig. 5, including both the correct and incorrect recognitions. As can been seen, the DTRN demonstrates excellent capability on recognising extremely ambiguous word images, some of which are even hard to human. This is mainly beneficial from its strong ability to leverage explicit order and meaningful word context information. The results on three benchmarks are compared with the state-of-the-art in Table 1.

Mid-level representation: Strokelet (Yao et al. 2014) and Lee *et al.*'s method (Lee et al. 2014) achieved leading performance based on the mid-level features. Though they show large improvements over conventional low-level features, their performance are not comparable to ours, with significant reductions in accuracies in all the three datasets.

Deep neural networks: As shown in Table 1, the DNN methods largely outperform the mid-level approaches, with close to 10% of improvement in all cases. The considerable performance gains mainly come from its ability to learn a deep high-level feature from the word image. Su and Lu's method obtained accuracy of 83% on SVT by building a RNN model upon the HOG features. DeepFeatures achieved leading results on both the SVT and IC03 datasets. However, the DeepFeatures are still built on isolate character classifier. By training a similar CNN model with the same training data, the DTRN achieved significant improvements over the DeepFeatures in all datasets. The results agree with our analysis conducted above. On the widely-used SVT, our model outperforms the DeepFeatures considerably from 86.1% to 93.5%, indicating the superiority of our recurrent model in connecting the isolated deep features sequentially for recognition. Furthermore, our system does not need to learn the additional language model and character location information, all of which are optimized jointly and automatically by our RNN in an end-to-end fashion.

Whole image representation: Almazan *et al.*'s approach, based on the whole word image representation, achieved 87.0% accuracy on the SVT (Almazán et al. 2014), slightly over that of DeepFeatures. In the IIIT5k, it yielded 88.6% and 75.6% on small and large lexicons, surpassing previous results with a large margin. Our DTRN strives for a further step by reaching the accuracies of 94% and 91.5% on the IIIT5k. The large improvements may benefit from the explicit order information included in our CNN sequence. It is the key to increase discriminative power of our model for word representation, which is highly sensitive to the order of characters. The strong discriminative power can be further verified by the consistent high-performance of our system along with the increase of lexicon sizes, where the accuracy of Almazan *et al.*'s approach drops significantly.

Training on additional large datasets: The PhotoOCR (Bissacco et al. 2013) sets a strong baseline on the SVT (90.4%) by using large additional training data. It employed about 10^7 character examples to learn a powerful DNN classifier, and also trained a strong language model with a corpus of more than a trillion tokens. However, it involves a number of low-level techniques to over-segment characters, and jointly optimizes the segmentation, character classification and language model with beam search. Furthermore, it also includes a number of post-processing steps to further improve the performance, making the system highly complicated. The DTRN achieved 3.1% improvement over the PhotoOCR, which is also significant by considering only a fraction of the training data (two orders of magnitude less data) we used. While our model works without a language model, and does not need any post-processing step.

Jaderberg *et al.* proposed several powerful deep CNN models by computing a deep feature from the whole word image (Jaderberg et al. 2014; 2015a; 2015b). However, directly comparing our DTRN to these models may be difficult. First, these models was trained on 7.2×10^6 word images, comparing to ours 3×10^3 word images (with 1.8×10^5 characters). Nevertheless, our model achieves comparable results against $Jaderberg2015a$ with higher accuracies on the SVT and IIIT5k-1K. Importantly, the DTRN also provides unique capability for unconstrained recognition of any number of characters and/or word strings in a text-line. Several examples are presented in the figure of Table 1. $Jaderberg2015b$ model achieves the best results in all databases. It casts the word recognition problem as a large-scale classification task by considering the images of a same word as a class. Thus the output layer should include a large number of classes, e.g. 90,000, imposing a huge number of model parameters which are difficult to be trained. Furthermore, it is not flexible to recognize a new word not trained. While the scene texts often include many irregular word strings (the number could be unlimited) which are impossible to be known in advanced, such as "AB00d". Thus

Figure 5: (Left) Correct recognitions; (Right) Incorrect samples.

Method	Cropped Word Recognition Accuracy(%)				
	IC03-50	IC03-FULL	SVT-50	IIIT5k-50	IIIT5k-1K
Wang et al. 2011	76.0	62.0	57.0	64.1	57.5
Mishra et al. 2012	81.8	67.8	73.2	-	-
Novikova et al. 2012	82.8	-	72.9	-	-
TSM+CRF(Shi et al. 2013)	87.4	79.3	73.5	-	-
Lee et al. 2014	88.0	76.0	80.0	-	-
Strokelets(Yao et al. 2014)	88.5	80.3	75.9	80.2	69.3
Wang et al. 2012	90.0	84.0	70.0	-	-
Alsharif and Pineau 2013	93.1	88.6	74.3	-	-
Su and Lu 2014	92.0	82.0	83.0	-	-
DeepFeatures	96.2	91.5	86.1	-	-
Goel et al. 2013	89.7	-	77.3	-	-
Almazán et al. 2014	-	-	87.0	88.6	75.6
DTRN	**97.0**	**93.8**	**93.5**	**94.0**	**91.5**
PhotoOCR	-	-	90.4	-	-
Jaderberg2015a	97.8	97.0	93.2	95.5	89.6
Jaderberg2015b	98.7	98.6	95.4	97.1	92.7

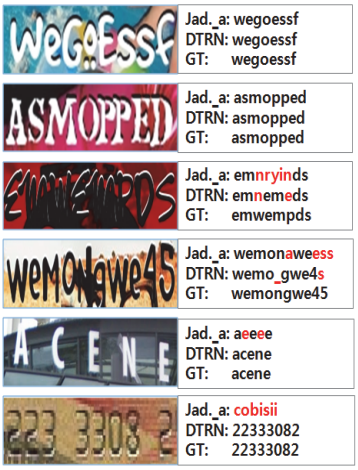


Table 1: Cropped word recognition results on the SVT, ICDAR 2003, and IIIT 5K-word. The bottom figure shows unconstrained recognitions of the DTRN and the publicly available model (Jaderberg et al. 2014), which is similar to $Jaderberg2015a$. Obviously, it seems to be sensitive to non-character spaces.

our DTRN can process unknown words and arbitrary strings, providing a more flexible approach for this task.

Conclusion

We have presented a Deep-Text Recurrent Network (DTRN) for scene text recognition. It models the task as a deep sequence labelling problem that overcomes a number of main limitations. It computes a set of explicitly-ordered deep features from the word image, which is not only robust to low-level image distortions, but also highly discriminative to word strings. The recurrence property makes it capable of recognising highly ambiguous images by leveraging meaningful word context information, and also allows it to process unknown words and arbitrary strings, providing a more principled approach for this task. Experimental results show that our model has achieved the state-of-the-art performance.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China (61503367, 91320101, 61472410), Guangdong Natural Science Foundation (2015A030310289), Guangdong Innovative Research Program (201001D0104648280, 2014B050505017) and Shenzhen Basic Research Program (KQCX2015033117354153). Yu Qiao is the corresponding author.

References

Almazán, J.; Gordo, A.; Fornés, A.; and Valveny, E. 2014. Word spotting and recognition with embedded attributes. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)* 36:2552–2566.

Alsharif, O., and Pineau, J. 2013. End-to-end text recognition with hybrid HMM maxout models. *arXiv:1310.1811v1*.

Andrej, K., and Li, F. 2015. Deep visual-semantic alignments for generating image descriptions. IEEE Computer Vision and Pattern Recognition (CVPR).

Bissacco, A.; Cummins, M.; Netzer, Y.; and Neven, H. 2013. Photoocr: Reading text in uncontrolled conditions. IEEE International Conference on Computer Vision (ICCV).

Breuel, T.; UI-Hasan, A.; Azawi, M.; and Shafait, F. 2013. High-performance ocr for printed english and fraktur using lstm networks. International Conference on Document Analysis and Recognition (ICDAR).

Donahue, J.; Hendricks, L.; Guadarrama, S.; and Rohrbach, M. 2015. Long-term recurrent covolutional networks for visual recognition and description. IEEE Computer Vision and Pattern Recognition (CVPR).

Goodfellow, I.; Warde-Farley, D.; Mirza, M.; Courville, A.; and Bengio, Y. 2013. Maxout networks. *arXiv:1302.4389v4*.

Graves, A., and Jaitly, N. 2014. Towards end-to-end speech recognition with recurrent neural networks. IEEE International Conference on Machine Learning (ICML).

Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18:602–610.

Graves, A., and Schmidhuber, J. 2008. Offline handwriting recognition with multidimensional recurrent neural networks. Neural Information Processing Systems (NIPS).

Graves, A.; Fernandez, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. IEEE International Conference on Machine Learning (ICML).

Graves, A.; Liwicki, M.; and Fernandez, S. 2009. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)* 31:855–868.

He, T.; Huang, W.; Qiao, Y.; and Yao, J. 2015. Text-attentional convolutional neural networks for scene text detection. *arXiv:1510.03283*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9:1735–1780.

Huang, W.; Lin, Z.; Yang, J.; and Wang, J. 2013. Text localization in natural images using stroke feature transform and text covariance descriptors. IEEE International Conference on Computer Vision (ICCV).

Huang, W.; Qiao, Y.; and Tang, X. 2014. Robust scene text detection with convolution neural network induced mser trees. European Conference on Computer Vision (ECCV).

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. Workshop in Neural Information Processing Systems (NIPS).

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2015a. Deep structured output learning for unconstrained text recognition. International Conference on Learning Representation (ICLR).

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2015b. Reading text in the wild with convolutional neural networks. *International Jounal of Computver Vision (IJCV)*.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Deep features for text spotting. European Conference on Computer Vision (ECCV).

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Lee, C.; Bhardwaj, A.; Di, W.; and andR. Piramuthu, V. J. 2014. Region-based discriminative feature pooling for scene text recognition. IEEE Computer Vision and Pattern Recognition (CVPR).

Lucas, S. M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; and Young, R. 2003. Icdar 2003 robust reading competitions. International Conference on Document Analysis and Recognition (ICDAR).

Mishra., A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. British Machine Vision Conference (BMVC).

Neumann, L., and Matas, J. 2013. Scene text localization and recognition with oriented stroke detection. IEEE International Conference on Computer Vision (ICCV).

Shi, C.; Wang, C.; Xiao, B.; Zhang, Y.; Gao, S.; and Zhang, Z. 2013. Scene text recognition using part-based tree-structured character detection. IEEE Computer Vision and Pattern Recognition (CVPR).

Su, B., and Lu, S. 2014. Accurate scene text recognition based on recurrent neural network. Asian Conference on Computer Vision (ICCV).

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. Neural Information Processing Systems (NIPS).

Visin, F.; Kastner, K.; Cho, K.; Matteucci, M.; Courville, A.; and Bengio, Y. 2015. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv:1505.00393*.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. IEEE International Conference on Computer Vision (ICCV).

Wang, T.; Wu, D.; Coates, A.; and Ng, A. Y. 2012. End-to-end text recognition with convolutional neural networks. IEEE International Conference on Pattern Recognition (ICPR).

Yao, C.; Bai, X.; Shi, B.; and Liu, W. 2014. Strokelets: A learned multi-scale representation for scene text recognition. IEEE Computer Vision and Pattern Recognition (CVPR).

Yin, X. C.; Yin, X.; Huang, K.; and Hao, H. W. 2014. Robust text detection in natural scene images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 36:970–983.

Zhang, Z.; Shen, W.; Yao, C.; and Bai, X. 2015. Symmetry-based text line detection in natural scenes. IEEE Computer Vision and Pattern Recognition (CVPR).