A Novel Integrated Framework for Learning both Text Detection and Recognition

Wanchen Sui	Qing Zhang	Jun Yang	Wei Chu
Alibaba	Alibaba	Alibaba	Ant Financial, Alibaba
wanchen.swc@alibaba-inc.com	sensi.zq@alibaba-inc.com	muzhuo.yj@alibaba-inc.com	weichu.cw@alibaba-inc.com

Abstract—In this paper, we propose a novel integrated framework for learning both text detection and recognition. For most of the existing methods, detection and recognition are treated as two isolated tasks and trained separately, since parameters of detection and recognition models are different and two models target to optimize their own loss functions during individual training processes. In contrast to those methods, by sharing model parameters, we merge the detection model and recognition model into a single end-to-end trainable model and train the joint model for two tasks simultaneously. The shared parameters not only help effectively reduce the computational load in inference process, but also improve the end-to-end text detection-recognition accuracy. In addition, we design a simpler and faster sequence learning method for the recognition network based on a succession of stacked convolutional layers without any recurrent structure, this is proved feasible and dramatically improves inference speed. Extensive experiments on different datasets demonstrate that the proposed method achieves very promising results.

I. INTRODUCTION

Text in images provides rich and precise high-level semantic information, which is important for numerous potential applications such as scene understanding, image and video retrieval, and content-based recommendation systems. Therefore, considerable efforts have been made towards the automatic detection and recognition of text in images, which is still an important challenge for visual understanding [1].

As stated in [2], detection and recognition components in conventional scene text recognition systems are sequential and isolated in the whole system pipeline, this makes it difficult to exploit interactive information between the two components. Several works proposed uniformed frameworks for text detection and recognition. Li *et al.* [3] presented a unified network that simultaneously localizes and recognizes text with a single forward pass, based on convolutional recurrent neural networks. Bartz *et al.* [4] designed a single deep neural network to learn to detect and recognize text from natural images in a semi-supervised way. Bušta *et al.* [5] proposed an end-to-end trainable scene text localization and recognition framework called Deep TextSpotter. Different from these methods, our system can perform text detection and recognition via a single framework based entirely on Convolution Neural Network (CNN), and is proved to perform well in cursive English handwriting recognition and Chinese text recognition.

In this work, we design an end-to-end trainable neural network to address this problem, condensing the detection network and recognition network into a single one, which can be trained in an end-to-end manner. By sharing the convolutional layers, we can compute the shared feature maps from entire input image only once, as a result, the text recognition system can be effectively accelerated. In order to make a further speedup for inference, we apply a succession of convolutional layers instead of recurrent layers for sequence learning task. Since our proposed recognition network is based entirely on convolutional neural networks, computations over all elements can be fully parallelized for inference optimization. Meanwhile, it is proved to perform on par with recurrent sequence learning method with respect to recognition accuracy.

In our work, we choose two scenarios to verify our proposed methods for end-to-end text recognition task. We first make a large collection of business cards and get manual labelling to develop a business card recognition system. In addition, we conduct extensive experiments on end-to-end handwritten document recognition using public IAM dataset [6] as benchmark.

In summary, the contributions of the paper are as follows:

(1) A novel integrated framework is proposed for text detection and recognition, which can accomplish these two tasks in an end-to-end trainable neural network.

(2) Given the detection and recognition networks, sharing their common convolutional features can get speedup in recognition computation.

(3) The shared convolutional feature maps, which exploit interaction between two different tasks, can be effective in text detection and recognition and improve performance of end-to-end model.

(4) We propose a simpler and faster recognition network based on a succession of convolutional layers without any recurrent layer which is often used for sequence learning. In this way, by combining with Region-based Fully Convolutional Networks (R-FCN) [7], which is a region-based object detection method, our system can perform text detection and recognition via a fully convolutional network.

II. THE PROPOSED NETWORK ARCHITECTURE

In this section, we present our main ideas and details of the proposed algorithm. Specifically, we give an overview of the proposed framework in Section II-A, explain the text
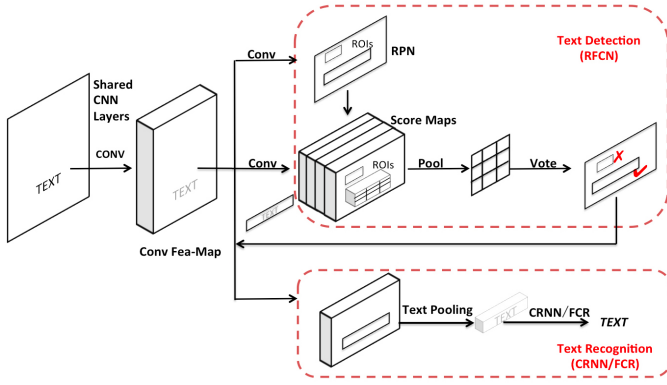
Fig. 1. Schematic overview of the proposed framework

detection network and text recognition network in Section II-B and Section II-D, illustrate the proposed text pooling layer in Section II-C, and describe the loss function and implementation details in Section II-E.

A. Overview

A schematic overview of the proposed framework is illustrated in Figure 1. Firstly, the whole image is processed with several convolutional layers and max pooling layers to generate convolutional feature maps, which are shared by text detector and recognizer. The network uses the very deep VGG-16 network [8] as its backbone (conv1_1 through conv4_3), while other networks such as Googlenet [9], Resnet [10] and so on are also applicable.

Then we perform text detection using a CNN based detection method. In this paper, we take R-FCN [7] as an example to describe our approach, which is readily applicable to other CNN based detection method, such as Faster Region-based CNN (R-CNN) [11], You Only Look Once (YOLO) [12], Single Shot MultiBox Detector (SSD) [13]. The detector returns the regions containing individual lines of text. In contrast to most existing algorithms that directly perform text recognition on each cropped text line of the input image, we design a text line pooling layer similar to Region of Interest (RoI) pooling layer [14], to extract feature maps of a fixed height from the aforementioned shared convolutional feature maps for each detected text line. Then, these small text-line feature maps are sent to a text recognition network.

For the text recognizer, we propose two different schemes to perform text recognition based on feature maps of each extracted text-line. One is a sequence recognition method Convolutional Recurrent Neural Network (CRNN) [15], a combination of Deep Convolutional Neural Networks (DCNN), Recurrent Neural Networks (RNN) layer and Connectionist Temporal Classification (CTC). In the other scheme, we design a simpler and faster recognition network, applying several successive convolutional layers instead of the recurrent layers in CRNN. We denote the newly designed network as Fully Convolutional Recognizer (FCR).

To summarize, our network takes an entire image as input, extracts convolutional features from input only once, makes

feature sharing for both detection and recognition, and finally outputs the detected text regions and the corresponding text contents.

B. Text Detection Network

In this paper, we deploy a region proposal based object detection method named R-FCN to perform text detection. R-FCN adopts a popular two-stage object detection strategy: region proposal and proposal classification. Correspondingly, in this detection network, the aforementioned shared convolutional feature maps finally branch into two sibling sub-networks: the R-FCN network uses a Region Proposal Network (RPN) for extracting candidate regions, and ends with a position-sensitive RoI pooling layer for generating scores and refining coordinates for each proposal.

R-FCN is designed to preform accurate and efficient object detections. Nevertheless, text regions are usually of their specific shapes which are different from that of general objects. In our framework, we modify the specific settings in the R-FCN architecture to cover a wide range of text lines: the scales of anchors are set to 4, 8 and 16, and the aspect ratios are changed to 1:2, 1:5 and 1:10.

C. Text Pooling Layer

To handle the detected text regions, we append another sub-network over the convolutional feature map outputs of the last shared convolutional layer. At the bottom of the sub-network, a text pooling layer is designed to extract a fixed-height feature maps for individual text region.

Specially, it uses max pooling to reshape the shared feature maps inside a valid text region to fixed height H, keeping the original height/width ratios. Here, H is the layer hyper-parameter, which is constant for any particular regions. In this work, the detected text regions are horizontal rectangular bounding boxes. The text pooling layer divides an $h \times w$ text region into $H \times \frac{w \times H}{h}$ sub-regions, the size of each grid is approximate $\frac{h}{H} \times \frac{h}{H}$, and the values in each sub-region are downsampled to the corresponding output grid cell by max-pooling. Each feature map channel processes pooling independently, which is the same as the standard max pooling.

In virtue of the text pooling layer, the network is able to jointly optimize the detector and recognizer with an end-to-end training strategy. Additionally, the end-to-end system computes the shared feature maps from the entire image only once, this can effectively reduce computational load during inference process.

D. Text Recognition Network

As mentioned above, the text regions are converted into fixed-height feature maps via a text pooling layer, and finally sent to a sequence recognizer to predict the corresponding text contents. As for the recognizer, we adopt two different schemes to perform text recognition based on feature maps of each extracted text-line.

The first one is a sequence recognition method named CRNN [15], which integrates feature extraction, sequence

modeling and transcription into an end-to-end trainable network. In our framework, we take advantage of the shared convolutional layers to process feature extraction. Thus, these small text-line feature maps can be directly fed into sequence modeling and transcription layers. To be specific, we deploy a Bidirectional Recurrent Neural Network (BiLSTM) to predict a label distribution for each frame, and then translate the per-frame predictions into the final text sequence.

Recently, Gehring *et al.* [16] proposed a fully convolutional model for sequence to sequence learning, which outperforms recurrent models on very large benchmark datasets at an order of magnitude faster speed. Inspired by this work, we design a simpler and faster recognition network named FCR, using a succession of convolutional layers in place of the BiLSTM layers in CRNN. Through combining R-FCN and FCR, our system can perform text detection and recognition via a single network that entirely consists of convolutional layers.

E. Loss Function

In this work, we employ multi-task learning to jointly optimize model parameters. According to the multi-task loss applied in Faster RCNN, we define the overall loss function for each image as:

$$L = \sum_i L_i^{det} + \lambda \sum_j L_j^{ctc}, \qquad (1)$$

where i is the index of a proposal in a mini-batch for detection, j is the index of a text line in a mini-batch for recognition, L_i^{det} is the detection loss on the i-th proposal, L_j^{ctc} is the CTC [17] loss on the j-th labeled text line, and $\lambda > 0$ is the trade-off parameter.

Following R-FCN, the first loss L^{det} on each proposal is defined as:

$$L^{det} = L^{cls} + \gamma c^* L^{reg}, \qquad (2)$$

where L^{cls} is the cross-entropy loss for classification, L^{reg} is the bounding box regression loss, c^* is the ground-truth label of the proposal ($c^* = 1$ means positive proposal, $c^* = 0$ means negative one), γ is the balance weight.

III. EXPERIMENT

A. Datasets

Chinese Business Card Database consists of 20 thousands Chinese business card photographs taken from different viewpoints under varying illumination conditions. It contains about 200 thousands text lines with a dictionary of 3800 characters. The business cards with different size, color and font are collected and labeled manually, we uniformly sample 16 thousands text lines as training data, the rest as testing data.

IAM Handwriting Database [6] is an English sentence database for offline handwriting recognition. It contains forms of unconstrained handwritten text, which are scanned at a resolution of 300dpi and saved as PNG images with 256 gray levels. All texts in the database are built using sentences provided by the Lancaster-Oslo/Bergen(LOB) Corpus. 657 writers produced between 1 and 59 handwritten documents

in their own handwriting. There are 747 documents (6,161 lines) in the training set, 105 documents (900 lines) in the validation1 set, 115 documents (940 lines) in the validation2 set and 232 documents (1,861 lines) in the test set. The texts in this database typically contain 450 characters in about nine lines.

B. Setup

In the experiments, we adopt the VGG-16 network as backbone of the network, conv1_1, conv1_2, conv2_1, conv2_2, conv3_1, conv3_2, conv3_3, conv4_1, conv4_2, conv4_3 are selected to be shared between detection and recognition tasks. The network can be trained end-to-end using the standard error back-propagation and ADAM optimizer [18]. In the training process, we set the balance weight $\lambda = 1$ in formula (1) and $\gamma = 1$ in formula (2). We perform two different training strategies: separate training and joint training. For separate training, the shared layers are initialized and fixed by pre-training a model for ImageNet classification, the rest layers (include the non-shared convolutional layers and the typical layers for detector and recognizer) are separately trained by the detector and recognizer. For joint training, we employ multi-task learning to jointly optimize model parameters. The shared layers are jointly trained by the detector and recognizer, and the whole network is trained in an end-to-end strategy using the standard error back-propagation.

We use these two training strategies to share different number of convolutional layers based on the Chinese Business Card database. For each strategy, a baseline method (share no layer) is compared with the proposed models. For Business Card database, a bounding box is considered as correct only if its Intersection over Union (IoU) ratio with any ground-truth is greater than 0.5 and the recognized character sequence also matches. In order to validate the performance further, we carry out experiments on a public database, IAM database, and compare our best model with the existing systems. For IAM database, the measurement of performance is the Word Error Rate (WER%), corresponding to the edit distance between the recognition result and ground-truth, normalized by the number of answers in ground-truth.

C. Evaluation Under Different Settings

Table I illustrates the results of our method using the combination of R-FCN and CRNN on Chinese Business Card database by separately training. The first column represents the shared layers, "None" indicating no layer is shared. The second column "Detection" gives text detection results, including Recall, Precision and Average Precision (AP). The column "Recognition" lists sequence and character accuracies given the ground-truth text regions. The last column shows the end-to-end recognition results, including Recall, Accuracy and F-measure at sequence line level. It can be seen that if we make convolutional layers shared until conv2_2, the end-to-end system can get improvement for both precision and computational efficiency. We also report the results of our

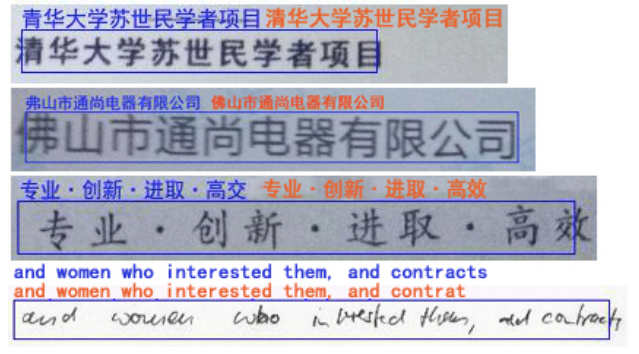Fig. 2. Examples of end-to-end text recognition results.

Fig. 3. The comparison between the models with shared layers and without shared layers, in case that the bounding boxes do not cover exactly. The orange character sequences are the recognition result of models with shared layers, the blue results from models without shared layers. The first three samples are selected from Chinese business card set and the last one is from IAM English handwriting set.

method using R-FCN and FCR on Chinese Business Card database in Table II , which obtain the similar trend.

Table III and Table IV show the joint training results, from which we can observe that:

(1) The models with a certain number of shared layers can achieve better performance than the baseline ("None", non-shared model). For the combination of R-FCN and CRNN, sharing convolutional layers from "conv1_1" to "conv2_2" performs best. Employing the FCR, sharing convolutional layers from "conv1_1" to "conv3_3" yields the highest F-measure.

(2) Compared with the separate training results, the shared layers contribute to the end-to-end recognition improvement by joint training. The knowledge learned from the correlated detection and recognition tasks may be effectively transferred between each other.

(3) The models with different numbers of shared layers have different performance. If sharing too many convolutional layers, the recognition accuracy may decline to some extent, such as sharing layers from "conv1_1" to "conv4_3" using R-FCN and CRNN.

Several end-to-end recognition examples are illustrated in Figure 2.

D. Comparison to Published Results

In this section, we perform experiments on IAM database and compare our models with the existing methods. Using the joint training strategy, we conduct experiments on combinations of R-FCN + CRNN and R-FCN + FCR, while the R-FCN can use VGG-16 or Resnet-50 [10] as its backbone. Feature Pyramid Network (FPN) [19] is applied to our recognizer as a generic feature extractor to improve the performance. The comparison of our experiments on IAM are shown in Table V, we can see the Resnet-50 based R-FCN + CRNN/FCR framework achieves better recognition results. Our best result is compared with reported results of existing methods in Table VI. When comparing the error rates, it is important to note that methods [20], [21], [22], [23] all used an explicit (ground-truth) line segmentation and a language model. Method [24] is an end-to-end handwritten paragraph recognition method, comparing to it, our system (using R-FCN and CRNN based

on Resnet-50) can achieve competitive result, which is slightly better in WER%.

E. Impact of Sharing Layers

In deep convolution neural networks, with the increasing convolutional layers, each point in the corresponding feature map has larger receptive field on the input image. Taking VGG-16 as an example, the size of receptive fields for "conv2_2" and "conv3_3" feature maps are 14×14 and 40×40, respectively.

In our framework, we use a text pooling layer to extract features from the feature map outputs of the last shared convolutional layer according to each detected region. And each point in the feature map has considerable receptive field on the input image. Thus, comparing to performing text recognition on each cropped text line of the input image, our method can obtain more effective information. It is important to note that if a detected box does not precisely cover the whole character area, directly cropping the detected region from input image for recognition will cause irreparable loss of information, especially the leftmost and rightmost boundary. In contrast, extracting from the feature map potentially acquires the information beyond the given bounding box and can contribute to improve the recognition performance, as shown in Figure 3.

F. Inference Efficiency

In this part, we evaluate inference efficiency of our architecture. All experiments are implemented on an NVIDIA Tesla M40 GPU with 12GB memory. As illustrated in Table VII and Table VIII, for the combination of R-FCN and CRNN, we can share convolutional layers from "conv1_1" to "conv3_3" without sacrificing the accuracy, this can make 53% computational saving and achieve $1.56\times$ speedup for text recognition. As for the combination of R-FCN and FCR, sharing layers from "conv1_1" to "conv4_3" acquires 75% computational saving and $2.41\times$ speedup in inference, while keeping the end-to-end result higher than baseline. In addition, comparing the

TABLE I

RESULTS OF OUR METHOD USING THE COMBINATION OF R-FCN AND CRNN ON CHINESE BUSINESS CARD DATABASE BY SEPARATELY TRAINING.

Shared Layers	Detection (R-FCN)			Recognition (CRNN)		End-to-End		
	Recall	Precision	AP	Seq Acc	Char Acc	Recall	Accuracy	F-measure
None	0.9723	0.8412	0.9485	0.8442	0.9799	0.7193	0.6755	0.6967
conv1_1 − conv1_2	0.9629	0.8577	0.9414	0.8397	0.9792	0.7233	0.6993	0.7110
conv1_1 − conv2_2	0.9660	0.8480	0.9423	0.8310	0.9777	**0.7223**	**0.6883**	**0.7048**
conv1_1 − conv3_3	0.9618	0.8552	0.9388	0.7911	0.9703	0.6802	0.6565	0.6881
conv1_1 − conv4_3	0.9563	0.8544	0.9313	0.6470	0.9387	0.5557	0.5389	0.5471

TABLE II

RESULTS OF OUR METHOD USING THE COMBINATION OF R-FCN AND FCR ON CHINESE BUSINESS CARD DATABASE BY SEPARATELY TRAINING.

Shared Layers	Detection (R-FCN)			Recognition (FCR)		End-to-End		
	Recall	Precision	AP	Seq Acc	Char Acc	Recall	Accuracy	F-measure
None	0.9723	0.8412	0.9485	0.8376	0.9780	0.7129	0.6695	0.6905
conv1_1 − conv1_2	0.9703	0.8449	0.9486	0.8407	0.9782	0.7302	0.6902	0.7096
conv1_1 − conv2_2	0.9708	0.8427	0.9491	0.8273	0.9759	**0.7213**	**0.6796**	**0.6998**
conv1_1 − conv3_3	0.9681	0.8492	0.9466	0.7945	0.9693	0.7008	0.6673	0.6837
conv1_1 − conv4_3	0.9536	0.8477	0.9288	0.5574	0.9121	0.4867	0.4696	0.4780

TABLE III

RESULTS OF OUR METHOD USING THE COMBINATION OF R-FCN AND CRNN ON CHINESE BUSINESS CARD DATABASE BY JOINTLY TRAINING.

Shared Layers	Detection (R-FCN)			Recognition (CRNN)		End-to-End		
	Recall	Precision	AP	Seq Acc	Char Acc	Recall	Accuracy	F-measure
None	0.9723	0.8412	0.9485	0.8442	0.9799	0.7193	0.6755	0.6967
conv1_1 − conv1_2	0.9668	0.8521	0.9449	0.8447	0.9800	0.7252	0.6938	0.7092
conv1_1 − conv2_2	0.9643	0.8600	0.9425	0.8388	0.9791	**0.7251**	**0.7020**	**0.7134**
conv1_1 − conv3_3	0.9638	0.8674	0.9442	0.8325	0.9780	**0.7212**	**0.7045**	**0.7128**
conv1_1 − conv4_3	0.9605	0.8623	0.9424	0.7976	0.9735	0.6734	0.6562	0.6647

TABLE IV

RESULTS OF OUR METHOD USING THE COMBINATION OF R-FCN AND FCR ON CHINESE BUSINESS CARD DATABASE BY JOINTLY TRAINING.

Shared Layers	Detection (R-FCN)			Recognition (FCR)		End-to-End		
	Recall	Precision	AP	Seq Acc	Char Acc	Recall	Accuracy	F-measure
None	0.9723	0.8412	0.9485	0.8376	0.9780	0.7129	0.6695	0.6905
conv1_1 − conv1_2	0.9710	0.8433	0.9492	0.8455	0.9793	0.7322	0.6903	0.7106
conv1_1 − conv2_2	0.9711	0.8466	0.9503	0.8393	0.9779	**0.7327**	**0.6934**	**0.7125**
conv1_1 − conv3_3	0.9712	0.8525	0.9508	0.8378	0.9779	**0.7402**	**0.7052**	**0.7223**
conv1_1 − conv4_3	0.9665	0.8459	0.9448	0.8294	0.9768	**0.7245**	**0.6883**	**0.7059**

TABLE V

RESULTS OF OUR METHOD ON IAM DATABASE.

Method	Network	Shared Layers	WER%
R-FCN+CRNN	Resnet-50	conv1 − res2c	24.4
	VGG-16	conv1_1 − conv2_2	30.9
R-FCN+FCR	Resnet-50	conv1 − res3d	28.8
	VGG-16	conv1_1 − conv3_3	31.2

TABLE VI

RESULTS OF DIFFERENT APPROACHES ON IAM DATABASE.

Mehods	Language Model	Line Segmentation	WER%
Pham *et al.* [23]	Yes	No	13.6
Kozielski *et al.* [22]	Yes	No	13.3
Doetsch *et al.* [21]	Yes	No	12.2
Bluche [20]	Yes	No	10.9
Bluche [24]	Yes	Yes	16.4
Bluche [24]	No	Yes	24.6
Our best result*	No	Yes	24.4

two tables, applying convolutional layers instead of BiLSTM in recognizer gains considerable speed improvements both theoretically and practically. Table IX shows the inference speed of our architecture on IAM database.

IV. CONCLUSION

We have presented an end-to-end trainable neural network, which solves text detection and recognition tasks in a novel integrated framework. By sharing the convolutional feature maps, we can simultaneously train these two models in a single unified pipeline. In addition, we designed a fully convolutional recognizer, which has been proved effective and efficient. In this way, given the computation of detection network, recognition network can achieve about 50% computation saving. Experiment results suggest that the proposed method is not only a cost-efficient solution for practical usage, but also an effective way of improving text detection and recognition accuracy. In this framework, we mainly focus on horizontal or near-horizontal texts, as for future work, we will pay more attention to handling texts of multiple orientations.

TABLE VII

THE INFERENCE SPEED OF OUR ARCHITECTURE USING THE COMBINATION OF R-FCN AND CRNN ON CHINESE BUSINESS CARD DATABASE.

Shared Layers	R-FCN	CRNN	Runtime speedup for CRNN	Computational saving for CRNN
None		0.1494±0.0008	-	-
conv1_1 − conv2_2	0.0888±0.0081	0.1166±0.0005	1.28×	26.78%
conv1_1 − conv3_3		0.0955±0.0003	1.56×	53.41%
conv1_1 − conv4_3		0.0804±0.0002	1.86×	80.04%

TABLE VIII

THE INFERENCE SPEED OF OUR ARCHITECTURE USING THE COMBINATION OF R-FCN AND FCR ON CHINESE BUSINESS CARD DATABASE.

Shared Layers	R-FCN	FCR	Runtime speedup for FCR	Computational saving for FCR
None		0.0583±0.0005	-	-
conv1_1 − conv2_2	0.0888±0.0081	0.0461±0.0004	1.26×	25.15%
conv1_1 − conv3_3		0.0368±0.0003	1.58×	50.16%
conv1_1 − conv4_3		0.0242±0.0002	2.41×	75.17%

TABLE IX

THE INFERENCE SPEED OF OUR ARCHITECTURE ON IAM DATABASE.

Method	Shared Layers	R-FCN Detection	Recognition	Runtime speedup for Recognition
R-FCN+CRNN (Resnet-50)	None	0.2790±0.0023	0.3455±0.0036	-
	conv1 − res2c		0.2847±0.0057	1.21×
R-FCN+FCR (Resnet-50)	None	0.2790±0.0023	0.2863±0.0019	-
	conv1 − res3d		0.1012±0.0003	2.83×

REFERENCES

[1] 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.

[2] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

[3] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5238–5246.

[4] C. Bartz, H. Yang, and C. Meinel, "Stn-ocr: A single neural network for text detection and text recognition," *arXiv preprint arXiv:1707.08831*, 2017.

[5] M. Busta, L. Neumann, and J. Matas, "Deep textspotter: An end-to-end trainable scene text localization and recognition framework," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[7] Y. Li, K. He, J. Sun *et al.*, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[14] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[15] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[16] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*, 2017, pp. 1243–1252.

[17] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[18] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[19] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[20] T. Bluche, "Deep neural networks for large vocabulary handwritten text recognition," Ph.D. dissertation, Université Paris Sud-Paris XI, 2015.

[21] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 279–284.

[22] M. Kozielski, P. Doetsch, and H. Ney, "Improvements in rwth's system for off-line handwriting recognition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 935–939.

[23] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 285–290.

[24] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Advances in Neural Information Processing Systems*, 2016, pp. 838–846.