2017 14th IAPR International Conference on Document Analysis and Recognition

A Robust Approach to Detecting Text from Images of Whiteboards and Handwritten Notes

Wei Jia[1,2*], Lei Sun[2], Zhuoyao Zhong[2,3*], Xiongjian Mo[2], Guoen Ma[2], Qiang Huo[2]
[1]Dept. of EEIS, University of Science and Technology of China, Hefei, China
[2]Microsoft Research Asia, Beijing, China
[3]School of EIE, South China University of Technology, Guangzhou, China
{v-jiawe, lsun, v-zhuzho, xiomo, v-guoenm, qianghuo}@ microsoft.com

***Abstract*—Detecting text from the images of whiteboards and handwritten notes is an important yet under-researched topic. In this paper, we present a robust approach to solving this challenging problem as follows. First, given a color image, color-enhanced Contrasting Extremal Regions (CERs) are extracted from its grayscale image as candidate text connected components (CCs). Second, four shallow neural networks are used to pre-prune efficiently most of unambiguous non-text CCs. Third, a Fast R-CNN based approach is proposed to filter out remaining non-text CCs by leveraging contextual information and to estimate the corresponding text-line orientation in the position of each remaining text CC. Fourth, each pair of the remaining text CCs within a certain distance and orientation constraint are connected to construct a directed graph. Finally, based on the estimated text-line orientations, candidate text-lines are generated easily by pruning greedily redundant edges in the graph to make each vertex have at most one direct successor and one direct predecessor, respectively. Our proposed approach has achieved promising results on an in-house testing set consisting of 285 camera-captured images of whiteboards and handwritten notes.**

Keywords—handwritten text detection; natural scene image; text-line grouping; Fast R-CNN

I. INTRODUCTION

Text detection in natural scene images has recently received considerate attention from computer vision and document analysis community [1, 2], largely due to its important roles in numerous useful applications, e.g., assistive technology for visually impaired people, OCR translation, image/video retrieval, etc. Many effective scene text detection approaches have been proposed in the literature, and very rapid progress has been made in this field. However, almost all these approaches focus only on printed text. The problem of handwritten text detection in natural scene images is relatively under-researched. Nevertheless, besides printed text, scene images may also contain lots of handwritten text in some specific scenarios, e.g., text on whiteboards, sticky notes and lecture notes. These handwritten contents usually contain useful information, therefore users often use their camera phones to take photos of them for revisiting. Detecting, recognizing and understanding these text contents can empower users to do more and achieve more. For example, popular apps like OneNote and Evernote index automatically handwritten notes in images to make them searchable. Vajda et al. [3] propose a camera-based whiteboard reading system for

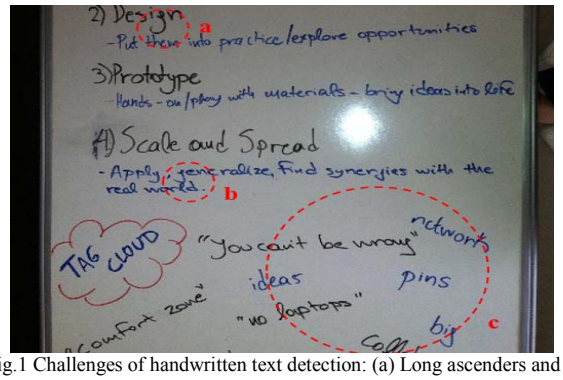

Fig.1 Challenges of handwritten text detection: (a) Long ascenders and descenders; (b) Touching strokes; (c) Unstructured layout.

understanding mind maps. To enable these new scenarios, robust handwritten text detection from natural scene images is a crucial prerequisite. However, like printed text detection, detecting handwritten text from natural scene images is another challenging problem. Other than the difficulties faced in printed text detection, e.g., high diversities of texts and the complexity of backgrounds [4], handwritten text detection has some extra challenges like long ascenders and descenders, touching strokes and unstructured layout (Fig. 1). To overcome these problems, more researches need to be done. In this paper, we take the "whiteboard" and "handwritten note" scenarios, which are two most typical scenarios of handwritten text detection from scene images, for example to study this problem.

Different from our scenarios, the problem of handwritten text detection in conventional document images has been studied for decades. Effective approaches have been proposed in the literature (e.g., [5, 6, 7]), and impressive results have been achieved on benchmark datasets (e.g., [8]). However, these methods cannot be adopted directly to address our problem as they rely on assumptions that images have plain backgrounds and text-lines are structured, which is not true for handwritten texts in images of whiteboards and handwritten notes. Another related work is handwritten scene text detection in video [9]. This work is based on lots of heuristic rules and less robust as evidenced by results in [9]. Hence, none of the above methods are suitable for robust handwritten text detection in our scenarios.

However, rapid progress has been made recently in the research field of natural scene text detection. The ideas of

*This work was done when Wei Jia and Zhuoyao Zhong were interns in the Speech Group of Microsoft Research Asia, Beijing, China.

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.138

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

Fig. 2 Pipeline of the proposed approach: (a) Input image; (b) Remaining CCs after text/non-text classification; (c) Constructed CC graph; (d) Result after local orientation based edge pruning; (e) Final detection result (Different CCs are presented in different colors).

newly proposed approaches can be borrowed to address our problem, although they focus only on printed text. Existing scene text detection methods can be roughly categorized into two mainstream groups: top-down methods (e.g., [10, 11, 12, 13, 14, 15]) and bottom-up methods (e.g., [16, 17, 18, 19, 20]). Top-down methods usually exploit fixed-size sliding windows to search for possible text regions in images. Recently, with the rapid development of deep learning, convolutional neural network (CNN) based top-down methods (e.g., [12, 13, 14]) have achieved very promising results. Bottom-up methods are usually composed of three major steps, i.e., candidate text CCs extraction (e.g., based on MSER [21] or SWT [22]), text/non-text classification and text-line grouping. The methods based on extremal-region (ER) or its variants (e.g., MSER [21], edge-enhanced MSER [23], color-enhanced CER [19]) are the most representative methods in this group due to their superior performance. Both CNN-based top-down methods and ER-based bottom-up methods can be used to solve the handwritten text detection problem.

In this paper, we try the ER-based methods first and present a robust bottom-up approach for text detection from images of whiteboards and handwritten notes. Following [19], we employ color-enhanced CER and shallow neural networks to extract candidate text CCs and filter out unambiguous non-text CCs, respectively. We find that these two simple methods can also work well for handwritten text. However, compared with printed text detection, text-line grouping problem for handwritten text detection is much more difficult due to the challenges illustrated in Fig. 1. Previous text-line grouping methods usually define some similarity metrics to estimate how likely two neighboring CCs are in the same text-line firstly, based on which all kinds of classical clustering algorithms are used to cluster isolated CCs into text-lines. The similarity metrics are based on handcrafted features like size differences, relative locations, orientation differences, etc. Although these features can work well for printed text-line grouping, they are not robust enough for handwritten text-line grouping. As stated in [17], some short, similar and sparse multi-line text scenarios, where CCs in multiple lines have both a close distance and a very similar structure, are not well handled. These cases are more common in handwritten scene images. To overcome these problems, we propose a new Fast R-CNN [24] based text-line grouping approach for multi-oriented handwritten text detection. We find that the text-line orientation in the position of each remaining CC can be estimated quite accurately by using Fast R-CNN, based on which the difficult text-line grouping problem can be simplified as a graph pruning problem. Only several simple pruning rules can make our approach robust enough. Moreover, the remaining non-text CCs which cannot be filtered out by the previous steps can be pruned effectively by this Fast R-CNN.

Our proposed approach has achieved promising results on an in-house testing set consisting of 285 camera-captured images of whiteboards and handwritten notes.

II. METHODOLOGY

A. Overview

As illustrated in Fig.2, the proposed approach includes four stages, i.e., candidate text CCs extraction, unambiguous non-text CCs pre-pruning, Fast R-CNN based text-line orientation estimation and non-text CC removal, and text-line grouping. The color-enhanced CER method [19] is used to extract candidate text CCs from the grayscale image directly. Details of other stages will be introduced in the subsequent sections.

B. Pre-pruing unambiguous non-text CCs

Following [19], extracted candidate text CCs are classified into five groups according to their aspect ratio and filled rate, namely, Long, Thin, Fill, Square-large and Square-small. For each of the first four groups, a shallow neural network with two hidden layers is used for text/non-text classification. As the neural networks are trained by an ambiguity-free learning strategy, only unambiguous non-text CCs are pre-pruned. Thanks to the low computation cost of the shallow neural networks, more than 96 percent of the non-text CCs in the first four groups are removed very efficiently. We don't use shallow neural network to classify Square-small CCs as more contextual information is needed to classify non-text CCs robustly. All the Square-small CCs are kept for later processing.

The printed Thin, Fill, Long, and Square-large models as well as the training data described in [19] are leveraged. In our implementation, only the more important Square-large and Long neural network models are retrained. The Thin and Fill models from [19] are reused directly. To train the Square-large model for handwritten text, the training data for the printed Square-large model from [19], which includes 1,633,024

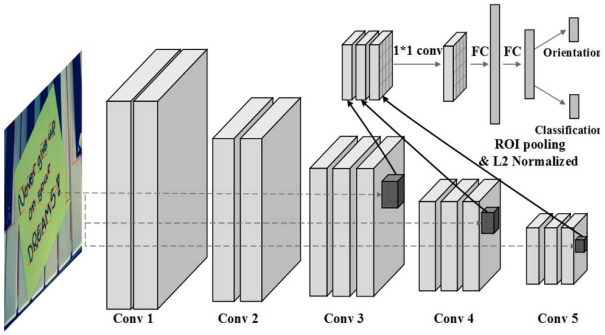

Fig. 3 Architecture of the proposed Fast R-CNN network for local skew estimation and text/non-text classification.

negative samples and 1,459,712 printed positive samples, are reused. Moreover, the printed Square-large model is also used as our seed model. Then we use the same "bootstrap" strategy [25] to collect an additional 152,320 positive samples for handwritten text, which are then added into the original training set. After that, new handwritten Square-large model is obtained by retraining the model with the enriched training data.

To train Long model for handwritten text, we made two modifications. Firstly, as illustrated in Fig. 2, many Long CCs are skewed. To simplify the succeeding text/non-text classification problem, we rectify the skew of each Long CC to make it in near horizontal direction before classification. The rough orientation of each Long CC is estimated by using principle component analysis (PCA). Secondly, unlike [19], Long CCs in this work are resized to 24x56 pixels instead of 12x28 pixels to ensure that thin strokes are not corrupted after image resizing step.

C. Text-line orientation estimation and non-text CC removal

Fast R-CNN is an effective approach for object detection. Given an input image and a set of pre-extracted object proposals, the network first processes the whole image with several convolutional (conv) and max pooling layers to produce a conv feature map. Then, for each proposal, a region of interest (ROI) pooling layer is used to extract a fixed-length feature vector from the feature map. Each feature vector is fed into a sequence of fully connected layers to predict its object category and refine the corresponding bounding box [24].

In our approach, we use Fast R-CNN to estimate the text-line orientation in the position of each remaining CC and the text/non-text score for that CC. Various CNN architectures, such as AlexNet [26], VGGNet [27], ResNet [28], can be exploited as the base networks in the Fast R-CNN framework. Here, we take the standard VGG16 network [27] for example to introduce our approach. The architecture of our proposed Fast R-CNN network is illustrated in Fig. 3. Given an input image, we use the bounding boxes of remaining candidate text CCs as region proposals. For each proposal, instead of pooling features from the last conv layer ("Conv5_3") only [24], we use a skip pooling [29] technique to combine features from different conv layers ("Conv3_3", "Conv4_3" and "Conv5_3") to enhance the representation ability of features for small text. Specifically, for each proposal, three feature descriptors with a fixed spatial extent of 4×4 are pooled from the "Conv3_3", "Conv4_3" and "Conv5_3" layers respectively using ROI pooling [29]. These descriptors are then L2-normalized and re-scaled back by a learnable per-channel scale parameter (initialized to 2 measured on our training set). After that, the normalized descriptors are channel-wise concatenated and dimension reduced with a 1×1 convolutional layer to obtain a fixed-length feature descriptor of size $512 \times 4 \times 4$, which is fed into two fully-connected layers with 1024 and 512 nodes respectively. Finally, two sibling output layers are used to predict the text-line orientation angle and text/non-text score for each proposal simultaneously. It should be noted that the pooled fixed-length feature descriptor has much larger receptive field on the input image than the original proposal. Therefore, this feature contains rich enough context information to deal with the ambiguous non-text CCs whose shapes or textures are similar to text instances. Based on this, CCs classified as non-text by the Fast R-CNN network are removed directly.

A multi-task loss function is defined for training the above model. Let c and c^* be the true and predicted label, θ and θ^* be the ground-truth and predicted text-line orientation angle. The loss function is defined as follows:

$$L(c, c^*, \theta, \theta^*) = \lambda_{cls} L_{cls}(c, c^*) + \lambda_{ori} L_{ori}(\theta, \theta^*) \quad (1)$$

where $L_{cls}(c, c^*)$ and $L_{ori}(\theta, \theta^*)$ are the classification loss and regression loss respectively. We adopt the standard softmax loss for $L_{cls}(c, c^*)$ and the smooth-L_1 loss [24] for $L_{ori}(\theta, \theta^*)$. λ_{cls} and λ_{ori} are two control parameters, which are set as 1 and 16 respectively.

The standard back-propagation and stochastic gradient descent (SGD) algorithm are used to train our model. The parameters in the first five conv layers ("Conv1"-"Conv5") are initialized by a pre-trained VGG16 model [27] for ImageNet classification. The other layers are initialized by using random weights with Gaussian distribution of 0 mean and 0.01 standard deviation. The model is fine-tuned by 150,000 iterations. In each iteration, at most 512 region proposals from one image are randomly sampled to constitute a mini-batch. The base learning rate is 0.001 and multiplied by 0.1 every 30,000 epochs. The momentum and weight decay are set to 0.9 and 0.0005 respectively. During training, raw images are rescaled such that its short side equals to 500 pixels while long side no more than 800 pixels.

D. Text-line grouping

The proposed text-line grouping approach is composed of two steps, i.e., graph construction and redundant edge pruning. Without loss of generality, we assume that text-line orientation angles range from -45° *to* 45° (with respect to the horizontal direction) in our task.

In the first step, each pair of CCs within a certain distance and orientation constraint are connected to construct a directed graph. As shown in Fig. 4, let CC_i, CC_j denote two neighboring CCs whose centroids are represented by c_i, c_j. Let $\overrightarrow{c_i c_j}$ be the line segment connecting c_i and c_j, θ_{ij} be the direction angle of $\overrightarrow{c_i c_j}$, d_{ij} be the length of $\overrightarrow{c_i c_j}$. If $|\theta_{ij}| < 50^\circ$ and $d_{ij} < R_d$, then c_i and c_j are connected by an edge. Here, R_d is a constant threshold set to three times of average height of all remaining CCs. Assume CC_i is on the left of CC_j, the direction of the edge

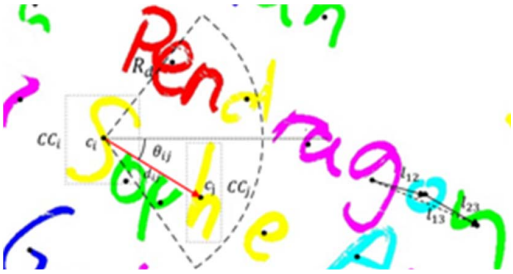

Fig.4 Illustration of the graph construction approach.

is defined as from CC_i to CC_j. Then CC_i is a direct predecessor of CC_j and CC_j is a direct successor of CC_i. Note that, if three edges form a loop and have similar orientation (e.g., l_{12}, l_{23}, l_{13} in Fig. 4), the longest edge l_{13} is pruned.

After the above steps, redundant edges need be pruned to generate text-lines. To achieve this purpose, each edge is assigned a weight, which is defined as follows:

$$w = \text{abs}(\theta - \theta_{est}) \times x_{diff} \quad (2)$$

where θ denotes the direction angle of the edge, θ_{est} is the estimated text-line orientation angle of the source CC of the edge (e.g., CC_i in Fig. 4), x_{diff} is the length of the horizontal projection of the edge. Consequently, the redundant edges are pruned by the following two rules: If w of an edge is larger than a threshold τ, it will be pruned firstly. Then, if a CC with more than one successor/predecessor, only the successor/predecessor with the lowest weight is kept.

III. EXPERIMENTS

As there is no standard database for handwritten text detection from images of whiteboards and handwritten notes, we create our own dataset of 285 camera-captured images of whiteboards and handwritten notes for evaluating the performance of our proposed method. The dataset has been made as diverse as possible. There are totally 4,893 text-lines in this dataset, among which the orientation angles of 4,681 text-lines are less than 10 degrees with respect to the horizontal direction. To prepare a multi-oriented text detection testing set, we rotate the above 285 images from -30 to 30 degrees at an interval of 10 degrees.

A. Text and non-text classification

We compare the performance of the Square-large model before and after model retraining. The positive samples in the testing set are labeled from the above 285 images. The negative samples from [19] are reused directly for saving labeling cost. Results can be seen in Table 1. The accuracy of positive samples increases from 74.2% to 94.9% after adding only 10% additional positive (handwritten text) samples into the original training set, while the accuracy of negative samples decreases a little.

B. Effect of Fast R-CNN based approach

The performance of Fast R-CNN based text-line orientation estimation and text/non-text classification is evaluated in this section. Moreover, other than the standard VGG16 architecture, two more efficient architectures (ZFNet [30] and DarkNet [31]) have also been evaluated. The experimental setup is as follows:

Table 1. Performance comparison of the Square-large models

Model type	Text set size	Acc.	Non-text set size	Acc.
Printed	13,285	74.2%	39,442	98.3%
Handwritten	13,285	94.9%	39,442	96.1%

Table 2. Comparison of different architectures

Model	Angle Difference		Classification Acc.		Speed
	<= ±3°	<= ±6°	Text	Non-text	
VGG16	90.3%	98.6%	98.3%	89.3%	8.77s
ZFNet	84.1%	97.0%	97.7%	84.4%	1.57s
DarkNet	87.6%	98.0%	98.3%	86.6%	1.34s

Table 3. Performance of our handwritten text detection approach

Rotation angle	Recall	Precision	F-score
0°	86.2%	88.8%	87.5%
±10°	82.0%	88.7%	85.2%
±20°	78.0%	85.7%	81.7%
±30°	73.2%	82.6%	77.6%

1) Training set: 5,740 raw images are labeled to construct our training set, in which most text-lines are near horizontal. The bounding box and orientation of each text-line are labeled by human labelers. Since most text-lines in these images are near horizontal, a new multi-oriented training set is generated by rotating these 5,740 images from -45° to 45° at an interval of 5°. When an image is rotated, the labeled bounding box and orientation of each text-line is adjusted accordingly. If an image contains a text-line whose orientation angle is not in the range [-45°, 45°], it will be removed from the training set. After these steps, we get 97,000 images for model training. Each image contains about 150 remaining CCs on average.

2) Ground-truth generation for each CC: We label acccurately the bounding boxes and orientations of text-lines in all images. Each CC's text/non-text label and text-line orientation can be assigned as follows. Let B_i be the bounding box of a CC (CC_i), $S(B_i)$ be the area of B_i, B_k^{GT} be the bounding box of a text-line (l_k), $S(B_i \cap B_k^{GT})$ be the overlapped area of B_i and B_k^{GT}. Then, for a CC, CC_i, if there exists a text-line l_k such that $S(B_i \cap B_k^{GT})/S(B_i) > 0.7$, CC_i is considered as a text CC, otherwise a non-text CC. If CC_i is a text CC, the ground truth orientation of the corresponding text-line is taken as the text-line orientation in the position of CC_i.

3) Evaluation criterion for orientation estimation: The above ground truth generation method for orientation could cause some small estimation errors. So, we use an error tolerant evaluation criterion. For a CC in the testing set, let θ_i and θ_i^* denote the estimated and ground truth text-line orientation in its position, respectively. If $|\theta_i - \theta_i^*| < T_\theta$, θ_i is considered as an accurate orientation estimation, otherwise an inaccurate estimation. T_θ is an adjustable threshold. Then, the percentage of the CCs in the testing set whose orientations are estimated accurately is calculated as the performance metric for orientation estimation.

The above multi-oriented text detection testing set is used as the testing set. Results can be seen in Table 2, which

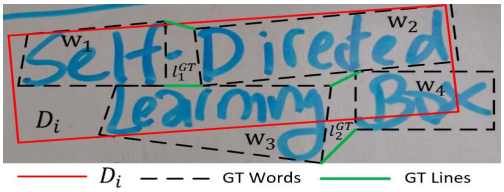

Fig. 5 D_i is a detection result, $w's$ are the GT words it covers, w_1 and w_2 are connected as a GT text-line l_1^{GT}, w_3 and w_4 as l_2^{GT} . F_{ik} is computed for l_1^{GT} and l_2^{GT} separately. Here, l_1^{GT} is considered to be matched to D_i as its matching score is higher.

demonstrate that the text-line orientation in the position of each CC can be estimated quite accurately by using Fast R-CNN. Moreover, more than 80% of remaining ambiguous non-text CCs can be rejected by Fast R-CNN. In terms of network topology, the VGG16 network achieves slightly better results on orientation estimation and text/non-text classification than ZFNet and DarkNet, but it is more than 5.5 times slower, which can be seen in the last column in Table 2. The speed is measured by calculating the average running time of the Fast R-CNN module of our approach for a 500*535 image on a standard CPU (Xeon E5-2665, 2.4G Hz) with Caffe framework [32]. After code optimization, the average running time on a standard CPU can be reduced to 0.28s for a 500*535 image with DarkNet.

C. *Whole system performance*

We choose the most efficient model, i.e., DarkNet, as the base network to evaluate the performance of our handwritten text detection approach. The evaluation method for multi-oriented text detection is still an open problem. The most widely used evaluation protocol is the one proposed by Wolf and Jolion [33], which is adopted by ICDAR-2011 and ICDAR-2013 Robust Reading Competitions. However, this evaluation protocol has some limitations [34] and is not suitable for multi-oriented text detection task [35]. Although ICDAR-2015 [2] and MSRA-TD500 [35] evaluation protocols can handle multi-oriented texts, they have limitations too. The ICDAR-2015 evaluation protocol performs word-level evaluation, which assigns high penalties to text-line-level detections. The MSRA-TD500 evaluation protocol requires de-rotating the ground truth and detected bounding boxes before calculating their overlap ratios, which will bring extra measurement errors into the evaluation protocol. Moreover, this protocol performs text-line-level evaluation only. To overcome these problems, Calarasanu et al. [34] proposed a more effective evaluation protocol for text detection tasks. However, when a detected bounding box enclosing a text-block that includes several text-lines, it could be considered as a correct detection too. This will cause the results of text detection module not consistent with the results of text recognition module, as text recognition engines generally take a single text-line as input. In this paper, we borrow the idea of Calarasanu's protocol and propose a stricter evaluation protocol for our task as follows:

1) Evaluation protocol for multi-oriented text detection: The ground truth (GT) bounding box of each word in testing set is labeled by a quadrilateral firstly. Then words in the same text-line are assigned the same text-line ID. We assume the space

Fig. 6 Some examples.

between neighboring words in a text-line is less than twice the height of the word with higher height. Let D_i be a detected bounding box, W be a set of GT words that have overlaps with D_i. The GT words in W could belong to different text-lines, so we group them into K text-lines denoted as L^{GT} according to their text-line ID (see Fig. 5 for an example). Next, for each candidate text-line l_k^{GT} in L^{GT}, an associated matching score F_{ik} is calculated to describe how well D_i is matched to l_k^{GT} as follows:

$$Acc_{ik} = \frac{Area(D_i \cap l_k^{GT})}{Area(D_i)}, \quad Cov_{ik} = \frac{Area(D_i \cap l_k^{GT})}{Area(L_k)} \tag{3}$$

$$F_{ik} = 2 \times \frac{Acc_{ik} \times Cov_{ik}}{Acc_{ik} + Cov_{ik}} \tag{4}$$

where $Area(D_i \cap l_k^{GT})$ denotes the overlapped area of D_i and l_k^{GT}. The bounding box of l_k^{GT} is calculated by connecting the vertices of the corresponding word bounding boxes in counterclockwise direction. The l_k^{GT} with the highest matching score in L^{GT} is considered to be matched to D_i. Furthermore, if $Acc_{ik} > 0.5$ & $Cov_{ik} > 0.5$, D_i is considered to be a correctly detected text-line and the words in l_k^{GT} are considered to be detected correctly; otherwise, D_i is a false alarm.

Based on the above matching strategy, each detected bounding box D_i is assigned a "correct detection" or "false alarm" label and each GT word is assigned a "detected" or "not detected" label. Then, the average precision, recall and F-score of a text detection approach in the whole testing set are computed as follows:

$$Precision = \frac{Number\ of\ correctly\ detected\ lines}{Number\ of\ detected\ lines} \tag{5-1}$$

$$Recall = \frac{Number\ of\ recalled\ words}{Number\ of\ GT\ words} \quad (5\text{-}2)$$

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5\text{-}3)$$

Experimental results of our approach based on the above evaluation protocol are presented in Table 3. Although our testing set is very challenging, our approach can still achieve 87.5% F-score on the near horizontal testing set, i.e., original 285 testing images. As the rotation angle of testing images increases, the performance of the proposed approach degrades slowly. Compared with the precision rate, the recall rate degrades faster, especially the "±30°" case, due to two reasons: 1) More text CCs are pre-pruned wrongly by shallow neural networks; 2) Connection angle computed by the centroids of CC pairs is not accurate enough, which affects edge pruning.

Some example results can be seen in Fig. 6, which demonstrate that our approach is robust to text-line orientations, long ascenders and descenders, unstructured layout, complex background, etc. However, as shown in the bottom row of Fig. 6, our approach is not robust enough to touching strokes, which is the major limitation of our approach.

IV. CONCLUSION

In this paper, we present a robust approach to detecting text from images of whiteboards and handwritten notes. Thanks to the proposed Fast R-CNN based text-line grouping method, our handwritten text detector is not only robust to horizontal text-lines, but also can deal with oriented text-lines. Our proposed approach has achieved promising results on a challenging in-house testing set.

V. REFERENCES

[1] D. Karatzas, et al., "ICDAR 2013 robust reading competition," in *ICDAR*, 2013, pp. 1484-1493.

[2] D. Karatzas, et al., "ICDAR 2015 robust reading competition," in *ICDAR*, 2015, pp. 1156-1160.

[3] S. Vajda, T. Plotz and G. A. Fink, "Camera-based whiteboard reading for understanding mind maps," *Int. J. Pattern Recognition and Artificial Intelligence,* vol. 29, no. 3, 1553003, 2015.

[4] Q.-X. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Tran. PAMI,* vol. 37, no. 9, pp. 1480-1500, 2015.

[5] Z. Razak, et al., "Off-line handwriting text line segmentation: A review," *Int. J. Compter Science and Network Security,* vol. 8, no. 7, pp. 12-20, 2008.

[6] D. Fernandez-Mota, J. Llados, and A. Fornes, "A graph-based approach for segmenting touching lines in historical handwritten documents," *IJDAR,* vol. 17, no. 3, pp. 293-312, 2014.

[7] N. Ouwayed and A. Belaid, "A general approach for multi-oriented text line extraction of handwritten documents," *IJDAR,* vol. 15, no. 4, pp. 297-314, 2012.

[8] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," in *ICDAR*, 2013, pp. 1402-1406.

[9] P. Shivakumara, A. Dutta, U. Pal, and C. L. Tan, "A new method for handwritten scene text detection in video," in *ICFHR*, 2010, pp. 387-392.

[10] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in *CVPR*, 2004, pp. 366-373.

[11] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *ICCV*, 2011, pp. 1457-1464.

[12] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localization in natural images," in *CVPR*, 2016, pp. 2315-2324.

[13] Z. Tian, W.-L. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *ECCV*, 2016, pp. 56-72.

[14] M.-H. Liao, B.-G. Shi, X. Bai, X.-G. Wang, and W.-Y. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *AAAI*, 2017, pp. 4161-4167.

[15] Z.-Y. Zhong, L.-W. Jin, and S.-P. Huang, "DeepText: A new approach for proposal generation and text detection in natural images," in *ICASSP*, 2017, pp. 1208-1212.

[16] X.-C. Yin, X.-W Yin, K.-Z Huang, and H.-W Hao, "Robust text detection in natural scene images," *IEEE Tran. PAMI,* vol. 36, no. 5, pp. 970-983, 2014.

[17] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. PAMI,* vol. 37, no. 9, pp. 1930–1937, 2015.

[18] H.-I. Koo and D.-H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE TIP,* vol. 22, no. 6, pp. 2296-2305, 2013.

[19] L. Sun, Q. Huo, W. Jia, and K. Chen, "A robust approach for text detection from natural scene images," *Pattern Recognition,* vol. 48, no. 9, pp. 2906-2920, 2015.

[20] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *ACCV*, 2010, pp. 770-783.

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC*, 2002, pp. 384-393.

[22] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963-2970.

[23] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *ICIP*, 2011, pp. 2609-2612.

[24] R. B. Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440-1448.

[25] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. PAMI,* vol. 20, no. 1, pp. 39-51, 1998.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097-1105.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[28] K.-M. He, X.-Y. Zhang, S.-Q Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770-778.

[29] S. Bell, C. L. Zitnick, K. Bala, and R. B. Girshick, "Inside-Outside Net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016, pp. 2874-2883.

[30] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818-833.

[31] J. Redmon, "Darknet: Open source neural networks in c," 2013–2016. [Online]. Available: https://pjreddie.com/darknet/.

[32] Y.-Q. Jia, et al., "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.

[33] C. Wolf and J. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *IJDAR,* vol. 8, no. 4, pp. 280-296, 2006.

[34] S. Calarasanu, J. Fabrizio, and S. Dubuisson, "What is a good evaluation protocol for text localization systems? Concerns, arguments, comparisons and solutions," *Image and Vision Computing,* vol. 46, pp. 1-17, 2016.

[35] C. Yao, X. Bai, W.-Y. Liu, Y. Ma, and Z.-W. Tu, "Detecting texts of arbitrary orientations in natural images," in *CVPR*, 2012, pp. 1083-1090.