A High Accuracy Text Detection Model of Newly Constructing and Training Strategies

Kha Cong Nguyen[a] and Ryosuke Odate[b]
Research and Development Group, Hitachi, Ltd., Tokyo, Japan

Keywords: Text Detection, Text Recognition, Anchor Boxes, Clustering, Feature Extraction, Non-Maximum Suppression.

Abstract: Normally, text recognition systems include two main parts: text detection and text recognition. Text detection is a prerequisite and has a big impact on the performance of text recognition. In this paper, we propose a high-accuracy model for detecting text-lines on a receipt dataset. We focus on the three most important points to improve the performance of the model: anchor boxes for locating text regions, backbone networks to extract features, and a suppression method to select the best fitting bounding box for each text region. Specifically, we propose a clustering method to determine anchor boxes and apply novel convolution neural networks for feature extraction. These two points are the newly constructing strategies of the model. Besides, we propose a training strategy to make the model output angles of text-lines, then revise bounding boxes with the angles before applying the suppression method. This strategy is to detect skewed and downward/upward curved text-lines. Our model outperforms other best models submitted to the ICDAR 2019 competition with the detection rate of 98.87% (F1 score) so that we can trust the model for detecting text-lines automatically. These strategies are also flexible to apply for other datasets of various domains.

1 INTRODUCTION

Recently, document recognition for information retrieving and digitally storing has garnered a large amount of interest from the deep learning and computer vision communities due to the important information included in documents and the huge amount of scanned and captured document accumulated during many decades. To recognize documents, the first task is text detection. Text detection locates and extracts text regions from documents that encompass many complex layouts such as text regions, tables, figures, and even noised regions. This task has a big influence on the performance of its consequent text recognition task.

Currently, we receive many requests for developing an automatically text-image recognizing system for scanned receipts, invoices, and form documents. By applying cutting-edge technologies such as the method composing Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Connectionist Temporal Classification (CTC) or the method combining CNN, RNN, and Attention-based sequence prediction (Attn) (Jeonghun, 2019), we achieved high recognition rates for text-line recognition. The remaining bottleneck is a high accuracy text-line detection model where we can trust machines for text-line detection mostly. Besides, the method needs to be effective and flexible to apply for detecting text on datasets including complex backgrounds, various styles, and languages.

In this paper, we propose a text-line detection model, based on the Faster R-CNN architecture for object detection (Ren, 2015). In this model, a CNN is used to extract feature maps at multiple deep levels, and a set of anchor boxes with different scales and aspect ratios is employed to locate text regions on the feature maps. Finally, a selection algorithm like Non-Maximum Suppression (NMS) is utilized to select a bounding box (denoted bbox shortly) with the highest confidence score of containing a text region (Bodla, 2017). It discards other bbox candidates that are overlapping the selected bbox.

[a] https://orcid.org/0000-0002-2760-8724
[b] https://orcid.org/0000-0002-9467-2275

Nguyen, K. and Odate, R.
A High Accuracy Text Detection Model of Newly Constructing and Training Strategies.
DOI: 10.5220/0010343406350642
In *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2021)*, pages 635-642
ISBN: 978-989-758-486-2

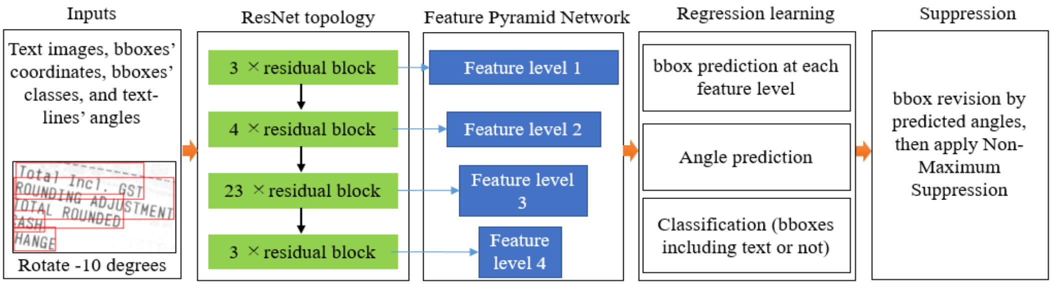


Figure 1: Overview of our proposed text-detection network.

The aforementioned architecture of the text detection model is typically used by other research groups for text detection, but there are three our main contributions: applying clustering methods to produce the best fitting anchor-set for training the model; implementing CNNs for feature extraction that keep the almost similar number of trainable parameters and computational complexity; proposing a training strategy for detecting skewed and downward/upward curved text-line regions.

The remaining of this paper is organized as follows. In Section 2, we present the related work of text detection. Then, in Section 3, we describe the overview of our text detection network and the detailed implementation with an emphasis on our main contributions. We show the experiment result of each contribution and comparison with other best models in Section 4. Finally, in Section 5, we conclude and summarize the main points in this paper.

2 RELATED WORK

This section presents the related work of text detection, including conventional methods and CNN-based methods. Conventional methods produce not good results on documents of complex backgrounds, so we mainly focus on advanced methods employing CNNs.

2.1 Conventional Methods

Before CNNs became popular, conventional methods for text detection had adopted text-component extraction by edge detection or extremal region extraction, and then text components were conjoined by geometric relations to make text-lines. The typical methods here include the method by (Dinh, 2007), the Stroke Width Transform method (SWT) by (Epshtein, 2010), and the Maximally Stable Extremal Region method (MSER) by (Huang, 2014). Those methods, however, are outdated when dealing with documents of complex backgrounds, low resolution, stroke distortion, or touching text-lines.

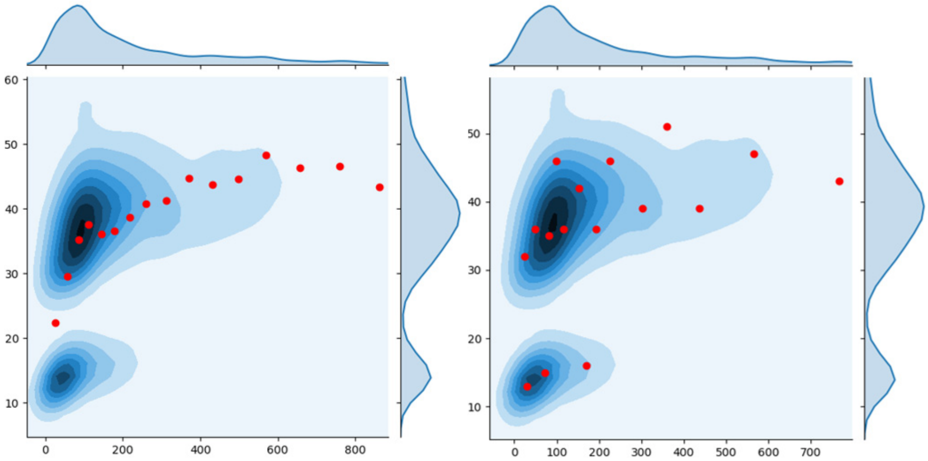
2.2 CNN-based Methods

With the great advances of CNN, a wider variety of text detection methods has been explored. (Zhou, 2017) proposed a faster and accurate model for scene text detection. The method utilizes an CNN that can directly produce either rotated rectangular or quadrangular bboxes of text regions. The loss functions for training to predict rotated rectangular bboxes are a region overlapping loss and a cosine similarity loss. The loss function for training to predict coordinates of quadrilateral bboxes is a regression loss – smooth L1 loss.

For methods using the same architecture as our work, (Zhong, 2019) applied direct regression (Wenhao, 2017) for Faster R-CNN to predict quadrilateral bboxes of arbitrarily oriented text-lines. Unlike regular Faster R-CNN using regression training to predict offsets between pre-defined anchor boxes and rectangular bboxes of text regions, the method predicted core points and the offsets from these points to quadrilateral bboxes of text regions. The method is effective for predicting quadrilateral bboxes but for rectangular bboxes, it works worse than the conventional Faster R-CNN method because the network needs to learn to output core the points besides the offsets. That is proved by the comparison between our method and the second rank method submitted to the ICDAR competition (Huang, 2019) as shown in Section 4.

3 METHODOLOGY

In this section, we present the overview of our text detection network and the detailed implementation of each component in the proposed network.



(a) Clustering both width and height of bounding boxes

(b) Clustering bounding boxes on *IoU* metrics

Figure 2: Sixteen estimated anchor boxes (red points) based on clustering the training bounding. Horizontal and vertical axes are width and height of bboxes respectively.

3.1 Overview of Proposed Network

Figure 1 displays an overview of our proposed text-detection network. The training inputs of the network include text images, coordinates of ground-truth rectangular bboxes, the class of bboxes (value of 1, indicating that they contain text regions), and rotated angles when applying a rotating data-augmentation method randomly. We use a ResNet architecture network to extract features of input images after four container layers, defined as conv2_x, conv3_x, conv4_x, and conv5_x by (Kaiming, 2016). This is known as the Feature Pyramid Network. We apply the newest techniques such as the split-transform-merge strategy (Xie, 2017) and Squeeze-and-Excitation (SE) (Hu, 2018) to bottleneck residual blocks of ResNet. The details of feature extraction networks are presented in Section 3.3.

Next, at each considering point of feature maps, we match pre-defined anchor boxes to ground-truth rectangular bboxes. The methods to determine the anchor boxes are presented in Section 3.2. Anchor boxes overlap ground-truth rectangular bboxes more than a certain threshold of *IoU* (defined in Section 3.2), are assigned a positive label (value of 1). In our configuration, the threshold is set up to 0.5. Only positive label bboxes are considered when training offsets and angles regressively. Others are marked by negative labels (value of zero, indicating that they do not contain text regions) and ignored. The network is trained to produce the probabilities of anchor boxes containing text regions. Besides, we perform regression training to make the network produce offsets between pre-defined anchor boxes of a positive label and their corresponding ground-truth bboxes. From pre-defined anchor boxes and their predicted offsets, the network can produce predicted bboxes in the testing phase. We also train regression for oriented angles of bboxes. The loss (L_{text}) to train the network to yield the probabilities of bboxes including text regions is the binary cross-entropy loss function. The regression losses for training the network to predict coordinates ($L_{bbox\ regr}$) and angles ($L_{agl\ regr}$) of bboxes are L1 loss functions. To balance loss values, we add a weight *w* = 0.01 to the loss of angle regression practically. *w* has a small impact to the performance of model. The total loss (L_{total}) is calculated as Eq. (1).

$$L_{total} = L_{bbox\ regr} + w * L_{agl\ regr} + L_{text} \qquad (1)$$

The network may produce many bboxes for each text region. We finally revise proposed bboxes with predicted angles before applying Non-Maximum Suppression to discard overlapped bboxes of smaller confident scores that show whether they include text or not. This step is described in Section 3.4 specifically.

3.2 Anchor Box Determination

Anchor box sizes are one of the most influential factors in the performance of text detection models. We perform a statistic from training data to determine which anchor boxes are the most suitable to allocate text regions. First, training bboxes are rescaled with scaled ratios of whole images when applying data augmentation methods such as resizing or cropping. Then, we can choose one of the following methods to produce the best fitting set of anchor boxes.

1) Method 1: clustering training bboxes into groups based on both their width and height

We apply the K-mean algorithm to the Euclidian distance between training bboxes and centroids for clustering training bboxes into k groups. The center point of each group is an anchor box for training the network. The larger number of anchor boxes helps the network allocate text regions more precisely, however, it also increases the execution load of the network. Therefore, we need to trade-off between the number of anchor boxes and the preciseness of the network.

2) Method 2: clustering by Intersection of Union

Method 1 does not put priorities on bigger bboxes and smaller bboxes. We propose a method, which clusters training bboxes into groups, based on Intersection of Union (IoU) metrics as Eq. (2) between them and centroids of groups.

$$IoU = \frac{Total\ pixels\ in\ ovellaped\ areas}{Total\ pixels\ in\ union\ areas} \quad (2)$$

Figure 2 shows the distribution of determined anchor boxes by two methods. The first method treats every bbox with the same priority, so even at thinly scattered areas, it also proposes anchor boxes. Otherwise, the second method proposes more anchor boxes in the highly-dense areas and fewer anchor boxes in other areas. Therefore, it can estimate anchor boxes better than the first method. We select the second method to estimate the anchor boxes when comparing our method with others in Section 4.

3.3 Networks for Feature Extraction

Normally, a network to extract feature maps for locating bboxes is designed in the ResNet architecture (Kaiming, 2016). The ResNet architecture is easy for mapping feature maps to input images when allocating text regions because it down-samples input features two times after each container layers. This is more complicated when using other deeper architectures like Inception ResNet (Szegedy, 2016) where each down-sampling block adds a different offset to output sizes. ResNet is constructed by many residual blocks as shown in Figure 3 (a). Each residual block is built in a bottleneck architecture which includes three convolutional layers: 1×1 convolution, 3×3 convolution, and 1×1 convolution. The first convolutional layer reduces the dimensionality and the last one restores it. The number of filters of the first convolutional layer is called bottleneck width (denoted d). Another feature of the residual block is the addition of the input to its output, known as skip connection or residual connection, to produce the input for its next layer. That makes the network can backpropagate the gradient to early layers so that it can avoid the gradient vanished problem and allows the network to learn deeper in comparison with traditional convolutional neural networks like VGG net (Simonyan, 2014).

One of the upgraded versions of ResNet is applying the split-transform-merge strategy of the Inception ResNet model to each original residual block to create a new type block named ResNeXt as shown in Figure 3 (b) (Xie, 2017). The block keeps the topology, the computation complexity, and spatial output sizes the same as the original residual block. In this block, a new dimension named cardinality (denoted as C) is introduced. It is the number of aggregated transformation paths. In Figure 3 (a), we can estimate the number of trainable parameters of the residual bottleneck block is 256×64 + 64×64×3×3 + 64×256 where d = 64. We can split and transform the original residual block to the corresponding block

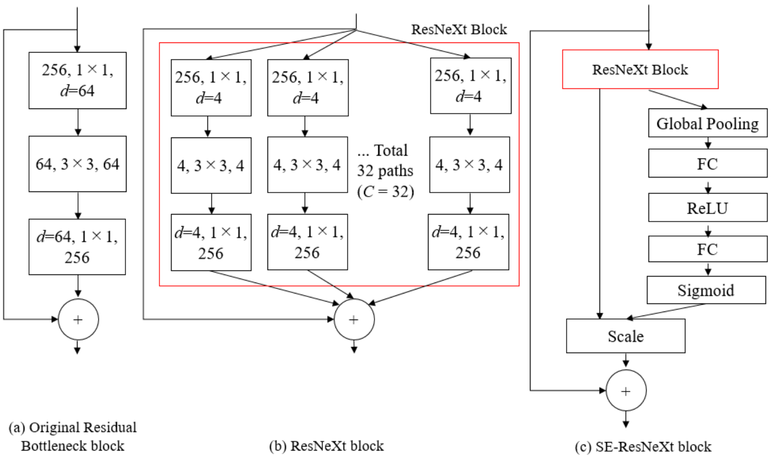


Figure 3: Different residual blocks. Each layer is shown as (number of input channels, filter size, number of output channels).

as shown in Figure 3 (b) with the same number of training parameters: *C* × (256 ×*d* + 3×3×*d*×*d* + *d*×256) where *C*=32 and *d* = 4.

Furthermore, (Hu, 2018) has recently proposed a Squeeze-and-Excitation block (SE block) that can be combined with the ResNeXt block and bring a significant improvement to the performance of the original ResNet for classification. Therefore, we also apply this idea to the ResNeXt block. The combination is shown in Figure 3 (c) in which the ResNeXt block (red box) is integrated into the SE block. The SE block carries out two operations: squeeze and excitation. The squeeze operation is aggregating feature maps across their spatial dimensions. The excitation operation is to capture channel-wise dependencies which improve channel interdependencies of features but change at least the computational cost. The squeeze operation is performed by the global pooling layer, and the excitation operation is executed by fully connected layers following by their activation layers as shown in Figure 3 (c). The first fully connected layer and its ReLU activation layer are to reduce the channel dimension so that they limit model complexity and enhance generalization. The second fully connected layer and its sigmoid activation layer are to restore the dimensionality.

3.4 Detecting Skewed, Upward and Downward Curved Text-lines

The Non-Maximum Suppression (NMS) algorithm is used to select one bbox of the highest confident score of text including classification when having many predicted bboxes overlapped each other at one position. The algorithm removes other bboxes that are overlapping the selected bbox more than a threshold. In our network, we set the threshold of 0.25. That, however, prunes bbox candidates of skewed, upward/downward curved text-lines wrongly. Especially, when these text-lines are close to each other vertically. Therefore, we apply a training strategy to make the network produce angles of text-lines. Then, the angles are utilized to revise the bboxes of text-lines before applying the NMS algorithm.

To do that, we take advantage that the provided training dataset often includes straight text-lines. We apply the rotation augmentation method to produce skewed text-lines as shown in the input part of Figure 1. The limitation of rotated angles is in a range [-10^o, 5^o] from the fact that text-lines are not rotating so much practically. The rotated angles are used to train the text detection network regressively. The regression learning allows the network to predict angels of text-lines besides bbox coordinates and confident scores of text including classification.

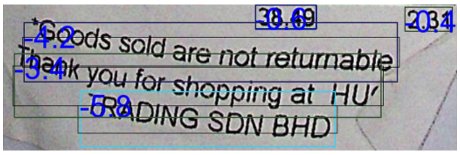


Figure 4: Revision of predicted bboxes. Black line bboxes are revised with predicted angles. Other color line bboxes are originally predicted bboxes by our network without revision. Blue numbers are predicted angles.

In Figure 4, black line bboxes are revised with their predicted angles while other color line bboxes are predicted directly by our text-detection network. The directly predicted bboxes of upward/downward curved text-lines are easy to be removed by the NMS algorithm because they are overlapping others. We need to revise the bboxes of these text-lines before applying the NMS algorithm. In our proposed network, only predicted bboxes of angles that are larger than one degree, are revised their angles. The directly predicted bboxes, corresponding to revised bboxes that are remaining after applying the MNS algorithm are kept as results.

We assume (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right corner coordinates of a predicted bbox. α and $(x_0 = (x_1 + x_2)/2, y_0 = (y_1 + y_2)/2)$ are a predicted angle (blue numbers in Figure 4) and the center point of the bbox respectively. We carry out rotating the bbox with $-\alpha^o$ to get top-left (x_1', y_1') and bottom-right corner coordinates (x_2', y_2') of the revised bbox as Eq. (3).

$$\begin{aligned} x_1' &= x_0 + (x_1 - x_0) * \cos(-\alpha) + (y_1 - y_0) \\ &\quad * \sin(-\alpha) \\ y_1' &= y_0 - (x_1 - x_0) * \sin(-\alpha) + (y_1 - y_0) \\ &\quad * \cos(-\alpha) \\ x_2' &= x_0 + (x_2 - x_0) * \cos(-\alpha) + (y_2 - y_0) \\ &\quad * \sin(-\alpha) \\ y_2' &= y_0 - (x_2 - x_0) * \sin(-\alpha) + (y_2 - y_0) \\ &\quad * \cos(-\alpha) \end{aligned} \tag{3}$$

4 EXPERIMENTS

We do three experiments on the receipt dataset in the ICDAR 2019 competition on scanned receipt OCR and information extraction (Huang, 2019). The dataset includes a training set of 626 English receipt images and a testing set of 361 English receipt images. The images come along with rectangular bbox coordinates of text-lines and ground-truth text. We just use ground-truth text to set text including labels for training.

We also compare our best model with the top rate methods submitted to the ICDAR 2019 competition. There is a total of 29 submissions to the ICDAR 2019 competition for detecting text regions of the above dataset. The top-4 methods are summarized as follows:

1) The first-ranking method named "SCUT-DLVC-Lab-Refinement" (ICDAR-MT1): The method uses a refinement-based Mask-RCNN model for text detection.
2) The second-ranking method named "Ping An Property & Casualty Insurance Company" (ICDAR-MT2): This method applies an anchor-free detection framework with FishNet as the backbone.
3) The third-ranking method named "H&H Lab" (ICDAR-MT3): This method is based on EAST (Zhou, 2017). They add a multi-oriented corner network to EAST to make network learning easier.
4) The fourth-ranking method named "GREAT-OCR Shanghai University" (ICDAR-MT4): This method uses a novel text detector called Progressive Scale Expansion Network (PSENet).

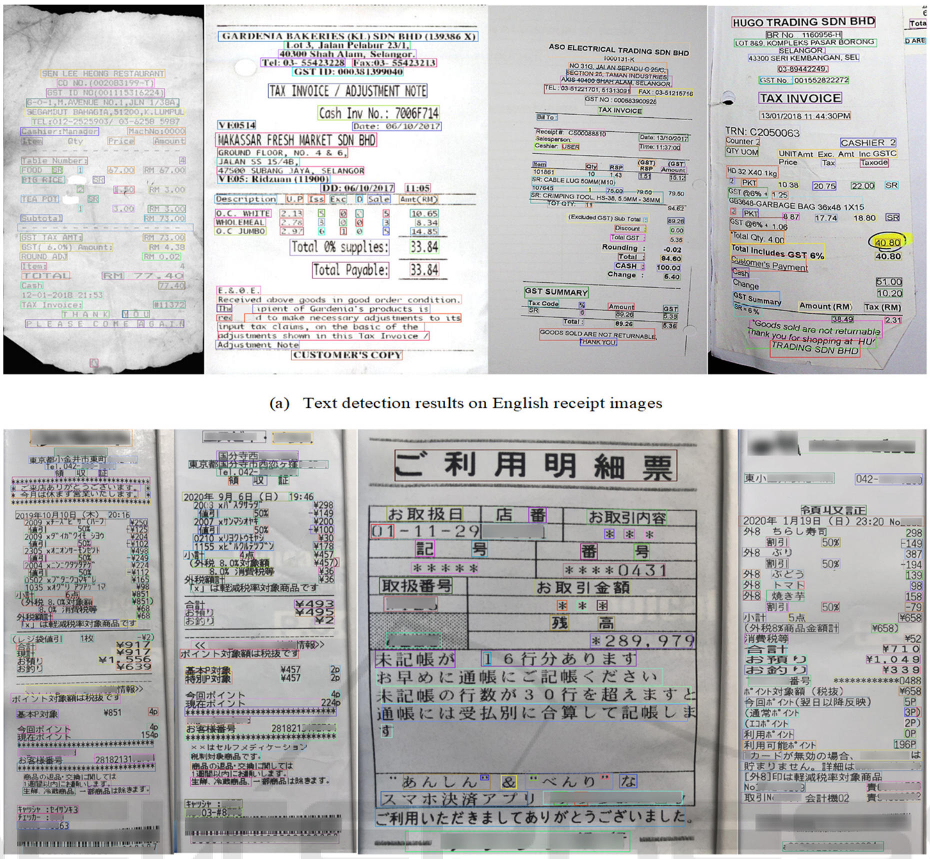
Our models are trained in 100 epochs and batch sizes of 8 images. We use the precision, recall, and harmonic mean (F1 score) as the ICDAR competition to evaluate the performance of our models. Predicted bboxes overlapped ground-truth bboxes more than 0.5 are evaluated as correct bboxes.

4.1 Evaluating Methods to Determine Anchor Boxes

The first experiment is to evaluate methods to determine anchor boxes for training the network. As shown in the first part of Table 1, clustering training bboxes, based on both their width and height is not good enough for determining anchor boxes. On the other hand, clustering by *IoU* between assumed anchor boxes and training bboxes proposes more anchor boxes at the densely distributed regions of bbox sizes, so it estimates the more fitting set of anchor boxes. When the number of anchor box is small, the efficiency of two methods is not evident, but it is clear when using more number for anchor boxes. The larger number of anchor boxes we use, the better performance we get. However, it seems that the efficiency increases slowly when using a large number of bboxes. Because of the limitation of GPU, we just train with a maximum of 96 anchor boxes.

Table 1: Contribution of different improvements to the performance of the text-detection network.

Experiments	**Models**	**Recall (%)**	**Precision (%)**	**F1 score (%)**
Anchor box determination methods and number of anchor boxes (Section 4.1)	ResNet 101 backbone + 16 anchors by method 1 + without bbox revision	97.61	97.45	97.53
	ResNet 101 backbone + 16 anchors by method 2 + without bbox revision	97.58	97.72	97.65
	ResNet 101 backbone + 36 anchors by method 1 + without bbox revision	97.71	97.57	97.64
	ResNet 101 backbone + 36 anchors by method 2 + without bbox revision	97.89	97.83	97.86
	ResNet 101 backbone + 96 anchors by method 2 + without bbox revision	**98.15**	**97.91**	**98.03**
Backbone networks for feature extraction (Section 4.2)	ResNet 152 backbone + 96 anchors by method 2 + without bbox revision	98.19	98.29	98.24
	ResNeXt 101 backbone + 96 anchors by method 2 + without bbox revision	98.17	98.25	98.21
	SE ResNeXt 101 backbone + 96 anchors by method 2 + without bbox revision	**98.83**	**98.49**	**98.66**
Bbox revision by predicted angles and comparison with other submitted methods (Section 4.3)	SE ResNeXt 101 backbone + 96 anchors by method 2 + bbox revision (our best model)	**98.79**	**98.95**	**98.87**
	ICDAR-MT1	98.64	98.53	98.59
	ICDAR-MT2	98.60	98.40	98.50
	ICDAR-MT3	97.93	97.95	97.94
	ICDAR-MT4	96.62	96.21	96.42



(a) Text detection results on English receipt images

(b) Text detection results on Japanese receipt images

Figure 5: Some text detection results. We use text bounding boxes of different colors to display them better.

4.2 Asserting Efficiency of Backbone Networks

The second experiment is to assert the efficiency of different backbone networks for feature extraction. As shown in the second part of Table 1, combining SE blocks and ResNeXt blocks in SE-ResNeXt backbone of 101 layers (~ 64 million trainable parameters) for extract features improves the performance of the text detection model significantly in comparison with using ResNet of 152 layers (~75 million trainable parameters) and ResNeXt of 101 layers (~ 60 million trainable parameters, the same as using ResNet of 101 layers). The ResNeXt backbone of 101 layers also shows improvement for the performance of the model, but it still cannot outperform the original ResNet network of 152 layers.

4.3 Comparison of with/without Bbox Revision and Other Methods

The last experiment is a comparison of bbox revision and without bbox revision. As shown in the second and third part of Table 1, the revision of bboxes with their predicted angles before applying the MNS algorithm also shows the contribution to the performance of the text detection model. This model shows a better result than the best model without bbox revision in Section 4.2. It is effective for detecting skewed, curved downward/upward text-lines.

By integrating all improvements in this model, it outperforms other models submitted to the ICDAR competition. Our best model achieves the text detection rate of 98.87% (F1 score), more than 0.28% in comparison with the best method submitted to the ICDAR 2019 competition.

4.4 Discussion of Detection Results

Figure 5 shows some text-detection results. Our model can work well with receipt images of complicated backgrounds and can detect skewed, upward/downward curved text-lines. We also test our best model with some of our taken Japanese receipts. The model, just trained with the English receipt dataset in the ICDAR 2019 competition, is also able to detect text-lines on a new domain of receipts. We blur characters in some bboxes because of sensitive information. We think fine-tuning with a small dataset may help the model work better on new domains of different languages, backgrounds, and styles.

5 CONCLUSIONS

In this paper, we propose newly constructing and training strategies for a text-detection model based on the Faster R-CNN architecture. We focus on three important factors that influence the accuracy of text-detection models. Firstly, we propose an anchor box determining method by clustering the *IoU* of assumed anchor boxes and bboxes. Secondly, we implement Squeeze-and-Excitation blocks (SE blocks) and ResNeXt blocks to create a very deep feature extraction network so that the model using this network outperforms the model using the ResNet 152 network, which has more trainable parameters. Finally, we train the text detection network with artificially skewed text-lines, then they can predict angles of skewed and upward/downward curved text-lines. We use the predicted angles to revise bboxes before applying the Non-Maximum Suppression algorithm, so that the model can detect skewed and upward/downward curved text-lines.

The model achieves a high accuracy of text-line detection, so we can integrate it with our text-line recognition model to create an automatically text-image recognizing system for receipt, invoice, and form images. Our approach is also flexible to apply for other datasets of complex backgrounds, different styles, and languages.

REFERENCES

Jeonghun, Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. “What is wrong with scene text recognition model comparisons? dataset and model analysis.” *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 4715-4723. 2019.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks.” *In Advances in neural information processing systems*, pp. 91-99. 2015.

Bodla, Navaneeth, Bharat Singh, Rama Chellappa, and Larry S. Davis. “Soft-NMS--improving object detection with one line of code.” *In Proceedings of the IEEE international conference on computer vision*, pp. 5561-5569. 2017.

Dinh, Viet Cuong, Seong Soo Chun, Seungwook Cha, Hanjin Ryu, and Sanghoon Sull. “An efficient method for text detection in video based on stroke width similarity.” *In Asian conference on computer vision*, pp. 200-209. Springer, Berlin, Heidelberg, 2007.

Epshtein, Boris, Eyal Ofek, and Yonatan Wexler. “Detecting text in natural scenes with stroke width transform.” *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2963-2970. IEEE, 2010.

Huang, Weilin, Yu Qiao, and Xiaoou Tang. “Robust scene text detection with convolution neural network induced mser trees.” *In European conference on computer vision*, pp. 497-511. Springer, Cham, 2014.

Zhou Xinyu, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. “East: an efficient and accurate scene text detector.” *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5551-5560. 2017.

Zhong, Zhuoyao, Lei Sun, and Qiang Huo. “An anchor-free region proposal network for Faster R-CNN-based text detection approaches.” *International Journal on Document Analysis and Recognition (IJDAR)* 22, no. 3 (2019): 315-327.

Wenhao, He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. “Deep direct regression for multi-oriented scene text detection.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 745-753. 2017.

Huang, Zheng, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. “ICDAR 2019 competition on scanned receipt OCR and information extraction.” *In 2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1516-1520. IEEE, 2019.

Kaiming, He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. “Aggregated residual transformations for deep neural networks.” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492-1500. 2017.

Hu, Jie, Li Shen, and Gang Sun. “Squeeze-and-excitation networks.” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132-7141. 2018.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. “Inception-v4, inception-resnet and the impact of residual connections on learning.” *arXiv preprint arXiv:1602.07261* (2016).

Simonyan, Karen, and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” *arXiv preprint arXiv:1409.1556* (2014).