Handwriting and Hand-Sketched Graphics Detection Using Convolutional Neural Networks

Song-Yang Cheng[1], Yu-Jie Xiong[1(✉)], Jun-Qing Zhang[1], and Yan-Chun Cao[2]

[1] School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, People's Republic of China
xiong@sues.edu.cn

[2] Faculty of Economics and Management, School of Public Administration, East China Normal University, Shanghai 200062, People's Republic of China

Abstract. Handwriting and hand-sketched graphics carry rich information to reveal the insights of the physical and emotional state of the writer. Before analyzing the personal traits of handwriting and hand-sketched graphics, detecting them from the image is the most immediate subproblem in handwritten document analysis and understanding. In this paper, we introduce two Convolutional Neural Networks (CNN) based methods to extract multimodal information (handwriting and hand-sketched graphics) from questionnaire documents. A Connectionist Text Proposal Network (CTPN) based method is proposed to detect handwriting. The first stage employs the VGG-16 model to generate the convolutional feature maps of the original images. Then the second stage adopts a BLSTM based detector to predict the scores of candidate zones. An instance segmentation method using the Mask Region Convolutional Neural Network (Mask-RCNN) is also proposed to solve hand-sketched graphics detection problem. The Mask-RCNN based approach has two parts: the backbone and the head. The backbone is to extract the features over the original image, and the head is to perform bounding boxes regression and mask prediction. At first, a simple Region Proposal Network (RPN) is adopted to generate the proposals of hand-sketched graphics efficiently. Then, the Region of Interest (RoI) features of the above proposals are fed into the Fast-RCNN branch and the mask branch to obtain the bounding boxes and the graphics segmentation results. The best handwriting detection performance of 200 test questionnaire images is that the precision rate is 99.5%, the recall rate is 99.2% and the F-measure score is 99.4%. The best detection performance of hand-sketched graphics is that the precision rate is 99.0%, the recall rate is 99.5% and the F-measure score is 99.3%. Experiments demonstrate that the proposed methods achieve promising results in both handwriting and hand-sketched graphics detection tasks.

Keywords: Handwriting detection · Hand-sketched graphics detection · Convolutional neural networks

Y. Lu et al. (Eds.): ICPRAI 2020, LNCS 12068, pp. 352–362, 2020.
https://doi.org/10.1007/978-3-030-59830-3_31

1 Introduction

Modern science has confirmed that handwriting and hand-sketched graphics play significant roles on human communication, perception, emotional behavior and so on. With the wide and rapid development of pattern recognition and artificial intelligence, it is reasonable to discover the automatic analysis of individual characteristics based on his/her handwriting and hand-sketched graphics. In order to achieve this ambitious goal, extracting structured information from handwritten documents is the first and fundamental process.

In daily life and business, handwritten documents are everywhere. Examples of handwritten documents are purchase receipts, questionnaires and so on. Handwritten documents contain information in the forms of both text (handwriting) and vision (hand-sketched graphics). Handwriting usually has a strong correlation between the horizontal and vertical texture properties around a certain patch. The position of the handwriting can be represented by the four coordinates of the handwriting bounding box, however, the hand-sketched graphics focus on the visual features of the global structural information. The zone of the hand-sketched graphics can be marked as the segmented region of the graphics. Thus, multimodal information extraction from handwritten documents via a unified framework is not easy, besides the various printed texts and lines, the main difficulty comes from the diversity of the textural and visual cues between handwriting and hand-sketched graphics.

Convolutional Neural Networks (CNN) have demonstrated excellent capabilities in the fields of machine learning, and are widely used in many vision-based applications [1]. With millions of cameras acting as sensors around the world, there are lots of significant opportunities for images analysis of these photos to provide actionable insights. These insights will benefit a wide variety of fields, from public safety to commercial activity. In this paper, we introduce two methods based on CNN to extract handwriting and hand-sketched graphics from the questionnaire documents. An end-to-end method based on CTPN [2] is proposed and used for spotting the handwriting. The first stage employs the VGG-16 model [3] as its backbone network to generate the convolutional feature maps of the initial samples. Then the second stage adopts a Bi-directional LSTM-based encoder, and connects it with the Fully Connected layer (FC) to predict the scores and locations of handwriting zones. An instance segmentation based approach using Mask-RCNN [4] is also proposed to deal with the problem of hand-sketched graphics detection. The approach is divided into two parts: the backbone and the head. The role of the backbone part aims to extract the features over the original images through the convolution layers, and the head part is to achieve three tasks: classification, bounding box regression and mask prediction. Firstly, a simple CNN called as Region Proposal Network (RPN) is assigned to generate the region proposals of hand-sketched graphics efficiently. Secondly, the Region of Interest (RoI) features of the above region proposals are fed into the Fast-RCNN [5] branch and the mask branch to obtain the bounding boxes and graphic instances segmentation results. Experiments demonstrate that the proposed two methods achieve promising results in both handwriting and

hand-sketched graphics detection tasks. The remainder of this paper is organized as follows: Sect. 2 gives a brief review of recent works on text and object detection; Sect. 3 describes details of the proposed methods; Sect. 4 provides experimental results, and Sect. 5 concludes this paper.

2 Related Work

Our work is inspired by recent researches in the field of text detection and object detection. Text and object detection have attracted great attention in document analysis and computer vision in recent years. There are two commonly used deep learning methodologies in text and object detection systems: one-stage detectors and two-stage detectors. Generally speaking, the dominant methods in text/object detection are based on a two-stage approach, and one-stage detectors are tuned for realtime applications. While, recent researches reported that two-stage detectors can be made fast simply by reducing input image resolution and the number of proposals [6]. Detection tasks are still very challenging due to the complexity of environments, flexible imagine acquisition styles and variations of text/object contents [7].

2.1 Text Detection

The earlier work on scene text detection usually focused on handcraft features, like Maximally Stable Extremal Regions (MSER) [8], Extremal Regions (ER) [9] and Stroke Width Transform (SWT) [10,11]. These methods utilized the texture and shape of text characters, and they were workable when the images were clean [12]. Deep learning is able to extract different discriminative features for improving the performance of text detection from images with complex backgrounds. For example, Single Shot Multiboxes Detector (SSD) [13] can be used to detect multi-scale texts in scenes. Faster-RCNN achieved good results on horizontal text detection with the help of the LocNet which was based on localization module [14]. Connectionist Text Proposal Network (CTPN) used the LSTM module to realize accurate text location [2]. B. Shi et al. combined the core ideas of small candidate proposals with SSD to deal with multi-oriented problem [15]. Mask TextSpotter [16] can recognize the instance sequence inside character maps rather than only predict an object region.

2.2 Object Detection and Instance Segmentation

Object detection and instance segmentation made great progress in the past decade. Faster-RCNN [17] classified object proposals and predicted their spatial locations jointly. YOLO [18] was simple to construct and can be trained directly on full images. It pushed the state-of-the-art into real-time object detection. FPN [19] presented a clean and simple framework for building feature pyramids inside networks, and showed good performance on object detection. J. Dai et al. exploited image local coherence to provide instance-level segment candidates [20].

3 The Proposed Methods

3.1 CTPN Based Handwriting Detection

The overall framework of the proposed method is shown in Fig. 1. CTPN receives the images of different sizes, and predicts the positions of handwriting by moving the sliding-windows over the extracted feature maps. For a document image, CNN features are firstly extracted using VGG16 pre-trained model. Secondly, a sliding-window (3×3) moves densely over the extracted feature to create sequential features as the input for B-LSTM. At last, the output of B-LSTM is connected to the following fully connected layer that calculates text/non-text score and coordinates information of each proposal.

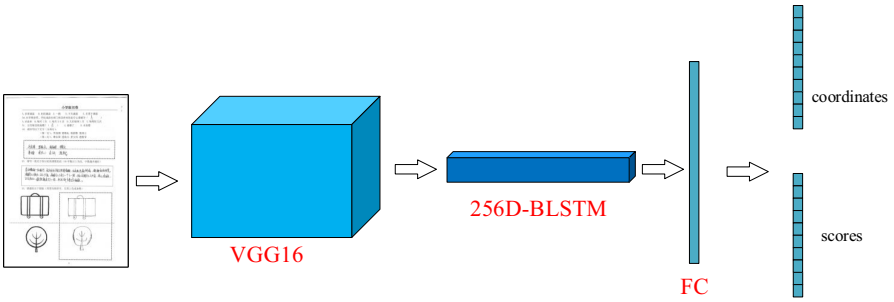

Fig. 1. The framework of CTPN based method for handwriting detection

Convolutional Feature Extraction. VGG-16 model is utilized as a feature extractor in our method. In order to share convolutional computation, a small 3×3 spatial window is slide over the output of VGG-16 model. The total stride and receptive field of the obtained feature map are fixed as 16 and 228 pixels by the network architecture.

Fine-Scale Proposals. Compared with general object, handwriting has strong correlation between the horizontal pixel sequences. Thus, the fine-scale text proposal is defined as a sequence of text pieces with the fixed (16-pixel) width. Each proposal contains a small patch (such as single strokes, a part of a character, or a single character) of a certain horizontal text lines. By fixing the horizontal coordinates, the prediction of vertical location will be easier and more effective. The specific implementation is as follows. Due to the structure of VVG-16 model, the receptive field of the obtained feature map is 16×16. Thus, the minimum horizontal interval is 16 pixels in the original image. Ten vertical anchors are created to predict coordinates of y-axis in each proposal, and the height of these anchors in the original image ranges from 11 to 273 pixels with equal ratio.

Then we get the coordinates measured by the height and y-axis center of a proposal. The relative predicted vertical coordinates are calculated:

$$v_c = (c_y - c_y^a)/h^a \quad v_h = log(h/h^a) \tag{1}$$

$$v_c^* = (c_y^* - c_y^a)/h^a \quad v_h^* = log(h^*/h^a) \tag{2}$$

where v_c, v_h and v_c^*, v_h^* are the predicted coordinates and ground truth coordinates. c_y^a and h^a are the y-axis center and height of each anchor box, it can be calculated from the original input image. c_y and h are the predicted value of y-axis, c_y^* and h^* are the ground truth value. In this way, the bounding box with size of $h \times 16$ in each proposal from original images is obtained.

Recurrent Connectionist Text Proposals. Text lines are split into a sequence in order to get more accurate location of each proposal. It has been verified that Recurrent Neural Network (RNN) can be used to encode context information for text recognition [21]. RNN gives us the chance to make accurate detection for every proposal by exploring the context information. A BLSTM which allows to encode the recurrent context in both directions is used to extend the RNN layer, so that the connectionist receipt field is able to cover the whole image width. Each LSTM has 128 nodes, resulting in a 256D RNN hidden layer in our network. The internal state of BLSTM is then mapped to the following FC to compute the predictions of each proposal.

As for the optimization of the model, the loss function L is the sum of the classification loss L_s^{cl} and the regression loss L_v^{re}:

$$L = \frac{1}{N_s}\sum_i L_s^{cl}(s_i, s_i^*) + \frac{1}{N_v}\sum_j L_v^{re}(v_j, v_j^*) \tag{3}$$

where i is the index of anchors in one batch, j is the index in a set of interested anchors for y-axis coordinates. An interested anchor represents the anchor whose Intersection-over-Union (IoU) with the ground truth text proposal is larger than 0.7. s_i is the confidence of judging an anchor as a handwriting, and s_i^* is the ground truth. v_j are the prediction y-axis coordinates while v_j^* are the ground truth coordinates.

Proposal Connection. In this part, the fine-scale proposals are connected into an integral proposal which contains all information of an area that we are interested. In this way, we can get proper results which are meaningful for human beings. As done in Ref. [2], text line construction is straightforward by connecting continuous text proposals whose text/non-text score is >0.7. At First, a paired neighbour (B_j) for a proposal is defined as B_i as $B_j \Rightarrow B_i$, when (i) B_j is the nearest horizontal distance to B_i, and (ii) this distance is less than 50 pixels, and (iii) their vertical overlap is larger than 0.7. Secondly, two proposals are grouped into a pair, if $B_j \Rightarrow B_i$ and $B_i \Rightarrow B_j$. Then a text line is constructed by sequentially connecting the pairs having a same proposal.

3.2 Mask-RCNN Based Hand-Sketched Graphics Detection

The Mask-RCNN [4] based framework for hand-sketched graphics detection is divided into four parts: Feature Pyramid Network (FPN), Region Proposal Network (RPN), Fast-Region Convolutional Neural Network (Fast-RCNN) and Fully convolutional network (FCN). The overall architecture of the proposed method is presented in Fig. 2. For a document image, CNN features are firstly extracted using FPN. During this period, the top-level features are merged with the bottom-level features by up-sampling, and each layer provides feature maps independently. Secondly, the RPN generates a lot of graphics proposals. Then, the features of the proposals are fed into the RCNN and the mask branch by applying RoIAlign [4] on the output of FPN. The RCNN branch performs classification and regression to produce horizontal bounding boxes, meanwhile, the Mask branch predicts the global graphics instance segmentation results.

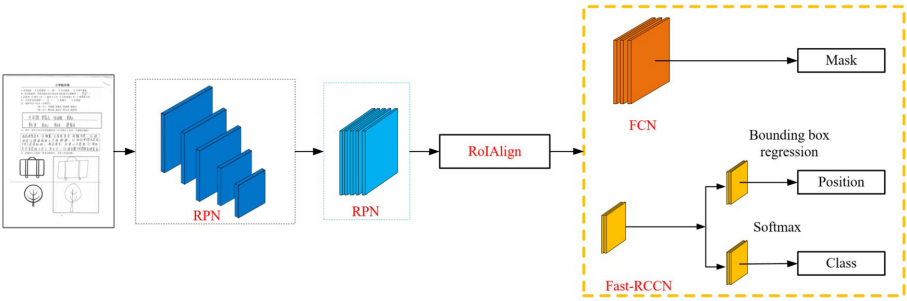

Fig. 2. The framework of Mask-RCNN based method for hand-sketched graphics detection

FPN. To demonstrate the generality of the original Mask-RCNN, multiple architectures of network such as ResNet-50, ResNet-101, ResNeXt-50 and ResNeXt-101 are used as the backbone [4]. It is well known that FPN takes the advantages of multi-scale feature spaces for accurate localization, thus, we utilize it to enhance the backbone of our approach. ResNet-101 consists of five stages, and each stage corresponds to one scales of feature map $[C_1, C_2, C_3, C_4, C_5]$. These five feature maps are used to establish the feature pyramid of FPN, and get new features respectively $[P_1, P_2, P_3, P_4, P_5]$:

$$\begin{cases} P_1 = Conv(Sum(Upsample(P_2, Conv(C_2)))) \\ P_2 = Conv(Sum(Upsample(P_3, Conv(C_3)))) \\ P_3 = Conv(Sum(Upsample(P_4, Conv(C_4)))) \\ P_4 = Conv(C_5) \\ P_5 = Downsample(P_4) \end{cases}$$

where $Conv$ represents the convolution, Sum represents the element-by-element alignment operation, $Upsample$ and $Downsample$ represent upsampling and downsampling respectively.

RPN. RPN is used to scan the above feature maps to estimate the Region of Interest (RoI) where the hand-sketched graphics may exist. The size of RoI on each stage depends on the scale of the layer of the feature pyramid. According to the Ref. [5], the RoI sizes are set to $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ pixels on five stages respectively, and two aspect ratios $\{0.5, 1, 2\}$ are also adopted in each stage. The output of RPN is a series of bounding boxes with their anchors. If bounding boxes are overlapped, the non-maximum suppression (NMS) is applied to obtain the refined bounding box. After that, RoIAlign [4] is used to pool the RoIs into fixed-size feature maps. Compared to RoI Pooling, RoIAlign aligns the extracted features with the original region proposal properly, and is beneficial to the segmentation task in the mask branch. For the training of RPN, two loss functions to indicate the difference between the generated RoIs and the ground truth are utilized: L_{class}^{RPN} used to classify the generated RoIs and L_{bbx}^{RPN} used to modify the coordinates of the anchors. L_{class}^{RPN} is computed by a softmax over the outputs of a fully connected layer:

$$L_{class}^{RPN} = -\frac{1}{N_{class1}} \sum_{i}^{N_{class1}} log[p_i^* p_i + (1 - p_i^*)(1 - p_i)], \tag{4}$$

L_{bbx}^{RPN} is defined over the output of the i-th anchor $(x_i^o, y_i^o, h_i^o, w_i^o)$ and the corresponding ground truth of bounding box $(x_i^g, y_i^g, h_i^g, w_i^g)$:

$$L_{bbx}^{RPN} = \frac{1}{N_{reg1}} \sum_{i,j \in \{x,y,w,h\}}^{N_{reg1}} \text{smooth}_{L_1}(j_i^o - j_i^g), \tag{5}$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

where N_{class1} indicates the number of RoIs; If the RoIs is positive sample, $p^* = 1$; otherwise, $p^* = 0$. p_i is the probability that the i-th RoI is predicted to be positive sample, and N_{reg1} is the number of corresponding anchors (background RoIs are ignored for training).

Fast-RCNN. In this part, two tasks are accomplished: box regression and classification. Thus, this part is called as box regression and classification branch. The fixed-size feature maps of RoIs are fed into fully connected layers, then the output vectors are used for the classification and bounding boxes of detected targets. Similar to the above RPN, two loss functions (L_{class}^{RCNN} and L_{bbx}^{RCNN}) are obtained by:

$$L_{class}^{RCNN} = -\frac{1}{N_{class2}} \sum_{i}^{N_{class2}} log[p_i^* p_i + (1 - p_i^*)(1 - p_i)], \tag{6}$$

$$L_{bbx}^{RCNN} = \frac{1}{N_{reg2}} \sum_{i,j \in \{x,y,w,h\}}^{N_{reg2}} \text{smooth}_{L_1}(j_i^o - j_i^g), \tag{7}$$

where N_{class2} indicates the number of detected targets, p_i is the probability that the i-th target is predicted to be positive sample, and N_{reg2} is the number of corresponding bounding boxes.

FCN. The difference between Mask-RCNN and Fast-RCNN is that the former not only predicts the class and bounding box, but also produces a binary mask for each detected target. As a result, this part is also called as mask branch. A mask should represent the whole spatial information of a target. Thus, FCN is the best choice that allows each layer in the mask branch to keep the spatial layout without compressing it into a feature vector that lose the spatial dimensions. The mask loss function L_{mask} is defined as:

$$L_{mask} = -\frac{1}{N}\sum_{n=1}^{N}\left[log(S(o_n))g_n + log(1 - S(o_n))(1 - g_n)\right], \tag{8}$$

where N is the area of the mask, g_n is the pixel label ($g_n \in 0, 1$), o_n is the output pixel, and $S(x) = \frac{1}{1+e^{-x}}$.

4 Experiments

To validate the effectiveness of the proposed methods, we conduct experiments on our own SUES-1000 database. **SUES-1000 database** is our own database of text and object detection. In this database, totally 1000 questionnaire images of the primary and secondary school students were collected as the handwritten samples (as shown in Fig. 3). Each student filled out one questionnaire independently and anonymously. The questionnaire contains two zones for writing

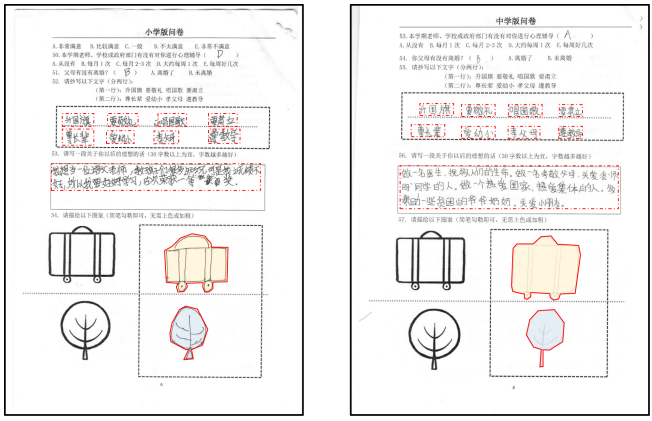

Fig. 3. Two questionnaire images of SUES-1000 database. The left one is written by a elementary school student, and the right one is written by a middle school student. The rectangles (red dashed lines) are the ground truth for the handwriting, and the polygons (red lines) are that of hand-sketched graphics. (Color figure online)

(Z_1, Z_2), and two areas for drawing (A_1, A_2). The handwriting in Z_1 is a copy of eight specified words (24 Chinese characters), and the handwriting in Z_2 is a free description about the future life (10–80 Chinese characters). The hand-sketched graphics in A_1 and A_2 are two tracings of specified illustrations (a suitcase and a tree).

During the experiment, the precision rate (P), recall (R) rate, and F-Measure are applied to evaluate the detection performance:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F = \frac{2 * P * R}{P + R},$$

where TP is the number of cases that are positive and detected positive, FP is the number of cases that are negative but detected positive, and FN is the number of cases that are positive but detected negative.

The experiments are performed under the framework of TensorFlow(1.14.0), with NVIDIA2080 for GPU acceleration, Inter (R) Core (TM) i7-9700k CPU and 16G memory. During the experiments, 800 document images were selected for training the models, and 200 document images are used to verify the stability and reliability of the trained models.

Table 1 shows the handwriting detection results of the proposed methods on SUES-1000 database. When the threshold of Intersection over Union (IOU) is set to 0.5 or 0.7, there is not much difference between the results of two methods. When the threshold of IOU is 0.9, CTPN based method has quite better performance than that of Mask-RCNN based method. lt demonstrates that CTPN keeps more local structural information of handwriting than Mask-RCNN.

Table 2 shows the results of the proposed methods for hand-sketched graphics detection. Like handwriting detection task, when the threshold of Intersection over Union (IOU) is small (<0.7), both methods work very well (>99%). With the increasing of threshold, the superiority of Mask-RCNN raises quickly. When IoU = 0.9, the overall precision and recall rates of 200 testing samples are 60.5% and 60.8%, respectively. However, the performance of CTPN based method is only 45.7% and 45.6%. lt means that Mask-RCNN is more suitable for the task of graphics detection.

Table 1. The handwriting detection results on SUES-1000.

	Precision	Recall	F-Measure
CTPN based@0.5	99.5%	99.2%	99.4%
Mask-RCNN baed@0.5	98.1%	99.0%	98.6%
CTPN based@0.7	95.0%	94.7%	94.9%
Mask-RCNN baed@0.7	97.1%	95.4%	95.0%
CTPN based@0.9	48.4%	48.3%	48.4%
Mask-RCNN baed@0.9	33.6%	33.9%	33.8%

CTPN based@0.5 means that CTPN based method is tested with the threshold of Intersection over Union (IoU) as 0.5.

Table 2. The hand-sketched graphics detection results on SUES-1000.

	Precision	Recall	F-Measure
CTPN based@0.5	99.5%	99.2%	99.3%
Mask-RCNN baed@0.5	99.0%	99.5%	99.2%
CTPN based@0.7	90.8%	90.6%	90.7%
Mask-RCNN baed@0.7	98.5%	99.0%	98.7%
CTPN based@0.9	45.7%	45.6%	45.6%
Mask-RCNN baed@0.9	60.5%	60.8%	60.6%

CTPN based@0.5 means that CTPN based method is tested with the threshold of Intersection over Union (IoU) as 0.5.

5 Conclusions

In this paper, we introduce two CNN based methods to extract handwriting and hand-sketched graphics from questionnaires. CTPN based method used fine-scaled proposals and vertical anchor mechanism to accurate and effective text detection. With the help of instance segmentation, Mask-RCNN based method dominate the global graphics detection. The results achieved by our proposed methods on the SUES-1000 database validate their effectiveness. The former is more workable in detecting handwriting, and the latter has better performance in the detection of hand-sketched graphics. In the future, we would like to apply the proposed methods into further handwriting analysis, such as writer identification and signature verification.

Acknowledgment. This work is sponsored by Shanghai Sailing Program (Grant No. 19YF1418400).

References

1. Gatys, L., Ecker, A., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceeding of IEEE Conference on Computer Vision Pattern Recognition, pp. 2414–2423 (2016)
2. Zhi, T., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. arXiv e-prints. arXiv:1609.03605 (2016)
3. Simonyan, K. and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556 (2014)
4. He, K., Gkioxari, G., Dollar, P., Girshick, R.B.: Mask R-CNN. In: Proceeding of IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
5. Girshick, R.B.: Fast R-CNN. In: Proceeding of IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
6. Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceeding of IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
7. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1480–1500 (2015)

8. Neumann, L., Matas, J.: A method for text localization and recognition in realworld images. In: Proceeding of Asian Conference on Computer Vision, pp. 770–783 (2010)
9. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Proceeding of IEEE Conference on Computer Vision Pattern Recognition, pp. 3538–3545 (2012)
10. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proceeding of IEEE Conference on Computer and Vision Pattern Recognition, pp. 2963–2970 (2010)
11. Yao, C., Bai, X., Liu, W.: A unified framework for multioriented text detection and recognition. IEEE Trans. Image Process. **23**(11), 4737–4749 (2014)
12. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: an attentional scene text recognizer with flexible rectification. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2035–2048 (2019)
13. Liu, W., et al.: Single shot multibox detector. In: Proceeding of European Conference on Computer Vision, pp. 21—37 (2016)
14. Zhong, Z., Sun, L., Huo, Q.: Improved localization accuracy by LocNet for faster R-CNN based text detection in natural scene images. In: Proceeding of IEEE Conference Document and Analysis, in press
15. Shi, B., Bai, X., Belongie, S.: Detecting oriented text in natural images by linking segments. In: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2550–2558 (2017)
16. Liao, M., Lyu, P., He, M., Yao, C., Wu, W., Bai, X.: Mask TextSpotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, in press
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
19. Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceeding IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
20. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: Proceeding European Conference on Computer Vision, pp. 534–549 (2016)
21. He, P., Huang, W., Qiao, Y., Loy, C., Tang, X.: Reading scene text in deep convolutional sequences. In: Proceeding of AAAI Conference on Artificial Intelligence, pp. 3501–3508 (2016)