IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 20, 2021, accepted June 21, 2021, date of publication June 30, 2021, date of current version July 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093568

Detecting Text Baselines in Historical Documents With Baseline Primitives

WEI JIA[1], CHIXIANG MA[1], LEI SUN[2], AND QIANG HUO[2], (Member, IEEE)
[1]Department of Electrical Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China
[2]Microsoft Research Asia, Beijing 100080, China

Corresponding author: Wei Jia (jiawei10@mail.ustc.edu.cn)

This work did not involve human subjects or animals in its research.

ABSTRACT Previous deep learning based approaches to text baseline detection in historical documents usually take it as a semantic segmentation task. These methods adopt a fully convolutional neural network to predict baseline pixels first and then group them into lines by heuristic post-processing steps, which tends to suffer from a wrongly merged or wrongly split problem owing to limited context information provided by pixels. To address these issues, we introduce the concept of a baseline primitive, which is defined as a virtual bounding box centered at each baseline pixel. After baseline primitive detection, a relation network is used to predict a link relationship for each pair of primitives. Consequently, text baselines are generated by detecting baseline primitives and grouping them with the corresponding link relationships. Owing to the design of baseline primitives, wider context information can be leveraged to improve link prediction accuracy. Therefore, our approach can effectively detect text baselines with small inter-line or large inter-word spacing. Quantitative experimental results demonstrate the effectiveness of the proposed baseline primitive design. Our approach achieves state-of-the-art performance on two public benchmarks, namely cBAD 2017 and cBAD 2019.

INDEX TERMS Handwritten documents, historical documents, text baseline detection, baseline primitive.

I. INTRODUCTION

Historical documents are valuable cultural heritage that connects the past with the present. In recent years, numerous historical documents have been captured and published online with the help of many libraries and archives worldwide. Digitalizing their content has made it more convenient for scholars as well as ordinary people to access them. It is widely accepted that robust and accurate text baseline detection is a critical first step [1]–[4] to digitalize historical document images automatically because errors made during this process will affect subsequent steps. With the recent emergence of two text baseline detection competitions in the ICDAR community, namely cBAD 2017 [5] and cBAD 2019 [6], more researchers have been attracted to this research field. Unlike text baseline detection in printed documents with a simple layout, the same process in unconstrained historical documents is still an unsolved problem due to some unique challenges, such as various handwriting styles (e.g., long ascenders and descenders, heterogeneous and touching strokes), complex layout (e.g., arbitrary oriented or curved text lines, marginalia, heterogeneous inter-line spacing), physical degradations (e.g., bleed-through, faded away characters) and distortions introduced by image capturing.

Text baseline detection in historical documents has been studied for decades, and a large number of algorithms have been proposed in the literature to solve the problems it poses. A comprehensive survey of the literature can be found in references [7] and [8]. Previously, researchers mainly focused on detecting text baselines in clean handwritten documents [9], [10], where degradations and complicated background are less considered. During that time, many image processing methods were proposed. For instance, some methods performed text block projection (e.g., [11]–[13]) or smearing operations (e.g., [14], [15]) on document images to detect text baselines directly; others first extracted connected components or interest points and then grouped them into individual text lines by clustering (e.g., [16]–[18]), performing Hough transform (e.g., [19]–[21]), or minimizing an energy function (e.g., [22]–[28]). Although these methods have achieved competitive results, their performance deteriorates a lot when

The associate editor coordinating the review of this manuscript and approving it for publication was Madhu S. Nair.

handling documents with degradations. For instance, the winner methods [27] of both the ICDAR 2013 handwriting segmentation contest [10] and the ICDAR 2015 text line detection in historical documents contest [29] can achieve an F-measure of 98.75% on the former contest while only an F-measure of 71.68% on the later contest. This huge performance gap mainly comes from the newly introduced degradations [29], [30] and more complex layouts, e.g., faded-out ink, bleed-through, marginalia, etc. Recently, astonishing developments have been made in text baseline detection thanks to those challenging benchmarks [5], [6] as well as the rapid development of deep learning methods (e.g., [2], [5], [6], [31]–[36]). These methods are generally composed of two separate procedures, namely baseline pixel prediction and baseline generation. In the first step, a pixel-wise classification is performed for each location on the feature maps generated from a document image to predict whether this location corresponds to a baseline pixel, where a fully convolutional neural network (FCN) [37], [38] framework is usually adopted for its capability to generate powerful feature representations. In the second step, various algorithms are utilized to group those candidate baseline pixels into individual baseline. Specifically, some methods simply group candidate baseline pixels according to the local pixel connectivity (e.g., 8-neighborhood), while others use handcrafted features (e.g., estimated inter-line interval, local orientation) with heuristic rules or clustering algorithms to generate baselines.

Although recent deep learning-based methods have shown superior performance on the public benchmarks, they tend to suffer from a wrongly merged or wrongly split problem. Wrongly merged problem arises when adjacent text baselines with small inter-line spacing are mistakenly merged together (Fig. 1 (a)), while wrongly split problem refers to a single text baseline with large inter-word spacing that is split into two or more broken lines (Fig. 1(c)). These two problems are caused by the unsatisfactory performance of the baseline pixel prediction module when dealing with nearby text lines or those with large inter-word spacing. Specifically, background pixels within small inter-line spacing tend to be misclassified as baseline pixels (Fig. 1(b)), while pixels within large inter-word spacing tend to be misclassified as background ones (Fig. 1(d)). These errors are then propagated to the following baseline generation module, which aims to group candidate baseline pixels into their corresponding text baselines. However, existing baseline generation methods cannot handle these cases robustly. For instance, local pixel connectivity based methods have intrinsic limitations in solving these problems; clustering-based methods (e.g., [2]) can alleviate these problems to some extent but usually rely on handcrafted features and complicated post-processing steps that limit their capabilities.

In this paper, we introduce the concept of a baseline primitive and solve these problems accordingly. Specifically, instead of viewing a text baseline l just as a thin line, we assign an enlarged virtual box b to the baseline that

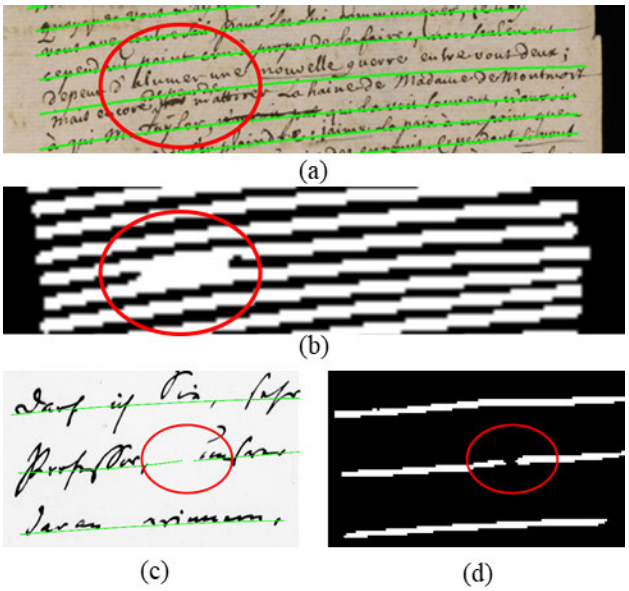

FIGURE 1. (a) Wrongly merged problems for dense touching text lines, where two adjacent text baselines are merged together; (b) The corresponding binarized classification score map of (a) for baseline pixel prediction; (c) Wrongly split problems for text baselines with large inter-word spacing; (d) The corresponding binarized classification score map of (c) for baseline pixel prediction.

centered at it, as illustrated in Fig. 2. Then, a virtual box bp, i.e., a baseline primitive, is generated according to each baseline pixel p and the enlarged virtual line box b. Back to the wrongly merged and wrongly split problems, it is clear that they can be solved by accurately determining whether each baseline pixel pair belongs to a same baseline. For convenience, we define this relationship as a link relationship. Baseline primitives have two advantages over pure baseline pixels as basic components. On the one hand, representative baseline pixels can be selected with a non-maximum suppression (NMS) algorithm performed on their corresponding baseline primitives since redundant baseline pixels have highly overlapped baseline primitives. This will help find confident candidate baseline pixels while filter less accurate ones, which reduces efforts for the following link prediction step. Besides, baseline primitives can capture wider context than pure baseline pixels, since context information can be encoded explicitly with the virtual box of each baseline primitive. Meanwhile, the union box of two baseline primitives can also be leveraged to determine the link relationship, which encodes relative position information.

With the baseline primitive design, many visual relationship learning methods can be used to learn the link relationship. In this paper, we leverage a relation network [39] to achieve this goal. Though conceptually simple, the relation network has shown its effectiveness in both visual relationship detection field (e.g., [39]–[43]) and scene text detection field [44]. In summary, a relation network takes a pair of baseline primitives as well as their union box as input and predicts the link relationship of this baseline primitive pair accordingly. When all the link relationships are obtained,

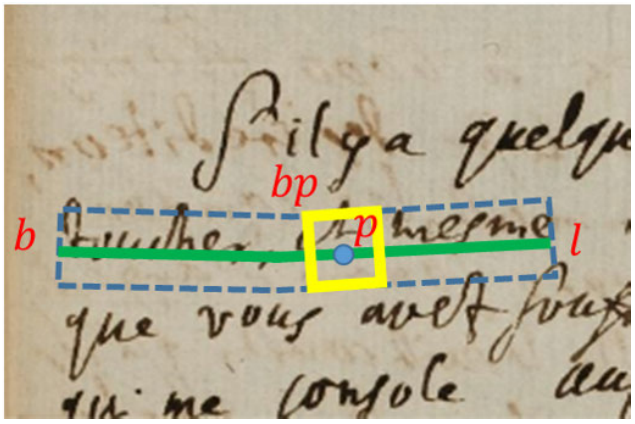

FIGURE 2. A baseline *l* (in green) and its corresponding enlarged virtual box *b* (in dashed blue); a baseline pixel *p* and its corresponding baseline primitive box *bp*. Best viewed in color.

text baseline primitives are grouped into individual primitive groups, from which text baselines are extracted accordingly.

In this paper, we make the following contributions:

(1) We analyze thoroughly the problems encountered by existing convolutional neural network (CNN) based text baseline detection approaches and introduce the concept of baseline primitives to leverage wider context information to address these problems.
(2) With the introduction of baseline primitives, we propose to use a relation network based framework to detect text baselines in historical documents, which identifies baseline primitives and learns a link relationship for each baseline primitive pair in a single neural network.
(3) Our approach can handle the wrongly merged and wrongly split problems effectively and achieves state-of-the-art performance on two challenging text baseline detection benchmarks, namely cBAD 2017 and cBAD 2019.

The rest of paper is organized as follows. Section II gives an overview of related work. Section III presents our method. Section IV reports experimental results. Section V concludes the paper with a discussion on future work.

II. RELATED WORK

A. TEXT BASELINE DETECTION IN HISTORICAL DOCUMENTS

Text baseline detection in historical documents has been studied for decades, and a large number of algorithms for undertaking the task have been proposed in the literature. These can be roughly classified into two categories: conventional image processing methods and deep learning methods.

1) CONVENTIONAL IMAGE PROCESSING METHODS

These methods can handle only a limited number of document layouts, and a comprehensive survey of them can be found in references [7] and [8]. We briefly introduce these methods and categorize them by the type of document layouts they can handle.

Algorithms in the first category focus on detecting text baselines in well-structured documents where the text lines are almost parallel to each other. They can be further categorized on the basis of the core techniques they adopt, i.e., projection-based methods (e.g., [11]–[13]), smearing-based methods (e.g., [14], [15]), filtering-based methods (e.g., [45]–[47]), and Hough transform based method (e.g., [19]–[21]). Projection-based methods compute projection profiles by summing pixel values along a given direction, and the peaks indicate individual text lines. To tackle the skew or moderate curved text lines, the input image can be divided into several vertical strips and profiles are computed at each strip [11]. Smearing-based methods like fuzzy RLSA [14] and adaptive RLSA [15] smear some consecutive black pixels along a given direction while the white space between them is filled with black pixels if the distance is within a predefined threshold. Filtering-based methods operate in a similar way as smearing-based ones. These methods use a Kalman filter [45], an adaptive local connectivity map [46], or a steerable direction filter [47] to reveal the local patterns of text lines, and the final text lines are generated by grouping foreground pixels accordingly. Hough transform based methods perform Hough transform on the centroids of the connected components or local minima to detect straight lines that fit these points best.

Algorithms in the second category try to handle more complex layouts, e.g., arbitrary oriented text lines, and they can be sub-categorized as clustering-based methods (e.g., [3], [16]–[18]) and function analysis methods (e.g., [22]–[27]). Clustering-based methods first extract basic elements (interest points, connected components, etc.) from document images, and then various clustering algorithms are adopted to group these elements into text lines accordingly. In references [16] and [17], minimum spanning tree (MST) is performed based on carefully designed distance measures. Gruuening *et al.* [18] first extracted super pixels with the FAST algorithm [48], and then applied a standard clustering method based on some text line characteristics, e.g., curvilinearity, inter-line spacing and local homogeneity. Pastor [3] first filtered noisy local minima points with Extremely Randomized Trees (ERT) and then performed a modified DBScan [49] algorithm. Function analysis methods try to segment text lines by finding an optimal path across the document image. Saabni *et al.* [22] computed the energy map of a document image and determined the seams that pass across and between text lines. Ryu *et al.* [23] estimated the states of connected components and built a cost function upon these states, which was minimized to yield text lines. Öztop *et al.* [24] proposed an energy minimizing dynamical system, which interacted with the document image through attractive and repulsive forces defined over baseline pixels. Yin and Liu [25] proposed a vibrational Bayes approach to segment the image after the number of text lines was estimated. Luthy *et al.* [26] utilized a hidden Markov model to model a text line as a sequence of word and space. Ahn *et al.* [27] leveraged the advantages of projection

methods, non-text filtering and energy-minimization to detect text baselines. Though algorithms in this category have achieved superior performance on some public benchmark datasets, their flexibility is usually constrained by their handcrafted features and manually tuned parameters. Thus, their performance worsens significantly when processing unconstrained historical documents.

2) DEEP LEARNING METHODS

With the emergence of two challenging competitions (cBAD 2017 and cBAD 2019) that focus on text baseline detection in unconstrained historical documents, many CNN-based methods have been proposed and shown to be superior over traditional methods in terms of both accuracy and capability. Generally speaking, these methods usually contain two steps. First, pixel-wise classification is conducted on the feature maps generated from input documents to detect baseline pixels. Second, the detected baseline pixels are grouped into individual text baselines based on the local pixel connectivity or some handcrafted features (e.g., inter-line distance, local orientation). BYU [5] used an FCN backbone to detect baseline pixels directly while heuristic rules guide the generation of baselines. TJNU [6] used an FCN backbone with dilated convolutions to predict baseline pixels directly and group them into text baselines with connectivity information. LITIS [31] adopted the same FCN backbone as TJNU, but it predicted text core regions (with extra x-height ground truths) instead, and the text baselines are extracted with an RDP (Ramer-Douglas-Peucker) algorithm. dhSegment [32] used an FCN backbone to predict baseline pixels and filtered them with a Gaussian filter; it then extracted the text baselines from each connected region. Based on dhSegment, IRISA+ [6], [33] made further efforts to better detect the text baselines in tabular documents. Specifically, the end-indicators along with the baseline pixels are predicted so that a grammatical definition can be used to describe and detect tables, and subsequently all the detected elements are combined in a structural way. DMRZ [6], [34] used a U-Net [38] to detect baseline pixels and carefully designed rules to generate baseline candidates. To reduce false alarms, regions around the candidate text baselines were extracted from the input image and classified with an extra CNN. Multi-task [35] also used a U-Net to predict baseline pixels and was trained in a generative-adversarial way. After this step, interest points are clustered into text baselines with a distance-based clustering method, namely DBScan [49] clustering. Planet [2] proposed an ARU-Net to detect baseline pixels and separators, from which super pixels were generated accordingly. A greedy clustering algorithm was adopted to generate baselines based on the states information (inter-line distance and local orientation) of super pixels. This method achieved much better performance on benchmarks [5], [6] than the above-mentioned methods. However, a lot of image processing steps based on handcrafted features are required (e.g., text profiles to estimate inter-line spacing), where errors may be accumulated.

For existing deep learning methods, the performance of the baseline generation module is usually improved by introducing handcrafted features and heuristic rules that are designed according to the characteristics of the documents. However, this also involves complicated post-processing steps which lead to poor generalization abilities. In this work, we overcome these limitations with a unified framework and achieve state-of-the-art performance on two challenging text baseline detection benchmarks.

B. VISUAL RELATIONSHIP DETECTION

In earlier works, visual relationships between objects have been long exploited to assist other computer vision tasks, such as object detection [50], semantic segmentation [51], and image caption [52]. The authors of VRD [40], a large-scale dataset with a variety of visual relationships that was recently released, formulated visual relationship detection as a task itself to identify objects and predict the relationships between pairs of objects. The unique characteristic of this task is to predict visual relationships, i.e., $< subject, predicate, object >$ triplets, where the ''subject'' is related to the ''object'' by the ''predicate'' relationship. A classical solution for this task is to take both subject and object proposals as well as their union simultaneously as input to predict the predicate relationship (e.g., [39]–[43]) so that wide context information can be leveraged to improve the relationship prediction accuracy.

Inspired by these works, the authors of a more recent study [44] utilized a relation network to detect curved texts in scene images by predicting the link relationships between text primitives. This is a different task from text baseline detection since accurate text bounding boxes are provided for this task. In ablation study part, we have conducted a set of comparative experiments by using a main body baseline primitive definition, which is similar to the method for text bounding box detection. The results validate the effectiveness of the baseline primitives we have proposed.

III. METHODOLOGY

A. MOTIVATIONS AND OVERVIEW

As illustrated in Fig. 1, existing deep learning methods for baseline detection usually suffer from the wrongly merged and wrongly split problems, which are caused by the unsatisfactory performance of the baseline pixel prediction module when dealing with nearby text lines or text lines with large inter-word spacing. Specifically, background pixels within small inter-line spacing tend to be misclassified as baseline pixels, while pixels within large inter-word spacing tend to be misclassified as background ones. These errors are then propagated to the following baseline generation module while existing baseline generation methods cannot handle these cases effectively. In essence, the baseline generation problem can be formulated as classifying the link relationship between each baseline pixel pair to determine whether or not they belong to the same text baseline. If we follow

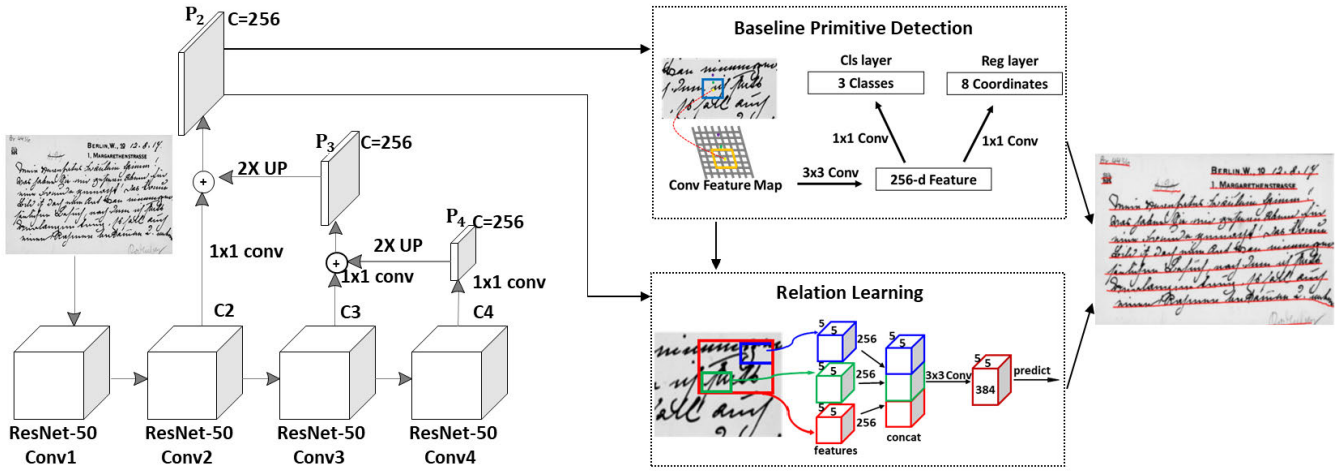

FIGURE 3. The flowchart of our proposed approach. Best viewed in color.

this logic, the wrongly merged and the wrongly split problems can be addressed by determining the link relationships between baseline pixel pairs. To leverage wider context information for better link relationship prediction, we introduce the concept of a baseline primitive, which is a virtual bounding box centered at each baseline pixel as illustrated in Fig. 2. To the best of our knowledge, we are the first to assign a virtual bounding box to a text baseline that centered at it. Previous works usually take text baseline detection problem as a baseline pixel prediction task since baselines are labeled as poly-baselines [53]. Quantitative ablation experiments have been conducted to show the effectiveness of our design. Specifically, with this design, the link relationships between baseline pixels can be equivalently transformed into the link relationships between baseline primitives, which can be exploited more effectively with a relation network, and then baseline primitives are grouped accordingly. Finally, the text baselines are generated by connecting the center points of the grouped baseline primitives efficiently.

The flowchart of our approach is illustrated in Fig. 3. It is composed of three steps, i.e., baseline primitive detection, relation learning and baseline extraction. Specifically, baseline primitive candidates are predicted with the finest feature map generated by a Feature Pyrimid Network (FPN) backbone [54]. Then, proper baseline primitive pairs are selected and a relation network is leveraged to learn the link relationships between baseline primitive pairs. Finally, the detected baseline primitives are grouped into individual text lines according to their predicted link relationships, and the text baselines are extracted accordingly. The details of the process are described in the following subsections.

B. BASELINE PRIMITIVE DETECTION

We adopt an Anchor-Free RPN (AF-RPN) method [55] to detect baseline primitives from the finest feature map generated by FPN, which is built on the top of ResNet-50 [56]. The finest feature map, i.e., P_2, has a stride of 4 pixels and $C = 256$ channels. This is different from the original AF-RPN implementation, which detects text instances of different scales from different feature pyramid levels. In this paper, we only use the finest feature pyramid level since the baseline primitives in our definition have a uniform scale.

In the training stage, we borrow the idea of border learning [57] to enhance the robustness of the detection module to nearby text lines. Specifically, for each baseline in a raw image, we first generate two virtual boxes, b_1 and b_2, centered at it, by enlarging the baseline in both vertical directions with h_1 and h_2 ($h_2 = 0.4h_1$) pixels respectively, as depicted in Fig. 4(a)-(b). Then we define the region inside box b_2 as the baseline region, the region outside box b_2 but inside box b_1 as the border region, and the region outside any box b_1 as the background region. As stated in [55], each pixel on a feature map can be mapped back to a sliding point in the raw image. During training, each pixel on the feature map will be assigned a label belonging to the three categories, i.e., baseline, border or background, according to the region in the raw image where its corresponding sliding point is located. For each "baseline" pixel, the corresponding ground-truth baseline primitive bounding box will be generated with the algorithm depicted in Fig. 4(c)-(d).

In the inference stage, for each pixel on the feature map P_2, the detection module will predict the class it belongs to, i.e., baseline, border, or background. For each detected "baseline" pixel, the detection module will further predict the offsets from it to the four vertices of its corresponding baseline primitive. As depicted in Fig. 3, the detection module is implemented as a 3×3 convolutional layer followed by two sibling 1×1 convolutional layers with 3-dimensional output channels for pixel-wise classification and 8-dimensional output channels for baseline primitive bounding-box regression, respectively. Finally, a Sigmoid output layer is adopted to generate the classification scores.

To reduce false alarms, we only keep "baseline" pixels whose classification scores are higher than a pre-defined score threshold of 0.6. We then use the standard NMS

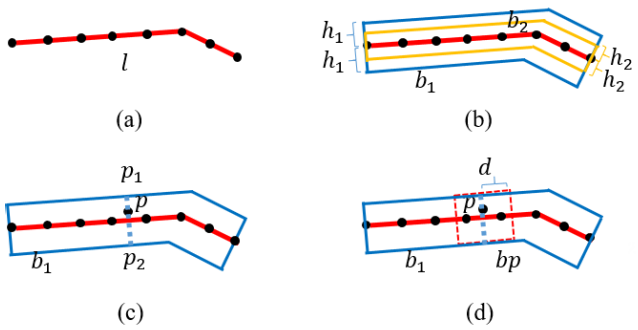

FIGURE 4. (a) We evenly divide each baseline into N (N = 7) segments with N + 1 points on it. (b) For each baseline, two virtual boxes b_1 (in blue) and b_2 (in yellow) are generated by enlarging the baseline in both vertical directions with h_1 and h_2 pixels, respectively. (c) For a "baseline" pixel p, the perpendicular line of the center line through p intersects with b_1 at p_1 and p_2. (d) We move p_1 and p_2 forward and backward along the upper and lower edges of b_1 with d pixels (d = 16), respectively, to generate the ground-truth bounding box (red dashed box) of a baseline primitive.

algorithm with an Intersection-over-Union (IoU) threshold of 0.5 to remove redundant baseline primitives.

C. RELATION LEARNING

To group the detected baseline primitives into individual baselines, there is no need to predict the link relationships between all possible baseline primitive pairs. In fact, a text baseline can be represented by a sequence of ordered baseline primitives with link relationships between only nearby baseline primitive pairs. Based on this observation, two baseline primitives are considered as a candidate pair only when they are within a certain distance. In our experiments, we set the distance as $10h_1$ pixels, which ensures that baseline primitives with large inter-word spacing can be selected as a candidate pair and avoids generating too many redundant pairs.

There are many effective ways to exploit the link relationship between a primitive pair. Following the work in [44], we choose to use the relation network as a first attempt for its simplicity and effectiveness. In a nutshell, three feature descriptors extracted from two baseline primitive boxes and their union box are utilized to predict the link relationship between this baseline primitive pair. In this way, both relative position information and wide context information can be leveraged to improve link prediction accuracy. Specifically, as illustrated in Fig. 3, three $256 \times 5 \times 5$ feature descriptors of those two baseline primitives and their union box are extracted with the RoI Align algorithm [58] from the P_2 feature map. Then, these three feature descriptors are concatenated along the channel dimension and fed into a 3×3 convolutional layer to generate a fixed-size ($384 \times 5 \times 5$) feature descriptor. Finally, a 2-hidden-layer MLP with 1,024 nodes at each hidden layer followed by a Sigmoid output layer is adopted to predict the link relationship between this baseline primitive pair.

We directly use the ground-truth baseline primitives to generate training samples for the relation learning module. It is based on the assumption that there is little difference between the predicted baseline primitive boxes from the trained model and its corresponding ground truth. This assumption makes sense because the performance of the baseline primitive detection module is satisfactory and the loss of the box regression term converges to a small value at the end of the training process. In this way, the training samples generation becomes much more accurate and easier because we can directly select baseline primitive pairs from the same text baseline as positive samples while taking baseline primitive pairs from different text baselines as negative samples without the ambiguity associated with matching the learned baseline primitives to the ground-truth text baselines.

D. BASELINE EXTRACTION

During inference time, baseline primitives are put into individual primitive groups based on the predicted link relationship with a threshold of 0.7 by using the Union-Find algorithm. According to our definition, the center of a baseline primitive lies on a text baseline, thus the final baseline can be generated by connecting these centers. In particular, for each baseline primitive that belongs to a specific group, its center is computed by the four predicted vertices. Then, a rough text line orientation is estimated by computing the variances of those centers along horizontal and vertical directions because its value can vary significantly along the text line direction. With a rough horizontal orientation, centers are sorted from left to right, otherwise centers are sorted from top to bottom. After these steps, a final baseline is generated by simply connecting the center points of the baseline primitives in a sequential order.

E. LOSS FUNCTIONS

1) MULTI-TASK LOSS FOR BASELINE PRIMITIVE DETECTION

There are two sibling output layers for the baseline primitive detection module, i.e., a baseline pixel prediction layer and a quadrilateral bounding box regression layer. The multi-task loss function can be denoted as follows:

$$L(c, c^*, t, t^*) = \lambda_c L_c(c, c^*) + \lambda_l L_l(t, t^*), \tag{1}$$

where c and c^* are the predicted and the ground-truth 3-dimensional labels for each sampling pixel respectively, and $L_c(c, c^*)$ is a binary cross-entropy loss for each category channel; t and t^* represent the predicted and the ground-truth 8-dimensional normalized coordinate offsets [55], and $L_l(t, t^*)$ is a Smooth-L_1 loss [60] for the bounding box regression task. λ_c and λ_l are two balancing parameters for multi-task learning. We set $\lambda_c = 1$ and $\lambda_l = 1$ because this worked well in our experiments.

2) LOSS FOR RELATION LEARNING

The loss for the relation learning module is defined as follows:

$$L(r, r^*) = \lambda_r L_r(r, r^*), \tag{2}$$

TABLE 1. Performance comparison on cBAD 2017 benchmark. (* indicates that numbers are quoted from reference [5]).

Methods	Complex Track			Simple Track		
	P-value	R-value	F-value	P-value	R-value	F-value
LITIS*	-	-	-	78.0%	83.6%	80.7%
UPVLC*	83.3%	60.6%	70.2%	93.7%	85.5%	89.4%
IRISA*	69.2%	77.2%	73.0%	88.3%	87.7%	88.0%
BYU*	77.3%	82.0%	79.6%	87.8%	90.7%	89.2%
Multi-task [35]	84.8%	85.4%	85.1%	-	-	-
DMRZ*	85.4%	86.3%	85.9%	97.3%	97.0%	97.1%
dhSegment [32]	82.6%	92.4%	87.2%	94.3%	93.9%	94.1%
IRISA+ [33]	85.8%	93.5%	89.5%	-	-	-
ARU-Net [2]	92.6%	91.8%	92.2%	97.7%	98.0%	97.8%
docExtractor [36]	88.3%	94.3%	91.3%	94.8%	97.8%	96.3%
Ours	92.8%	94.7%	93.8%	96.5%	97.8%	97.2%

TABLE 2. Performance comparison on cBAD 2019 benchmark. (* indicates that numbers are quoted from reference [6]).

Methods	P-value	R-value	F-value
Baseline (DMRZ-17)*	77.3%	74.3%	75.8%
TJNU*	85.2%	88.5%	86.8%
UPVLC*	91.1%	90.2%	90.7%
DMRZ-19*	92.5%	90.5%	91.5%
Planet(ARU-Net)*	93.7%	92.6%	93.1%
docExtractor [36]	92.0%	93.1%	92.5%
Ours-17	89.7%	88.4%	89.1%
Ours-19	93.3%	94.3%	93.8%

where r and r^* are the predicted and ground-truth relationship labels for each sampling baseline primitive pair, and $L_r(r, r^*)$ is also a binary cross-entropy loss for link relationship classification.

The total loss of the framework is a sum of $L(c, c^*, t, t^*)$ and $L(r, r^*)$. In our experiments, we also set $\lambda_r = 1$.

IV. EXPERIMENTS

A. DATASETS AND EVALUATION PROTOCOLS

To evaluate the performance of the proposed approach and compare it with other works, we conduct experiments on two publicly available text baseline detection benchmarks, cBAD 2017 and cBAD 2019.

cBAD 2017 [5] consists of 2,036 document images written between the years 1,470 and 1,930, which are collected from 9 different European archives. These documents are split into two tracks, simple documents and complex documents. The simple documents track focuses on baseline detection in documents with simple layouts, and the documents are annotated with extra text region information. It contains 216 images for training and 539 images for testing. The complex documents track takes more challenging layouts into consideration, including full page tables, multi-column and rotated text lines. This track has no extra text region information and contains 270 document images for training and 1,010 document images for testing.

cBAD 2019 [6] is a successor of cBAD 2017 with a larger dataset that contains more diverse document pages. These document pages have different layouts and origins, such as heavily structured pages, sparsely inscribed pages, drawings and engravings. This dataset consists of 3,021 document images sampled from 175,567 archival documents. Among them, 755 images are used as a training set, 755 images as a validation set and 1,511 images as a testing set.

We follow the official evaluation protocols to make our results comparable to those from other methods. In simple terms, this scheme aligns detected baselines with ground truths by using a defined coverage function. It has three indicators, R-value, P-value and F-value, which are similar to the well-known terms recall, precision and F-score, respectively. More details on the scheme can be found in [1].

B. IMPLEMENTATION DETAILS

We implement our approach based on PyTorch[1] v0.4.1 while the experiments are conducted on a workstation with 4 Nvidia V100 GPUs. The weights of ResNet-50 related layers in the backbone network are initialized with a pre-trained ResNet-50 model for the ImageNet classification task [56]. The weights of newly added layers in FPN, baseline primitive detection module and relation learning module are initialized with a Gaussian distribution of mean 0 and standard deviation 0.01. Our models are trained in an end-to-end manner and optimized by the standard SGD algorithm, where the momentum is 0.9 and weight decay is 0.0005. Note that all the models are trained for 40K iterations with the initial learning rate of 0.004, which is divided by 10 at each 10K iterations. In each training iteration, we sample one image for each GPU. For each image, we randomly select 128 baseline, 128 border and 128 background pixels for the baseline primitive detection module, and 64 positive and 64 negative relation pairs for the relation learning module. During training, we adopt a multi-oriented and multi-scale data augmentation strategy. Specifically, each training image is first randomly rotated by an angle in the range of $(-45°, 45°)$ and then its shorter side is randomly rescaled to a number in the set of {800, 928, 1024, 1200} while keeping its aspect ratio.

In the testing phase, we adopt a single model and a single scale testing strategy. In all the experiments, the shorter side of each testing image is rescaled to be 1,024 pixels.

C. OVERALL PERFORMANCE

We compare the proposed approach with other competitive methods by applying them to the benchmarks cBAD 2017 and

[1] https://pytorch.org/

FIGURE 5. Qualitative results of the proposed approach. Detected baselines are shown in red. Results in the first row are from cBAD-17 and the following ones are from cBAD-19.

cBAD 2019. Quantitative results are listed in Table 1 and Table 2, respectively.

For the cBAD 2017 benchmark, our approach achieves the best results of 92.8%, 94.7% and 93.8% in P-value, R-value and F-value, respectively, on the complex documents track, which demonstrates the effectiveness of our approach. For the simple documents track, we do not use the informative text region annotations during the inference stage since in this paper we mainly focus on the baseline detection problem in unconstrained historical documents. Even without this information, our approach can still achieve comparable results as prior arts, which can further demonstrate the superior performance of our approach.

For the cBAD 2019 benchmark, our approach also achieves the best F-value of 93.8%. The ''baseline'' results are from the winner method of cBAD 2017 complex track (DMRZ in Table 1, denoted as DMRZ-17 in Table 2), where the trained model on cBAD 2017 is tested directly on cBAD 2019. Similarly, we also test our model trained on the cBAD 2017 complex track, denoted as Ours-17 in Table 2, on cBAD 2019 for comparison. It is worth noting that the performance of DMRZ-17 decreases by 10.1% in F-value when it tested on cBAD 2019 testing set while ours decreases only by 4.7%, which shows that our method has a better generalization ability with regard to unseen datasets.

We observe that our approach can detect text baselines effectively under various challenging conditions, such as multi-column and sparsely inscribed pages, documents with multiple text-line spacings as well as those with complicated backgrounds and degradations. Some qualitative results are shown in Fig 5.

TABLE 3. Comparison of the connectivity-based baseline generation strategy with our relation learning based strategy on the cBAD-19 benchmark.

Methods	Validation Set			Testing Set		
	P-value	R-value	F-value	P-value	R-value	F-value
Connectivity	89.0%	90.1%	89.6%	89.6%	90.2%	89.9%
Ours	92.7%	94.4%	93.5%	93.3%	94.3%	93.8%

TABLE 4. Comparison of different heights h_1 (number of pixels) of a defined virtual box on the cBAD 2019 validation set.

box height h_1	P-value	R-value	F-value
10	91.1%	94.3%	92.7%
20	92.2%	94.9%	93.5%
25	92.7%	94.4%	93.5%
30	92.9%	93.3%	93.1%
40	91.7%	90.2%	90.9%

TABLE 5. Comparison of different baseline primitive definitions in our framework on the cBAD 2019 validation set.

Strategy	P-value	R-value	F-value
Fixed-size box	90.1%	91.5%	90.8%
Text main body [44]	89.5%	93.0%	91.2%
Ours	92.7%	94.4%	93.5%

D. ABLATION STUDY

In this section, we perform ablation studies on the key components of our approach to analyze their effects.

1) EFFECTIVENESS OF RELATIONSHIP PREDICTION BASED BASELINE GENERATION

As stated in Sec. 3.1, the performance of CNN-based baseline detection methods is affected by the baseline generation strategy. To demonstrate the effectiveness of the proposed relationship prediction based baseline generation module, we compare it with an 8-neighborhood connectivity-based method, which is widely used in many prior arts. Besides conducting ablation experiments on the validation set, we also conduct them on the testing set of the cBAD 2019 benchmark for comparison with other works. For a fair comparison, we replace the relation learning module in our codes with an 8-neighborhood connectivity-based baseline generation module, and the hyper-parameters are carefully tuned to get its best possible results. As shown in Table 3, our implemented connectivity-based method achieves an F-value of 89.9%. This is higher than the F-value of TJNU-19 (86.8%) in Table 2, which is a method that also adopts the connectivity-based baseline generation strategy. Compared with this strong baseline, our method still improves the F-value by 3.9% on both the validation set and the testing set of cBAD 2019. This performance increase mainly comes from improvements in the abovementioned wrongly split and wrongly merged problems. Two comparative examples are shown in Fig. 6.

2) INFLUENCE OF THE VIRTUAL BOX HEIGHT

Intuitively, our approach can achieve its best performance when the enlarged virtual box height h_1 is set according to the actual height of each text line and the line interval. However, the abovementioned information is not provided and we investigate the influence of h_1 when it is set as a constant number across datasets in this part. The experimental results are shown in Table 4, from which we can observe that our system is not sensitive to this hyper-parameter as long as it is in a proper range. If h_1 is too small, less context information is encoded so that our baseline primitive design cannot produce the maximum effect; if h_1 is too big, the baseline primitive will contain much noise, leading to a significant performance degradation. Therefore, we set $h_1 = 25$ pixels in all the experiments as a trade-off for its satisfactory performance.

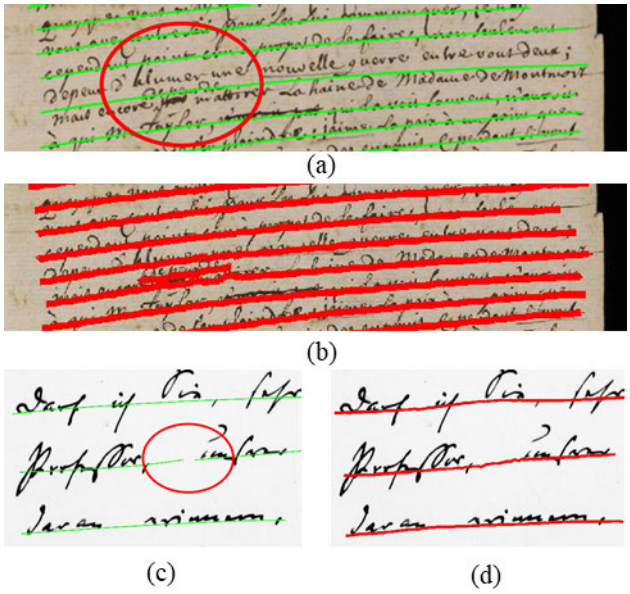

FIGURE 6. Effectiveness of the proposed relation learning module: (a) and (c) are results generated by the connectivity based method, (b) and (d) are results generated by our approach.

3) EFFECTIVENESS OF THE PROPOSED BASELINE PRIMITIVE DEFINITION

A baseline primitive is formed according to a baseline pixel and the virtual line bounding box, which is generated by enlarging the baseline in both vertical directions with a constant height, as illustrated in Fig. 4. In this part, we study some other ways to define a baseline primitive.

The most straightforward strategy would be to assign a fixed-size box directly to each baseline pixel. In this way, there is no need to regress a virtual box for each baseline pixel. In practice, we replace our learned baseline primitive box with this fixed-size box, thus allowing the relationships between pixel pairs to be explored with these pre-defined boxes at both training and inference time. Its performance, shown in the first row of Table 5, is worse than the strategy we adopt. The biggest difference between this strategy and

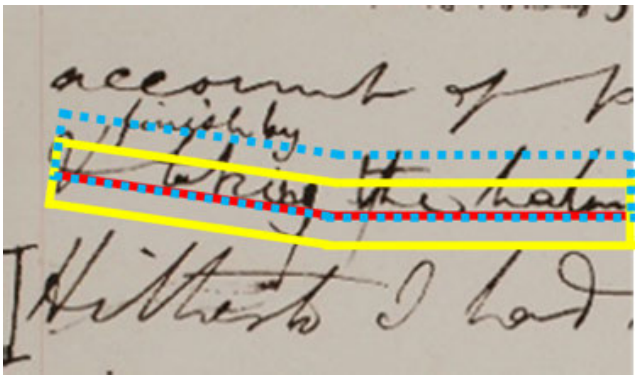

FIGURE 7. Illustration of two ways to generate virtual boxes for a baseline. Baseline is depicted in red; enlarging the baseline only at the text main body direction is depicted by the dashed blue box; the solid yellow box indicates our definition, which enlarges the baseline at both directions. Best viewed in color.

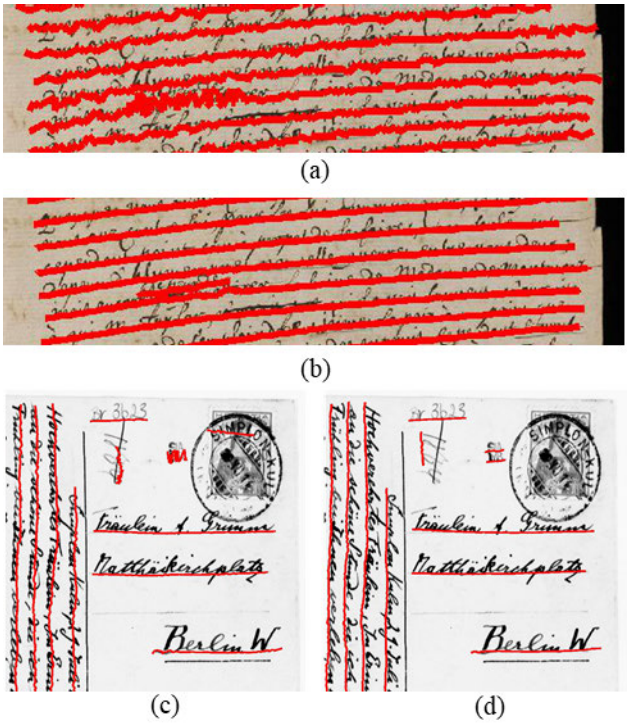

FIGURE 8. Effectiveness of the proposed definition of baseline primitive. (a) is generated from the fixed-size box definition; (c) is generated from the main body definition; (b) and (d) are results generated by our proposed definition for comparison.

ours is that our defined baseline primitives are centered at the baselines for all the pixels in the core region, while the former does not. It is easier for these pre-defined boxes to cover adjacent text lines, but this will make it confusing to learn the link relationship. Therefore, this fixed-size box strategy lacks the ability to solve the wrongly merged problem, as illustrated in Fig. 8(a).

Besides, we can also generate the virtual box of each text baseline with an enlargement in the text main body direction as shown in Fig. 7, and this virtual box can be roughly seen as a text main body. To verify this strategy, we replace the ground truths generation in our pipeline while keep others

TABLE 6. Comparison of different baseline primitive definitions based on the framework in [59] on the cBAD 2019 validation set.

Strategy	P-value	R-value	F-value
Fixed-size box	89.3%	90.7%	90.0%
Text main body [59]	88.9%	92.4%	90.6%
Ours	92.6%	92.9%	92.7%

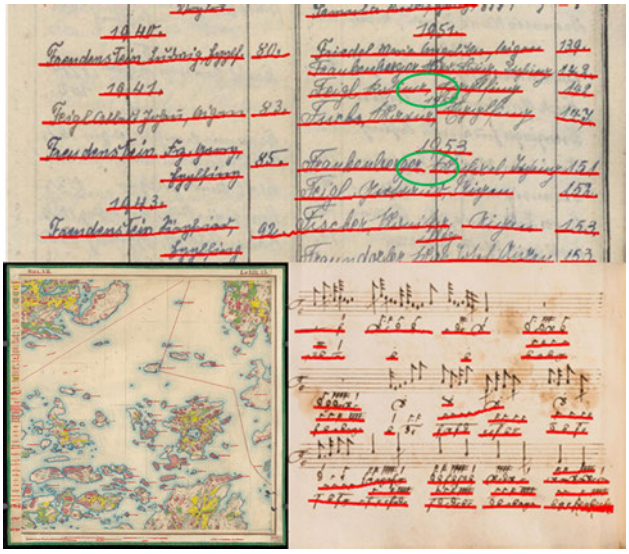

FIGURE 9. Some cases where our approach fails.

unchanged, which is similar to the method designed in reference [44] for detecting text bounding boxes of curved scene texts. However, this method is worse than our approach, as shown in the second row of Table 5. In fact, this text main body strategy lacks the ability to differentiate between a baseline and an upper line in an arbitrarily oriented text line if no extra information is provided. As a comparison, the text baseline is located at the center of baseline primitives by our definition, thus the ambiguous problem is avoided. An example of this limitation is illustrated in Fig. 8(c), where the detected vertical text baselines are wrong.

We also implement [59] as another base framework to further validate the generalization ability of our baseline primitive design. Specifically, [59] is designed for handwritten text bounding boxes detection in natural scene images, where they use an 8-neighborhood link prediction module instead of a relation network to exploit the link relationship. For a fair comparison, we keep all the configurations fixed except the baseline primitive definition strategy. As shown in Table 6, our baseline primitive design has consistent gains over the other two strategies.

These two sets of comparative experiments demonstrate the effectiveness of our baseline primitive definition.

E. LIMITATIONS OF OUR APPROACH

Although our approach has achieved superior results in most challenging scenarios as depicted in Fig. 5, it has failed in some difficult cases. Some of these are illustrated in Fig. 9. The first row of Fig. 9 shows a hard case where a text baseline is split into two when there is a line separator in the word

spacing (emphasized with green circles). Although this split seems to be correct when only local context is considered, it is wrong when nearby text lines are taken into consideration. Solving this limitation is currently beyond the capability of our approach because it may require global context information to exploit the link relationships effectively. The second row of Fig. 9 shows other examples of scenarios where our approach does not work well, such as musical notes and maps, which are rarely included in training sets. The cases in which our approach fails do not seem to follow a specific deterministic principle and may be handled more effectively when more training samples are available.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a baseline primitive based approach to text baseline detection in unconstrained historical documents. Previous methods usually took text baseline detection as a semantic segmentation task, thus suffered from the wrongly merged and wrongly split problems. Unlike them, we first introduced the concept of a baseline primitive, which allowed us to solve the baseline generation problem by learning the link relationships between baseline primitive pairs. The notion of baseline primitives led us to choose the relation network to exploit their link relationships, which helped connect distant baseline primitives in the same text baseline and separate close baseline primitives belonging to different text baselines. Consequently, our approach achieved superior performance on two challenging benchmarks, namely cBAD 2017 and cBAD 2019.

However, in some cases our approach was not effective, as shown in the examples in the first row of Fig. 9. This implies that wider context information is needed to generate text baselines. In future research, we will explore other visual relationship learning methods involving global context information to address these limitations.

ACKNOWLEDGMENT

(Wei Jia and Chixiang Ma contributed equally to this work.) This work was done when Wei Jia and Chixiang Ma were interns in Speech Group, Microsoft Research Asia, Beijing, China.

REFERENCES

[1] T. Grüning, R. Labahn, M. Diem, F. Kleber, and S. Fiel, "READ-BAD: A new dataset and evaluation scheme for baseline detection in archival documents," in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 351–356.

[2] T. Grüning, G. Leifert, T. Strauß, J. Michael, and R. Labahn, "A two-stage method for text line detection in historical documents," *Int. J. Document Anal. Recognit.*, vol. 22, no. 3, pp. 285–302, Sep. 2019.

[3] M. Pastor, "Text baseline detection, a single page trained system," *Pattern Recognit.*, vol. 94, pp. 149–161, Oct. 2019.

[4] L. Ma, C. Long, L. Duan, X. Zhang, Y. Li, and Q. Zhao, "Segmentation and recognition for historical tibetan document images," *IEEE Access*, vol. 8, pp. 52641–52651, 2020.

[5] M. Diem, F. Kleber, S. Fiel, T. Grüning, and B. Gatos, "CBAD: ICDAR2017 competition on baseline detection," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1355–1360.

[6] M. Diem, F. Kleber, R. Sablatnig, and B. Gatos, "CBAD: ICDAR2019 competition on baseline detection," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1494–1498.

[7] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: A survey," *Int. J. Document Anal. Recognit.*, vol. 9, nos. 2–4, pp. 123–138, Apr. 2007.

[8] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "A comprehensive survey of mostly textual document segmentation algorithms since 2008," *Pattern Recognit.*, vol. 64, pp. 1–14, Apr. 2017.

[9] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR2009 handwriting segmentation contest," *Int. J. Document Anal. Recognit.*, vol. 14, no. 1, pp. 25–33, Mar. 2011.

[10] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1402–1406.

[11] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, "Arabic hand-written text-line extraction," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 281–285.

[12] N. Ouwayed and A. Belaïd, "A general approach for multi-oriented text line extraction of handwritten documents," *Int. J. Document Anal. Recognit.*, vol. 15, no. 4, pp. 297–314, Dec. 2012.

[13] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to handwritten line segmentation," *Proc. SPIE*, vol. 6500, Jan. 2007, Art. no. 65000T.

[14] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," in *Proc. 1st Int. Workshop Document Image Anal. Libraries*, 2004, pp. 306–312.

[15] M. Makridis, N. Nikolaou, and B. Gatos, "An efficient word segmentation technique for historical and degraded machine-printed documents," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2007, pp. 178–182.

[16] F. Yin and C.-L. Liu, "Handwritten text line extraction based on minimum spanning tree clustering," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, vol. 3, Nov. 2007, pp. 1123–1128.

[17] F. Yin and C.-L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," *Pattern Recognit.*, vol. 42, no. 12, pp. 3146–3157, Dec. 2009.

[18] T. Gruuening, G. Leifert, T. Strauß, and R. Labahn, "A robust and binarization-free approach for text line detection in historical documents," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 236–241.

[19] L. Likforman-Sulem, A. Hanimyan, and C. Faure, "A Hough based algorithm for extracting text lines in handwritten documents," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 2, 1995, pp. 774–777.

[20] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognit.*, vol. 42, no. 12, pp. 3169–3183, Dec. 2009.

[21] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of historical handwritten documents into text zones and text lines," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 464–469.

[22] R. Saabni, A. Asi, and J. El-Sana, "Text line extraction for historical document images," *Pattern Recognit. Lett.*, vol. 35, pp. 23–33, Jan. 2014.

[23] J. Ryu, H. I. Koo, and N. I. Cho, "Language-independent text-line extraction algorithm for handwritten documents," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1115–1119, Sep. 2014.

[24] E. Öztop, A. Y. Mülayim, V. Atalay, and F. Yarman-Vural, "Repulsive attractive network for baseline extraction on document images," *Signal Process.*, vol. 75, no. 1, pp. 1–10, Jan. 1999.

[25] F. Yin and C.-L. Liu, "A variational Bayes method for handwritten text line segmentation," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 436–440.

[26] F. Luthy, T. Varga, and H. Bunke, "Using hidden Markov models as a tool for handwritten text line segmentation," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2007, pp. 8–12.

[27] B. Ahn, J. Ryu, H. I. Koo, and N. I. Cho, "Textline detection in degraded historical document images," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, pp. 1–13, Dec. 2017.

[28] D. Chakraborty and U. Pal, "Baseline detection of multi-lingual unconstrained handwritten text lines," *Pattern Recognit. Lett.*, vol. 74, pp. 74–81, Apr. 2016.

[29] M. Murdock, S. Reid, B. Hamilton, and J. Reese, "ICDAR 2015 competition on text line detection in historical documents," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1171–1175.

[30] Y. Akbari, S. Al-Maadeed, and K. Adam, "Binarization of degraded document images using convolutional neural networks and wavelet-based multichannel images," *IEEE Access*, vol. 8, pp. 153517–153534, 2020.

[31] G. Renton, Y. Soullard, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, "Fully convolutional network with dilated convolutions for handwritten text line segmentation," *Int. J. Document Anal. Recognit.*, vol. 21, no. 3, pp. 177–186, Sep. 2018.

[32] S. A. Oliveira, B. Seguin, and F. Kaplan, "DhSegment: A generic deep-learning approach for document segmentation," in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Aug. 2018, pp. 7–12.

[33] C. Guerry, B. Couasnon, and A. Lemaitre, "Combination of deep learning and syntactical approaches for the interpretation of interactions between text-lines and tabular structures in handwritten documents," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 858–863.

[34] M. Fink, T. Layer, G. Mackenbrock, and M. Sprinzl, "Baseline detection in historical documents using convolutional U-Nets," in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 37–42.

[35] L. Quirós, "Multi-task handwritten document layout analysis," 2018, *arXiv:1806.08852*. [Online]. Available: http://arxiv.org/abs/1806.08852

[36] T. Monnier and M. Aubry, "DocExtractor: An off-the-shelf historical document element extraction," in *Proc. 17th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Sep. 2020, pp. 91–96.

[37] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[39] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal, "Relationship proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5678–5686.

[40] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 852–869.

[41] Y. Li, W. Ouyang, X. Wang, and X. Tang, "ViP-CNN: Visual phrase guided convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1347–1356.

[42] Y. Zhu and S. Jiang, "Deep structured learning for visual relationship detection," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[43] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Visual relationship detection with internal and external linguistic knowledge distillation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1974–1982.

[44] C. Ma, Z. Zhong, L. Sun, and Q. Huo, "A relation network based approach to curved text detection," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 707–713.

[45] A. Lemaitre and J. Camillerapp, "Text line extraction in handwritten document with Kalman filter applied on low resolution image," in *Proc. 2nd Int. Conf. Document Image Anal. Libraries (DIAL)*, 2006, p. 8.

[46] Z. Shi, S. Setlur, and V. Govindaraju, "Text extraction from gray scale historical document images using adaptive local connectivity map," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 794–798.

[47] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten Arabic text lines," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 176–180.

[48] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.

[49] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996, vol. 96, no. 34, pp. 226–231.

[50] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3217–3224.

[51] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 300–316, Dec. 2008.

[52] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 15–29.

[53] V. Romero, J. A. Sanchez, V. Bosch, K. Depuydt, and J. de Does, "Influence of text line segmentation in handwritten text recognition," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 536–540.

[54] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[55] Z. Zhong, L. Sun, and Q. Huo, "An anchor-free region proposal network for faster R-CNN-based text detection approaches," *Int. J. Document Anal. Recognit.*, vol. 22, no. 3, pp. 315–327, Sep. 2019.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[57] Y. Wu and P. Natarajan, "Self-organized text detection with minimal post-processing via border learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5000–5009.

[58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[59] W. Jia, Z. Zhong, L. Sun, and Q. Huo, "A CNN-based approach to detecting text from images of whiteboards and handwritten notes," in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Aug. 2018, pp. 1–6.

[60] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

WEI JIA received the B.Eng. degree in electronic information engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include machine learning, pattern recognition, image processing, and especially for handwritten text line detection.

CHIXIANG MA received the B.Eng. degree in electronic information engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include machine learning, pattern recognition, text detection, and table detection and recognition.

LEI SUN received the B.Eng. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2009 and 2015, respectively. He is currently a Principal Researcher with the Speech Group, Microsoft Research Asia (MSRA), Beijing, China. His research interests include text detection, layout analysis, table detection and recognition, OCR, document image understanding, and information retrieval.

QIANG HUO (Member, IEEE) received the B.Eng. degree in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1987, the M.Eng. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1989, and the Ph.D. degree in electrical engineering from USTC, in 1994. He is currently the Partner Research Manager with the Speech Group, Microsoft Research Asia (MSRA), Beijing, China. Prior to joining MSRA, in August 2007, he had been a Faculty Member with the Department of Computer Science, The University of Hong Kong, where he also did his Ph.D. research on speech recognition, from 1991 to 1994. From 1995 to 1997, he worked with the Advanced Telecommunications Research Institute (ATR), Kyoto, Japan. Over the past 30 years, he has been active in research and making contributions in the fields of speech recognition, handwriting recognition, OCR, document understanding, gesture recognition, biometric-based user authentication, and hardware design for speech and image processing.