Information Processing and Management 60 (2023) 103286

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

An effective method for figures and tables detection in academic literature

Fengchang Yu, Jiani Huang, Zhuoran Luo, Li Zhang, Wei Lu *

School of Information Management, Wuhan University, Wuhan, 430072, China
Institute for Information Retrieval and Knowledge Mining, Wuhan University, Wuhan, 430072, China

ARTICLE INFO

Keywords:
Figure and table detection
Academic literature
Object detection
Semantic segmentation

ABSTRACT

Figures and tables in scientific articles serve as data sources for various academic data mining tasks. These tasks require input data to be in its entirety. However, existing studies measure the performance of algorithms using the same IoU (Intersection over Union) or IoU-based metrics that are used for natural situations. There is a gap between high IoU and detection entirety in scientific figures and tables detection tasks. In this paper, we demonstrate the existence of this gap and suggest that the leading cause is the detection error in the boundary area. We propose an effective detection method that cascades semantic segmentation and contour detection. The semantic segmentation model adopted a novel loss function to enhance the weights of boundary parts and a categorized dice metric to evaluate the imbalanced pixels in the segmentation result. Under rigorous testing criteria, the method proposed in this paper yielded a page-level F1 of 0.983 exceeding state-of-the-art academic figure and table detection methods. The research results in this paper can significantly improve the data quality and reduce data cleaning costs for downstream applications.

1. Introduction

Figures and tables in academic literature are critical visual resources that convey the study's primary content and crucial findings. They provide the foundational data for various fine-grained analysis studies on the content of academic literature. Therefore, accurately detecting figures and tables in academic literature is a prerequisite for all subsequent tasks.

Academic literature is published and distributed in various formats, but the most popular format is PDF, the primary distribution format for the world's five most significant volumes of publications. Thus, extracting figures and tables from the literature primarily equals extracting them from PDF documents. Images and table contents can be easily located and extracted from structured documents like MS Word and HTML. However, the PDF format is a set of print commands and lacks a layout description. Textual and non-textual elements are separated and stored unrelatedly, meaning commonly used vector figures are not stored as a single structure but as several lines and texts. Therefore, representing the textual and non-textual elements in a unified format is the major challenge of detecting figures and tables in PDF-format academic literature.

Heuristic-based and deep-learning-based object detection algorithms are two mainstream detection methods. Heuristic-based algorithms take PDF files as input and extract elements from PDF files of journal and conference papers with a similar layout. They utilize manually constructed rules to derive feature vectors and merge them to obtain the position coordinates of the figures and tables. This element-level method usually detects figures and tables entirely if they are detectable. However, according to Siddiqui, Malik, Agne, Dengel, and Ahmed (2018), extending to a large volume of academic literature with varying layouts is difficult because

* Corresponding author at: Institute for Information Retrieval and Knowledge Mining, Wuhan University, Wuhan, China.
E-mail address: yufc2002@whu.edu.cn (F. Yu).

https://doi.org/10.1016/j.ipm.2023.103286
Received 16 June 2022; Received in revised form 10 December 2022; Accepted 18 January 2023
Available online 1 February 2023

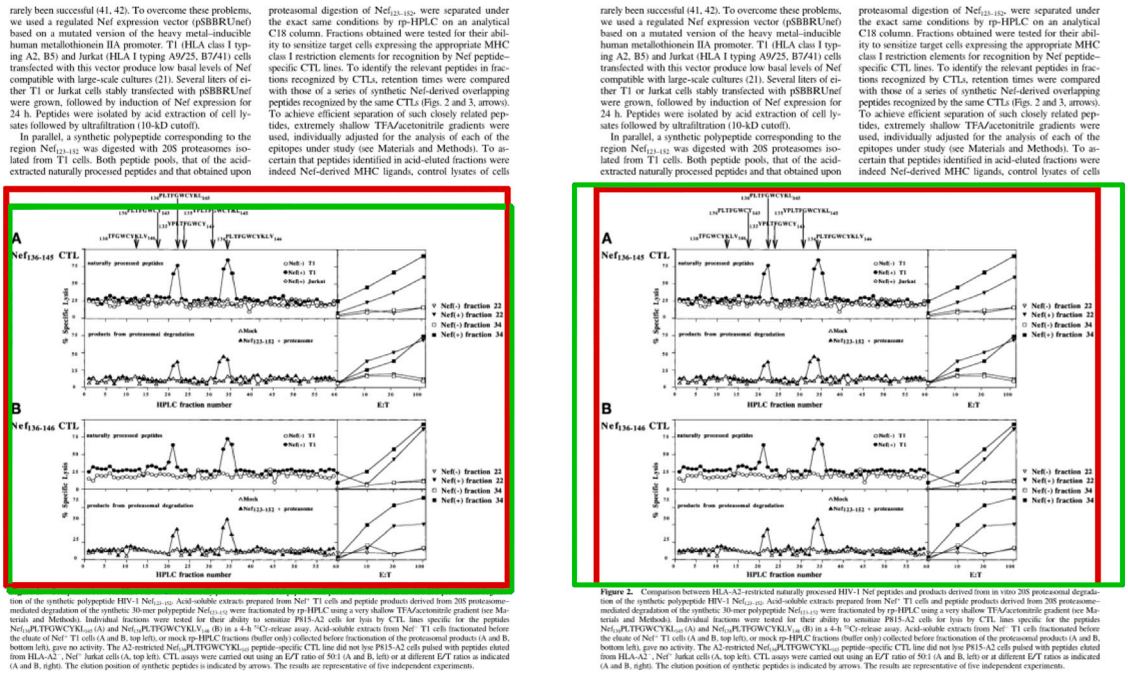


Fig. 1. Examples of lousy detection and sound detection in academic literature. The red rectangle denotes the ground truth bounding box and the green rectangle denotes the predicted bounding box. Various IoU metrics are listed in Table 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
The values of IoU and derivatives for bad detection and good detection in Fig. 1.

	Bad detection	Good detection
IoU	0.9247	0.8843
GIoU	0.7246	0.6898
DIoU	0.9241	0.8840
CIoU	0.9242	0.8841

of the low recall issue. Deep-learning object detection algorithms overcome the low recall shortage of heuristic-based algorithms by automatically learning high-dimensional features. However, experiment results from considerable research, including (Chen, Lee, Lin, Wang, & Chen, 2021; Huang et al., 2019; Saha, Mondal, & Jawahar, 2019) indicate that present models are unable to frame the figures and tables completely.

The IoU (Intersection over Union) metrics and IoU-based metrics, such as AP and mAP, are popular evaluation metrics in this task (Jimeno Yepes, Zhong, & Burdick, 2021). They are originally used in natural scenarios like the PASCAL VOC2007 challenge (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010) and the COCO 2017 challenge (Lin et al., 2014). The IoU is a number from 0 to 1 that specifies the amount of overlap between the predicted and ground truth bounding box. The thresholds of IoU are often set at 0.5, 0.7, and 0.8, and Lee and Chen (2021) points out that a prediction can be regarded as a good one if the IoU is greater than 0.5.

We argue that evaluation standards for natural scenarios are inappropriate for this task. Because downstream applications require input to be in its entirety, all details in scientific figures and tables matter. A typical example is that a diagram is incomprehensible without an axis label, while a cat picture without legs is still a cat. Higher IoU does not always equal better detection. As shown in Fig. 1, a higher IoU detection result (Fig. 1A) cuts off a part of texts in the figure, which makes this result useless for downstream tasks. The detection result with a lower IoU (Fig. 1B) can be used for downstream tasks, though some white background area is included. There are several derivatives of IoU, such as GIoU, DIoU, and CIoU. We listed their values for each example in Table 1. All the results of lousy detection are greater than those of sound detection.

The main issue addressed in this paper is the following: How do we entirely detect scientific figures and tables and evaluate detection results under downstream application requirements? This paper proposes a detection method using a cascading image semantic segmentation model and a contour detection algorithm. A U-Net model with an attention mechanism is utilized to classify

each pixel on the rendered page of an academic paper into backgrounds, figures, and tables. The classification regions are then computed by the computer vision contour detection method to determine the position of the figures and tables. The results of our method are compared under strict evaluation standards consistent with practical applications with state-of-the-art heuristic-based and pure-vision object-detection-based methods.

The contributions of this paper are as follows.

1. We proposed an effective academic figure and table detection method that cascades semantic segmentation and contour recognition.
2. We defined a boundary enhance loss function that can improve the semantic segmentation performance of figure and table edge regions.
3. We applied the categorized dice coefficient to address the problem of pixel imbalance in the academic literature rendered images.

The remainder of the paper is organized as follows: the second section presents the literature related to this paper, the third section is the introduction of the proposed method, the fourth section is the description of the experimental and discussion sections, and the final section is the conclusion of this paper.

2. Related work

Scientific figures and tables detection is a sub-problem of document analysis and content identification, which aims to calculate the location of figures and tables within the pages of the literature (Srihari, Lam, Govindaraju, Srihari, & Hull, 1986). As typical visual elements, figures and tables contain significant findings and experimental results from academic research (Choudhury et al., 2013), help readers understand the research content (Lebourgeois, Bublinski, & Emptoz, 1992), and play a critical role in scholarly communication and dissemination. Accurately detecting figures and tables in academic articles is vital for various visual content studies (Augusto Borges Oliveira & Palhares Viana, 2017; Bhatia & Mitra, 2012; Poco & Heer, 2017).

The heuristic-based algorithm, object detection, and semantic segmentation are three major detection methods. The former accepts PDF format files as input, parses PDF stream content, and utilizes heuristic algorithms to detect figures and tables. The latter two take rendered document images as input and make predictions using deep learning models.

The core idea of the heuristic-based algorithm is to extract text and non-text elements from PDF streams, combine them into a composite object, then use elements attributes as features to categorize composite object types into figures or tables. Text element attributes consist of coordinates, font, color, and size. Non-textual element attributes include position, type, and shape (Perez-Arriaga, Estrada, & Abad-Mota, 2016; Ray Choudhury, Mitra, & Giles, 2015). The approach for combining and classifying objects differs significantly between studies. Most studies employ targeted feature engineering for particular journal and conference papers with similar layouts (Ajij, Pratihar, Roy, & Hanne, 2022; Corrêa & Zander, 2017; Li, Jiang, & Shatkay, 2019; Praczyk & Nogueras-Iso, 2013). One of the most well-known tools is PDFFigures 2.0 (Clark & Divvala, 2016). The main advantage is that relatively high detection precision can be reached, and detection entirety is better than deep learning models. However, creating handcrafted features is labor and time intensive, and the literature layout significantly affects detection recall.

The object detection networks are trained to directly predict the vertex coordinates of the region where the figure or table is found. Two-stage object detection models task into potential region proposal generation and potential region classification and correction. A few typical examples follow. Sun, Zhu, and Hu (2019) combined Faster R-CNN and corner location method to detect tables in the document image. Agarwal, Mondal, and Jawahar (2021) used ResNet as the backbone of Mask R-CNN and adopted deformable convolution instead of convolutional convolution on the table detection task. Fernandes, Simsek, Kantarci, and Khan (2022) used deformable convolution backbone R-CNN and a modified IoU loss function to capture tables. Single-stage object detection models employ regression to obtain the predefined anchor location and classification results directly from the network, Huang et al. (2019) conducted anchor optimization on YOLO v3 and added two post-processing methods also on the table detection task. Traquair, Kara, Kantarci, and Khan (2019) adopted pretrained Faster-RCNN and RetinaNet as backbone networks and Feature Pyramid Network as scale-independent feature extractors to discover tabular objects from electronic component datasheets.

The semantic segmentation methods, broadly used in historical and scanned documents, train end-to-end networks to predict the label of every pixel in the image. Neighbor pixels with the same label are detected as figure/table instances. Studies deploying this strategy differ in how the specific models are constructed. Shelhamer, Jonathan, and Trevor (2017) transferred pretrained classification networks into fully convolutional networks to make the per-pixel classification. Chen, Seuret, Liwicki, Hennebert, and Ingold (2015) used convolutional autoencoders and SVM classifiers to label pixels in the historical document into the periphery, background, text block, and decoration. Augusto Borges Oliveira and Palhares Viana (2017) proposed a one-dimensional approach for document layout analysis considering the text, figures, and tables based on CNN. Mechi, Mehri, Ingold, and Amara (2019) modified U-Net architecture to extract text lines from historical documents. Kavasidis et al. (2019) applied dilated convolutions in the segmentation network and introduced the CRF model to enhance the prediction. Tang et al. (2022) used line segment detection and merging algorithms to detect the triangle coordinate diagrams. Ma et al. (2021) proposed a lightweight $L-E^3$ Net to process non-Manhattan layout documents, and achieved 0.79 F1 on DSSE-200 dataset (200 images) and 0.70 F1 on FPD dataset (66 images). Liu, Si, Jin, Shen, and Hu (2020) attempted to adopt instance segmentation models to detect figures and captions. Wu, Hu, Du, Yang, and He (2021) designed Dynamic Residual Fusion Network to address training issues on limited data and got 89.5% F1 on DSSE-200. Li et al. (2021) tried to introduce the idea of metric learning and few-shot to this task and developed a novel regularization method called FS-PARN. The experimental results reached mean-IoU scores of 28.8% and 31.7% on DSSE-200 for 1-shot and 5-shot, respectively.

A frequently reported problem was the detection error in the boundary area of figures or tables (Agarwal et al., 2021; He, Cohen, Price, Kifer, & Giles, 2017; Huang et al., 2019). This problem has an adverse effect on the performance of downstream tasks because they require the input to be in its entirety. The possible reasons can be summarized in two aspects. The first comes from the data's inherent characteristics. Melinda and Bhagvati (2019) suggested that because commonly seen open tables and figures have no explicit borders, it is difficult to detect them. The second aspect comes from the model training strategy and objectives. According to Tychsen-Smith and Petersson (2018), most object detection methods use IoU-based loss function, and they produce a fairly precise target location but not an absolutely accurate bounding box. Rezatofighi et al. (2019) also indicated a disparity between optimizing bounding box position and maximizing the IoU or its variation's metric value, as IoU focuses on overlap areas rather than the scale. Several studies, such as GIoU (Rezatofighi et al., 2019), Distance-IoU (Zheng et al., 2020), and Complete-IoU (Zheng et al., 2021), have enhanced the IoU, but they have not altered this fundamental concept.

In addition to the fact that the loss function used by the existing models is inappropriate for this scenario, the detection metrics nowadays fail to reflect this inaccuracy. In the ICDAR 2013 competition, completeness and purity were used to evaluate the detection region in the table detection competition. A correctly detected region was defined as one that includes all sub-objects in the GT region and none of the sub-objects of other objects (Göbel, Hassan, Oro, & Orsi, 2013). Due to the computational complexity of this evaluation, manual intervention is often required. With the rise in the popularity of deep learning approaches, the evaluation method has shifted to using IoU-based metrics. For example, precision, recall, and F1 scores were calculated with several IoU thresholds in the ICDAR 2019 competition on table detection (Gao et al., 2019). Therefore, there is a gap between the current commonly used detection metrics and the actual requirements of downstream applications.

Based on the previous work, we propose a semantic segmentation and contour recognition cascading detection method. To address the detection error in the boundary area, we defined a loss function for enhancing the figures/tables' edges. In order to bridge the gap between detection testing and practical application requirements, we apply appropriate metrics to evaluate detection results during the model training and prediction phases.

3. Proposed method

3.1. Workflow of the method

The most common drawback of existing methods is the detection error on the edge portion of figures/tables. The two main reasons are (a) the absence of explicit boundaries on available figures/tables and (b) the gap between IoU-based evaluation metrics and detection entirety. We propose an effective figure and table detection method to address these issues. This method uses a semantic segmentation model to classify each pixel in the literature-rendered image into three classes: background, figure, and table. A loss function that assigns a higher penalty to the edge part, called Boundary Enhance Loss, is proposed to train the segmentation model. We also designed a novel segmentation evaluation metric called Categorized Dice efficient for the imbalance classification problem. This metric is used to choose the best performance model weight in the evaluation process. The detection result is given by applying a contour detection algorithm to the segmentation result. Fig. 2 shows the workflow of the proposed method in this paper.

3.2. Segmentation model

We use Attention U-Net (Oktay et al., 2018) as the segmentation model. The model contains an encoder consisting of 4 downsample convolution layers and a decoder consisting of 4 upsample convolution layers. Attention Gates are added to the skip connection to suppress irrelevant regions in the input image while highlighting salient features useful for a specific task. The model receives an $H \times W$ sized image rendered from a single page in scientific literature as input. The output is a $R^{H \times W}$ matrix indicating the classification of each pixel inside the background, figure, or table. In this research, we maintain the network configuration in the Attention U-Net but alter the loss function to the proposed Boundary Enhance loss. We then utilize Categorized Dice to select the best-performing model during the evaluation stage.

3.3. Boundary enhance loss

Previous research has discovered that the model using position-independent loss functions suffered from misclassification around the figures and tables. Inspired by Kervadec et al. (2019) and Caliva, Iriondo, Martinez, Majumdar, and Pedoia (2019), we propose a position-related loss function called Boundary Enhance Loss to boost classification performance. It emphasizes the boundary area by more heavily penalizing classification errors. The difference between our loss function and mentioned research is that we do not apply a heavier penalty to a specific edge line like (Kervadec et al., 2019), nor a position-related penalty on the whole image like (Caliva et al., 2019) but to the boundary margin of figures and tables. There are two considerations for this design. First, unlike medical images, available figures and tables in scientific literature need more unambiguous borders, making it hard for annotators (human or machine) to draw an exact bounding box. Second, from the perspective of downstream applications, a figure or table input can be regarded as valid if it reaches the standard of completeness and purity defined in Göbel et al. (2013). Therefore, the bounding box can be expanded or contracted to a certain extent. For those reasons, our loss function exclusively imposes a higher penalty on the margin area.

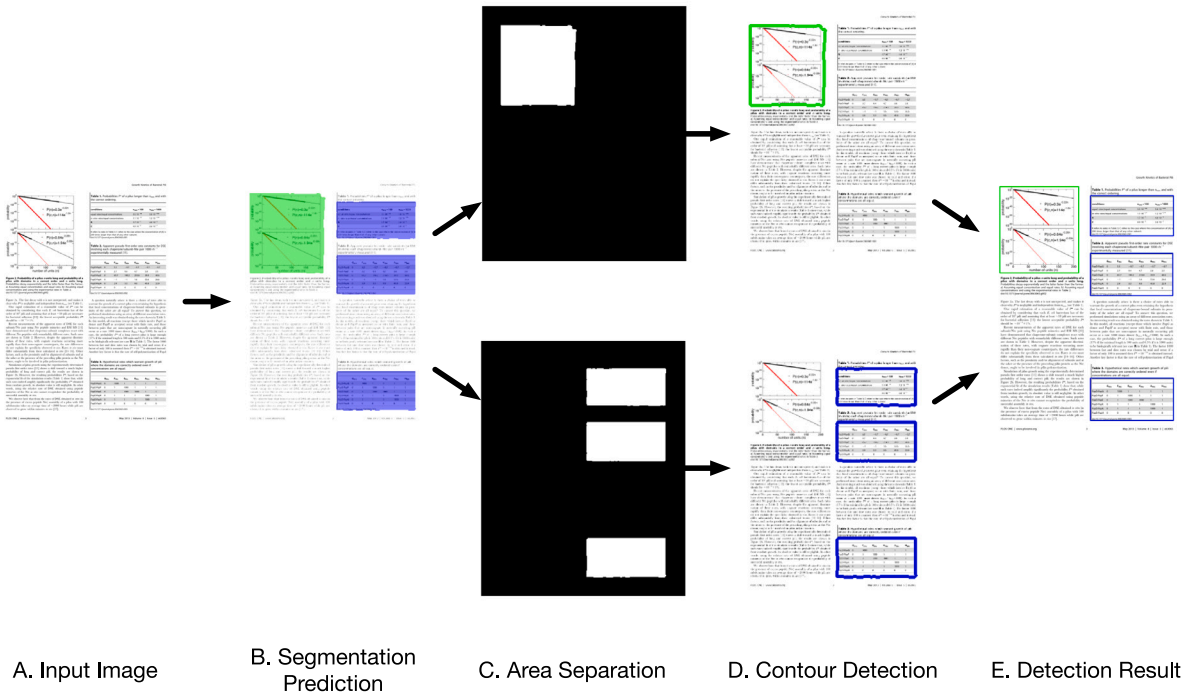


Fig. 2. The workflow of our method. The input is a rendered image of academic literature (Sub-figure A). The classification results are first obtained for each pixel by the attention U-Net model (Sub-figure B). Then the contours of the figure and table area are calculated separately (Sub-figure C and D). Finally, the vertex coordinates of the smallest bounding rectangles of both types are given as the detection result (Sub-figure E). The figure and table are shown in green and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The Boundary Enhance Loss for each rendered image is calculated based on its annotation result. The boundary margin area is defined by the inner and outer margins of the annotated bounding box. Two margins are calculated using a distance transform and an inverse distance transform. These two transforms require the input image to be binary, so a render page size matrix is constructed, and all pixels in the figure or table are set to 1 based on the annotation result, while background pixels are set to 0. The distance transform (as described in Eq. (1)) calculates the Euclidean distance of all background pixels to the nearest pixel on the annotated bounding box (Fabbri, Costa, Torelli, & Bruno, 2008). Additionally, the result is normalized to the image diagonal length to remove bias induced by fluctuating image sizes. The output indicates the distance between each outside pixel and the closest annotation boundary (as shown in Fig. 3B). In the inverse distance transform, the object and background pixel values are inverted; the same process is conducted. The resulting matrix $IDT(B)[x]$, which is visualized in Fig. 3C, is used to describe the relative distance between the inside pixel of figures and tables and the boundary.

$$DT(B)[x] = \min_{y \in B} \text{dist}(x, y) \tag{1}$$

Where x denotes the inside point of figures and tables (value 1), and y denotes the points on the annotated boundary in the input image.

A pixel-wise maximum operation (Eq. (2)) is conducted on these two matrices to combine the inner and outer margins into a single unified one, resulting in a new boundary distance matrix (as illustrated in Fig. 3D), which defines the shortest distance from every pixel to the annotation bounding box. To emphasize the importance of the boundary and control the margin width, elements in the boundary distance matrix lower than 0.02 are set to 5, while the rest are assigned to 1, as shown in Fig. 3E. Then the loss function in our model is a pixel-wise weighted cross entropy, as described in Eq. (3).

$$BE(B)[x] = \max(DT(B)[x], IDT(B)[x]) \tag{2}$$

$$loss = \sum_{x \in \Omega} BE(B)[x] \log\left(P_{l(x)}(x)\right) \tag{3}$$

3.4. Categorized dice

In order to choose the best performance model weights in the evaluation phase of different training epochs, a proper evaluation metric is needed. A logical question is whether the loss function proposed in this paper can be used as an evaluation metric. In the

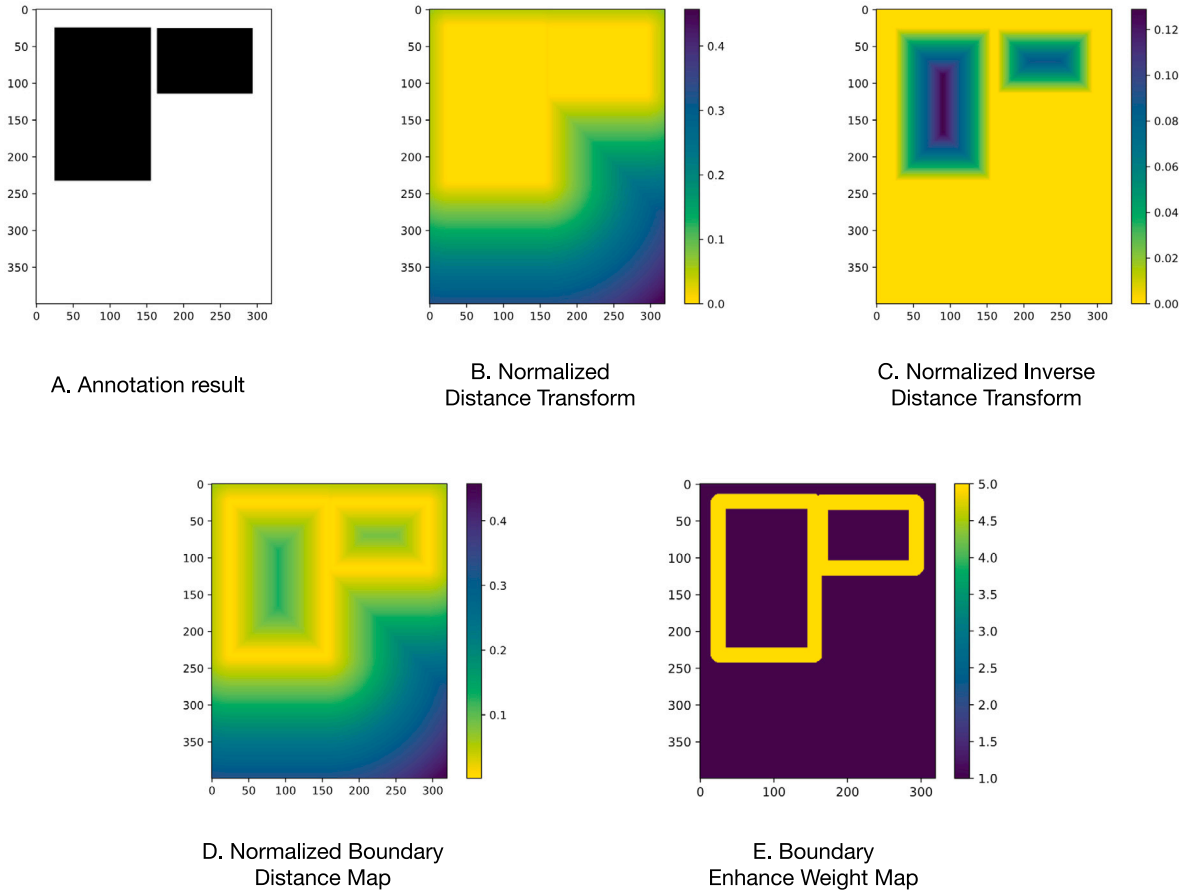


Fig. 3. The boundary enhance loss calculation process.

workflow of our method, a contour detection algorithm is applied to the segmentation result to determine the position of figures and tables. The proposed loss function evaluates classification in a position-sensitive fashion while the following contour detection algorithm equally weights every pixel in the segmentation result. So the proposed loss function is inappropriate. A further question is whether the Dice Coefficient is proper. The Dice Coefficient is a widely used region- and category-independent evaluation metric for semantic segmentation. It quantifies the similarity between prediction and ground truth segments. Shamir, Duchin, Kim, Sapiro, and Harel (2019) suggested Dice Coefficient is directly related to target structure size. The smaller the target (given a fixed resolution), the less sensitive the Dice Coefficient is. In this task, background points occupy most images, and the figure and table areas are relatively small. So the dice coefficient is insufficient to distinguish different errors when all pixels are evaluated indiscriminately.

We propose a category-related dice coefficient in this paper called Categorized Dice. As indicated in Eq. (4), the Categorized Dice computes the dice coefficients for the background, figure, and table pixels separately and then linearly combines them to obtain the global semantic segmentation evaluation. Considering the share of these three categories in the image, we set the figure, table, and background weights as 0.4, 0.4, and 0.2, aiming to reflect the classification results of the figure and table pixels during model training more effectively.

$$Categorized\ \ Dice = \sum_{c} \alpha_c \frac{2\left|X_c \cap Y_c\right|}{\left|X_c\right| + \left|Y_c\right|} \tag{4}$$

Fig. 4 compares the Dice coefficient and Categorized Dice coefficient trends during the model evaluation phase. Because the proportion of background pixels is substantial, the Dice coefficient is initially elevated (around 0.947) and then gradually increases to 0.985. On the other hand, the Categorized Dice coefficient begins from a lower point (around 0.865) due to the model's lack of task knowledge. As training advances, the coefficients of Dice and background pixels follow a nearly identical trajectory. However, the Categorized Dice coefficient incorporates three types of pixels. It compensates for the poor representation in Dice by assigning greater weights to non-dominance figure-type and table-type pixels. What stands out is that when the model performs poorly for categorizing table pixels at step 54 K, the Categorized Dice coefficient responds better to the issue than the Dice coefficient.

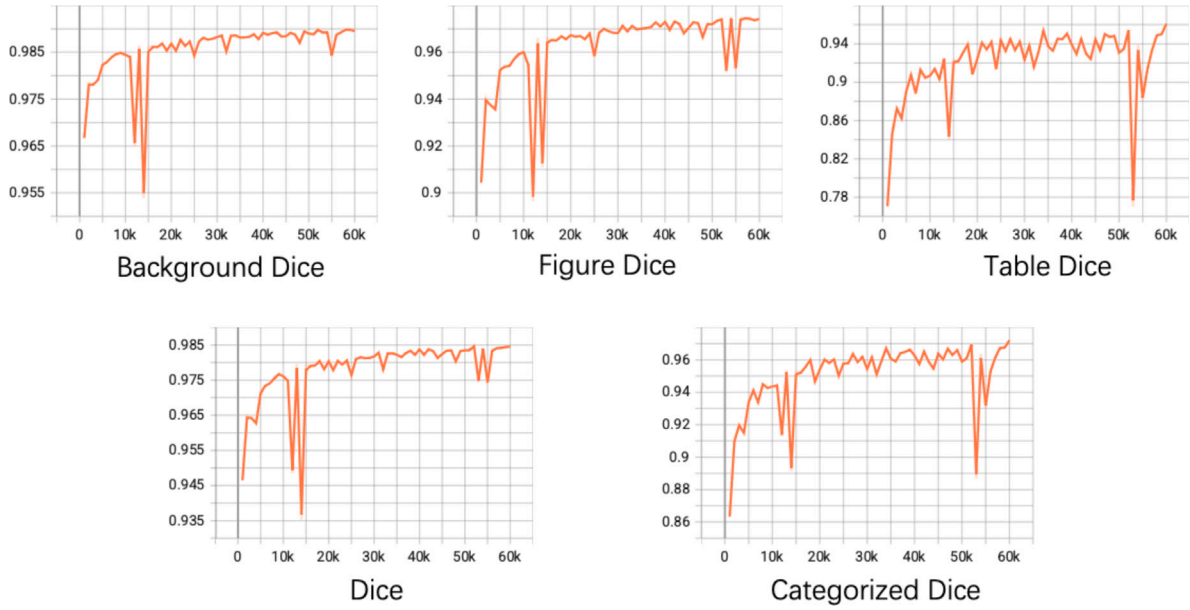


Fig. 4. We compare the proposed Categorized Dice with the original Dice in the model evaluation. The x-coordinate is the number of training steps, and the y-coordinate is the corresponding dice coefficient value. The first row plots the dice coefficients against the training steps calculated separately for the background, picture, and table categories. The left side of the second row shows the result of indiscriminately computing dice coefficients for all pixels. The right side of the second row illustrates the results of the Categorized Dice coefficients proposed in this paper, which assign higher weights to figure/table pixels and more accurately depict the model detection performance change as training steps increase.

3.5. Localization method

A contour detection algorithm is adopted on the segmentation result to obtain the vertex coordinates of the figure and table. We split one segmentation prediction image into separate binary images for the figure and table, so the figures and tables can be detected separately. As seen in Fig. 2C, the separated images only consist of the white foreground (figure or table) and black background. Noise reduction methods like expansion and erosion are utilized. Then, the canny edge detection algorithm recognizes irregular contours of the figure and table region, as shown in Fig. 2D. Finally, the vertices coordinates of the smallest horizontal bounding rectangle for the contours are given as the position coordinates of the detected figure or table, as shown in Fig. 2E.

4. Experiment

4.1. Dataset

The dataset used in this study consisted of 14,678 papers with figures and tables. They were randomly picked from the PubMed Open Access database. Figures are more prevalent than tables in the medical literature. According to our statistics, the number of figures outnumbers the number of tables by a factor of 6.5. To address the data imbalance issue in the PubMed OA dataset, we included 2966 randomly selected annotated data from the TableBank dataset (Li, Cui, et al., 2019).

An individual article package in the PubMed OA dataset includes metadata in XML, the article PDF, and the media files. A data annotation method similar to Siegel, Lourie, Power, and Ammar (2018) was applied. However, we used a SIFT-based image feature matching algorithm on the figure/table image files and PDF-rendered images, rather than the pyramid template matching algorithm in the original paper, for higher annotation accuracy. All annotated data was manually checked to ensure the training material fed to the proposed model was correct. The annotated data was partitioned into the training, validation, and test sets of images: 24,045, 4517, and 4517, respectively.

4.2. Experimental details

The experiment was conducted in the PyTorch 1.7.1 environment with an NVIDIA GeForce RTX 3090Ti GPU. All PDF literature was rendered into 400 × 320 pixels images page by page. Annotation results were saved as vertex coordinates and their types. They were converted to the form required by the proposed and compared models.

In the training process of the proposed method, the Attention U-net network was randomly initialized, and the mini-batch was set to 4. An RMSprop optimizer with a learning rate of 0.0001 and weight decay of 10^{-8} was used. We have trained our model for 10 epochs. A validation test on the validation set was conducted, and the best-performance weight was saved for every 1000 iterations.

Table 2
Evaluation of the boundary enhance loss and categorized dice proposed in this research.

Method	mAP IoU = 0.9	mAP IoU = 0.95
UNet	0.717	0.473
UNet+B	0.718	0.491
UNet+C	0.704	0.475
UNet+B+C	**0.745**	**0.508**

The loss functions were Cross Entropy Loss and proposed Boundary Enhance Loss. Validation metrics were the Dice coefficient and proposed Categorized Dice.

4.3. Evaluation standard and metric

As previously stated, there is a gap between the IoU-based metric and downstream application requirements, like Fig. 1B. It is particularly true when the IoU threshold values in earlier research were set to 0.5 and 0.8. Unless the IoU threshold is set to 1.0, increasing it does not ensure the purity and completeness of prediction results. So IoU or IoU-based AP and mAP are not suitable as the primary metrics to test the model's performance for downstream application scenarios.

In this experiment, apart from IoU or mAP, we adopt page-level precision, recall, and F1 metric to evaluate the performance of different algorithms. A detection for an individual page is considered correct when all object-level predictions on that page are valid. This page will be considered incorrect or missing if any object detection is invalid or not detected. An object-level prediction is valid when it satisfies completeness and purity. The standard of completeness and purity were defined in Göbel et al. (2013). We recall them for convenience. An object prediction is classified as complete if it includes all sub-objects in the GT region. An object prediction is classified as pure if it does not include any sub-objects which are not also in the GT region. The completeness and purity are checked manually.

The reason for using page-level instead of figure/table-level like in previous studies is to follow the practical application standard. Because in a real-world application, if a detection error occurs on one page, the page must be manually rediscovered, and the figure or table on it must be manually located. To simulate this usage scenario, we adopt this rigorous evaluation criterion.

4.4. Ablation experiment

To validate the impacts of the Boundary Enhance loss and Categorized Dice proposed in this study, we compare the original Attention U-Net model with the model containing both. The mean average precision under two IoU thresholds (0.9 and 0.95) is tested on the test set data (described in Section 4.1) with four different model setups.

In Table 2, +C denotes the addition of Categorized Dice, whereas +B denotes the addition of Boundary Enhance Loss. The results in Table 2 demonstrate that utilizing Categorized Dice or Boundary Enhance loss alone has a minimal effect on the model performance of high-precision figure and table detection. However, incorporating both into the Attention U-Net model can considerably enhance the performance of our method. Subsequent comparison tests will incorporate the experimental data obtained under this paradigm.

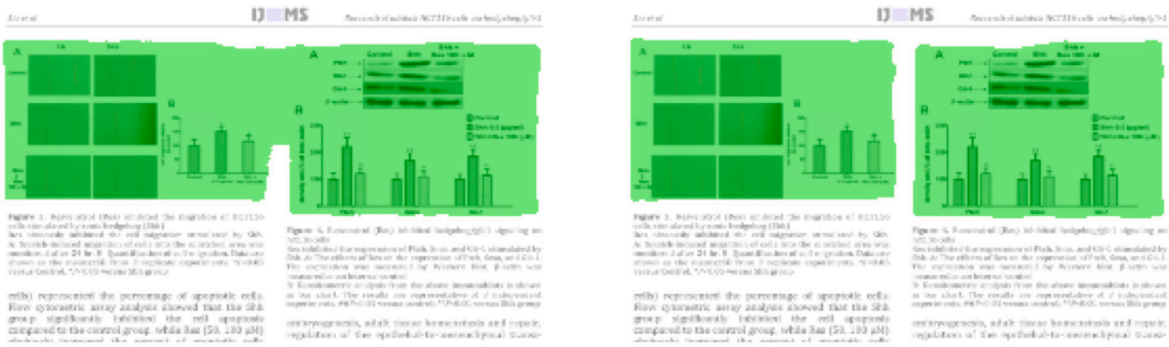
We also compare segmentation results from Attention U-Net with different loss functions to clearly observe the effect of the Boundary Enhance loss function. In Fig. 5, pixels between two adjacent figures are mistakenly classified under the Cross-Entropy loss function. While in the prediction under the proposed Boundary Enhance loss function, two figure areas are correctly separated.

4.5. Comparing experiment

We compare the performance of the proposed method to the state-of-the-art figure and table detection algorithms in a real-world figure and table detection scenario. Comparing approaches include a heuristic-based algorithm PDFFigures 2.0 (Clark & Divvala, 2016), a single-stage object detection algorithm YOLO v3 (Huang et al., 2019), and a two-stage object detection algorithm Faster RCNN (Sun et al., 2019). We use four algorithms to predict the locations of figures and tables on the test set data (described in Section 4.1) and calculate the corresponding mAP and page-level metrics (described in Section 4.3).

The training details are as follows. For YOLO v3, the Adam optimizer with a learning rate of 0.001 is adopted. Input images were resized to 416×416. The batch size was set to 16 for 100 epochs. For Faster RCNN, the SGD optimizer with a learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0005 was applied. Input images were resized to a random width and height between 600 to 1000. The batch size was set to 1 for 14 epochs. Both use the dataset in Section 4.1 and are trained from scratch.

Table 3 reports the results of four methods on mAP metric at different IoU thresholds, which is the evaluation metric in most similar studies. Table 4 shows the page-level statistics results of the four methods. The standard of correct, incorrect, and missing are defined in Section 4.3. The total number of PDFFigures 2.0 is less than others because it accepted PDF files as input, and 315 PDF pages were identified as invalid input by the program. Of the 4202 valid inputs, no detection was reported on 1344 PDF pages. The results in the table are for the rest of the 2858 pages we manually checked. One page cannot be handled in YOLO. Corresponding page-level precision, recall, and F1 are shown in Table 5. It should be noted that the results of PDFFigures 2.0 are calculated excluding the 315 unprocessable inputs.



A. Prediction under Cross Entropy Loss

B. Prediction under Boundary Enhance Loss

Fig. 5. A segmentation result comparison between models with different loss functions.

Table 3
Detection results on mAP metrics under different IoU thresholds.

Method	mAP IoU = 0.5	mAP IoU = 0.75	mAP IoU = 0.9	mAP 0.5:0.95
PDFFigures 2.0	0.457	0.457	0.417	0.332
Faster RCNN	0.953	0.821	0.325	0.711
YOLO v3	**0.969**	**0.914**	0.359	0.764
Ours	0.913	0.863	**0.745**	**0.824**

Table 4
Statistics of results calculated at page level for different algorithms under practical application criteria.

Method	Correct	Incorrect	Missing
PDFFigures 2.0	2726	**63**	69
Faster RCNN	4032	484	**1**
YOLO v3	3258	968	290
Ours	**4422**	77	18

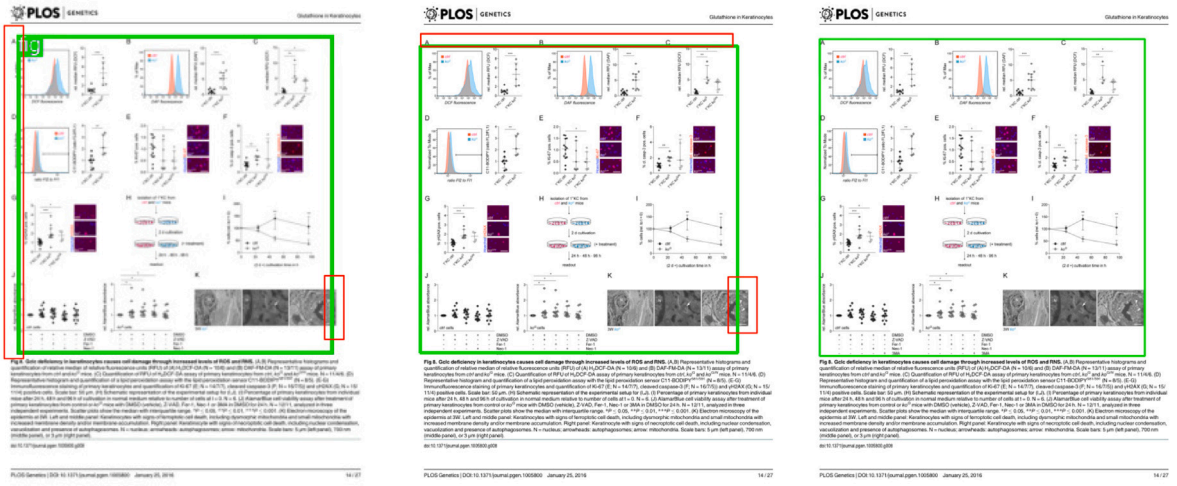
Table 5
Performance comparison of different algorithms under practical application criteria.

Method	Precision	Recall	F1
PDFFigures 2.0	0.954	0.649	0.772
Faster RCNN	0.893	0.893	0.893
YOLO v3	0.771	0.721	0.745
Ours	**0.983**	**0.983**	**0.983**

4.6. Discussion

Gap between IoU and detection entirety Comparing Tables 3 and 5, one can notice the gap between IoU-based metrics and actual application criteria. In Table 3, the IoU threshold influences the performance ranking. In contrast, the performance of the proposed method in real-world circumstances is significantly superior to comparative methods. What is interesting about the data in this table is that YOLO v3 outperforms Faster R-CNN in the mAP metric at all IoU thresholds. However, it is inferior to Faster R-CNN in practical applications. YOLO v3 and PDFFigures 2.0 meets a comparable conclusion. Consequently, IoU-based metrics alone are inadequate for comparing the performance of different models in downstream applications.

Effectiveness of proposed method In large-scale data testing, as shown in Table 3, two object detection methods outperform our method and PDFFigures at lower IoU thresholds (0.5 and 0.75) which are the test condition in most similar studies. However, they drop dramatically in a high detection accuracy condition (IoU = 0.9), which is consistent with the problems reported in several papers. Meanwhile, our method remains high across different IoU thresholds and has a significant advantage in high IoU threshold, indicating that our detection results have better completeness. The page-level results in Table 5 demonstrates that proposed method can produce excellent results under rigorous practical application criteria. The F1 value of 0.983 at the page level indicates that labor and time expenses associated with data cleaning can be decreased significantly in real applications.



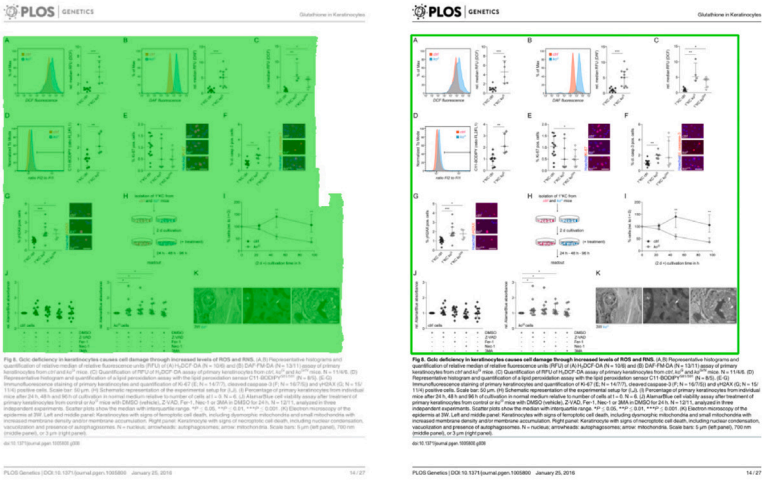


Fig. 6. The comparison of the four algorithms results on one literature page. The green bounding boxes denote detection results. The red rectangles denote the cut-off errors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The main problem with the two object detection algorithms is their inaccuracy in detecting the figure or table boundary regions, which is consistent with the issues reported in prior studies. Fig. 6 is a typical example of the four methods. YOLO v3 and Faster R-CNN miss several letters at the figure's edge, preventing downstream applications from utilizing the detection result. The proposed method achieves a good classification in the edge region so that all figure contents are ultimately included in the result region. The segmentation results show that the three edges of the figure are smooth, and the region without content on the right side is also precisely excluded. This demonstrates the efficacy of the proposed Boundary Enhance Loss, and the segmentation model can identify the content region. One may wonder why the non-content regions in Fig. 5 are not excluded. This is because the area of the figure in Fig. 5 is small. In contrast, the area of the figure in Fig. 6 is much larger, demonstrating that the model has learned the characteristics of the spatial distribution of academic figure content at various scales.

Shortcoming of proposed method Although the detection result of proposed method meet the purity and completeness standard, the accuracy can still be improved. The detection result of PDFFigures 2.0 is even more precise than our model if one zooms in to measure the distance between the bounding box and letter "A" on the left top in Fig. 6. A possible explanation for this

might be that we set a fixed 2% margin width on the Boundary Enhance loss, so the ground truth in training data could be more accurate. On the other hand, the input of the heuristic algorithm is the accurate position of all elements.

Other Observations When further looking at Table 5, we can find that even under rigorous criteria, the heuristic-based algorithm has a relatively good detection precision but suffers from low recall. It can also be beneficial for data cleaning of the downstream applications. Because the program proactively reports most detection errors, time is saved in testing them individually. In addition, this is a clear example demonstrating that literature publications follow specific design rules, and using PDF files as input can be beneficial. Three deep learning methods reach higher recall than the heuristic one, indicating better generalization ability of the deep learning methods.

5. Conclusion and future work

This paper theoretically and experimentally demonstrates a gap between widely used IoU-based detection metrics and the downstream application standard of figure and table detection in academic literature. An academic figure and table detection method that cascades semantic segmentation and contour detection is proposed. Boundary Enhance loss and Categorized Dice are proposed to address the detection issue of existing algorithms at the boundary region. In the experiment under rigorous criteria for downstream practical applications, the proposed method outperforms existing algorithms by a significant margin and reaches a page-level of 0.9833 F1. Our work can provide high-quality input for downstream applications and reduce the workload of data cleaning.

There are limitations of our study. First, the manual examination was applied in the experiment to check the purity and completeness of detection prediction, which was labor and time intensive. Second, the margin width of the proposed Boundary Enhance loss function is set to a fixed 2% of the diagonal length because the positions of neighbor elements are unknown. In future studies of layout analysis, detection types will be expanded to all element types on the page. Then the purity and completeness can be calculated automatically based on the positions of neighbor elements. Moreover, the margin width can be set dynamically, which is expected to improve the robustness of the model.

CRediT authorship contribution statement

Fengchang Yu: Conceptualization, Methodology, Funding acquisition. **Jiani Huang:** Data Curation, Writing – original draft. **Zhuoran Luo:** Visualization, Validation. **Li Zhang:** Software, Validation. **Wei Lu:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the 2021 Hubei Province Postdoctoral Innovation Research Positions Program.

References

Agarwal, M., Mondal, A., & Jawahar, C. (2021). Cdec-net: Composite deformable cascade network for table detection in document images. In *2020 25th international conference on pattern recognition* (pp. 9491–9498). IEEE.

Ajij, M., Pratihar, S., Roy, D. S., & Hanne, T. (2022). Robust detection of tables in documents using scores from table cell cores. *SN Computer Science*, *3*(2), 1–19.

Augusto Borges Oliveira, D., & Palhares Viana, M. (2017). Fast CNN-based document layout analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 1173–1180).

Bhatia, S., & Mitra, P. (2012). Summarizing figures, tables, and algorithms in scientific publications to augment search results. *ACM Transactions on Information Systems (TOIS)*, *30*(1), 1–24.

Caliva, F., Iriondo, C., Martinez, A. M., Majumdar, S., & Pedoia, V. (2019). Distance map loss penalty term for semantic segmentation. arXiv preprint arXiv:1908.03679.

Chen, K. C., Lee, C. C., Lin, M. P. H., Wang, Y. J., & Chen, Y. T. (2021). Massive figure extraction and classification in electronic component datasheets for accelerating PCB design preparation. In *2021 ACM/IEEE 3rd workshop on machine learning for CAD* (pp. 1–6). IEEE.

Chen, K., Seuret, M., Liwicki, M., Hennebert, J., & Ingold, R. (2015). Page segmentation of historical document images with convolutional autoencoders. In *2015 13th international conference on document analysis and recognition* (pp. 1011–1015). IEEE.

Choudhury, S. R., Tuarob, S., Mitra, P., Rokach, L., Kirk, A., Szep, S., et al. (2013). A figure search engine architecture for a chemistry digital library. In *Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries* (pp. 369–370).

Clark, C., & Divvala, S. (2016). Pdffigures 2.0: Mining figures from research papers. In *2016 IEEE/ACM joint conference on digital libraries* (pp. 143–152). IEEE.

Corrêa, A. S., & Zander, P. O. (2017). Unleashing tabular content to open data: A survey on pdf table extraction methods and tools. In *Proceedings of the 18th annual international conference on digital government research* (pp. 54–63).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Fabbri, R., Costa, L. D. F., Torelli, J. C., & Bruno, O. M. (2008). 2D Euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, *40*(1), 1–44.

Fernandes, J., Simsek, M., Kantarci, B., & Khan, S. (2022). TableDet: An end-to-end deep learning approach for table detection and table image classification in data sheet images. *Neurocomputing*, *468*, 317–334.

Gao, L., Huang, Y., Déjean, H., Meunier, J. L., Yan, Q., Fang, Y., et al. (2019). ICDAR 2019 competition on table detection and recognition (cTDaR). In *2019 international conference on document analysis and recognition* (pp. 1510–1515). IEEE.

Göbel, M., Hassan, T., Oro, E., & Orsi, G. (2013). ICDAR 2013 table competition. In *2013 12th international conference on document analysis and recognition* (pp. 1449–1453). IEEE.

He, D., Cohen, S., Price, B., Kifer, D., & Giles, C. L. (2017). Multi-scale multi-task fcn for semantic page segmentation and table detection. In *2017 14th IAPR international conference on document analysis and recognition, vol. 1* (pp. 254–261). IEEE.

Huang, Y., Yan, Q., Li, Y., Chen, Y., Wang, X., Gao, L., et al. (2019). A YOLO-based table detection method. In *2019 international conference on document analysis and recognition* (pp. 813–818). IEEE.

Jimeno Yepes, A., Zhong, P., & Burdick, D. (2021). ICDAR 2021 competition on scientific literature parsing. In *International conference on document analysis and recognition* (pp. 605–617). Springer.

Kavasidis, I., Pino, C., Palazzo, S., Rundo, F., Giordano, D., Messina, P., et al. (2019). A saliency-based convolutional neural network for table and chart detection in digitized documents. In *International conference on image analysis and processing* (pp. 292–302). Springer.

Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., & Ayed, I. B. (2019). Boundary loss for highly unbalanced segmentation. In *International conference on medical imaging with deep learning* (pp. 285–296). PMLR.

Lebourgeois, F., Bublinski, Z., & Emptoz, H. (1992). A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *11th IAPR international conference on pattern recognition. Vol. II. Conference B: Pattern recognition methodology and systems, vol. 1* (pp. 272–273). IEEE Computer Society.

Lee, S. H., & Chen, H. C. (2021). U-SSD: Improved SSD based on U-Net architecture for end-to-end table detection in document images. *Applied Sciences*, *11*(23), 11446.

Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., & Li, Z. (2019). Tablebank: A benchmark dataset for table detection and recognition. arXiv preprint arXiv:1903.01949.

Li, P., Jiang, X., & Shatkay, H. (2019). Figure and caption extraction from biomedical documents. *Bioinformatics*, *35*(21), 4381–4388.

Li, Y., Zhang, P., Xu, X., Lai, Y., Shen, F., Chen, L., et al. (2021). Few-shot prototype alignment regularization network for document image layout segementation. *Pattern Recognition*, *115*, Article 107882.

Lin, T. Y., et al. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Springer.

Liu, Y., Si, C., Jin, K., Shen, T., & Hu, M. (2020). FCENet: An instance segmentation model for extracting figures and captions from material documents. *IEEE Access*, *9*, 551–564.

Ma, T., Wu, X., Li, X., Du, X., Zhou, Z., Xue, L., et al. (2021). Document layout analysis with aesthetic-guided image augmentation. arXiv preprint arXiv:2111.13809.

Mechi, O., Mehri, M., Ingold, R., & Amara, N. E. B. (2019). Text line segmentation in historical document images using an adaptive U-Net architecture. In *2019 international conference on document analysis and recognition* (pp. 369–374). IEEE.

Melinda, L., & Bhagvati, C. (2019). Parameter-free table detection method. In *2019 international conference on document analysis and recognition* (pp. 454–460). IEEE.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., et al. (2018). Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.

Perez-Arriaga, M. O., Estrada, T., & Abad-Mota, S. (2016). TAO: system for table detection and extraction from PDF documents. In *The twenty-ninth international flairs conference*.

Poco, J., & Heer, J. (2017). Reverse-engineering visualizations: recovering visual encodings from chart images. *Computer Graphics Forum*, *36*(3), 353–363.

Praczyk, P. A., & Nogueras-Iso, J. (2013). Automatic extraction of figures from scientific publications in high-energy physics. *Information Technology and Libraries*, *32*(4), 25.

Ray Choudhury, S., Mitra, P., & Giles, C. L. (2015). Automatic extraction of figures from scholarly documents. In *Proceedings of the 2015 ACM symposium on document engineering* (pp. 47–50).

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658–666).

Saha, R., Mondal, A., & Jawahar, C. (2019). Graphical object detection in document images. In *2019 international conference on document analysis and recognition* (pp. 51–58). IEEE.

Shamir, R. R., Duchin, Y., Kim, J., Sapiro, G., & Harel, N. (2019). Continuous dice coefficient: a method for evaluating probabilistic segmentations. arXiv preprint arXiv:1906.11031.

Shelhamer, E., Jonathan, L., & Trevor, D. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 640–651.

Siddiqui, S. A., Malik, M. I., Agne, S., Dengel, A., & Ahmed, S. (2018). Decnt: Deep deformable cnn for table detection. *IEEE Access*, *6*, 74151–74161.

Siegel, N., Lourie, N., Power, R., & Ammar, W. (2018). Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries* (pp. 223–232).

Srihari, S. N., Lam, S. W., Govindaraju, V., Srihari, R. K., & Hull, J. J. (1986). Document image understanding. In *FJCC* (pp. 87–95). Citeseer.

Sun, N., Zhu, Y., & Hu, X. (2019). Faster R-CNN based table detection combining corner locating. In *2019 international conference on document analysis and recognition* (pp. 1314–1319). IEEE.

Tang, B., Jiang, J., Xu, X., Qi, L., Zhou, X., & Chen, Y. (2022). Triangle coordinate diagram localization for academic literature based on line segment detection in cloud computing. In *International conference on cloud computing* (pp. 47–59). Springer.

Traquair, M., Kara, E., Kantarci, B., & Khan, S. (2019). Deep learning for the detection of tabular information from electronic component datasheets. In *2019 IEEE symposium on computers and communications* (pp. 1–6). IEEE.

Tychsen-Smith, L., & Petersson, L. (2018). Improving object localization with fitness nms and bounded iou loss. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6877–6885).

Wu, X., Hu, Z., Du, X., Yang, J., & He, L. (2021). Document layout analysis via dynamic residual feature fusion. In *2021 IEEE international conference on multimedia and expo* (pp. 1–6). IEEE.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07* (pp. 12993–13000).

Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., et al. (2021). Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*.