IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 38, No. 1, February, 2024, pp. 115-125
Journal homepage: http://iieta.org/journals/ria

Geometric Deep Learning for Enhancing Irregular Scene Text Detection

Madhuri Aluri[1,2*], Uma Devi Tatavarthi[1]

[1] Department of Computer Science, GITAM (Deemed to be University), Visakhapatnam 530045, India
[2] Department of Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada 520007, Andhra Pradesh, India

Corresponding Author Email: madhuria@pvpsiddhartha.ac.in

https://doi.org/10.18280/ria.380112

Received: 31 October 2023
Revised: 25 November 2023
Accepted: 28 December 2023
Available online: 29 February 2024

Keywords:
image text detection, scene image; irregular, relation inference, graph convolution network

ABSTRACT

Text detection in natural scene images presents significant challenges, particularly in detecting irregular shapes. As a result of the limited receptive field of CNNs, existing methods have difficulty capturing long-range relationships between distant component regions. This study introduces an innovative method for identifying irregular text in images of natural scenes. The approach utilizes a U-net architecture combined with connected component analysis, resulting in improved accuracy in detecting text components and reducing the identification of non-character text components. Additionally, our strategy incorporates the use of graph convolution networks (GCN) to deduce adjacency relations among text components. The integration of GCNs introduces a sophisticated mechanism for inferring adjacency relations, contributing significantly to the advancement of text detection in natural scene images. Our method's efficacy is showcased through experimental assessments on three publicly available datasets: "ICDAR2013," "CTW-1500," and "MSRA-TD500."

1. INTRODUCTION

Irregular Text detection [1] has been a prominent research area in computer vision and image processing in recent years. The task of automatically locating and extracting text from images has gained significant attention and has seen remarkable advancements. This surge in interest is driven by the increasing demand for applications that involve text analysis, such as document understanding, scene recognition, image captioning, augmented reality, and text-based information retrieval. In natural scene images, text can take on various irregular or arbitrary shapes that differ from the standardized and structured text typically found in documents or books. Unlike conventional text, which is typically aligned horizontally and follows a consistent layout, irregular text in natural scenes can be distorted, curved, skewed, rotated, or have other deformations.

Many factors can contribute to irregular text shapes in natural scene images. First, the text may be present on objects with non-planar surfaces, such as street signs, product packaging, or vehicle number plates. These objects can have varying shapes and orientations, leading to text that conforms to the object's contours or perspectives. Second, the presence of text in natural scenes can be influenced by environmental factors and human interventions. For example, graffiti or handwritten messages may appear on walls, sidewalks, or public spaces, adopting unique shapes and styles. Similarly, text displayed on billboards, banners, or advertisements may undergo distortions due to wind, wear and tear, or artistic choices. Moreover, irregular text shapes can also emerge from the inherent characteristics of the scene itself. In landscapes or scenic images, text may be incorporated into the natural elements or structures, such as text carved on rocks, etched on trees, or integrated into architectural designs. Natural scene images often contain complex backgrounds and varying lighting conditions. This adds to the challenge of detecting irregular text shapes as the text may blend into the background or be affected by shadows, occlusions, or other visual distractions. Overcoming these challenges requires robust algorithms and models specifically designed for detecting irregular text shapes.

Irregular text shapes [2] can be found in different languages and scripts. The ability to accurately identify irregular text shapes across multiple languages and scripts is essential for applications involving multilingual environments, such as international signage recognition, translation services, or cross-lingual information retrieval.

It is crucial for several applications to be able to detect and understand irregular text shapes in natural scene images. It enables tasks such as scene understanding, visual content analysis, autonomous driving, text-based information retrieval, augmented reality, and more. Researchers and practitioners strive to develop robust computer vision algorithms and models capable of effectively handling these irregular text shapes, contributing to advancements in various fields.

GNN-based irregular text detection methods have shown promising results in handling complex text layouts, such as curved or perspective text, text in cluttered backgrounds, or text with various orientations. By capturing the structural dependencies among text components, GNNs can effectively

model the contextual information necessary for accurate detection and localization of irregular text.

Hence in this we are aggregating both the transformer and GNN based methods for irregular text detection.

We have made three main contributions:

• Utilizing the U-Net architecture, we perform feature extraction, and character center point estimation is achieved through connected component analysis.

• We represent each text region as a node and employ Graph Neural Networks (GNNs) to build a local inference graph.

• The integration of the inference graph and the deep relational inference network enhances our ability to comprehend the relationships and interactions among character text components in a more holistic manner. Here is the structure of the research. Section 2 examines various papers on text detection. Section 3 presents the proposed architecture for scene text detection. Section 4 entails an experimental analysis on datasets, along with the evaluation metrics for the proposed methodology. The paper concludes with a final section summarizing the findings.

2. RELATED WORK

Irregular text detection [3-6], which focuses on identifying text in non-standard or non-horizontal orientations, has gained significant attention in recent years. This challenging task has been tackled in various ways by researchers. Here are some notable methods:

(1) **Stroke Width Transform (SWT)**: The SWT algorithm identifies text regions based on the variations in stroke width. It detects regions where the stroke width is relatively constant, which is indicative of text, and distinguishes them from non-text regions.

(2) **Connected Component Analysis:** This approach segments the image into connected components and analyzes their properties to identify irregular text. By considering attributes like aspect ratio, height, or geometric relationships between components, irregular text regions can be detected.

(3) **Hough Transform:** The Hough Transform is a widely used technique for detecting lines and shapes in images. By applying the Hough Transform specifically for text detection, researchers have successfully identified irregular text by detecting lines or curves that represent the text shape.

(4) **Deep Learning Based Methods:** The development of deep learning has contributed to the development of CNNs that can detect irregular texts [7-10]. These models are trained on annotated datasets to learn the complex patterns and characteristics of irregular text, enabling them to accurately detect and localize such text in images.

(5) **Graph Neural Networks (GNN**): GNNs have also been utilized for irregular text detection tasks [11]. By representing the image as a graph and leveraging the graph structure, GNNs can capture the relationships between text elements and effectively identify irregular text regions.

(6) **Hybrid Approaches:** Some methods combine multiple techniques to improve irregular text detection.

In particularly, Transformers are neural network models that excel in capturing long-range dependencies and modeling contextual information. Transformer-based methods have emerged as powerful techniques for irregular text detection [12, 13], leveraging their ability to capture long-range dependencies and handle complex spatial relationships. These methods have shown promising results in accurately identifying and localizing irregular text regions in images. Here are some notable approaches:

(1) **Mask TextSpotter**: This method combines the Transformer architecture with a mask-based text detection framework. It first generates text proposals using region-based methods and then refines them using a Transformer-based network. By modeling the contextual information and capturing the relationships between text elements, Mask TextSpotter achieves precise irregular text detection.

(2) **Border Detectors**: Border detectors based on Transformers aim to detect the boundaries of irregular text regions. By using the self-attention mechanism of Transformers, irregular text regions of arbitrary shapes [14, 15] can be accurately localized by capturing both local and global context.

(3) **TextPerceiver**: TextPerceiver is a recent approach that combines the Perceiver model, a variant of the Transformer, with a segmentation head. It operates on the entire image and learns to attend to relevant text regions, allowing for effective irregular text detection. The model can adapt to various text shapes and orientations, making it suitable for challenging scenarios.

(4) **TextFuseNet**: CNNs and Transformers are both used in TextFuseNet. It employs a multi-branch architecture where CNNs capture local features, while Transformers model global context for irregular text detection. By fusing the information from different branches, TextFuseNet achieves robust performance in detecting irregular text regions.

(5) **LayoutLM**: Although primarily designed for document layout analysis, LayoutLM, which is based on the Transformer, can also be utilized for irregular text detection. By treating text detection as a sequence labeling task, LayoutLM captures the spatial dependencies of irregular text elements and accurately identifies them.

As a technique for detecting irregular text in images, graph neural networks have emerged as one of the most powerful. As graphs are modeled as edges and nodes in a graph, GNN-based methods use text components to represent images.

Our goal in this article is to provide you with a brief summary of the latest advances in detecting text in images that have arbitrary and irregular shapes [16, 17]. Recent research has been focusing on the detection of text in scenes of various orientations, forms, and layouts. Several studies [18-23] have been published on the topic of detecting irregular text because of the considerable interest in this area of text detection. The following four groups best describe the scope of these investigations.

Regression Based Approaches: Regression-based approaches have been used to identify scene text using text bounding boxes. Various approaches have been developed in this category, such as Textboxes [24], ABCnet [25], EAST [26], and the adaptive boundary proposal network proposed by authors [27].

To handle rectangular candidate boxes with long sliding windows and convolution kernels, textboxes use horizontal text processing and has trouble with text that isn't perfectly rectangular. The output of ABCnet is influenced by the over-reliance on control points in the description of non-rectangular text shapes. EAST is built to provide quick and precise results for text detection in natural settings. Further, a boundary proposal network was developed in the previous paper to help detect arbitrary-shaped text. It produced accurate boundaries without the need for further post-processing.

While these regression-based approaches have shown

promising results in detecting horizontal and multi-oriented text, they may struggle to do so when presented with scene texts that have very wide aspect ratios and are oriented in unexpected ways.

Segmentation Based Approaches: These have emerged as another approach for text detection, relying on classification at pixel-level [28-31]. Text segmentation zones are identified using deep convolutional neural networks, and then the boxes are created using postprocessing. PSEnet [28] presents a progressive scale expansion post-processing approach that greatly enhances detection precision. In contrast, Pixellink [29] overcomes the problem of textual closeness by foreseeing pixel connections between distinct instances of text. According to the study [30], pixels are classified into groups using feature distances through pixel embedding.

There are obvious benefits to using segmentation-based approaches for both text & non-text segmentation [32, 33]. It is possible, however, for irrelevant non-characters to be misclassified as characters during the text segmentation areas training process. This can lower the quality of the segmentation findings by causing problems with text line adherence.

Connected component Based Approaches: Text detection systems employ these methods, which first detect individual text entries, then link or group these into complete text instances after a post-processing step. As a result of their flexible representation and adaptability [34-37], these methods have gained popularity in the detection of arbitrarily shaped text.

Using ordered discs and text centerlines to model text instances, TextSnake [37] represents text of varying shapes successfully. When it comes to inference, however, TextSnake still needs to rely on laborious post-processing procedures like centralising, striding, and sliding. Each text instance is built from ordered rectangular components, including text and non-text, in DRRG's text detection approach.

Text regions are typically divided into several pieces consisting of both text and non-text components by these methods that work on specific text parts. The computational cost and difficulty may rise if many non-text parts must be generated all at once. The authors [38] present a method for multidirectional text detection that uses exhaustive segmentation to provide potential character candidates. To foretell character region maps and affinity maps, CRAFT [39] uses semi-supervised learning. These techniques can decrease computing complexity and difficulty by limiting their attention to character regions inside text components.

Relational inference is an essential aspect of connected component-based methods, as their performance relies heavily on the grouping of text lines. Methods like Pixellink utilize embedding features to generate text areas and provide instance information. In the case of CRAFT [39], affinity maps are predicted through weakly supervised learning.

However, the receptive field of the CNN limits the efficacy of these approaches, making it difficult to capture relationships between distant component areas utilising local convolutional operators. Graph convolutional networks (GCNs) were created by the authors [40] to overcome this shortcoming by allowing for local graph-based reasoning and deduction of the likelihood of links between a component and its neighbours. On open-source datasets, their technique outperformed previous best practices.

To accomplish iterative boundary deformation, the authors [40] present a model that combines GCN with recurrent neural network (RNN). The goal of this iterative procedure is to produce a text instance with a more precise form. Their approach performed exceptionally well on difficult text-in-the-wild datasets like TotalText [41].

Transformer Based Approaches: Computer vision [42-45] has grown in popularity since their introduction for machine translation [46]. With DETR [47], object detection was treated as a set prediction problem instead of being a complex post-processing problem. In spite of these challenges, DETR continued to investigate detection transformers due to inefficient utilization of high-resolution features and slow training convergence. For instance, Deformable-DETR [48] addressed these issues by focusing on sparse features. DE-DETR [49] identified sparse feature sampling as a crucial factor for data efficiency. In the Transformer decoder, dynamic anchor boxes were introduced to enhance training through DAB-DETR [50].

Limitations of the Exisiting approaches are:

(1) Existing methods struggle with detecting irregular shapes in natural scene images.

(2) Limited receptive fields of Convolutional Neural Networks (CNNs) make it difficult to capture long-range relationships between distant text component regions.

(3) Existing methods may inaccurately identify non-character text components, leading to false positives.

(4) Capturing adjacency relations among text components is crucial for accurate text detection.

Advantages of our Proposed approach

(1) Our approach employs a U-net architecture, which is particularly effective in capturing irregular shapes. This architecture, combined with connected component analysis, enhances the detection of irregular text shapes, addressing a significant challenge in text detection.

(2) Our method integrates graph convolution networks (GCN), enabling the deduction of adjacency relations among text components. This innovation allows for a more comprehensive understanding of long-range relationships, enhancing the model's ability to connect and identify distant text components.

(3) By combining U-net architecture with connected component analysis, our approach enhances the accuracy of text component detection and reduces the likelihood of misidentifying non-character text components. This results in a more precise and reliable text detection system.

(4) The integration of GCNs introduces a sophisticated mechanism for inferring adjacency relations. This step significantly contributes to the advancement of text detection by providing a more nuanced understanding of the spatial relationships between text components.

3. PROPOSED METHOD

In this section, we delve into several key aspects of our methodology for text detection in natural scenes. Firstly, we elucidate the intricacies of Character Center Point Estimation, employing the U-Net architecture to achieve precise identification. This step ensures accurate localization of character center points, a fundamental element for effective text detection. Secondly, we detail the Construction of the Local Inference Graph, where each identified text region is represented as a node, and Graph Neural Networks (GNNs) are utilized to establish a comprehensive local graph. This graph captures adjacency relations, enhancing our model's

ability to understand intricate connections among text components. Additionally, we explore the Comprehensive Exploration of Proximity Relationships, aiming to provide a holistic understanding of the spatial relationships between text elements. Lastly, we discuss Text Line Formation, emphasizing the strategic organization of identified text components into coherent lines. The combination of these components forms a robust and innovative approach to address challenges in text detection, particularly in scenes with irregular shapes and long-range dependencies.

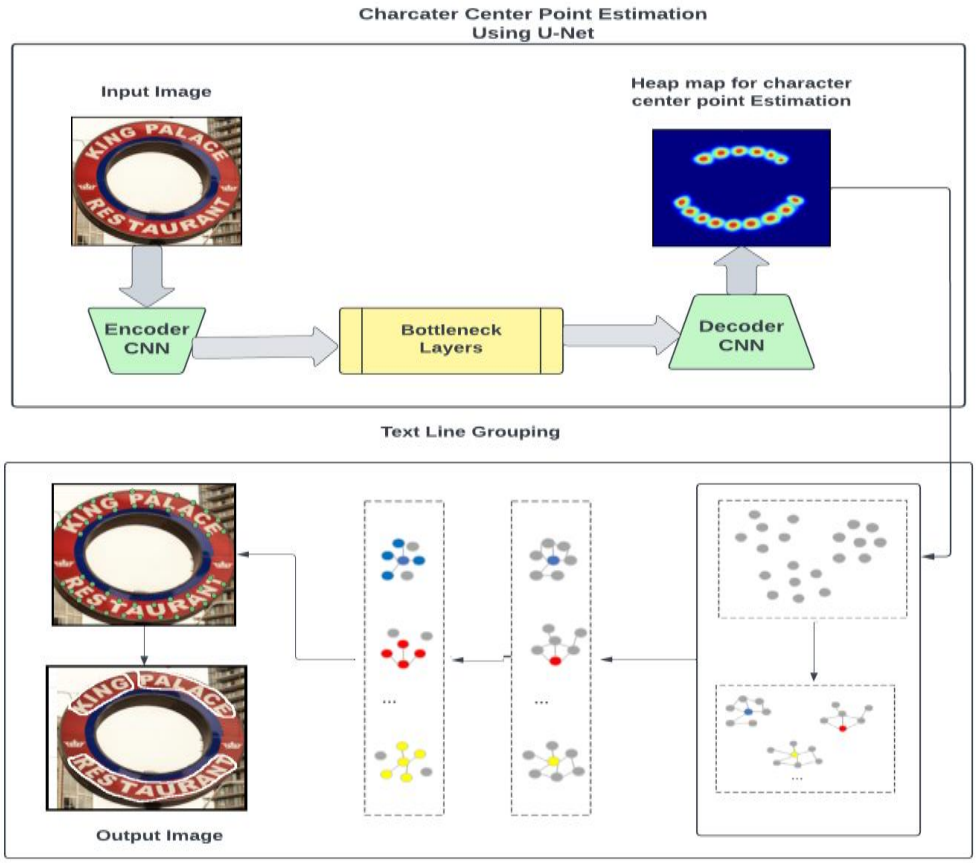

Figure 1. Proposed architecture for scene text detection

Figure 1 depicts the general architecture of our method, outlining the several processes involved in the framework. Extraction of text components, construction of a local inference graph, inference of deep adjacency relations, and production of text lines make up the framework. At first, we use the U-net architecture is applied for feature extraction and connected component analysis is used for character center point estimation to the last layer of U-Net [51]. Then, we create a local inference network that stands in for the innate connection relationships among the character text components by capitalising on their fundamental features. The deep relational inference network uses this local inference graph to reason about the causal relationships between the constituent parts of a character string. At last, the separated connected regions are used to categorise the reasoning outcomes obtained into individual text instances.

3.1 Character center point estimation using U-Net

An input image with characters or text regions can be processed by the U-Net architecture (Figure 2). Both an encoder path and a decoder path make up this system. Convolutional and pooling operations are applied to the input image in the encoder route in order to extract pertinent features at various levels. The input's abstract representations are captured by the pooling layers as they progressively decrease the spatial dimensions.

At the bottleneck layer, the spatial dimensions are significantly reduced, but the learned features are highly abstract and semantically rich. The decoder path starts with up sampling operations to restore the spatial dimensions, followed by convolutional layers that refine the features.

Skip connections, which link comparable feature maps between the encoder and decoder paths, are a crucial component of the U-Net design. The localization of character centre points is facilitated by these links, which allow low-level and high-level features to be combined. They also aid in maintaining spatial information and fine-grained features from previous network stages.

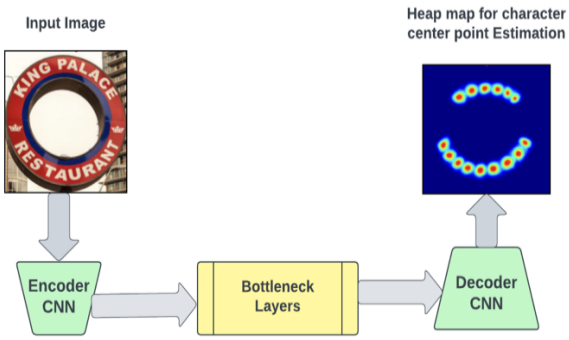

Figure 2. Block diagram of U-Net utilizing the static heatmap for scene text detection

The final layer of the U-Net architecture produces predictions, typically in the form of a heatmap. For character center point estimation, the output layer can be designed to predict the likelihood or probability of each pixel being a character center point. This is accomplished using a sigmoid activation function that generates pixel-wise predictions between 0 and 1.

To identify the character center points, a thresholding operation is applied to the heatmap. Pixels with values above a certain threshold are considered potential character center points. This thresholding step creates a binary mask, where values above the threshold are set to 1 and the rest to 0.

Connected component analysis is then applied to the binary mask. Connected component analysis is represented in Algorithm 1. This analysis identifies and labels connected regions in the binary image, where each labeled region represents a group of adjacent pixels.

Algorithm 1 Character Center Point estimation using Connected Component Analysis (CCA)

Input: Binary image (after thresholding)

Output: Connected components

```
Algorithm CCA (binaryImage):
components= [ ]
labelCounter=1
Algorithm dfs(pixel, component):
component.add(pixel)
binaryImage[pixel]=labelCounter
neighbors=getNeighbors(pixel)
for neighbor in neighbors:
if binaryImage[neighbor]==1:
dfs(neighbor, component)
for each pixel in binaryImage:
if binaryImage[pixel]==1:
component=new empty component
dfs(pixel, component)
components.add(component)
labelCounter+=1
return components
```

To filter out small connected components that may correspond to noise or artifacts, a filtering step based on component area is performed. Components below a certain area threshold are removed, retaining larger and more meaningful components that are more likely to represent character center points.

For each remaining connected component, the centroid (center of mass) is computed by averaging the x and y coordinates of all pixels within the component. These computed centroids represent the estimated character center points.

Optionally, additional refinement techniques can be applied to improve the accuracy of the character center points. These techniques may include centroid shift correction, sub-pixel precision estimation, or the incorporation of geometric constraints.

By following the procedure of connected component analysis and the subsequent steps, the U-Net architecture can effectively identify and extract the character center points from the binary mask, providing a more precise localization of the character positions.

3.2 Construction of the local inference graph

The next step, after character center point estimation, we use a graph convolution network for inferring adjacency relationships between text components. Text components are represented by character center points according to this method. Each piece of text represents a node in the network. Inference time and complexity would increase if all nodes were used for inference directly. A local inference graph is constructed for this purpose in DRRG, which includes the pivot node and its neighbours up to the second order. First-order neighbours are the eight nodes immediately adjacent to the pivot, while second-order neighbours are the four nodes immediately adjacent to the first-order neighbours. Our method, in contrast to DRRG, takes into account only the immediate neighbours of each node. This reduces the number of nodes involved in the reasoning process by choosing the pivot node, four neighbouring nodes of the first order, and two neighbouring nodes of the second order. Figure 3 elaborates on the steps taken to construct the local inference graph. A node's adjacency is determined by evaluating the affinity, between it and the pivot node. The affinity, A_s between a pivot node p_n and another node is defined as follows:

$$A_s = 1 - \frac{A_{pr}}{\max(H, W)} \tag{1}$$

$$A_{pr} = \sqrt{(M_p - M_r)^2 + (N_p - N_r)^2} \tag{2}$$

where, H and W represents height and width of the images and A_{pr} represents Euclidean distance between two nodes p and r.

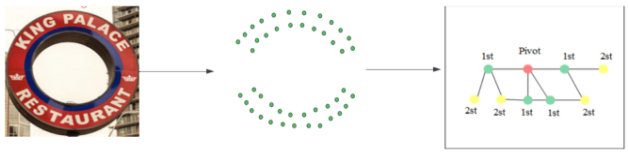

Figure 3. Construction of local inference graph

3.3 Comprehensive exploration of proximity relationships

Text nodes are connected in the local inference graph in an accurate manner. However, the adjacency relations between these nodes cannot be accurately represented by a link mapping or embedding mapping approach. To overcome this shortcoming, we present a Graph Convolutional Network (GCN)-based deep relational inference network. Inferring proximity relations between text component nodes is possible with the help of this network. A pivot's relationships with its first-order neighbours are an important part of the deep adjacency relation inference procedure. It is also important to note that the characteristics of a node can be influenced by its neighbors. Hence, fusion features are supplied for the first-order neighbours by second-order neighbours. Two common types of inputs to the GCN are a feature matrix (denoted by Fm) and an adjacency matrix (denoted by Am). Here is how these two matrices are calculated:

Feature-Matrix (F_m): Each text component of the same text instance is represented by a rotating rectangle, and these rectangles share certain geometric properties. We use a combination of deep features and geometric properties as

features for the textual parts. After a text component has been extracted, we can acquire its deep features by mapping its characteristics to the RROI-Align layer. At the same time, we calculate the text component's geometric properties using its X, Y, W, H, and attributes. We embed text components geometric qualities into high-dimensional spaces in order to derive geometric characteristics from text [8, 24, 25]. Eq. (3) and Eq. (4) provide the formulas for determining these embeddings. The feature matrix F_m, which represents the text components and it is the outcome of combining the features with the geometric characteristics.

$$\epsilon_{2b}^{(z)} = \left(\cos\left(\frac{z}{1000^{2b/C_\epsilon}}\right)\right), \text{j}\varepsilon\ (0, C_\varepsilon/2a\text{-}1) \quad (3)$$

$$\epsilon_{2a+1}^{(z)} = \sin\left(\frac{z}{1000^{2b/C_\epsilon}}\right), \text{j}\varepsilon\ (0, C_\varepsilon/2a\text{-}1) \quad (4)$$

Adjacency-Matrix (A_m): Inference graph nodes are connected to produce the adjacency matrix A_m. If node a of the text component is connected to node bof the local inference graph, then A_m (a, b) =1, and otherwise A_m (a, b) =0. Adjacency analysis between a node and itself is superfluous, thus we set A_m (a, a) =0.

Graph convolutional network: The local inference graph is inferred using a GCN-based inference network based on the feature matrix (F_m) and the adjacency matrix (A_m). Layer k's feature matrix is referred to as F^k, and its corresponding convolutional layer is defined as follows.

$$F^k = \sigma\left((F_m^k \oplus GX^k)W^k\right) \quad (5)$$

$$G = \left(D^{-1/2} A D^{-1/2}\right) \quad (6)$$

$$D_{i,i} = \sum_j A_{i,j} \quad (7)$$

In the equation, X^k represents the feature matrix of size $N \times d_{in}$, where N represents the text components number and d_{in} is the feature dimension of the input nodes. Similarly, F^k represents the feature matrix of size $N \times d_{out}$, where d_{out} is the feature dimension of the output nodes. Λ represents diagonal matrix, & G illustrates the symmetric normalized Laplacian of size ((N×N)). The symbol ⊕ denotes concatenation. W^k is the weight matrix of layer k, & σ represents a nonlinear activation function. Training involves only computing gradients for the nodes that are 1-order neighbours, since we are primarily interested in connecting the pivot node with its first-order neighbours, while testing involves the classification of 1-order nodes.

3.4 Text line formation

As part of the Comprehensive Exploration of Proximity Relationships, we summarize the probabilities from all the local inference graphs to derive the adjacency probability matrix (S). When deciding whether or not to keep an edge between two nodes, the threshold (TH) is used. If S(a, b) is greater than a threshold value, S(a, b) is set to 1; otherwise, S(a, b) is set to 0. By using BFS, we find the related subgraphs (L = L1, L2, ..., Lk) in the whole, which we will call L=L1, L2, ..., Lk. Each line of text in the set L is represented by a subgraph in L. Nodes inside each subgraph are subsequently sorted to complete the procedure.

4. EXPERIMENTAL SETUP

4.1 Datasets

ICDAR2013 Dataset: By separating the training from the testing sets and removing duplicate images, the ICDAR2013 dataset was created from the ICDAR2011 benchmark. Annotations have been modified for a subset of ground-truth annotations. We used 229 images for training and 233 for testing, resulting in a dataset of 482 images. The vast majority of the pictures are from nature, and the majority of the texts are horizontal or nearly horizontal.

MSRA(TD500) Dataset: Pocket camera indoor (office, mall) and outdoor (street) photos make up the bulk of the MSRA-TD500. Signs, doorplates, and warning signs predominate indoors, while guide-boards & bill-boards take up the bulk of exterior imagery. Images are available in dimensions ranging from 1296x864 to 1920x1280. The collection contains text in several formats, including a wide range of languages, scripts, sizes, colours, and orientations (including but not limited to Chinese, English, and combinations).

CTW (1500) Dataset: It contains 1500 images: 1000 for training & 500 for testing. There are 10,751 photographs of cropped text included, with an additional 3,530 images of bent text. The pictures were collected by hand from various sources, including the web, image databases like Google Open-Image, and mobile phone cameras. There is a lot of horizontal text in the dataset, as well as text in other orientations.

Total text dataset: The Total-Text dataset includes 1,255 high-dimensional images for training and 300 for testing. Text in a variety of orientations, including horizontal, multi-oriented, and curved text, are included in this collection. The text examples include both polygon and word-level annotations, providing additional information about the marked areas. The Totaltext dataset is an essential resource for developing and evaluating text detection and recognition algorithms.

4.2 Implementation details

Our network relies on the Resnet-50 architecture, which has undergone pre-training utilizing the ImageNet dataset. We use a two-stage training procedure that begins with two epochs of pre-training on the SynthText dataset and concludes with 600 epochs of fine-tuning on a targeted benchmark dataset. In the first round of training, we randomly crop text sections, scale them up to 512 pixels wide, and divide them into 12 batches. To train the model, we employ the Adam optimizer with a learning rate set at 10^4.

During the fine-tuning phase, we employ a multi-scale training strategy. Text regions are randomly cropped and resized to three distinct dimensions: 640×640 with a batch size of 8, 800×800 with a batch size of 4, and 960×960 with a batch size of 4. During the fine-tuning process, we transition to using the SGD optimizer with an initial learning rate of 0.01. This learning rate is decreased by a factor of 0.8 every 100 epochs. Moreover, we incorporate fundamental data augmentation methods, including rotations, crops, color variations, and partial flipping. The hyperparameters associated with the local graph remain constant during both the training and testing stages. All experiments are performed on a single GPU (RTX-2080Ti) utilizing PyTorch 1.2.0.

4.3 Assessment criteria

The role of evaluation metrics is pivotal when assessing the performance of algorithms designed for irregular text detection. These criteria serve as quantitative measures for evaluating the accuracy and efficacy of the detection system. Various evaluation metrics are commonly employed to assess irregular text detection in a standardized manner.

One commonly used metric is bounding box-based evaluation, where metrics such as “precision”, “recall”, & “F1-score” are computed based on the accuracy of the predicted bounding boxes compared to ground truth annotations. An indication of precision is the proportion of instances of irregular text that have been correctly localized out of all the predicted instances. Recall calculates the proportion of correctly detected instances out of all the ground truth instances, while F1-score provides a balanced evaluation by taking into account both precision and recall.

Another metric is pixel-level evaluation, which involves measuring the accuracy of the pixel-wise segmentation masks for irregular text. In this particular context, the widely utilized evaluation metric is Intersection over Union (IoU), which calculates the overlap between the predicted mask and the ground truth mask. Higher IoU values indicate better segmentation accuracy.

Other evaluation metrics commonly employed in the assessment of irregular text detection encompass Average Precision (AP), which evaluates precision at various recall levels, and the F-measure, which combines precision and recall to provide a consolidated assessment.

The selection of appropriate evaluation metrics depends on factors such as the specific characteristics of irregular text, the complexity of the detection task, and the desired trade-off between precision and recall. It is important to choose metrics that align with the objectives and requirements of the irregular text detection system being evaluated.

Precision evaluates the ratio of accurately identified text instances to all the text instances detected by the system. It emphasizes the accuracy of positive predictions, serving as an indicator of how effectively the algorithm recognizes true positive text regions. Recall, also known as sensitivity, measures the proportion of correctly detected text instances out of all the actual text instances present in the dataset. It emphasizes the ability of the algorithm to capture all the positive instances, minimizing false negatives.

The F1-score represents a balanced assessment of the algorithm's performance as it is a harmonic mean of precision and recall. It offers a comprehensive measure that takes into account both precision and recall simultaneously. It takes into account both precision and recall, giving equal importance to false positives and false negatives.

Additional evaluation metrics for text detection could encompass Intersection over Union (IoU), which quantifies the degree of overlap between the predicted text regions and the ground truth regions, as well as Average Precision (AP), which determines the average precision across various recall levels.

4.4 Ablation study

4.4.1 Exploring the impact of u-net architecture for text component extraction through ablation study

We performed ablation experiments on three datasets, namely Total text, CTW (1500), MSRA (TD500), to assess the effectiveness of the relational reasoning network. The experimental results are presented in Table 1. In order to mitigate the influence of data on the results, our model was initially pre-trained using SynthText and subsequently fine-tuned using Totaltext & CTW (1500). For MSRA (TD500), which includes both English and Chinese text, we pre-trained our network using ICDAR2017-MLT. The maximum dimensions of the images in Totaltext, CTW (1500), and MSRA (TD500) were restricted to 1,280, 1,024, and 640, respectively, while maintaining their original aspect ratios.

We have enhanced our method based on the DRRG and DPText-DETR approach and conducted a comparative analysis of the experimental results with DRRG. Figure 4 presents a visual comparison of the text components generated by various methods. A statistical comparison has also been conducted between the two approaches regarding the no. of text components & detection results. Table 1 shows that our text component generation method led to significant reductions in text component numbers and detection times. Furthermore, the results indicate that our method effectively reduces the number of non-character text components while improving the overall performance of text detection.

Table 1. Experimental results focusing on the extraction of text components

Dataset	Models	P	R	F
CTW (1500)	DRRG [11]	83.8	81.5	82.6
	TD-GCN [40]	86.7	85.4	86.1
	Proposed	**89.5**	**88.4**	**89.2**
MSRA(TD500)	DRRG [11]	88.1	82.3	85.1
	TD-GCN [40]	89.7	85.1	87.4
	Proposed	**90.2**	**88.9**	**89.2**
Total text	DRRG [11]	83.1	85.9	84.5
	TD-GCN [40]	89.1	84.4	86.1
	Proposed	**90.4**	**88.4**	**89.1**

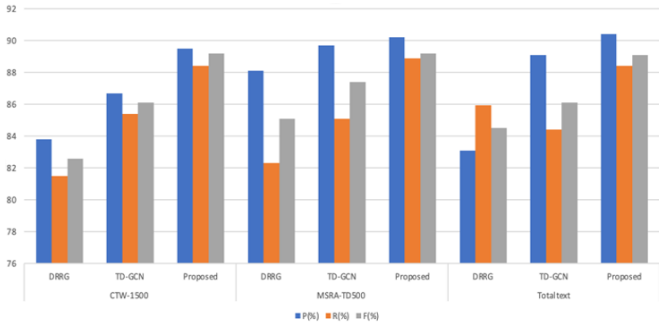

Figure 4. Bar graph illustration of text component extraction

4.4.2 Ablation experiment on local inference graph

Methods that employ feature extraction networks for direct text region detection often face challenges when it comes to accurately segmenting text lines. Instances where two text regions are mistakenly merged into one region. To address these issues and improve text region segmentation, our approach utilizes a relation inference network that leverages the adjacency relationships between text components. Experimental results on the MSRA (TD500) & CTW (1500) datasets demonstrate the effectiveness of our adjacency inference network were shown in Table 2. Figure 5 presents a visual comparison of the text components generated by various methods. The local inference ablation experiments reveal significant improvements, with precision, recall, and F-measure on MSRA (TD500) and on CTW (1500). These

performance improvements further validate the efficacy of our proposed adjacency inference network.

Table 2. Experimental results focussing on inference graph

Dataset	Models	P	R	F
CTW (1500)	DRRG [11]	83.1	80.6	81.8
	TD-GCN [40]	86.7	85.4	86.1
	Proposed	**91.2**	**89.9**	**87.3**
MSRA(TD500)	DRRG [11]	83.2	78.5	80.8
	TD-GCN [40]	89.7	85.1	87.4
	Proposed	**90.5**	**87.4**	**89.5**
Total text	DRRG [11]	83.1	85.9	84.5
	TD-GCN [40]	89.1	84.4	86.1
	Proposed	**90.3**	**88.9**	**88.4**

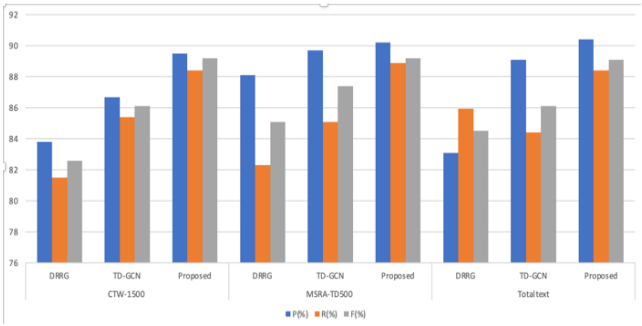

Figure 5. Bar graph illustration of local inference graph

4.3 Results & Interpretations

4.3.1 Empirical investigations conducted on the ICDAR (2013) dataset

We conducted experiments utilizing the ICDAR (2013) dataset, and Figure 6 showcases selected examples of the obtained outcomes. The experimental results highlight that the approach proposed in this paper showcases exceptional performance in detecting text on this specific dataset. To further evaluate its effectiveness, we compared our method's detection results with those of other existing text detection approaches. Our method performs best in the text detection performance evaluation with precision 92.8%, recall 87.1%, and F-measure 89.9% as demonstrated in Table 3. Significantly, our method surpasses other methods in terms of recall rate and F-measure.

Figure 6. Detection samples from the proposed method on the ICDAR (2013) dataset

Table 3. Experimental results conducted on the ICDAR (2013) dataset using different methods

Reference	P	R	F
[52]	83.5	77.2	80.2
[38]	87.3	81.1	84.3
[53]	90.0	80.0	85.0
[54]	94.0	69.0	80.0
[40]	92.8	87.1	89.9
Proposed model	**94.1**	**89.4**	**89.8**

4.3.2 Empirical investigations conducted on the dataset MSRA(TD500)

Using the MSRA(TD500) dataset, we evaluated our method on a multilanguage dataset. Figure 7 showcases selected examples of the experimental outcomes obtained, while Table 4 presents a comparative analysis between our method and other existing approaches. As shown in Table 4, our method achieved 89.7% precision, 85.1% recall, and 87.4% F-measure values in the MSRA(TD500) dataset. Notably, our approach outperforms other methods in terms of both recall rate and F-measure, showcasing its superiority in detecting multilanguage scene text.

Figure 7. Detection samples from the proposed method on the dataset MSRA(TD500)

Table 4. Experimental results conducted on the MSRA(TD500) dataset using different methods

Reference	P	R	F
[55]	86.0	70.0	77.0
[6]	87.4	75.9	81.3
[39]	88.2	78.2	82.9
[14]	81.6	77.2	79.3
[15]	85.0	82.0	83.0
[35]	88.1	82.3	85.1
[12]	90.2	81.9	85.8
[13]	90.9	83.8	87.2
[32]	91.5	83.3	87.2
[40]	89.7	85.4	86.1
Proposed model	**92.3**	**87.8**	**89.3**

4.3.3 Empirical investigations conducted on the CTW (1500):

Moreover, we selected the CTW (1500) dataset so that we could assess our method's robustness to detect irregular scene text. Figure 8 presents several examples showcasing the experimental outcomes achieved using our method. As shown in Table 5, our method outperforms other methods. Remarkably, the results presented in Table 5 highlight that our method surpasses alternative approaches in terms of recall rate and F-measure, attaining impressive values of 85.4% and 86.1% respectively. As a result, we were able to detect irregular and multidirectional scene text accurately with the help of our method.

Figure 8. Detection samples from the proposed method on the CTW (1500) dataset

Table 5. Results obtained from experiments conducted on the CTW (1500) dataset using different methods

Reference	P	R	F
[33]	84.5	82.8	83.6
[6]	83.0	79.8	81.4
[27]	87.7	80.6	84.0
[39]	86.0	81.1	83.5
[14]	85.1	78.2	81.5
[11]	85.9	83.0	84.4
[40]	86.7	85.4	86.1
Proposed model	**91.6**	**87.3**	**89.6**

4.3.4 Empirical investigations conducted on the dataset Totaltext

The primary emphasis of the Total-Text dataset lies in curved and multi-oriented texts, offering annotations at the word level. During testing, we resize the images to ensure the shortest side is 512 pixels, and we maintain the longest side not exceeding 1,280 pixels. We present some visually impressive results in Figure 9. The effectiveness of our approach is evaluated by comparing its performance with other methods, as presented in Table 6. The images demonstrate that our method excels at accurately detecting irregular texts at the word level and effectively separating closely positioned text instances of various shapes. The performance surpasses that of other existing methods by a significant margin, indicating the effectiveness and superiority of our approach.

Table 6. Results obtained from experiments conducted on the Totaltext dataset using different methods

Reference	P	R	F
[33]	85.6	75.7	80.3
[6]	81.2	79.9	80.6
[28]	84.02	77.9	80.87
[55]	82.1	80.9	81.5
[23]	87.6	79.3	83.3
[11]	86.54	84.93	85.73
Proposed model	**89.9**	**89.2**	**87.96**

Figure 9. Detection samples from the proposed method on the dataset Totaltext

5. CONCLUSION

In summary, this paper introduces an innovative approach to irregular text detection using a graph convolution network along with U-Net for character center point estimation and connected component analysis. Our method demonstrates robustness and effectiveness in accurately detecting irregular scene text across various datasets. For future work, we aim to address challenges such as overlapping-text, low resolution-text, and partially occluded-text. Additionally, we plan to integrate our approach with text recognition to create an end-to-end solution covering both text detection and recognition. It's crucial to acknowledge the limitations related to datasets, languages, and text types and explore ways to enhance the model's generalizability across diverse scenarios.

REFERENCES

[1] Chen, J., Lian, Z. (2021). TextPolar: Irregular scene text detection using polar representation. International Journal on Document Analysis and Recognition (IJDAR), 24(4): 315-323. https://doi.org/10.1007/s10032-021-00373-5

[2] Sheng, T., Chen, J., Lian, Z. (2021). Centripetaltext: An efficient text instance representation for scene text detection. Advances in Neural Information Processing Systems, 34: 335-346.

[3] Qin, S., Chen, L. (2022). Arbitrary-shaped scene text detection with keypoint-based shape representation. International Journal on Document Analysis and Recognition (IJDAR), 25(2): 115-127. https://doi.org/10.1007/s10032-022-00396-6

[4] Wu, G., Zhang, Z., Xiong, Y. (2022). CarveNet: A channel-wise attention-based network for irregular scene text recognition. International Journal on Document Analysis and Recognition (IJDAR), 25(3): 177-186. https://doi.org/10.1007/s10032-022-00398-4

[5] Yang, C., Chen, M., Yuan, Y., Wang, Q. (2023). Text growing on leaf. IEEE Transactions on Multimedia. https://doi.org/10.1109/TMM.2023.3244322

[6] Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., Bai, X. (2019). Textfield: Learning a deep direction field for irregular scene text detection. IEEE Transactions on Image Processing, 28(11): 5566-5579. https://doi.org/10.1109/TIP.2019.2900589

[7] Sirisha, U., Sai Chandana, B. (2022). Semantic interdisciplinary evaluation of image captioning models. Cogent Engineering, 9(1): 2104333. https://doi.org/10.1080/23311916.2022.2104333

[8] Sirisha, U., Bolem, S.C. (2022). Aspect based sentiment & emotion analysis with ROBERTa, LSTM. International Journal of Advanced Computer Science and Applications, 13(11). https://doi.org/10.14569/IJACSA.2022.0131189

[9] Sirisha, U., Chandana, B.S. (2023). Privacy preserving image encryption with optimal deep transfer learning based accident severity classification model. Sensors, 23(1): 519. https://doi.org/10.3390/s23010519

[10] Madhuri, M.A., Devi, T.U. (2023). Statistical analysis of design aspects on various graph embedding learning classifiers. In 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), IEEE, pp. 98-105. https://doi.oSliding rg/10.1109/ICCMC56507.2023.10083741

[11] Zhang, S.X., Zhu, X., Hou, J.B., Liu, C., Yang, C., Wang, H., Yin, X.C. (2020). Deep relational reasoning graph network for arbitrary shape text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9699-9708.

[12] Wang, X., Zheng, S., Zhang, C., Li, R., Gui, L. (2021). R-YOLO: A real-time text detector for natural scenes with arbitrary rotation. Sensors, 21(3): 888. https://doi.org/10.3390/s21030888

[13] Raisi, Z., Naiel, M.A., Younes, G., Wardell, S., Zelek, J.S. (2021). Transformer-based text detection in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3162-3171.

[14] Wan, Q., Ji, H., Shen, L. (2021). Self-attention-based text knowledge mining for text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5983-5992.

[15] Wang, X., Jiang, Y., Luo, Z., Liu, C.L., Choi, H., Kim, S. (2019). Arbitrary shape scene text detection with adaptive text region representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6449-6458.

[16] Long, S., He, X., Yao, C. (2021). Scene text detection and recognition: The deep learning era. International Journal of Computer Vision, 129: 161-184. https://doi.org/10.1007/s11263-020-01369-0

[17] Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T. (2021). Text recognition in the wild: A survey. ACM Computing Surveys (CSUR), 54(2): 1-35. https://doi.org/10.1145/3440756

[18] Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., Zhang, W. (2021). Fourier contour embedding for arbitrary-shaped text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3123-3131.

[19] Wang, Y., Mamat, H., Xu, X., Aysa, A., Ubul, K. (2022). Scene uyghur text detection based on fine-grained feature representation. Sensors, 22(12): 4372. https://doi.org/10.3390/s22124372

[20] Arava, K., Paritala, C., Shariff, V., Praveen, S.P., Madhuri, A. (2022). A generalized model for identifying fake digital images through the application of deep learning. In 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, pp. 1144-1147. https://doi.org/10.1109/ICESC54411.2022.9885341

[21] Sindhura, S., Praveen, S.P., Safali, M.A., Rao, N. (2021). Sentiment analysis for product reviews based on weakly-supervised deep embedding. In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, pp. 999-1004. https://doi.org/10.1109/ICIRCA51532.2021.9544985

[22] Praveen, S.P., Sindhura, S., Madhuri, A., Karras, D.A. (2021). A novel effective framework for medical images secure storage using advanced cipher text algorithm in cloud computing. In 2021 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1-4. https://doi.org/10.1109/IST50367.2021.9651475

[23] Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., Ding, X. (2019). Look more than once: An accurate detector for text of arbitrary shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10552-10561.

[24] Liao, M., Shi, B., Bai, X., Wang, X., Liu, W. (2017). Textboxes: A fast text detector with a single deep neural network. In Proceedings of The AAAI Conference on Artificial Intelligence, 31(1). https://doi.org/10.1609/aaai.v31i1.11196

[25] Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L. (2020). Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In Proceedings of The Ieee/Cvf Conference on Computer Vision and Pattern Recognition, pp. 9809-9818.

[26] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J. (2017). East: An efficient and accurate scene text detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5551-5560.

[27] Zhang, S.X., Zhu, X., Yang, C., Wang, H., Yin, X.C. (2021). Adaptive boundary proposal network for arbitrary shape text detection. In Proceedings of The IEEE/CVF International Conference on Computer Vision, pp. 1305-1314.

[28] Li, X., Wang, W., Hou, W., Liu, R. Z., Lu, T., Yang, J. (2018). Shape robust text detection with progressive scale expansion network. arXiv Preprint arXiv: 1806.02559. https://doi.org/10.48550/arXiv.1806.02559

[29] Deng, D., Liu, H., Li, X., Cai, D. (2018). Pixellink: Detecting scene text via instance segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.12269

[30] Tian, Z., Shu, M., Lyu, P., Li, R., Zhou, C., Shen, X., Jia, J. (2019). Learning shape-aware embedding for scene text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4234-4243.

[31] Zhang, S.X., Zhu, X., Hou, J.B., Yang, C., Yin, X.C. (2022). Kernel proposal network for arbitrary shape text detection. IEEE Transactions on Neural Networks and

Learning Systems. https://doi.org/10.1109/TNNLS.2022.3152596
[32] Liao, M., Zou, Z., Wan, Z., Yao, C., Bai, X. (2022). Real-time scene text detection with differentiable binarization and adaptive scale fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1): 919-931. https://doi.org/10.1109/TPAMI.2022.3155612
[33] Feng, W., He, W., Yin, F., Zhang, X.Y., Liu, C.L. (2019). Textdragon: An end-to-end framework for arbitrary shaped text spotting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9076-9085.
[34] Keserwani, P., Dhankhar, A., Saini, R., Roy, P.P. (2021). Quadbox: Quadrilateral bounding box-based scene text detection using vector regression. IEEE Access, 9: 36802-36818.
[35] Yin, F., Wu, Y.C., Zhang, X.Y., Liu, C.L. (2017). Scene text recognition with sliding convolutional character models. arXiv preprint arXiv:1709.01727. https://doi.org/10.48550/arXiv.1709.01727
[36] Tian, Z., Huang, W., He, T., He, P., Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. In Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part VIII. Springer International Publishing, 14: 56-72. https://doi.org/10.1007/978-3-319-46484-8_4
[37] Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C. (2018). Textsnake: A flexible representation for detecting text of arbitrary shapes. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 20-36.
[38] Wei, Y., Shen, W., Zeng, D., Ye, L., Zhang, Z. (2018). Multi-oriented text detection from natural scene images based on a CNN and pruning non-adjacent graph edges. Signal Processing: Image Communication, 64: 89-98. https://doi.org/10.1016/j.image.2018.02.016
[39] Baek, Y., Lee, B., Han, D., Yun, S., Lee, H. (2019). Character region awareness for text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365-9374.
[40] Zhang, S., Zhou, C., Li, Y., Zhang, X., Ye, L., Wei, Y. (2023). Irregular scene text detection based on a graph convolutional network. Sensors, 23(3): 1070. https://doi.org/10.3390/s23031070
[41] Ch'ng, C.K., Chan, C.S., Liu, C.L. (2020). Total-text: Toward orientation robustness in scene text detection. International Journal on Document Analysis and Recognition (IJDAR), 23(1): 31-52. https://doi.org/10.1007/s10032-019-00334-z
[42] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv Preprint arXiv: 2010.11929. https://doi.org/10.48550/arXiv.2010.11929
[43] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of The IEEE/CVF International Conference on Computer Vision, pp. 10012-10022.
[44] Zhou, Y., Xie, H., Fang, S., Wang, J., Zha, Z., Zhang, Y. (2021). TDI TextSpotter: Taking data imbalance into account in scene text spotting. In Proceedings of the 29th ACM International Conference on Multimedia, pp. 2510-2518. https://doi.org/10.1145/3474085.3475423
[45] Zhang, Q., Xu, Y., Zhang, J., Tao, D. (2023). Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. International Journal of Computer Vision, 1-22. https://doi.org/10.1007/s11263-022-01739-w
[46] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.
[47] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020). End-to-end object detection with transformers. In European Conference on Computer Vision. Cham: Springer International Publishing, pp. 213-229. https://doi.org/10.1007/978-3-030-58452-8_13
[48] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. arXiv Preprint arXiv: 2010.04159. https://doi.org/10.48550/arXiv.2010.04159
[49] Wang, W., Zhang, J., Cao, Y., Shen, Y., Tao, D. (2022). Towards data-efficient detection transformers. In European Conference on Computer Vision. Cham: Springer Nature Switzerland, pp. 88-105. https://doi.org/10.1007/978-3-031-20077-9_6
[50] Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L. (2022). Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv Preprint arXiv: 2201.12329. https://doi.org/10.48550/arXiv.2201.12329
[51] Lu, X., Jian, M., Wang, X., Yu, H., Dong, J., Lam, K.M. (2022). Visual saliency detection via combining center prior and U-Net. Multimedia Systems, 28(5): 1689-1698. https://doi.org/10.1007/s00530-022-00940-8
[52] Wei, Y., Zhang, Z., Shen, W., Zeng, D., Fang, M., Zhou, S. (2017). Text detection in scene images based on exhaustive segmentation. Signal Processing: Image Communication, 50: 1-8. https://doi.org/10.1016/j.image.2016.10.003
[53] Gao, J., Wang, Q., Yuan, Y. (2019). Convolutional regression network for multi-oriented text detection. IEEE Access, 7: 96424-96433. https://doi.org/10.1109/ACCESS.2019.2929819
[54] Jeon, M., Jeong, Y.S. (2020). Compact and accurate scene text detector. Applied Sciences, 10(6): 2096. https://doi.org/10.3390/app10062096
[55] Tang, J., Yang, Z., Wang, Y., Zheng, Q., Xu, Y., Bai, X. (2019). Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. Pattern Recognition, 96: 106954. https://doi.org/10.1016/j.patcog.2019.06.020