iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

OCR-ENHANCED KERNEL PROPOSAL NETWORK FOR ARBITRARY-SHAPED SCENE TEXT DETECTION AND RECOGNITION

Vuyyuru Asritha Triveni,
Velpula Surya Prakash Reddy,
Gunji Gayathri,
Students, Department of CSE, R.V.R & J.C College of Engineering, Guntur, A.P.

Dr.Ch. Aparna,
Professor, Department of CSE, R.V.R & J.C College of Engineering, Guntur, A.P.

ABSTRACT
Scene text detection for arbitrary-shaped text remains challenging, especially in separating neighbouring instances. This article proposes a Kernel Proposal Network (dubbed KPN). This novel method addresses this issue by classifying texts into instance-independent feature maps through dynamic convolution kernels extracted from Gaussian center maps. To promote kernel independence, an Orthogonal Learning Loss (OLL) is introduced. Each kernel proposal encodes self-information and positional features, enabling effective separation of text instances without heavy post-processing.
In this extension, we further integrate an OCR module that leverages the KPN-detected regions to perform text recognition. The detected arbitrary-shaped text masks are processed through a OCR model, achieving a complete pipeline for scene text detection and recognition. Experimental results validate the effectiveness of our unified approach on multiple challenging datasets. The code is publicly available at https://github.com/ATV-77/OCR-Enhanced-KPN.

Keywords:
Scene Text Detection, Scene Text Recognition, Arbitrary-Shaped Text, OCR, Deep Neural Network, Dynamic Convolution Kernel, Kernel Proposal

I. INTRODUCTION

Scene text detection plays a crucial role in computer vision, serving as a fundamental step in a wide range of text-related applications, including translation, text-based visual question answering, text recognition, and text mining. With the rapid advancements in deep learning-based object detection techniques [1]–[3] and segmentation networks [4], [5], the performance of scene text detection has significantly improved [6]–[8]. Among the various challenges in this field, detecting text with arbitrary shapes — including varied scales, curved, and irregularly shaped text — remains particularly difficult, drawing increasing attention from both the research community and industry. In addition to the general challenges inherent in scene text detection, arbitrary-shaped text detection must also handle these additional complexities.
Recently, segmentation-based approaches [9]–[11] have demonstrated strong performance in detecting texts of arbitrary shapes. Thanks to the flexibility of pixel-level predictions, such as pixel-wise classification of text and non-text regions, segmentation-based methods [10], [12], [13] based on fully convolutional networks (FCNs) can easily adapt to irregularly shaped text instances. However, despite this adaptability, segmentation-based methods often struggle with separating neighbouring text instances in complex scenes, as shown in Fig. 1(a). This difficulty can arise from factors such as annotation inaccuracies or visual similarities between adjacent text instances.
To overcome these problems, many segmentation-based methods [9]–[11], [13], [14] incorporate sophisticated post-processing techniques to group and separate different text instances based on the predicted masks. For example, PSENet [13] uses progressively scaled kernels to reconstruct individual text instances by expanding the minimal scale kernels. Similarly, methods like [9]–[11], [14] learn pixel-pair embedding vectors to cluster pixels into distinct text instances during post-processing. Although these techniques can alleviate the problem of

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

neighbouring text adhesion to some extent, they typically introduce a heavy computational burden due to their complex post-processing steps.

Seeking to enhance efficiency, DB [12] avoids complex post-processing altogether. Instead, it shrinks annotated text boundaries using the Vatti clipping algorithm [15] to create probability maps, which are subsequently used to recover the separated boundaries of text instances through an inverse transformation of the same algorithm. However, because DB [12] relies on fixed boundary scaling rules, it often struggles to maintain accurate boundary delineations for text of varying sizes.

Furthermore, several methods based on region-based convolutional neural networks (R-CNNs) [16], such as [17]–[19], adopt a two-step strategy: initially detecting bounding rectangles around candidate text regions, followed by pixel-level segmentation of the text instances within those boxes. While approaches based on Mask R-CNN can mitigate the issue of adjacent instance overlap, their effectiveness is heavily dependent on the quality of the bounding box proposals and involves the computationally expensive RoI (Region of Interest) operations.

In addition to the challenge of detecting text regions, recognizing the textual content within the detected instances is crucial for enabling full scene text understanding. we further integrate an OCR-based recognition module into our framework, allowing the extracted regions from the kernel proposals to be recognized effectively. Recognizing text from arbitrary-shaped regions has been explored in works such as Total-Text [58] and STN-OCR [59], and our approach leverages similar ideas by directly applying OCR techniques on the detected instance masks generated by our KPN.

In this work, we introduce a Kernel Proposal Network (KPN**)** aimed at detecting arbitrary-shaped text in scene images. The proposed KPN framework addresses one of the key challenges in text detection — the separation of closely located text instances — by classifying different text instances into independent feature maps, thus eliminating the need for the complex aggregation procedures typically employed in segmentation-based text detection methods.

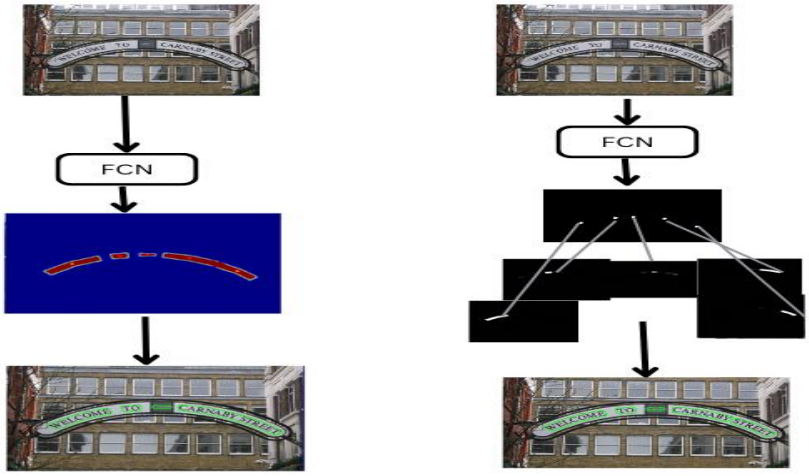


Fig-1: Comparison with FCN-based methods. (a) FCN-based methods suffer from covering adjacent instances (red contour in the bottom image). (b) Demonstrates that our KPN can successfully disperse texts on feature maps, thus generating separated yet accurate masks for text instance

Specifically, KPN predicts a Gaussian center map for each input image, which is then used to extract a set of candidate kernel proposals (i.e., dynamic convolution kernels) from the embedding feature maps based on their corresponding key point positions. Crucially, for accurate separation of neighbouring text instances, it is necessary to ensure the independence of these kernel proposals. To achieve this, we introduce a novel Orthogonal Learning Loss (OLL), which applies orthogonal constraints to enforce independence among the kernels.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

The kernel proposals are designed to encapsulate key self-learned features from the network as well as positional information through position embedding, rather than simply sharing information across all text instances. After filtering out noisy kernel proposals using predefined rules, the remaining valid proposals individually convolve the embedding feature maps, thereby generating instance-specific feature maps for each detected text region. Through this approach, KPN effectively separates adjacent text instances and significantly improves robustness in scenarios with ambiguous or unclear boundaries.
To the best of our knowledge, this work is the first to incorporate the concept of dynamic convolution kernels for addressing the adhesion problem between neighbouring text instances in the field of text detection, offering an efficient and powerful solution.

Our main contributions are summarized as follows:

1. We present a **segmentation-based arbitrary-shape text detection framework** that eliminates the need for costly post-processing steps, thereby achieving both high effectiveness and efficiency.
2. We propose a **dynamic convolution kernel mechanism**, where kernel proposals embed critical self-information and spatial position features, enabling the effective separation of neighbouring text instances even under challenging conditions with small spacing or blurred boundaries.
3. We introduce a **novel Orthogonal Learning Loss (OLL)** to directly promote independence among kernel proposals via orthogonal constraints, ensuring clear distinction between text instances.
4. We extend the detection framework by **integrating an OCR module for recognizing text** from the detected arbitrary-shaped regions, thus providing an end-to-end solution for scene text detection and recognition.
5. We validate the proposed method through extensive experiments on publicly available datasets, demonstrating its superior performance and efficiency.

The rest of this article is organized as follows: Section II overviews the related work. Section III elaborates our method. In Section IV, we demonstrate experimental results on several datasets. Finally, we conclude our work in Section V.

II. LITERATURE REVIEW

A. Regression-Based Methods

Regression-based methods typically predict text boxes by regressing the offsets of bounding rectangles, either based on predefined anchors or directly on original pixels. These methods can generally be divided into two types: anchor-based and anchor-free methods. anchor-free methods [6], [22] aim to predict text boxes without relying on predefined anchors, thereby improving model flexibility. The EAST [6] employs a fully convolutional network (FCN) [4] branch for classification and a regression branch to directly predict bounding quadrilaterals using IoU loss [23]. HAM [24] proposes a hidden anchor mechanism that integrates the advantages of anchor-based methods into an anchor-free framework. In contrast, anchor-based approaches focus on designing or learning appropriate anchors to accurately regress text boxes. TextBoxes [20] and TextBoxes++ [21] employ a series of anchors with varying aspect ratios to cover texts of different lengths. Specifically, TextBoxes++ regresses the offsets of four-point quadrilaterals from predefined anchors to better adapt to arbitrarily oriented texts. RRPN [7] adopts rotated anchors (across three scales, three aspect ratios, and six angles) and applies rotated RoI (RRoI) pooling for detecting arbitrarily oriented texts. However, the regression distance and angle formulations in most regression-based methods are strictly constrained to quadrilateral representations, making it challenging for them to effectively handle texts with arbitrary shapes.

B. Connected Component-Based Methods

In object detection, connected component-based techniques [25] are widely used, particularly in scene text detection. These methods typically begin by identifying individual text parts or characters, followed by postprocessing steps to link or group them into final text instances. One such approach, the Connectionist Text Proposal Network (CTPN) [26], modifies the faster R-CNN framework [1] to extract horizontal text components of a fixed width, which are then connected to form dense text groups and generate horizontal text lines. SegLink [27] takes a different approach, decomposing text into two components: segments and links, where a link indicates that two adjacent segments belong to the same text. Character Region Awareness for Text Detection (CRAFT) [29] focuses on detecting text regions by exploring the affinities between characters. TextDragon [30] detects local bounding boxes for text and groups them into distinct text instances based on their geometric

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

relationships. Zhang et al. [8] introduced the use of a graph convolutional network (GCN) to learn and predict the relationships between different text components, allowing for effective grouping. While CC-based methods offer flexibility in adapting to texts of arbitrary shapes, the postprocessing step required for grouping components into final text instances is often slow and complex.

C. Contour-Based Methods

Goal of contour-based techniques [31]–[37] is to directly represent text boundaries in order to recognize arbitrary text. The contours of text instances are modelled by ABCNet [33] and FCENet [34] using curve modelling, such as Bezier-curve and Fourier-curve. Through successive fitting, this method aids in the correct approximation of closed contours. Simple polygon-based text detections are achieved by using TextRay [35], which represents text outlines in the polar coordinate system and proposes an anchor-free, single-shot framework to forecast the geometric parameters. A progressive contour regression framework created especially for text detection in intricate, asymmetrical structures is used by PCR [36]. An adaptive model for boundary deformation is presented by TextBPN [37], which modifies the text boundaries iteratively. However, contour-based approaches typically suffer from poor performance and efficiency when compared to segmentation-based approaches.

D. Segmentation-Based Methods

Text instance contours are extracted from segmentation masks using segmentation-based techniques. These techniques are used in [17], [19], and [38] to segment the pixel-level mask of the text inside each box after first predicting the bounding boxes surrounding the text. Fully Convolutional Networks (FCN) [4] are used in other methods [9], [11], and [14] to predict text masks and to create extra embedding vectors to group pixels inside those masks. PixelLink [9] uses a disjoint-set data structure to link the pixels after predicting eight linkages to evaluate pixel connectivity. TextField [14] groups pixels using a direction field and then does morphological postprocessing. To provide more distinct borders, PSENet [13] first reduces the text region to several scales before progressively enlarging the smallest scale kernel to cover the entire text form. Tian et al. [11] use pixel clustering to predict an embedding map, treating each text occurrence as a cluster. The complex postprocessing needed to group pixels into whole text instances presents a difficulty for segmentation-based approaches, despite their great adaptability to texts of different shapes.

E. Dynamic Kernel Methods

In instance segmentation, AdaptIS [39] predicts point proposals iteratively to produce instances. It chooses a point proposal in each step and creates the corresponding instance by using the AdaIN method [40]. For the same input image, AdaptIS can produce different results by giving AdaIN alternative configurations. AdaptIS's low efficiency stems from its repetitive prediction during inference, despite its flexibility. CondInst [42] uses conditional convolution, also known as dynamic convolution, to build instance-specific filters that capture object information, building on the fully convolutional one-stage (FCOS) framework [41]. By dynamically proposing $s \times s$ kernels (where the input image is divided into $s \times s$ grids) to convolve the feature maps, SOLOv2 [43] simultaneously produces and separates instances, resulting in $s \times s$ instances over $s \times s$ channels. However, the memory consumption would exceed $(w \times h)^2$ if the resolution $(s \times s)$ in SOLOv2 is set to $w \times h$, which corresponds to the width and height of feature maps used in general scene text identification. This would result in a significant computational and memory burden. Moreover, non-maximum suppression (NMS), a costly procedure, is still used by both CondInst and SOLOv2 to remove unnecessary bounding boxes.

F. Text Recognition Networks

Text recognition in scene images faces challenges due to distortions, irregular layouts, and complex backgrounds. In [58], segmentation-based methods using DeconvNet were fine-tuned on the Total-Text dataset, showing success with multi-oriented and curved text but struggling with repetitive backgrounds and poor word separation. Grouping nearby words and limited spatial resolution further reduced recognition accuracy. In [59], STN-OCR combined spatial transformers with recognition in a single network, reducing preprocessing needs. However, it struggled with cluttered scenes, was restricted by a fixed number of detected text lines, and required extensive staged pretraining for convergence. Both methods reveal a need for better separation, improved handling of curved text, and more efficient recognition in complex environments.

III. METHODOLOGY

A. Overview

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Our approach is composed of 5 main parts, as seen in Fig. 2: Feature extraction, Kernel proposal, instance-independent feature map extraction, contour generation, and Text recognition.
To capture precise and resilient feature representations, we use a fully convolutional network (FCN) [4] in combination with a feature pyramid network (FPN) [2]. Figure 3 further illustrates the architecture of the feature extraction subnetwork. For kernel proposal creation, we first identify the connected components (CC) in the expected center map and choose the pixel with the greatest score inside each component as a key point. The embedding feature at each key point's position acts as the kernel proposal. These kernel ideas are then convolved to create instance-independent feature maps. These feature maps are binarized using a predetermined threshold, resulting in the extraction of the contours of recognized text instances. Finally, the selected text sections are fed to an OCR module, which performs text recognition.

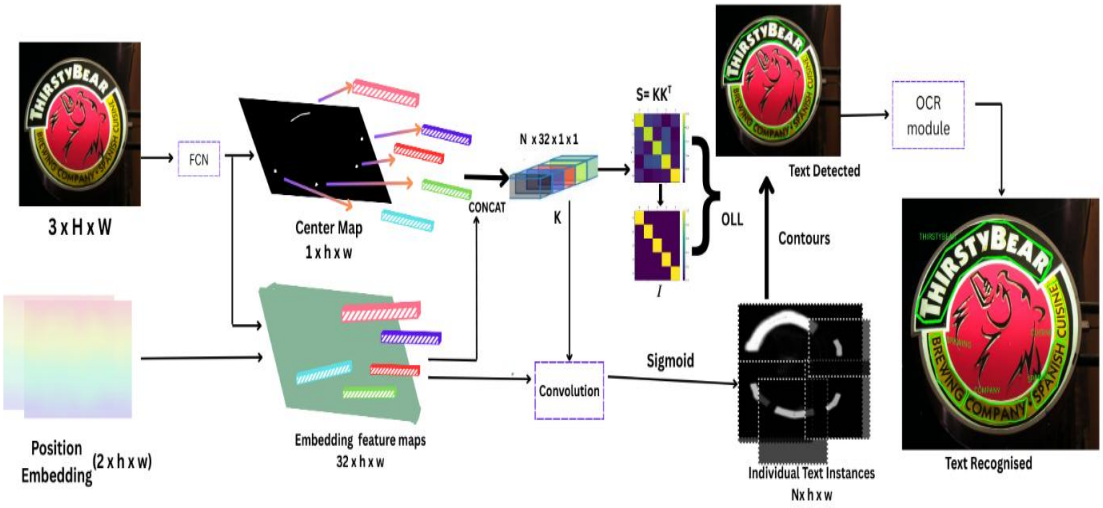


Fig-2: Overview of our framework.

The feature extraction backbone (FCN) is illustrated in Fig. 3. The network adaptively predicts a varying number N key points and generates corresponding embedding vectors to construct kernel proposals. To ensure the independence of these kernel proposals, we propose an Orthogonal Latent Loss (OLL), encouraging the similarity matrix S K K^T to approximate the identity matrix I. Afterward, text masks are predicted to delineate text contours. Later, an OCR module is incorporated to recognize the detected text.

B. Position embedding

Our feature extraction module is built on the widely used FCN [4] architecture, with ResNet-50 [47] serving as the backbone and including a feature pyramid network (FPN) [2] for improved multi-scale feature representation, which is extensively used in text identification domain. However, segmentation approaches that depend on normal convolutional pixel embeddings sometimes struggle to discriminate between several identical instances [48].
To solve this issue, as inspired by [11], [43], [48], and [49], we incorporate a position embedding mechanism into our network to maintain crucial locations' spatial coordinates, as shown in Fig. 2. We create two extra feature map channels that encode each pixel's x- and y-axis positional information.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

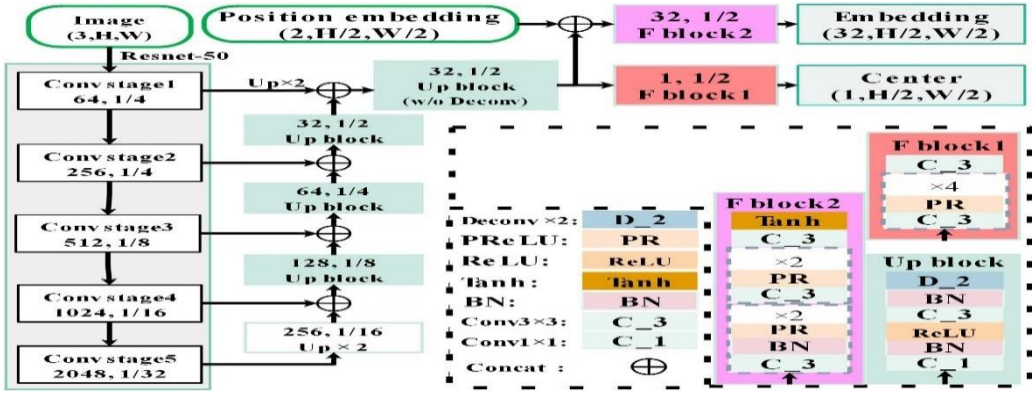


Fig-3: Feature extraction subnetwork has two output branches: 1 channel for center map, other 32 channels for embedding feature maps

The positional values are normalized within the range [−1,1], where the pixel at the start of the x-axis is assigned − 1 and the pixel at the end is assigned 1.
Mathematically, the position embedding of X_i and Y_i can be formulated as:

$$x_i = -1 + \frac{2i}{w-1} \quad \text{where} \quad i \in (0, w-1) \tag{1}$$

$$y_i = -1 + \frac{2i}{h-1} \quad \text{where} \quad i \in (0, h-1) \tag{2}$$

where w and h, respectively, represent the width and height of the output feature maps.

C. Kernel proposal

For segmenting text instances in arbitrary shapes, most existing methods rely on the popular FCN architecture. However, these methods often struggle with texts that have small gaps or unclear boundaries, as shown in Fig. 1(a). To address these challenges, some approaches use embedding strategies to either link [9], [14] or cluster [10], [11] pixels into distinct text instances. The key to their success lies in designing a metric function F(∗,∗)that can determine whether two pixels belong to the same text instance. This function can be formulated as follows:

$$F(e_i,e_j) = \begin{cases} 1, & T(p_i) = T(p_j) \\ 0, & T(p_i) \neq T(p_j) \end{cases} \tag{3}$$

where p_i and p_j represent the pixels at positions i and j, respectively; $T(*)$ indicates the text associated with the pixel; e_i and e_j denote the embedding features of p_i and p_j; and $F(*,*)$ is a learnable function.
From our observation, by identifying an appropriate $F(*,*)$ as a binary classifier, we can effectively classify pixels of different text instances into instance-independent feature maps. Fortunately, this can be implemented using a convolutional layer combined with a sigmoid function. Specifically, we can use a dynamic convolution kernel to classify pixels belonging to a specific text instance and generate instance-independent feature maps for efficient segmentation of each text instance.
As noted earlier, we utilize a set of convolution kernels to classify different text instances into separate instance-independent feature maps. However, traditional convolution methods have a fixed number of kernels, which leads to a fixed number of output feature map channels. Since the number of text instances varies across different images, this fixed setup may cause missed or redundant detections. Drawing inspiration from dynamic kernel methods [39], [43], we propose a dynamic convolution kernel strategy (kernel proposal) to separate text instances into different feature maps. Specifically, we first predict N key points for N text instances and then extract feature maps based on these key points to generate dynamic convolution kernels, which we call kernel

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

proposals. Unlike other dynamic kernel methods, our kernel proposals are orthogonal to each other and form a set of orthogonal basis vectors. Each kernel proposal contains self-information and position information specific to its corresponding text instance. These kernel proposals are then used to convolve the embedding feature maps and generate distinct feature maps for each text instance. This process can be formulated as:

$$O = K * E = [K_1 \ldots K_i \ldots K_N] * E = [P_1 \ldots P_i \ldots P_N] \quad (4)$$

where O represents the output feature maps, with each channel corresponding to the prediction p_i for a specific text instance, and k_i denotes the i-th kernel proposal. The convolution operation * projects the high-dimensional embedding vector E onto the orthogonal basis K. The embedding feature maps Econsist of shared features extracted from the backbone (denoted Fs) and position embeddings (denoted Fp):

$$E = \text{Block2}(F_s \oplus F_p) \quad (5)$$

where $\oplus$ represents the concatenation operation, as shown in Fig. 3. After full training, Fs primarily contains features related to text instances, while non-text features are driven towards zero.

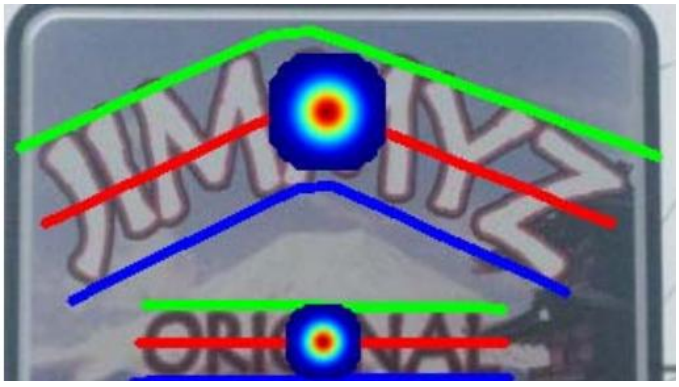
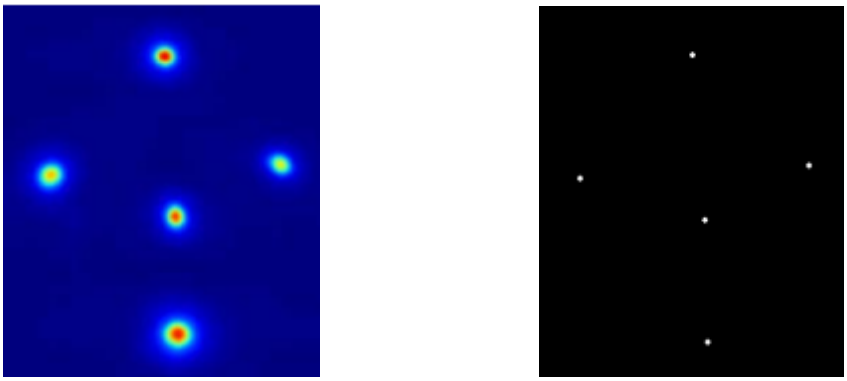


Fig-4: Overview of key points labeling: top lines (green lines), bottom lines (blue lines), center lines (red lines), and center Gaussian heat maps for text instances.

D. Key Points Generation

Using the function F (*,*) from Equation (3) to categorize all pixel pairs would result in a computational complexity of $(w \times h)^2$, making it unsuitable for real-world applications. To overcome this, we choose a single key point from the anticipated Gaussian center maps for each text instance, with each Gaussian peak typically representing a single text instance. These kernel ideas capture essential self-representative properties learned by the network, as well as spatial information provided by position embeddings. This architecture helps us to efficiently segregate neighbouring text instances.



(a) ***(b)***

Fig-5: Examples of key point generation: (a) predicted heat maps; (b) CCs after thresholding; (c) key points that with the scores in each CC.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Unlike normal objects in instance segmentation tasks, text instances frequently lack well-defined closed bounds and may include huge areas that look similar to the background. As a result, our approach may become sensitive to mistakes in key point selection. To address this, we apply a "broad key point" strategy to each text instance. Specifically, we define the center key point using the text's geometric centerline. Using the method described in TextSnake [44], we locate the top (green) and bottom (blue) borders of the text region, compute the centerline (red), and then choose the midpoint of the centerline as the center point.

However, determining a single exact center point remains difficult. Inspired by [45] and [46], we use a Gaussian heatmap to mark key points, as shown in Fig. 4. The Gaussian's radius is defined as the smallest distance between the center point and the text boundary. This produces a larger and smoother heatmap label (Fig. 5(a)). Nonetheless, the heatmap still shows many possible points each instance. To refine the selection, we use thresholding to identify connected components (CCs) in the heatmap, and then choose the pixel with the greatest confidence score inside each CC as the final key point (Fig. 5(b)). The matching kernel proposals are then retrieved from the embedding features at these critical point positions.

E. Loss function

To optimize the prediction of the Gaussian center map and the segmentation of each text instance, we employ the following loss function:

$$L_c(y, \hat{y}, \alpha) = \alpha \times L_{\text{dice}}(y, \hat{y}) + L_{\text{focal}}(y, \hat{y}) + L_{\text{OHEM}}(y, \hat{y}) + L_{\text{BBCE}}(y, \hat{y}) \quad (6)$$

Here, L_{dice} represents the Dice loss [50], L_{focal} is the focal loss [51],L_{OHEM} refers to the cross-entropy loss with Online Hard Example Mining (OHEM) [52], where the ratio of negative to positive pixels is set at 3:1, and L_{BBCE} is the balanced binary cross-entropy loss. y denotes the predicted output, y^ is the ground truth, and α is the weight for the Dice loss term. The individual loss functions for the Gaussian center map and the segmentation task are defined as:

$$L_c^{gc} = L_c(y, \hat{y}, 0) \quad (7)$$

$$L_c^{s} = L_c(y, \hat{y}, 1) \quad (8)$$

To ensure effective separation of different text instances, we first impose the constraint that kernel proposals must be independent of each other within a given image. This guarantees that each kernel proposal only contains the unique features of its corresponding text, without sharing features with other instances. By convolving the embedding feature maps with these kernel proposals, our method can generate distinctive feature maps for each individual text instance.

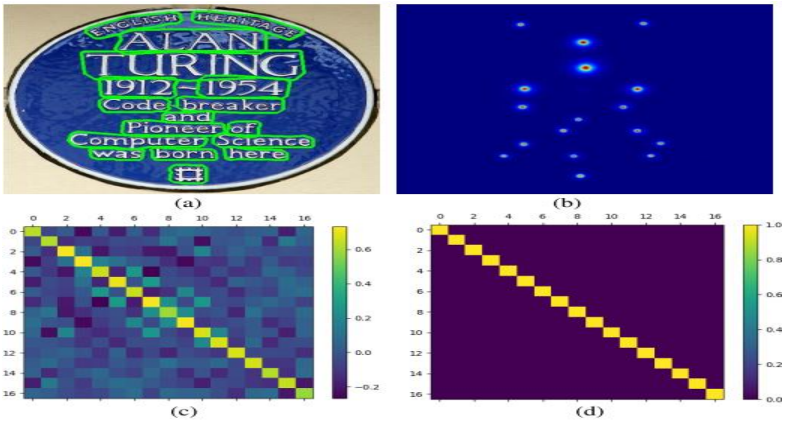


Fig-6: (a) Detected contours in the image. (b) Prediction of Gaussian center map. (c) Similarity matrix (S) between kernel proposals. (d) I is an identity matrix

To express the similarity between kernel proposals in the image, we define the similarity matrix S as:

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

$$S = K K^T = [K_1 ... K_i ....K_N] \times [K_1 ... K_i ....K_N]^T = I \quad (9)$$

Here, K is the set of kernel proposals in the image, S is the resulting similarity matrix (of size N×N), and N is the number of kernel proposals.
To enforce the independence of the kernel proposals, we introduce an orthogonal loss (OLL) L_{OLL} defined as:

$$L_{OLL} = L_{dice}(S, I) + L_{BCE}(S, I) \quad (10)$$

where I is the identity matrix, as shown in Fig. 6(d). L_{dice} and L_{BCEL}_help to measure the similarity between the kernel proposals and the identity matrix, encouraging the kernel proposals to become orthogonal to one another.
Finally, the total loss of our method is computed as:

$$L = L_c^{gc} + L_c^s + \lambda_O \times L_{OLL} \quad (11)$$

Where λ_O is a hyperparameter set to 0.1.

F.OCR Module

The OCR module operates as a key component following the kernel-based text detection system, which generates instance-aware masks that capture the complete geometry of arbitrary-shaped text. These kernel-generated text instances preserve the original shape and spatial arrangement of the text, ensuring clean boundaries even between closely positioned instances, providing essential input for the OCR module.
Using a sequence-to-sequence approach, the OCR module estimates character sequences from the feature maps of word instances. This attention-based recognition adapts to various text orientations and curvatures, leveraging contour information from kernel masks to naturally follow the text's path without needing text rectification.
The OCR module utilizes a recurrent architecture to process kernel-separated text instances. The encoder converts visual features into a fixed dimensional embedding, capturing critical character features and their spatial relationships. This embedding helps the OCR module distinguish character features, even in distorted or curved text.

Fig. 7. Text recognition using OCR

For curved text, kernel masks trace the contours, enabling the OCR to follow the natural sequence of characters without flattening or rectifying the text. The attention-based decoder dynamically allocates attention to different regions of the feature embedding, ensuring the correct reading order even in non-linear text paths.Orthogonal constraints applied during kernel generation through Orthogonal Learning Loss (OLL) ensure clear separation of text instances, improving recognition accuracy. This end-to-end integration of kernel-based detection and OCR recognition eliminates the need for complex post-processing, streamlining the text recognition pipeline and maintaining high accuracy for arbitrary-shaped text.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

IV.EXPERIMENTS

a. Datasets :

TotalText (TotalText) contains 1255 training and 300 testing images. It especially provides curved texts, which are annotated by polygons and word-level transcriptions. It is collected from various scenes, including text-like background clutter and low-contrast texts, and are word-level annotated by polygons for curve text detection task.

CTW-1500 (CTW) consists of 1000 training and 500 testing images. Every image has curved text instances, which are annotated by polygons with 14 vertices.

b. Implementation Details

The pretrained ResNet-50 [47] serves as the network's backbone. Our technique is optimized using Adam [56], with an initial learning rate of 0.0001 and a decline of 0.9 per 100 epochs. Our network is trained on the MLT 2017 dataset [57] before randomly cropping photos to 640 × 640, 832×832 and 1024×1024 sizes.

We will fine-tune our network on the benchmark dataset of 832 × 832 crop images. In addition to DRRG [8], we use general augmentation techniques including crops, rotations, colour variations, and partial flipping. We employ two methods in the training process to obtain the essential information for training kernel proposals: 1) We randomly select one key point per text instance. The sampling probability is calculated using a Gaussian heat map. 2) We select the top-k (k = 50) points in the Gaussian heat map for each text instance, as shown in Fig. 5. In testing, just one point with the highest score is chosen as a kernel proposal for a text instance (CC), as illustrated in Figure 5. Text recognition is then performed using the OCR module. This OCR module uses the kernels from KPN to generate accurate text. All experiments are performed on CPU (Intel Xeon E5-2620 v4 @ 2.10 GHz), numpy 1.24.4 , cv2 4.11.0 and PyTorch 2.6.0+ cpu

c. Ablation Study

To validate our method, we tested images at various scales. All images will be adjusted between short and huge. In this step, we only train our Models on Total-Text with 100 epochs using Adam [56] as the optimizer.

1. KPN vs. FCN:

To confirm our KPN's efficiency, we contrast it with an FCN-based technique, which is comparable to our feature extraction subnetwork in Figure 3. The FCN-based method only uses the center branch to learn text masks, excluding the embedding branch. We trained and evaluated our KPN and FCN-based methods on Total-Text over 300 epochs to ensure fair comparisons. Table I shows that our KPN outperforms FCN on Total-Text while maintaining efficiency (FPS). Figure 8 displays representative detection findings.

Method	Recall	Precision	Hmeans
FCN-640	62.65	66.58	64.55
KPN-640	**65.48**	**83.19**	**73.28**
FCN-832	66.56	65.40	65.98
KPN-832	**71.26**	**84.58**	**77.35**
FCN-1024	64.16	67.99	66.02
KPN 1024	**74.55**	**85.00**	**79.43**

Table I: Ablation Study of KPN & FCN on TotalText

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

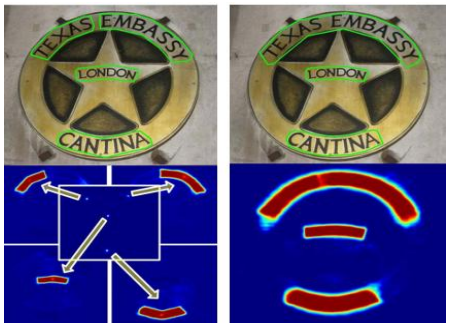


Fig. 8 Representative detecting results of KPN and FCN. (a) KPN. (b) FCN

2. Center Point Versus Center Region:

Segmentation-based approaches frequently use the strategy of decreasing the central section of text [11], [12]. Here, we perform an ablation study to compare our center-point strategy to the center-region strategy. Table II shows that the center point strategy outperforms the center region strategy by a significant margin. At 1024 resolution, our method outperforms FCN by 13.41% in H-means (KPN-1024 79.43% vs. FCN-1024 66.02%). Figure 9 demonstrates that our center point method accurately separates adjacent text occurrences. However, the center region strategy suffers from overlapping candidate text regions.

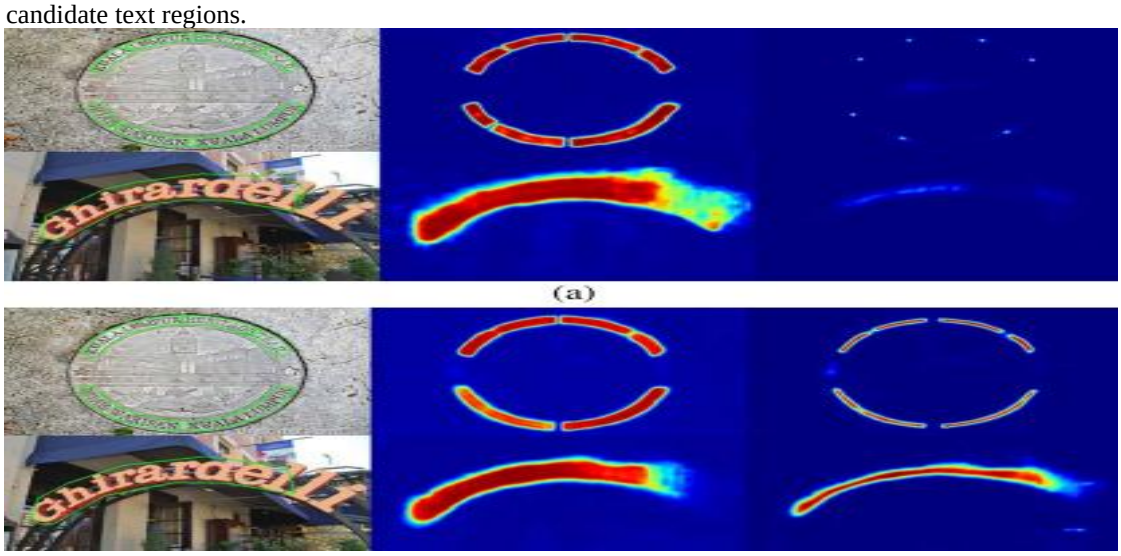


Fig-9: Left column: detection results; middle column: sum of all the masks in KPN; right column: center points/regions. (a) Center point. (b) Center region

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Scale	Cr	Cp	Pe	Recall (R)	Precision (P)	H-mean (H)	FPS
KPN-640	✓	✗	✓	59.36	71.53	64.88	22.15
KPN-640	✗	✓	✓	**65.48**	**83.19**	**73.28**	**23.15**
KPN-832	✓	✗	✓	66.22	71.88	68.93	14.39
KPN-832	✗	✓	✓	69.64	82.13	75.53	15.84
KPN-832	✗	✓	✓	**71.26**	**84.58**	**77.35**	15.17
KPN-1024	✓	✗	✓	68.27	71.76	69.97	9.76
KPN-1024	✗	✓	✓	73.35	83.54	78.11	10.88
KPN-1024	✗	✓	✓	**74.55**	**85.00**	**79.43**	10.54

Table-2: ablation study of center region (cr), center point (cp), position embedding (ps), and test image scale on total text (only trained on total-text with 300 epochs). "r," "p," and "h" represent "recall," "precision," and "h-mean," respectively

d. Comparisons With the State-of-the-Arts :

1. Comparison with Datasets :

Comparisons of the proposed KPN with the state-of-theart (SOTA) methods on Total-Text [53], CTW-1500 [54], are listed in Tables III,IV. TotalText and CTW-1500 consists of images with curve texts.

Total-Text primarily consists of photos with word-level annotations on curve texts. We evaluated two sets of resolutions [512, 640] and [512, 832], known as KPN-640 and KPN-832, with thresholds Tc = 0.2 and Ti = 0.6. 9(a) demonstrates that the predicted results for text instances are clearly separated. We utilize the official evaluation. Table III shows the results of running the script in Total-Text [53]. Our KPN outperforms DRRG [8] by 1.38% in H-mean, and ContourNet by 11.23 FPS in speed. The Pixel Aggregation Network (PAN) outperforms ResNet-50 and DB-640, which only has one FCN branch.KPN surpasses PAN-640 and DB by 2.11% and 2.41%, respectively, in H-means analysis. KPN has comparable efficiency to PAN and DB-640. Overall, our KPN outperforms SOTA on Total-Text.

Methods	Paper	Ext	R	P	H	FPS
TextSnake	ECCV'18	Syn	74.5	82.7	78.4	-
PSENet-1s	CVPR'19	MLT	79.7	84.8	82.2	-
CRAFT	CVPR'19	MLT	79.9	87.6	83.6	-
DB	AAAI'20	MLT	83.2	86.9	85.0	22.7
ABCNet†	CVPR'20	MLT	81.4	84.4	82.9	16.3
TextRay	MM'20	-	77.9	83.2	80.4	-
KPN-640	-	MLT	82.33	83.08	82.70	22.73
KPN-832	-	MLT	82.83	84.00	83.41	16.30
KPN MS	-	MLT	**87.04**	**84.17**	**87.60**	-

Table-3: Experimental results on Total-Text. ∗ and "MS" represents multiscale test and † denotes the textspotter-based methods

CTW-1500 primarily includes photos with curve texts and line-level comments. We tested sizes [512, 640] and [512,832] with thresholds Tc=0.2 and Ti=0.625. Specifically, KPN-640 and KPN-832. CTW-1500 annotations at the line level may make predicting the text line's center problematic. Our KPN may not detect really long text. Table IV includes extensive experimental data. Table IV shows that our KPN surpasses ContourNet [60] by 1.32% in H-means and significantly outperforms it in efficiency

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

(16.30 FPS vs. 4.5 FPS). And our KPN is comparable to DRRG [8] (84.27% vs. 84.45%). Our KPN shows promising results on CTW-1500.

Fig-10: Detected results on Total-Text, CTW-1500, and ICDAR 2015. The second row shows the predicted centerpoint maps (in center red boxes), and the individual text instances in different channels. (a) Total-Text. (b) CTW-1500.

Methods	Paper	Ext	R	P	H	FPS
TextSnake	ECCV'18	Syn	85.3	67.9	75.6	-
PSENet-1s	CVPR'19	MLT	79.7	84.8	82.2	3.9
CRAFT	CVPR'19	MLT	81.1	86.0	83.5	-
DB	AAAI'20	MLT	83.2	86.9	85.0	22.7
ContourNet	CVPR'20	MLT	83.9	84.0	83.9	2.5
ABCNet†	CVPR'20	MLT	81.4	84.4	82.9	16.3
KPN-640	-	MLT	81.4	84.0	83.4	24.25
KPN-832	-	MLT	81.8	84.4	83.1	16.30
KPN MS	-	MLT	86.44	84.04	85.22	-

Table-4: Experimental results on CTW-1500

Experiments on Total-Text and CTW-1500 show that KPN reaches high efficiency and performance levels. Our postprocessing-free segmentation system eliminates the need for costly postprocessing, resulting in increased efficiency. In addition, as demonstrated Fig. 10(a) and (b) show that KPN effectively separates neighbouring text instances by categorizing them into instance-independent feature maps. The effectiveness resides in the orthogonality of kernel proposals. When kernel proposals are orthogonal, the network learns self-knowledge and embeds position information, rather than sharing information from other texts.

2. Visual Comparison

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

In Fig. 11, we compare the presented method to other SOTA systems, such as PSENet [13] and DB [12].
Intermediate visual comparison. Figure 11 shows the final detections, the Gaussian center in KPN, the minimum text kernel in PSENet, and the probability map in the database. PSENet employs the min text kernel to separate nearby text, followed by the scale expansion algorithm to reconstruct text instances. However, this method is inefficient and frequently fails. Therefore, PSENet's detection speed and performance are not sufficient. DB separates neighbouring text using a probability map created by reducing annotations with the Vatticlipping method [15]. Then, It employs the inverse transformation of the Vatti clipping method [15] to obtain the detection boundaries. DB's detection speed is quite rapid. DB's detection bounds may not adequately encompass text due to human scaling restrictions, as illustrated in Fig. 11(c).

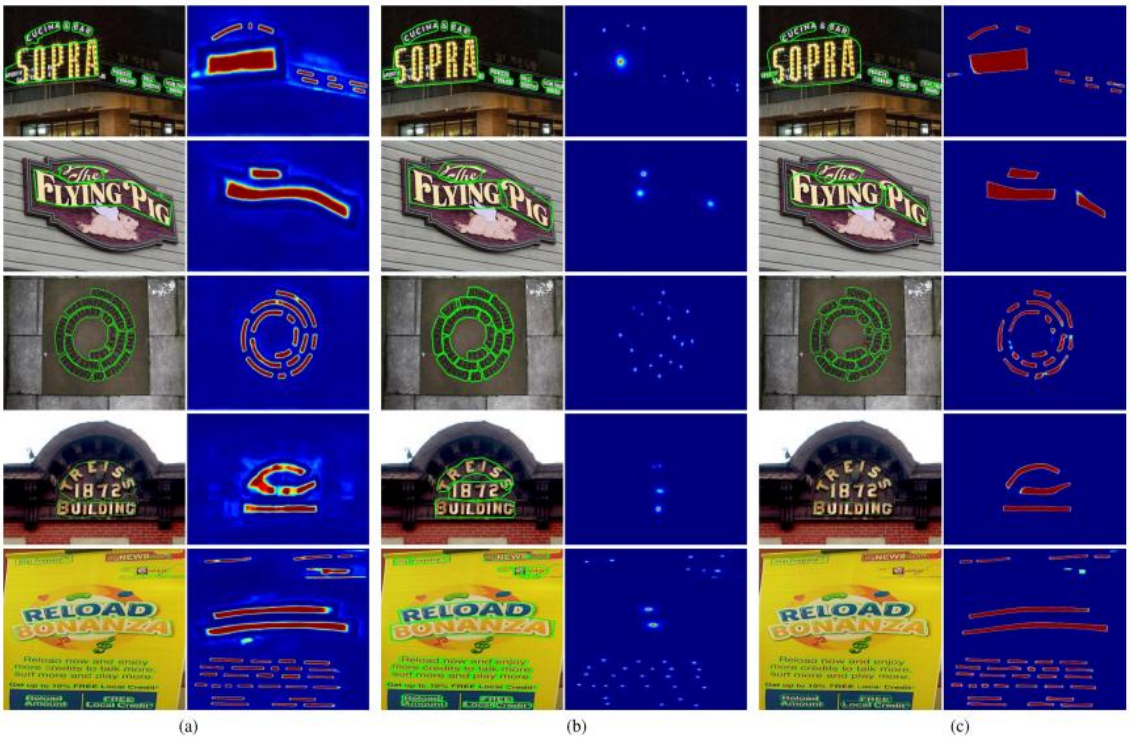


Fig. 11. Some visual comparison results with PSENet [13] and DB [12], where the results of PSENet and DB are reproduced by their official open-source code and model, where DB use the ResNet-50 with deformable convolution as Bockone and PSENet use the ResNet-50 as Bockone. (a) PSENet. (b) KPN. (c) DB.

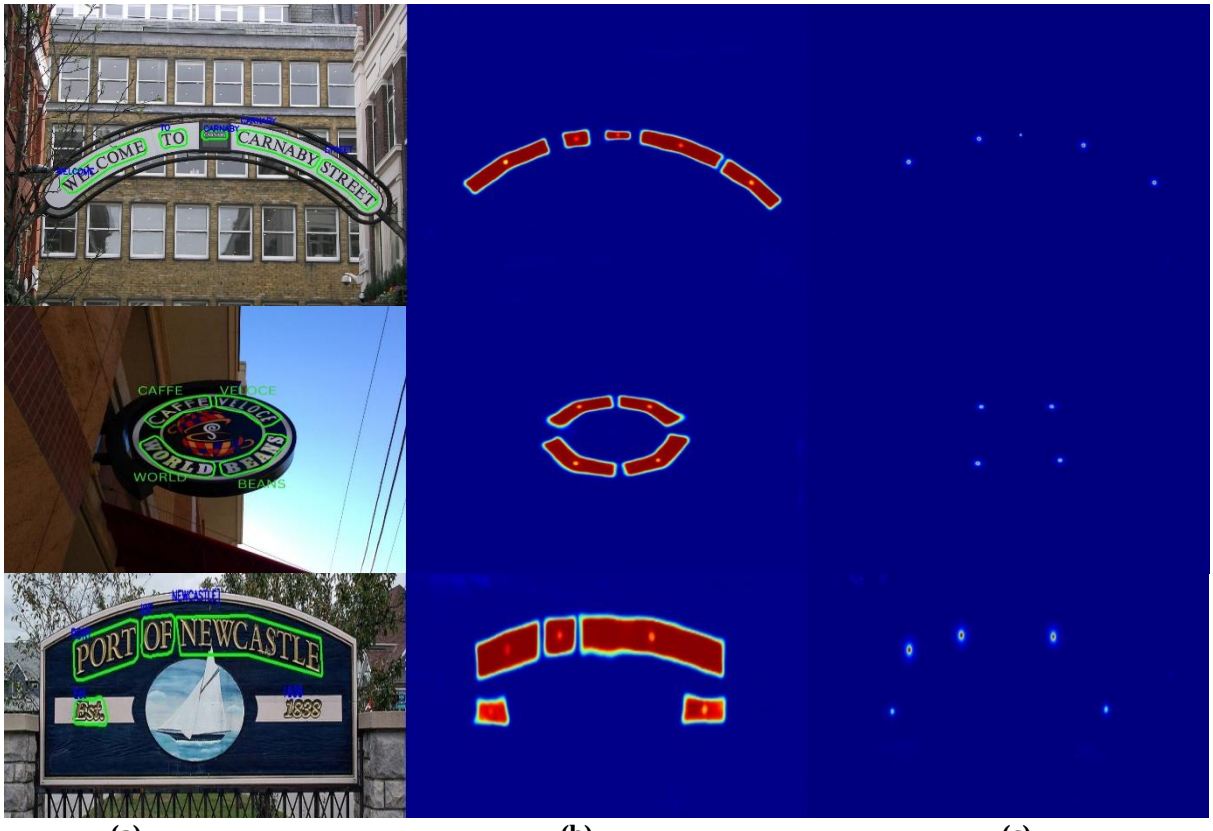
Inaccurate detection boundaries can cause issues in applications, such as incorrect character recognition in optical character recognition (OCR) systems, without impacting assessment performance. Our technique effectively separates surrounding texts and detects boundaries accurately and quickly, as seen in Fig. 11(b) and Tables III and IV.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

e. Results of Text Recognition

Total-Text Dataset Performance

Introduced OCR-KPN method demonstrates remarkable advancements in curved text recognition, particularly excelling on the Total-Text dataset. OCR-KPN achieves a precision of 88.7%, recall of 88.0%, and an F-measure of 88.5% in table V, significantly surpassing strong baselines like TextSnake, EAST and CRAFT, which previously set the benchmark for handling curved and irregular text. Unlike earlier methods that often struggled with complex backgrounds and highly curved text, OCR-KPN's enhanced kernel prediction enables more precise localization and separation of text instances as shown in the fig 12. Its high precision ensures minimal false positives, while the strong recall value indicates fewer missed detections, making it exceptionally reliable for real-world curved text scenarios.



(a) (b) (c)

Fig. 12. OCR Enhanced KPN results from Total Text Datset (a) Final result with text recognition (b) Kernels generated (c) Guassian Centre map

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Method	Precision	Recall	F-Measure (H)
SegLink[60]	30.3	23.8	26.7
EAST[61]	50.0	36.2	42.0
Baseline(DeconvNet)[62]	33.0	40.0	36.0
DMPNET[63]	69.9	56.0	62.2
CTD[64]	74.3	65.2	69.5
CTD+TLOC[65]	77.4	69.8	73.4
MarkSpotter[66]	69.0	55.0	61.3
TextSnake[67]	82.7	74.5	78.4
CRAFT[68]	87.6	79.9	83.6
OCR-KPN	**88.7**	**88.0**	**88.5**

Table V: Experimental results on Total Text

CTW-1500 DataSet Performance

On the CTW-1500 dataset, OCR-KPN continues to maintain its dominance with a precision of 87.3%, recall of 85.6%, and an F-measure of 87.7% in table VI. These results clearly outperform leading methods like CRAFT and TextSnake, reinforcing OCR-KPN's ability to handle long, curved, and complex text instances. The method's robust kernel-based localization strategy proves effective even against challenging backgrounds and varying text orientations, providing consistently superior detection and recognition performance across diverse and difficult scenes.as shown in fig 13

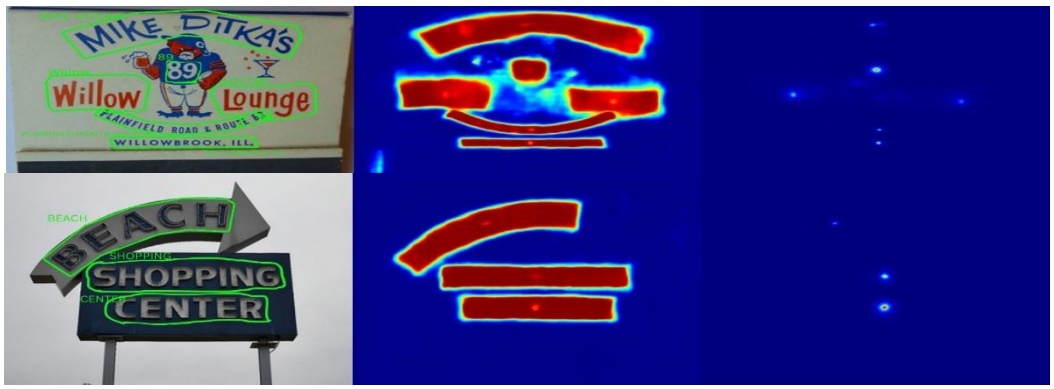


Fig. 13. OCR Enhanced KPN results from CTW-1500 Datset (a) Final result with text recognition (b) Kernels generated (c) Guassian Centre map

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Method	Precision	Recall	F-Measure (H)
SegLink[60]	42.3	40.0	40.8
EAST[61]	78.7	49.1	60.4
DMPNET[63]	69.9	56.0	62.2
CTD[64]	74.3	65.2	69.5
CTD+TLOC[65]	77.4	69.8	73.4
MarkSpotter[66]	-	61.3	-
TextSnake[67]	67.9	85.3	75.6
CRAFT[68]	86.0	81.1	83.5
OCR-KPN	**87.3**	**85.6**	**87.7**

Table VI: Experimental results on CTW-1500

Overall, OCR-KPN sets a new standard for scene text recognition, providing an effective and practical solution for extracting text from natural images with unmatched accuracy and reliability. Its consistent superiority across datasets like Total-Text and CTW-1500 highlights its strong generalization capabilities and adaptability to a wide range of text shapes, sizes, and backgrounds. The balanced combination of high precision and recall ensures OCR-KPN is both accurate and dependable, making it a highly suitable choice for real-world applications requiring robust curved and irregular text recognition.

CONCLUSION

In this paper, we proposed OCR Enhanced Kernel Proposal Network (OCR-KPN) designed for arbitrary-shaped text detection and recognition. Our KPN introduces a dynamic convolution kernel strategy that efficiently separates neighbouring text instances by classifying them into instance-independent feature maps, eliminating the complex post-processing. The Orthogonal Loss Layer (OLL) enforces independence between kernel proposals through orthogonal constraints, enhancing robustness against tiny gaps and unclear boundaries. The kernel proposals effectively capture both self-information and positional cues, significantly improving text instance separation. Building upon, by integrating the kernels generated by KPN into the OCR module, resulting in a powerful system capable of accurately recognizing arbitrary-shaped scene text. Extensive experiments on challenging datasets, including Total-Text and CTW-1500, demonstrate the superior.

REFERENCES

[1]. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[2]. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. CVPR, Jul. 2017, pp. 936–944.
[3]. Q. He, X. Sun, Z. Yan, and K. Fu, “DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images,” IEEE Trans. Geosci. Remote Sens., vol. 60, pp. 1–16, 2022.
[4]. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. CVPR, Jun. 2015, pp. 3431–3440.
[5]. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. MICCAI, vol. 9351, N. Navab, J. Hornegger, W. Wells, and A. F. Frangi, Eds., 2015, pp. 234–241.
[6]. X. Zhou et al., “EAST: An efficient and accurate scene text detector,” in Proc. CVPR, Jul. 2017, pp. 2642–2651
[7]. J. Ma et al., “Arbitrary-oriented scene text detection via rotation proposals,” IEEE Trans. Multimedia, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[8]. S.-X. Zhang et al., "Deep relational reasoning graph network for arbitrary shape text detection," in Proc. CVPR, Jun. 2020, pp. 9696–9705.
[9]. D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting scene text via instance segmentation," in Proc. AAAI, 2018, pp. 6773–6780.
[10]. W. Wang et al., "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in Proc. CVPR, Oct. 2019, pp. 8439–8448.
[11]. Z. Tian et al., "Learning shape-aware embedding for scene text detection," in Proc. CVPR, Jun. 2019, pp. 4234–4243.
[12]. M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in Proc. AAAI, 2020, pp. 11474–11481.
[13]. W. Wang et al., "Shape robust text detection with progressive scale expansion network," in Proc. CVPR, Jun. 2019, pp. 9336–9345.
[14]. Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," IEEE Trans. Image Process., vol. 28, no. 11, pp. 5566–5579, Nov. 2019.
[15]. B. R. Vatti, "A generic solution to polygon clipping," Commun. ACM, vol. 35, no. 7, pp. 56–63, 1992.
[16]. K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in Proc. ICCV, 2017, pp. 2980–2988.
[17]. Y. Dai et al., "Fused text segmentation networks for multi-oriented scene text detection," in Proc. ICPR, Aug. 2018, pp. 3604–3609.
[18]. C. Zhang et al., "Look more than once: An accurate detector for text of arbitrary shapes," in Proc. CVPR, Jun. 2019, pp. 10552–10561.
[19]. M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask TextSpotter v3: Segmentation proposal network for robust scene text spotting," CoRR, vol. abs/2007.09482, pp. 1–20, Jul. 2020.
[20].] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in Proc. AAAI, 2017, pp. 4161–4167.
[21]. M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," IEEE Trans. Image Process., vol. 27, no. 8, pp. 3676–3690, Aug. 2018.
[22]. W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in Proc. ICCV, Oct. 2017, pp. 745–753.
[23]. J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in Proc. ACM MM, Oct. 2016, pp. 516–520.
[24]. J.-B. Hou et al., "HAM: Hidden anchor mechanism for scene text detection," IEEE Trans. Image Process., vol. 29, pp. 7904–7916, 2020.
[25]. X. Sun, P. Wang, C. Wang, Y. Liu, and K. Fu, "PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery," " ISPRS J. Photogramm. Remote Sens., vol. 173, pp. 50–65, Mar. 2021.
[26]. Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in Proc. ECCV, 2016, pp. 56–72.
[27]. B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in Proc. CVPR, Jul. 2017, pp. 3482–3490.
[28]. J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, and X. Bai, "SegLink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping," Pattern Recognit., vol. 96, Dec. 2019, Art. no. 106954.
[29]. Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in Proc. CVPR, Jun. 2019, pp. 9365–9374.
[30]. W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in Proc. ICCV, Oct. 2019, pp. 9075–9084.
[31]. X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in Proc. ICCV, Jun. 2019, pp. 6449–6458.
[32]. Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, and Y. Zhang, "ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection," in Proc. CVPR, Jun. 2020, pp. 11753–11762.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[33]. Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Realtime scene text spotting with adaptive Bezier-curve network," in Proc. CVPR, Jun. 2020, pp. 9806–9815.
[34]. Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in Proc. CVPR, Jun. 2021, pp. 3123–3131.
[35]. F. Wang, Y. Chen, F. Wu, and X. Li, "TextRay: Contour-based geometric modeling for arbitrary-shaped scene text detection," in Proc. 28th ACM Int. Conf. Multimedia, C. W. Chen et al., Eds., Oct. 2020, pp. 111–119.
[36]. P. Dai, S. Zhang, H. Zhang, and X. Cao, "Progressive contour regression for arbitrary-shape scene text detection," in Proc. CVPR, Jun. 2021, pp. 7393–7402.
[37]. S.-X. Zhang, X. Zhu, C. Yang, H. Wang, and X.-C. Yin, "Adaptive boundary proposal network for arbitrary shape text detection," in Proc. ICCV, Oct. 2021, pp. 1305–1314.
[38]. Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "IncepText: A new inception-text module with deformable PSROI pooling for multioriented scene text detection," in Proc. IJCAI, Jul. 2018, pp. 1071–1077.
[39]. K. Sofiiuk, K. Sofiyuk, O. Barinova, A. Konushin, and O. Barinova, "AdaptIS: Adaptive instance selection network," in Proc. ICCV, Oct. 2019, pp. 7354–7362.
[40]. T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proc. CVPR, Jun. 2019, pp. 4401–4410.
[41]. Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9626–9635.
[42]. Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in Proc. ECCV, in Lecture Notes in Computer Science, vol. 12346, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., 2020, pp. 282–298.
[43]. X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic and fast instance segmentation," 2020, arXiv:2003.10152.
[44]. S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in Proc. ECCV, 2018, pp. 19–35.
[45]. H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in Proc. ECCV, vol. 11218, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018, pp. 765–781.
[46]. K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in Proc. ICCV, Oct. 2019, pp. 6568–6577.
[47]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. CVPR, Jun. 2016, pp. 770–778.
[48]. D. Novotný, S. Albanie, D. Larlus, and A. Vedaldi, "Semi-convolutional operators for instance segmentation," in Proc. ECCV, in Lecture Notes in Computer Science, vol. 11205, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018, pp. 89–105.
[49]. R. Liu et al., "An intriguing failing of convolutional neural networks and the CoordConv solution," in Proc. NeurIPS, 2018, pp. 9628–9639.
[50]. F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in Proc. 3DV, Oct. 2016, pp. 565–571.
[51]. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proc. ICCV, Oct. 2017, pp. 2999–3007.
[52]. A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in Proc. CVPR, Jun. 2016, pp. 761–769.
[53]. C. K. Ch'ng and C. S. Chan, "Total-Text: A comprehensive dataset for scene text detection and recognition," in Proc. ICDAR, Nov. 2017, pp. 935–942.
[54]. Y. Liu, L. Jin, S. Zhang, and S. Zhang, "Detecting curve text in the wild: New dataset and new solution," CoRR, vol. abs/1712.02170, pp. 1–9, Dec. 2017.
[55]. D. Karatzas et al., "ICDAR 2015 competition on robust reading," in Proc. ICDAR, Aug. 2015, pp. 1156–1160.
[56]. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, 2015, pp. 1–15.
[57]. Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, and Y. Zhang, "ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection," in Proc. CVPR, Jun. 2020, pp. 11750–11759

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[58]. Chee Kheng Ch'ng and Seng Chan, "Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition" arXiv:1710.10400v1 [cs.CV] 28 Oct 2017
[59]. Christian Bartz , Haojin Yang, Christoph Meinel "STN-OCR: A single Neural Network for Text Detection and Text Recognition" arXiv:1707.08831v1 [cs.CV] 27 Jul 2017
[60]. Shi, B., Bai, X., Belongie, S.: Detecting oriented text in natural images by linking segments. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
[61]. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: An efficient and accurate scene text detector. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
[62]. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. (2015) 1520–1528
[63]. Liu, Y., Jin, L.: Deep matching prior network: Toward tighter multi-oriented text detection. (2017)
[64]. Yuliang, L., Lianwen, J., Shuaitao, Z., Sheng, Z.: Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170 (2017)
[65]. L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng. Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170, 2017. 2, 6, 11
[66]. P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. arXiv preprint arXiv:1807.02242, 2018.
[67]. S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. arXiv preprint arXiv:1807.01544, 2018.
[68]. Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee∗ Clova AI Research, NAVER Corp. "Character Region Awareness for Text Detection" arXiv:1904.01941v1 [cs.CV] 3 Apr 2019