Engineering and Technology Journal for Research and Innovation (ETJRI)
ISSN 2581-8678, Vol. VII, Issue II, July 2025

BA-Text: Boundary Aware Text Framework for Text Detection in the Wild

Shilpi Goyal[1], Deepak Motwani[2]
Amity School of Engineering & Technology, Amity University Madhya Pradesh, Gwalior, India, 474005.
agarwal.shilpi1@gmail.com, dmotwani@gwa.amity.edu,

***Abstract*— Text detection in natural scene images, commonly referred to as wild text detection, has emerged as a crucial and challenging domain in computer vision due to its wide range of applications in different areas such as autonomous driving, scene understanding, augmented reality, and assistive technologies. Wild text characterises high variation in text orientation, scale, shape, font, color, layout, background clutter, and illumination conditions. These varied irregularities pose challenges to existing detection approaches. Traditional bottom-up segmentation methods tend to predict pixel-level features and suffer from high sensitivity to noise, while regression-based methods frequently fail to capture complex geometric shapes of text. To handle challenges, this work introduces BA-Text, a novel boundary-aware text detection framework designed to generate tight and accurate boundaries for text in natural scene images.**

The proposed BA-Text framework adopts a top-down, kernel-aware architecture that integrates multi-scale features to represent text instances more effectively. At its core, BA-Text employs a ResNet50 backbone for robust feature extraction combined with a Feature Pyramid Network (FPN) to enhance multi-level feature fusion. This design ensures that fine-grained local features, intermediate-level representations of text regions, and global contextual attributes are all simultaneously considered. Unlike conventional methods, BA-Text introduces a kernel shrinking mechanism and mid-line prediction strategy to better capture the geometric attributes of text, particularly curved or irregularly shaped instances. Kernel shrinking helps eliminate noise by focusing on the central structure of text regions, while mid-line prediction provides structural guidance for aligning text contours more accurately. Together, these innovations enable BA-Text to represent arbitrary-shaped text with high fidelity.

The pipeline of BA-Text consists of four key components: feature extraction using ResNet50, multi-path feature fusion via FPN, kernel and text map generation, and post-processing. The network outputs both text region maps and kernel maps, which are then refined to predict tight text boundaries. The design emphasizes a boundary-aware mechanism that goes beyond conventional bounding-box representations and instead focuses on contour-level accuracy. The objective function is formulated with a combination of classification and regression losses, optimized through a smoothed loss function to enhance stability during training. Extensive experiments were conducted using two widely recognized benchmarks—Total-Text and CTW1500, which include horizontal, multi-oriented, and curved text instances. BA-Text achieved a precision of 90.1%, recall of 83.1%, and F-measure of 86.5% on Total-Text, while obtaining precision of 87.7%, recall of 85.3%, and F-measure of 86.5% on CTW1500. These results confirm that BA-Text consistently outperforms or remains competitive with state-of-the-art methods such as PSENet, TextSnake, CRAFT, and DBNet. In particular, the recall performance highlights the framework's ability to capture more text instances in challenging wild environments, a critical improvement over prior methods that frequently miss curved or closely spaced text. The contributions of this work are threefold: (1) the introduction of a boundary-aware framework that enhances the robustness of wild text detection, (2) the integration of kernel shrinking and mid-line prediction to provide fine-grained geometric guidance, and (3) a comprehensive evaluation on multiple benchmarks demonstrating competitive performance against state-of-the-art approaches. Overall, BA-Text offers a generalizable and efficient solution for detecting arbitrary-shaped text in natural images and holds strong potential for integration into real-world applications where accurate text detection is critical.

***Keywords*—** Scene Text Detection, Wild text, Boundary aware Text, Kernel Shrinking, Feature Pyramid Network, Computer Vision

I. INTRODUCTION

TEXT embedded in wild images is a key topic in computer vision with number of applications. This kind of processing has several applications encompassing car plate detection along with signboard recognition and self-driving technology and environment comprehension. With the speedy advancement in object detection and image segmentation, scene text detection[1][2][3] achieves significant progress. But due to the irregularity in text in the form of its nature, font, color (foreground/ background), scale and orientation as shown in Figure 1, more efforts are required to unlock the complexities. Deep learning areas facilitate to deal with problems like object detection, text classification, and segmentation-based approaches. Text based object detection and text classification methods use to annotate each and every character while labelling the text. A normal informative image typically has at-least 20 characters, which is tedious

* Corresponding Author:- agarwal.shilpi1@gmail.com/

Engineering and Technology Journal for Research and Innovation (ETJRI)
ISSN 2581-8678, Vol. VII, Issue II, July 2025

task to label and handle at pre-stage. Segmentation-based approaches are pixel based prediction of instances that adopt bottom to top methodology and are very prone to noise. The main focus of the segmentation-based algorithms in use today is the prediction of isolated single pixels.

To get detection results, CT-Net[4] forecasts the centripetal shift map and kernel probability map. By anticipating the lateral and thin veins, TextLeaf[5] reconstructs the instances while concentrating on the text kernel mask. Here, to detect wild text, the focus is to localize the text regions and the number of text instances. To localize the regions and instances, a novel framework is proposed in this paper for identifying text boundaries in wild image scenarios (BA-Text), which leverages segmentation-based top-down scheme to determine the attributes or parameters of text. This paper utilizes combination of TextSnake [6] and Textfusenet [7] pipeline for wild-text text detection. ResNet [8] backbone for feature extraction followed by feature pyramid network (FPN) [9] to obtain multi-scale feature maps are used as feature fusion module to enhance the detection capabilities with varying scales. The proposed BA-Text model works to attain the features at pixel level, intermediate level and global level. Pixel level information is suitable for fine-grained local features; intermediate level is worth capturing text-regions and global level to capture all instances in scene image.

Text form in the form of dotted lines	Sparse Text Different font	Complex Background	School Board	Arbitrary Orientation

The main contributions of this work are as follows: 1) BA-Text model has been designed to boost text detection task. 2) A study was conducted to compare deep learning algorithms for determining the text detection performance of BA-Text in applications. 3) To assess the impact of the designed model, qualitative results are displayed. 4) On two publicly available benchmark datasets, the final proposed model was contrasted with state-of-the-art (SOTA) text detection methods, which include numerous horizontal, rotated, and irregular-shaped texts.

The rest of the paper is structured as follows. Section 2 presents some related work. Section 3 describes the detail of feature extraction, feature fusion module, and BA-Text model. Furthermore, the training details and loss functions will be discussed. Section 4 covers the experimental setup and shows the ability of proposed model. Furthermore, the outcomes of extensive experiments are contrasted with SOTA techniques. Section 5 concludes with a summary of the entire paper.

II. Related Work

Deep learning techniques have accelerated text detection research which has resulted in important developments during recent years. Existing techniques are generally divided into three categories: segmentation-based, regression-based, and hybrid approaches.

A. Regression-based methods

Most regression-based text detection techniques obtain their inspiration from object detection frameworks Faster-RCNN[11] and SSD[12] that identify text-candidates bounding boxes from text regions directly. DeepText[13], utilizes Location-Awareness-Attention Network to focus on text proposals. TextBoxes[14] use convolution kernels and anchor revision to detect texts. In order to handle multi-directional text, TextBoxes++ [15] was then suggested, adding an angle parameter. [16] separated text as rotated text and quadrangle text to predict different parameters based on FCN[17]. In order to identify oriented texts, Liao et al. proposed RRD[18] which made use of rotation-sensitive features. To achieve detection results, He et al. proposed SSTD[19], which made use of an auxiliary loss and an attention module. Text instances were represented by SegLink [20] as segments and links that combined segments based on link predictions. The development of most of the aforementioned techniques is hampered by complex post-processing. The irregular structure of the text makes it hard to use the already available techniques. The method of progressive contour regression deals with this issue by performing iterative updates of text contours[21].

Horizontal text is the initial output, which is progressively optimized to produce irregularly shaped and multidirectional text. Tian et al. proposed CTPN[22], which uses a Balanced Region Proposal Network for geometric proposals and a Deformable Morphology Semantic Network for semantic proposals that allow an extensive investigation of text layouts. In order to generate text proposals, Wang et al. proposed ContourNet[23] which took into account the fact that the text aligned between two orthogonal directions and concentrated on IoU values between predicted and ground-truth bounding boxes. Text contours were represented by the Fourier Signature Vector in [24] and the Bezier Curve in [25]. Irregular text shapes are manageable with current methods but these methods become less effective because of their complex structure.

B. Segmentation-based methods

The critical goal of segmentation-based approaches is to predict whether a pixel is text. PSENet[26] represented text as different scale kernels and reconstructed text instances according to a progressive expansion algorithm. Lyu [27] proposed an approach that generates candidate boxes by grouping corner points and assess them using region segmentation. Then, the candidate boxes were assessed by region segmentation and suppressed by NMS. TextField[28] predicted the direction field while segmenting the text. It utilized the direction field to separate geographically close texts. CRAFT[29] modelled text instance by judging the proximity of the characters to each other. PixeLink[30] predicted pixel score map and the relationships with surrounding pixels to detect text. LeafText[5] treated text instance as leaf and utilized main, lateral, and thin veins to form text. The above approaches perform well for dealing with irregular-shaped texts but still lack efficiency. DBNet[31] segmented the score and regressed the threshold to surprise the result of DB. Benefiting from that, DB module can be removed during inference and adopted a lightweight backbone, it achieved excellent performance while maintaining a high inference speed. On top of that, DBNet++[32] introduced an attention mechanism that improves detection accuracy with minimal effect on speed. CM-Net[33] proposed a novel text kernel representation named concentric mask and learned some auxiliary features to assist in detecting text. The PAN[34] model built its backbone lightweight while incorporating FFM and FPEM to improve feature strength. The framework suggested equivalent vector predictions together with trainable text post-processing for restructuring. TextSnake [6] represents text as a snake. It utilized circles to describe text components. Although connected-component-based approaches work well when dealing with irregular-shaped texts, the complex merging process remains an open problem.

C. Hybrid Methods

Hybrid methods take the benefits of regression and segmentation-based methods. Zhou et al. proposed EAST [16] system first operates at the pixel level for the identification of text-containing regions. This helps in locating the text areas within the image. Then, use regression to forecast the coordinates of the bounding boxes that enclose the text. This step refines the text detection by providing precise quadrilateral shapes around the text regions. Liao et al. proposed DB [32] that integrates the binarization process into the segmentation network, which simplifies the post-processing and enhances the accuracy of text detection. MSR [34] focuses on accurately locating text lines of various lengths, shapes, and curvatures by predicting dense text boundary points then extracts and fuses features at different scales to handle variations in text sizes and shapes. Such a multi-scale detection system delivers enhanced performance regarding text detection of small and large instances simultaneously. The PMTD [35] implementation uses the core advantages of Mask R-CNN in its design. The pixel-level regression in PMTD produces soft text masks and the obtained 2D soft masks get reinterpreted into 3D space while a plane clustering algorithm identifies the optimal text box through the derived 3D shapes. The enhanced mask information enables better detection accuracy because of its detail and information content.

Engineering and Technology Journal for Research and Innovation (ETJRI)
ISSN 2581-8678, Vol. VII, Issue II, July 2025

III. METHODOLOGY

This section introduced the overall pipeline of the proposed approach and further discussed pipeline components like feature extraction, feature fusion and proposed BA-Text module in detail.

1) Overall Structure

The overall structure of BA-Text is shown in Fig.2, which comprises of the feature extracting module (FEM), feature fusion module (FFM), BA-Text module, kernel prediction and text prediction layer map, and post- processing module.

In BA-Text, the images are pre-processed to handle noise and employ ResNet [36] pre-trained architecture as a backbone of FPN [9]. Feature extracting module extract multi-level feature representations, then feature fusion module perform multi-path fusion to embed the low-level features into high level features to enhance the capabilities of text detection. We used up-sampling to maintain the size of image and use soft max as last layer to predict the without compromising precision detect tight bound text. Kernel prediction and text- prediction modules are used to output the predictions by computing text maps and kernel maps. The prediction receives its final adjustments through a post-processing step that determines losses before adjusting training parameters.

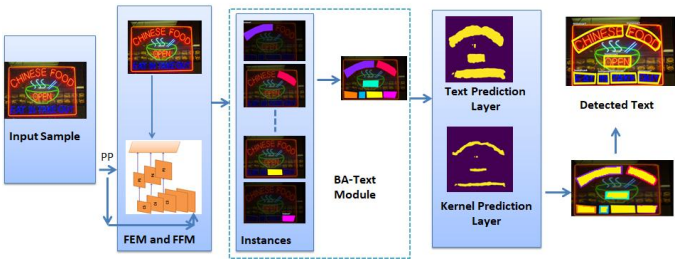

Fig. 2. Overall Structure of the proposed BA-Text model. PP, FEM and FFM represents pre-processing of input sample, feature extraction module and feature fusion module

2)Feature-Extraction and Feature-Fusion Module

As demonstrated in Fig. 3, throughout the feature extraction process, we employ CNN-based ResNet50 architecture followed by feature pyramid network (FPN) [9] to obtain multi-scale feature maps at varying scales by taking kernel 3x3, 1x1. In the pre-stage layers of ResNet, low level features are generated having much information and post- stage layers include high level features.

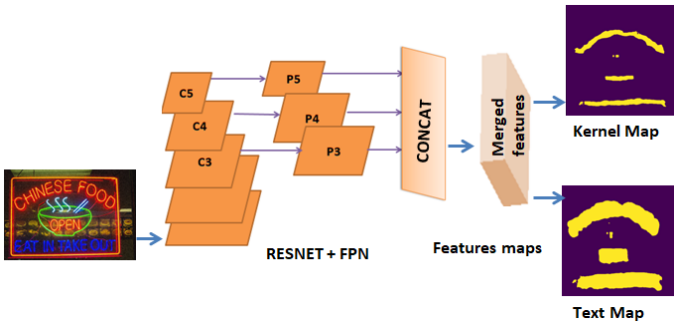

Fig. 3 Illustration of Feature Extraction module and Feature Fusion module. Filters used in ResNet are 64, 128, 256, 512. Up-sampling(conv 1*1 -> conv 3*3 -> Deconv) is used with batch normalization to improve accuracy. Shortcut connections are used to support fusion

This activity is mandatory to localize text of different sizes and orientations and getting contextual local features. The extracted features are then fused to identify text regions in the image. The accuracy and robustness of text detection are enhanced through the integration of multi-layer and multi-scale features, enabling effective detection in complex scenes with diverse text orientations and varying lighting conditions. Fusion is performed on the basis of merging context information from adjacent areas. This elaborates the findings of segmented instances of text within the image. Feature Fusion module is helpful to minimize the false positives and improves in detection of text.

3)BA-Text

BA-Text module focuses on parameters through which we can detect tight bound text. Merged features from FEM and FFM module are used to output the predictions by computing text-region and text-mid-line that are further known as text maps and kernel maps. We feed text maps and kernel maps to BA-Text to compute various geometric parameters that are used to train the model. The post-processing module fine tunes the model by determining smoothed losses which then modifies training parameters. The proposed module BA-Text works to attain the features at pixel level, intermediate level and global level and calculating the geometric attributes to handle parameters related to text present in the image. Pixel level information is suitable for fine-grained local features, intermediate level is

Engineering and Technology Journal for Research and Innovation (ETJRI)
ISSN 2581-8678, Vol. VII, Issue II, July 2025

worth capturing text-regions and global level to capture all instances in scene image. After applying the feature fusion with ResNet as shown in Fig. 3, output consists of a set of multi-scale feature maps (e.g., P2, P3, P4, P5) where P2 corresponds to a finer resolution (smaller receptive field, for detecting small objects) and P5 corresponds to a coarser resolution (larger receptive field, for detecting larger objects). Each feature map contains spatial and semantic information designed for text of different sizes. These multi-scale feature maps are used to generate proposals for wild text object locations.

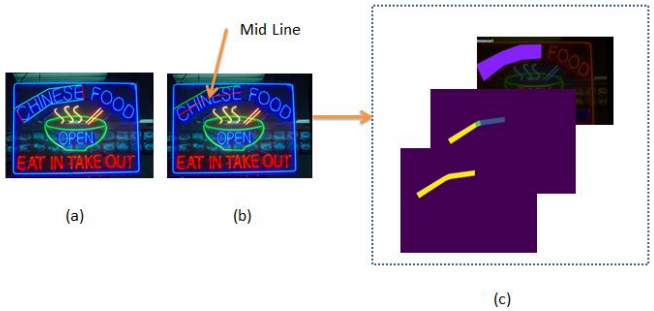

Fig. 4 The visualization of BA-Text. (a) BA-Text focuses on the scale of first instance. The text region is labelled green in the left image. (b) BA-Text calculates the midline by using mean value of sidelines. (c) Different instances are marked with distinct color. The kernel map is marked with yellow and text map is marked with blue

In the classification phase, first prediction of per-pixel masks of Text Regions (TR) is to be made, then text mid-line (TML) prediction can further improve the performance. It can be obtained through shrinking of texts with the shrinking factor being 0.3 as shown in Fig. 4. In an image of height H and width W, for a text instance i, represented by group of circles {c0, c1, c2,…., cn). from left to right, as shown in Fig. 5, group of circles are treated as text region or text map (Tm) and we further determine a text midline of polygons (kernel map) using corresponding top and bottom sides of TR by shrinking text map contours in the inner direction. Algorithm 1 is used to generate kernel maps. The shrinking factor is taken as 0.3 to deal with the effect of noise. In the regression phase, we compute other parameters that are further used to reconstruct the text regions while testing. Various parameters are the radius of circles and orientations in the direction of subsequent appearing texts. It can be computed by calculating its sine and cosine.

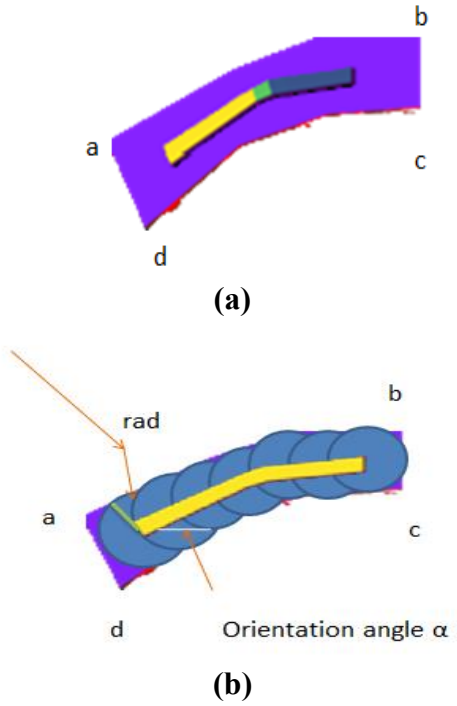

Fig.5 Extracting Mid-value line from corresponding polygons (a)Text instance is represented through a text region (blue region) and text midline (combination of yellow, green and dark blue (b)Text region is represented through number of circles, each is represented through mid-point, and orientation(α). Cosine(α) and Sine(α) are computed to identify the orientation of text instance

Algorithm 1: Kernel Map Label Generation
Input: text region map T_m, minimum area threshold Th_{min}, shrinking ratio Sr, instance area A, instance perimeter P, height H and width W.
Output: kernel map K_m
Procedure:
1 Initialize $K_m \in R^{W,H}$;
2 for each text instance i ϵ Tm do
3 $\text{offset}_i \leftarrow \frac{A_i}{P_i}(1 - Sr)^2$
4 $\text{text kernel}_i \leftarrow$ shrinking contour inward by offset_i;
5 if area of $\text{text kernel}_i > Th_{min}$ then
6 sketch text kernel_i on K_m;
7 end
8 end

Engineering and Technology Journal for Research and Innovation (ETJRI)
ISSN 2581-8678, Vol. VII, Issue II, July 2025

4)Label Generation

Text regions of text image are described by a collection of geometric attributes with overlapping circles {c0, c1, c2,…., cn). The parameters from each circle are represented through mid-point of circle, the rad and orientation. The red can be calculated as half the distance between border lines of text regions B1 and B2. Orientation (α) is tangent line to the midline around the mid-point of circle. By computing the distance to the bordering segment in B, we can create the label for the threshold map. Reconstruction of text boundaries involves calculating the mid-point line/ kernel mask and text region mask and then combining the area of circles. After predicting text regions, the detected pixels or regions are grouped to form polygons, closely approximating the ground truth masks. The placement of polygons closely aligns with text contours which makes BA-Text an efficient solution for detecting curved or shapes that are not regular.

5)Optimization Function

Lastly, we formulated the overall objective of proposed BA-Text module by calculating loss functions for text detection branch; kernel map and mask branch i.e. text map and improve the loss function by updating the weights.

$$L = L_{class} + L_{reg} \quad (1)$$

$$L_{class} = x_1 L_{TR} + x_2 L_{TML} \quad (2)$$

$$L_{reg} = y_1 L_{rad} + y_2 L_{sine} + y_3 L_{cosine} \quad (3)$$

L_{class} represents classification loss for text region and text mid-line. L_{reg} represents regression loss for rad, sine(α) and cosine(α).

We update the loss values using smoothed loss as:

$$L_{rad} = \frac{\widetilde{rad} - rad}{rad} \quad (4)$$

$$L_{sine} = \widetilde{sine(\alpha)} - sine(\alpha) \quad (5)$$

$$L_{cosine} = \widetilde{cosine}(\alpha) - cosine(\alpha) \quad (6)$$

$\widetilde{rad}$, $\widetilde{sine}(\alpha)$, $\widetilde{cosine}(\alpha)$ are new values, while rad, sine(α), cosine(α) are ground truth values from datasets respectively. The weights constants x_1, x_2, y_1, y_2 and y_3 are all set to 1.

IV. Results & Discussion

This section shows how performance of BA-Text is evaluated on the publicly available benchmark datasets that cover range of challenges while detecting text. Then, qualitative detection results are shown to prove the superiority of the recommended method. The comparison between BA-Text and top-notch text detection benchmarks takes place at this point. Next, we demonstrated the robustness of the method using evaluation metrics- Recall, Precision and F-measure and finally discussed the utility of BA-Text in text recognition.

1) Implementation Details

We implemented our model in Tensorflow on Google Colaboratory. The ResNet50 network architecture that is pre-trained on ImageNet [37] has been used with Feature Pyramid Network (FPN). The network is pre-trained on SynthText [38] for an epoch and fine-tuned on other datasets with 1000 epochs based on weights. We used 1255 total-text training images and 300 testing images to train the model. We train BA-Text with batch size 8 on 1 GPU for 5000 iterations. While training the model, we disregard the blurred regions of text that are labelled as DO NOT CARE from datasets and used Adam optimizer for learning rate and set as .001 initially and degrade exponentially at the rate of 0.8 after 1000 iterations. Random resize scale (0.25- 1.70), Random rotation, mirror flipping and lightening noise are used for data augmentation. During inference, no augmentation is used during validation. SynthText dataset is used for training only. Text Mid Line is masked with text region and use threshold value as 0.4. All the experiments are conducted on regular workstation NVIDIA Tesla T4 GPU. We train our model with batch size of 8 on 1 GPU which provides a balance between memory efficiency and stable gradient updates on a Tesla T4 GPU and evaluate model on 1 GPU with batch size 1 to accommodate image size variability and ensure efficient post-processing.

2) Datasets

To detect wild text in scene text detection, several datasets are specifically designed to address this challenge. Several noteworthy examples exist below:

Total-Text [39] is a perfect dataset to deal with wild text detection that contains sufficient number of text aligned in horizontal, curved and text oriented in multiple directions. It is composed of total 1555 images and ground-truths which are available in rectangular and polygon formats with .mat extension. These images and ground-truths are further divided into training and testing directories in approximately in the 80-20 ratio. **SynthText** [38] database is the real efforts to fulfil the demand of training the model with natural images by providing large number of manually created text dataset having 800K images. This is made possible through collection of images, color-model (foreground/background), different fonts, and by calculating images depth and segmentation. It is used to pre-train the model in some epochs to improve performance. **CTW1500** [40] is a curve text dataset containing 1500 images. Images are chosen in such a way that at least one curve-text-instance is included. This instance is labelled with 14 points. Training and testing images are kept at 66.6: 33.3 ratio.

3) Evaluation of BA-Text for text detection

To assess the performance of BA-Text on datasets - Total-Text and CTW1500, first dataset images are converted from JPEG to PNG format so that image quality and data is retained while processing. We have taken 0.4 and 0.5 as threshold values for TR and TML and for training and testing, process will fine-tune for 4K iterations. To calculate the text detection results on datasets, we have calculated the precision (P), recall (R), and f-measure (F).

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

where TP, FP, and FN are representations for the true positives, false positives and false negatives, respectively. Here, true positives are the correctly detected bounding boxes over texts, false positives are incorrect detected bounding box, and false negatives as missed bounding boxes that should be cover by BA-Text. F-measure balances both, precision and recall. We have evaluated the Total-Text and CTW1500 dataset on 300 and 500 testing images by formulating precision, recall and f-measure as shown in Table I and show them graphically as well in Fig. 6

Model	Datasets	Images taken for testing	True Positives	False Positives	False Negatives	Precision	Recall	F-measure
BA-Text	Total-Text	300	256	28	52	90.1	83.1	86.5
	CTW1500	500	435	61	75	87.7	85.3	86.5

Table I Evaluation of BA-Text on datasets

4) Comparisons with State-of-the-Art Methods

This section comprises of comparison of SOTA methods on two datasets: Total-Text, and CTW1500. Our technique results in state-of-the-art achievement on two evaluated datasets based on the data presented in Table II. Qualitative results of the test sets are shown in Fig. 7 that shows that BA-Text achieves high recall, precision for text including multi-oriented, curve text and horizontal text.

5) Discussion

To determine the geometric parameters to cover the text instance and forecast the boundaries along text, the text region and text mid-line are utilized. Similar to the ground truth labelling, boundaries along curved wild text, horizontal text, or multi-oriented text are aligned. These elements become more readable when their geometrical characteristics receive horizontal adjustment for text recognition purposes.

Engineering and Technology Journal for Research and Innovation (ETJRI)
ISSN 2581-8678, Vol. VII, Issue II, July 2025

Method	BB	Total-Text			CTW1500		
		P	R	P	P	R	F
PSENet[26]	ResNet50	84.2	77.9	84.6	84.6	84.6	82.2
TextSnake[6]	VGG16	82.7	74.5	67.9	67.9	67.9	75.6
FCENet[24]	ResNet50	89.3	82.5	85.4	85.4	85.4	83.1
DB[31]	ResNet50	87.1	82.5	86.9	86.9	86.9	83.4
PAN[41]	ResNet18	89.3	81.0	86.4	86.4	86.4	83.7
CRAFT[29]	VGG16	87.6	79.9	86.0	86.0	86.0	83.5
TextDCT[42]	ResNet50	87.2	82.7	85.0	85.0	85.0	85.1
ContourNet[23]	ResNet50	86.9	83.9	83.7	83.7	83.7	83.9
TextRay[43]	ResNet50	83.5	77.9	82.8	82.8	82.8	81.6
LeafText[5]	ResNet50	88.9	83.2	87.1	87.1	87.1	85.5
FEPE[44]	ResNet50	90.8	79.1	88.0	88.0	88.0	85.5
BA-Text (Ours)	ResNet50	90.1	83.1	86.5	87.7	85.3	86.5

Table II The summary of results on Total-Text and CTW1500 by different methods and approaches. "BB" represents Backbone. "P", "R" and "F" represents the precision, recall and f-measure respectively. "RED" AND "BLUE" represent the optimal and sub-optimal performance, respectively.

Fig. 7 Qualitative detection results of our proposed method on CTW1500[45] and Total-Text[39]

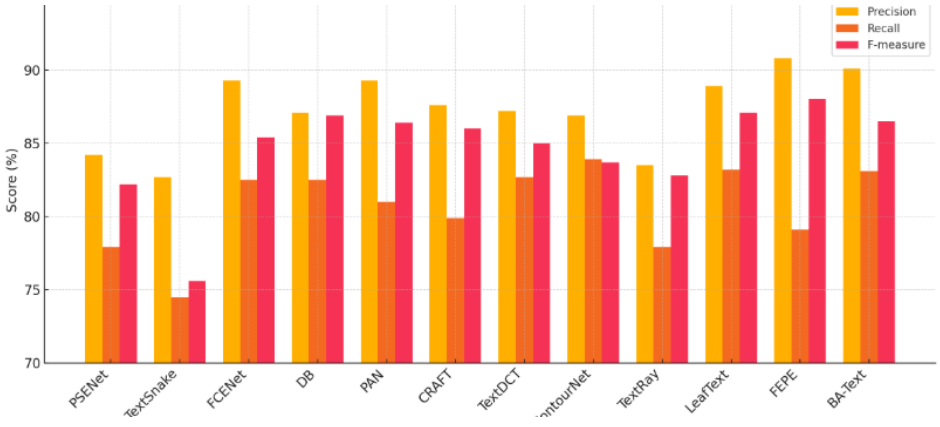

Fig.8 Performance Comparison on Total-Text Dataset

6) Ablation Study

To analyse the contribution of individual components in the BA-Text architecture, we conducted ablation studies on the Total-Text dataset. The results shown in Table III demonstrate the incremental value added by key components: Feature Fusion Module (FFM), Kernel Shrinking, and Mid-Line Prediction. These results indicate that each module contributes to performance enhancement. The use of kernel shrinking and mid-line guidance provides tighter boundary representation, especially for curved and multi-oriented text.

Configuration	Precision (%)	Recall (%)	F-measure (%)
ResNet50 + FPN (baseline)	85.2	77.1	80.9
+ Feature Fusion Module (FFM)	87.6	79.4	83.3
+ Kernel Shrinking (BA-Text)	89.0	82.0	85.3
+ Mid-Line Prediction (Full model)	**90.1**	**83.1**	**86.5**

Table III: Ablation Study Results on Total-Text Dataset

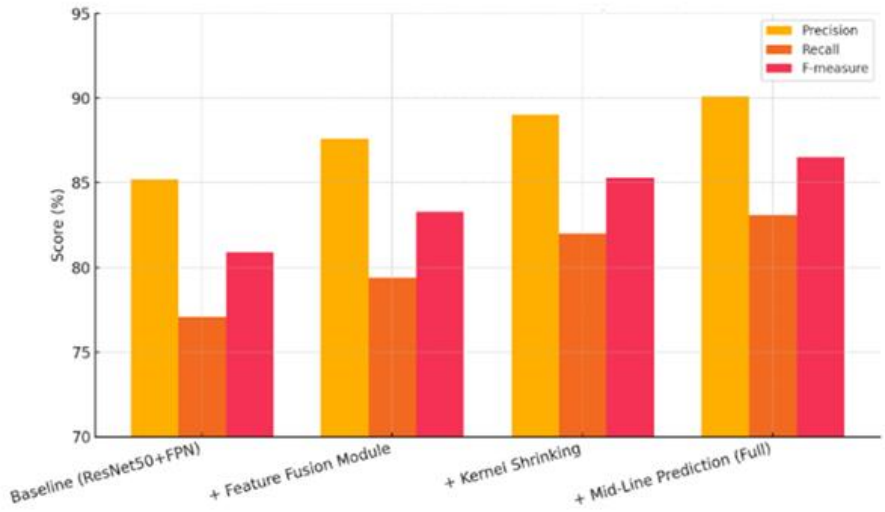

Fig. 9 Ablation Study of BA-Text Components

V. Conclusion

This paper develops a new framework which detects text boundaries in wild scene text images (BA-Text). BA-Text encourages the training model to distinguish instances by considering the proximity of pixels in the successive appearing polygon. The BA-Text representation approaches from top to bottom i.e. from polygon to pixel level information. It enables the model to capture the text proposal regions and generates text maps and kernel maps which when overlay on text instances gives satisfactory result. BA-Text then computes internal parameters that are used to reconstruct the text maps at the time of testing new wild images. Extensive execution of experiments proves the ability of BA-Text to correctly spot the text instances area while comparing with SOTA methods on two public benchmarks. We will continue to explore more to resolve the complexities associated with wild-text text detection in future.

REFERENCES

[1] Y. Qu, Yadong and Xie, Hongtao and Fang, Shancheng and Wang, Yuxin and Zhang, "ADNet: rethinking the shrunk polygon-based approach in scene text detection," IEEE Trans. Multimed., vol. 25, pp. 6983--6996, 2022.

[2] Q. Yang, Chuang and Chen, Mulin and Yuan, Yuan and Wang, "Reinforcement shrinkmask for text detection," IEEE Trans. Multimed., vol. 25, pp. 6458--6470, 2022.

[3] X. Dai, Pengwen and Li, Yang and Zhang, Hua and Li, Jingzhi and Cao, "Accurate scene text detection via scale-aware data augmentation and shape similarity constraint," IEEE Trans. Multimed., vol. 24, pp. 1883--1895, 2021.

[4] Z. Sheng, Tao and Chen, Jie and Lian, "Centripetaltext: An efficient text instance representation for scene text detection," Adv. Neural Inf. Process. Syst., vol. 34, pp. 335--346, 2021.

[5] Q. Yang, Chuang and Chen, Mulin and Yuan, Yuan and Wang, "Text growing on leaf," IEEE Trans. Multimed., vol. 25, pp. 9029--9043, 2023.

[6] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes," Eur. Conf. Comput. Vis., vol. 11206 LNCS, pp. 19–35, 2018, doi: 10.1007/978-3-030-01216-8_2.

[7] B. Ye, Jian and Chen, Zhe and Liu, Juhua and Du, "TextFuseNet: Scene Text Detection with Richer Fused Features.," in IJCAI International Joint Conference on Artificial Intelligence, 2020, pp. 516--522.

[8] B. Koonce, "ResNet 50," in Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, Berkeley, CA: Apress, 2021, pp. 63–72.

[9] S. Lin, Tsung-Yi and Doll{\'a}r, Piotr and Girshick, Ross and He, Kaiming and Hariharan, Bharath and Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117--2125.

[10] S. Goyal and D. Motwani, "A Study of Text Extraction Algorithms for Natural Scene Images," SN Comput. Sci., vol. 5, no. 6, p. 731, Jul. 2024, doi: 10.1007/s42979-024-03068-w.

[11] L. H N, S. Rudresh, D. Sampreeth, S. M. Otageri, and S. S. Hedge, "Image understanding: Semantic segmentation of graphics and text using faster-RCNN," 2018 Int. Conf. Networking, Embed. Wirel. Syst. ICNEWS 2018 - Proc., pp. 1–6, 2018, doi: 10.1109/ICNEWS.2018.8903963.

[12] A. C. Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, "Ssd: Single shot multibox detector," in Computer Vision--ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11--14, 2016, Proceedings, Part I 14, 2016, pp. 21--37.

[13] Z. Zhong, Z and Jin, L and Zhang, S and Feng, "Deeptext: A unified framework for text proposal generation and text detection in natural images. arXiv 2016," arXiv Prepr. arXiv1605.07314, vol. 2.

[14] W. Liao, Minghui and Shi, Baoguang and Bai, Xiang and Wang, Xinggang and Liu, "Textboxes: A fast text detector with a single deep neural network," in Proceedings of the AAAI conference on artificial intelligence, 2017.

[15] M. Liao, B. Shi, and X. Bai, "TextBoxes ++ : A Single-Shot Oriented," IEEE Trans. IMAGE Process., vol. 27, no. 8, pp. 3676–3690, 2018.

[16] X. Zhou et al., "EAST: An Efficient and Accurate Scene Text Detector," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.03155.

[17] T. Long, Jonathan and Shelhamer, Evan and Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431--3440.

[18] M. Liao, Z. Zhu, B. Shi, G. S. Xia, and X. Bai, "Rotation-Sensitive Regression for Oriented Scene Text Detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5909–5918, 2018, doi: 10.1109/CVPR.2018.00619.

[19] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single Shot Text Detector with Regional Attention," Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-Octob, pp. 3066–3074, 2017, doi: 10.1109/ICCV.2017.331.

[20] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 3482–3490, 2017, doi: 10.1109/CVPR.2017.371.

[21] X. Dai, Pengwen and Zhang, Sanyi and Zhang, Hua and Cao, "Progressive contour regression for arbitrary-shape scene text detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 7393--7402.

[22] Y. Tian, Zhi and Huang, Weilin and He, Tong and He, Pan and Qiao, "Detecting text in natural image with connectionist text proposal network," in Computer vision--ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part VIII 14, 2016, pp. 56--72.

[23] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, and Y. Zhang, "Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 11750–11759, 2020, doi: 10.1109/CVPR42600.2020.01177.

[24] W. Zhu, Yiqin and Chen, Jianyong and Liang, Lingyu and Kuang, Zhanghui and Jin, Lianwen and Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 3123--3131.

[25] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-Time Scene Text Spotting with Adaptive Bezier-Curve Network," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 9806–9815, 2020, doi: 10.1109/CVPR42600.2020.00983.

[26] W. Wang et al., "Shape robust text detection with progressive scale expansion network," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, no. c, pp. 9328–9337, 2019, doi: 10.1109/CVPR.2019.00956.

[27] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented Scene Text Detection via Corner Localization and Region Segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 7553–7563, 2018, doi: 10.1109/CVPR.2018.00788.

[28] 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, 2019, doi: 10.1109/TIP.2019.2900589.

[29] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 9357–9366, 2019, doi: 10.1109/CVPR.2019.00959.

[30] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting scene text via instance segmentation," 32nd AAAI Conf. Artif. Intell. AAAI 2018, pp. 6773–6780, 2018.

[31] X. Liao, Minghui and Wan, Zhaoyi and Yao, Cong and Chen, Kai and Bai, "Real-time scene text detection with differentiable binarization," in Proceedings of the AAAI conference on artificial intelligence, 2020, pp. 11474--11481.

[32] X. Liao, Minghui and Zou, Zhisheng and Wan, Zhaoyi and Yao, Cong and Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 919--931, 2022.

[33] Q. Yang, Chuang and Chen, Mulin and Xiong, Zhitong and Yuan, Yuan and Wang, "Cm-net: Concentric mask based arbitrary-shaped text detection," IEEE Trans. Image Process., vol. 31, pp. 2864--2877, 2022.

[34] C. Xue, S. Lu, and W. Zhang, "MSR: Multi-scale shape regression for scene text detection," IJCAI Int. Jt. Conf. Artif. Intell., vol. 2019-Augus, pp. 989–995, 2019, doi: 10.24963/ijcai.2019/139.

[35] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid Mask Text Detector," 2019, [Online]. Available: http://arxiv.org/abs/1903.11800.

[36] J. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770--778.

[37] L. I. Deng, J and Dong, W and Socher, R and Li, LJ and Li, K and Fei-Fei, "A large-scale hierarchical image database. En: 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009," 2009.

[38] A. Gupta, Ankush and Vedaldi, Andrea and Zisserman, "Synthetic data for text localisation in natural images," in Gupta, Ankush and Vedaldi, Andrea and Zisserman, Andrew, 2016, pp. 2315--2324.

[39] C. S. Ch'ng, Chee Kheng and Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in 2017 14th IAPR international conference on document analysis and recognition (ICDAR), 2017, pp. 935--942.

[40] Z. Yuliang, Liu and Lianwen, Jin and Shuaitao, Zhang and Sheng, "Detecting curve text in the wild: New dataset and new solution," arXiv Prepr. arXiv1712.02170, 2017.

[41] C. Wang, Wenhai and Xie, Enze and Song, Xiaoge and Zang, Yuhang and Wang, Wenjia and Lu, Tong and Yu, Gang and Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 8440--8449.

[42] Y. Su et al., "TextDCT: Arbitrary-Shaped Text Detection via Discrete Cosine Transform Mask," IEEE Trans. Multimed., vol. 25, no. X, pp. 5030–5042, 2023, doi: 10.1109/TMM.2022.3186431.

[43] F. Wang, Y. Chen, F. Wu, and X. Li, "TextRay: Contour-based Geometric Modeling for Arbitrary-shaped Scene Text Detection," MM 2020 - Proc. 28th ACM Int. Conf. Multimed., pp. 111–119, 2020, doi: 10.1145/3394171.3413819.

[44] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Focus Entirety and Perceive Environment for Arbitrary-Shaped Text Detection," IEEE Trans. Multimed., pp. 1–13, 2024, doi: 10.1109/TMM.2024.3521797.

[45] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting Curve Text in the Wild: New Dataset and New Solution," 2017, [Online]. Available: http://arxiv.org/abs/1712.02170.