Journal of Information Hiding and Multimedia Signal Processing ©2025 ISSN 2073-4212
Ubiquitous International Volume **16**, Number **2**, June **2025**

Text Detection in Multi-Oriented and Curved Formats: Addressing Adjacent Text Regions with Attention and Boundaries Separation

Soufiane Naim*

Data Science and Artificial Intelligence, LIM
University Hassan II, Faculty of science and technology
BP 146, Mohammedia, 28806, Morocco
naimsoufiane@gmail.com

Noureddine Moumkine

Data Science and Artificial Intelligence, LIM
University Hassan II, Faculty of science and technology
BP 146, Mohammedia, 28806, Morocco
noureddine.moumkine@fstm.ac.ma

Lachhab Youssef

C3S Research Laboratory High School of Technology
Hassan II University
El Jadida Road, KM 7, Casablanca, 20410, B.P. 8012,Morocco
youssef.lachhab.2020@gmail.com

Ziyati Elhoussaine

C3S Research Laboratory High School of Technology
Hassan II University
El Jadida Road, KM 7, Casablanca, 20410, B.P. 8012,Morocco
ziyati@gmail.com

*Corresponding author: Soufiane Naim
Received January 25, 2025, revised March 18, 2025, accepted March 19, 2025.

Abstract. *Unlike the general object detection sub-domain, where the main objective is to draw horizontal bounding boxes around detected objects inside a natural scene image, text detection presents greater challenges. Text can appear not only horizontally but also in inclined and curved formats. Additionally, when text regions are adjacent, general models may mistakenly detect them as a single instance, which is typically considered as a false detection. In this paper, we propose addressing these challenges by implementing a semantic segmentation architecture designed to detect text regions and their boundaries, ensuring proper separation. Our proposed architecture integrates an attention module that gathers both local and global contextual features by leveraging spatial and channel dimensions. Additionally, we incorporate dilated convolutions to capture a broader context, which is essential for accurately classifying each pixel in the image. The effectiveness of our architecture has been demonstrated through evaluations on various datasets, achieving a well-balanced performance in terms of recall and precision.*
Keywords: deep learning, computer vision, scene text detection, convolutional neural network, Dice loss

FIGURE 1. Example images from the CTW1500 Dataset, showcasing text regions that are positioned in close proximity to one another.

1. **Introduction.** Scene text detection is a subdomain of computer vision that plays an important role in identifying text within natural images. This task is typically followed by text recognition, which enables various applications, such as translation, autonomous driving, and augmented reality. Unlike other object detection tasks, detecting text in complex, real-world environments is particularly challenging. Images often suffer from occlusions, blurring, or other images weaknesses. Text can also appear in different sizes, colors, orientations, and fonts within the same image that makes the detection task more difficult. Another layer of complexity in scene text detection arises from the way text is annotated. Convolution neural networks (CNN) models are typically designed to detect text at different levels, such as character, word, or line level. However, variations in spacing between characters, words, or even lines across images can lead to inaccurate detections. When space between characters or words are very narrow, or when the interline spacing is minimal. This often leads the model to fuse text regions, which penalize it during testing, as these fused detections are counted as false positives (Fig. 1). Additionally, this fusion complicates the post-processing step, requiring additional effort to separate the regions and restore the detections to their correct form before the recognition phase.

To address these challenges, numerous research papers have been published. With the advent of convolutional neural networks, two main streams of techniques have emerged to tackle the problem: regression-based models[1][2] [3] and semantic segmentation models[4][5] [6]. Both techniques have been employed to detect text, particularly in multi-oriented formats, by either predicting rectangular coordinates and the inclination angle or detecting the vertices of a quadrilateral surrounding the text regions. Unfortunately, most of these frameworks struggle to detect text instances with arbitrary shapes (e.g., curved text), as this type of text requires the detection of more than four vertices, which can be achieved using a polygonal representation.

In this paper, we present a novel methodology for text detection that effectively handles multi-oriented and curved text, with a particular focus on separating closely positioned text. To achieve this, we incorporate a semantic segmentation module designed not only

to detect text regions but also to delineate boundaries around them. By leveraging these boundary regions, we enhance the model's ability to separate adjacent text. This approach transforms the traditional binary semantic segmentation task, which classifies each pixel as either text or background, into a three-class segmentation problem. Here, each pixel is identified as either background, part of a text region, or part of a boundary region.

Our contribution can be summarized as follows :

1. **A New Semantic Segmentation Architecture:** We propose a new semantic segmentation architecture based on the Pyramid Scene Parsing Network (PSPNet)[7]. This architecture is chosen because it uses a Pyramid Pooling Module (PPM), which allows the detection of objects at multiple scales within an image. This capability is crucial since, generally, text appears in the same image at various scales. For each pixel in the image, our model attempts to predict whether it belongs to a text region, the boundary of a text region, or the background. The objective is to better separate closely located text regions, leading to improved predictions.
2. **A New Attention Channel-Space Module :** We introduce a novel attention mechanism that operates across both spatial and channel dimensions, integrating it into the PPM block. The primary objective of this module is to evaluate the features present in each channel of the input feature map. By assigning a weight to each channel based on the quality of the data it contains, the module emphasizes the most informative channels while assigning lower weights to less relevant ones.
3. **A New Pipeline for Polygon Coordinate Generation :** We introduce a post-processing pipeline designed to generate polygon coordinates for each text instance within an image. This pipeline is tailored for detecting curved text by producing a set of points that define the closest polygon enclosing the text regions. Furthermore, to ensure the accurate detection of adjacent text, which the model may merge into a single region, the pipeline utilizes boundary predictions to effectively separate them, resulting in a more accurate representation.

2. **Related works :** Research in natural scene text detection has a long-standing history and did not originate only recently. Early approaches primarily relied on gradient-based methods combined with handcrafted rules to make predictions[8] [9] . While these strategies achieved notable results, the most significant advancements have emerged in the deep learning era, where many works have been inspired by general object detection frameworks such as Fast R-CNN, YOLO, Mask R-CNN, and FCN. This research can be broadly categorized into two principal streams.

Regression-Based Methods: These approaches focus on predicting the coordinates of bounding boxes that enclose text regions. Depending on the text's shape and orientation, models are designed to output various bounding box formats, including horizontal rectangles, multi-oriented rectangles, quadrangles, or polygons. Liao et al.[3] proposed an end-to-end architecture inspired by the VGG network to detect rectangular coordinates surrounding text. This process is followed by a Non-Maximum Suppression (NMS) algorithm to eliminate overlapping predictions. Text inside images generally appear in multi-oriented orientations, Keserwani et al.[10] addressed this challenge by proposing a method to regress the quadrilateral coordinates through a combination of direct and indirect regression techniques. Initially, the four vertices of the quadrilateral are shifted toward its centroid. Then, the model focus on regressing the relative positions of the vertices by utilizing vectors originating from the shifted points. In the case of curved text, Wu et al. [11] use a combined architecture that leverages both semantic and regression approaches. This architecture employs a two-branch design: the first branch focuses on learning semantic information, while the second handles geometric information. These

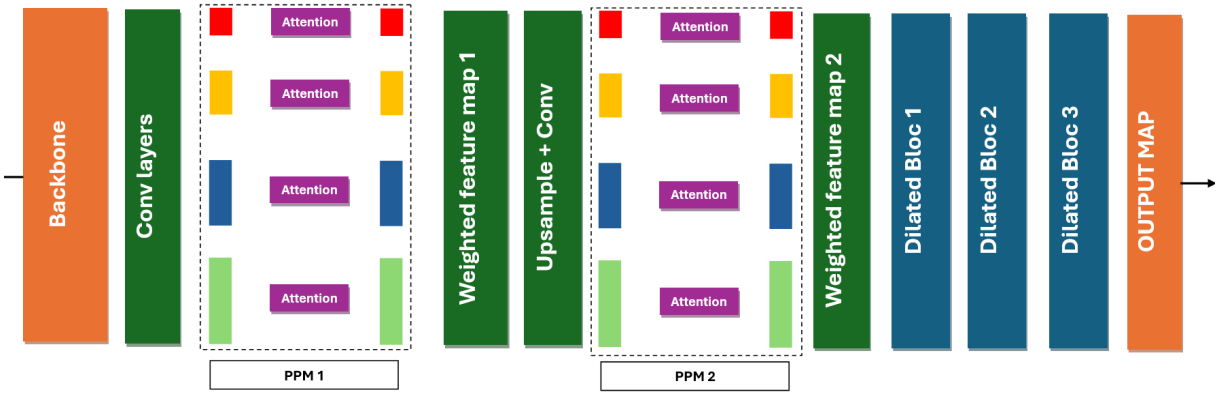

FIGURE 2. In the encoder part of our architecture, ResNet50 serves as the backbone to capture global features. In the decoder part, we integrate two PPM modules combined with an attention space-channel mechanism, which adjusts channel weights by prioritizing those rich in relevant features. Furthermore, three convolutional blocks with dilated convolutions are employed to improve boundary refinement.

two branches interact with each other through an interleaved feature attention module based on deformable convolutions.

Semantic segmentation-based methods : Focus on predicting whether each pixel in an image belongs to a text region or not. While this approach provides a better understanding at the pixel level and achieves improved results with irregular shapes, it often struggles with compact or overlapping regions. Liao et al. [12] attempt to combine predictions at various scales using a two-branch architecture. The first branch focuses on predicting text regions, while the second branch is dedicated to detecting boundaries. These two branches are then merged. Additionally, they propose a differentiable binarization module to optimize the training process. Li et al. [13] address the challenge of distinguishing closely spaced text regions, which models often detect as a single region. To tackle this, they propose the Progressive Scale Expansion Network (PSENet). This method generates multiple text region kernels of varying sizes and progressively expands the smallest kernels to reconstruct complete text shapes.

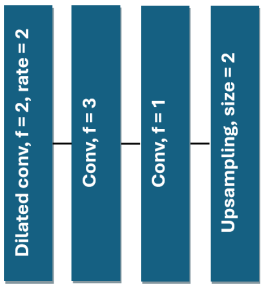

FIGURE 3. The dilated block consists of a dilated convolutional layer to capture more global context, followed by a 3×3 and a 1×1 convolutional layers to reduce the input feature map depth before the upsampling step.

3. **Methodology.** In this section, we describe the key components of our proposed architecture, which is inspired by the Pyramid Scene Parsing Network (PSPNet). PSPNet was chosen for its ability to extract features at multiple scales within an image and its

effectiveness to combine global and local context information to enhance detection performance. (Fig. 2) provides a detailed explanation of our model, which is based on an encoder-decoder semantic segmentation architecture. In the encoder, ResNet-50 is employed to extract global features. This choice is motivated by ResNet-50's capability to learn hierarchical features, from simple edges to complex patterns, which is crucial for handling text that may vary in color, shape, and font within the same image. Moreover, ResNet-50's residual connections are instrumental in facilitating gradient flow during backpropagation, ensuring efficient training of deeper layers.

In the decoder part, we extend the PSPNet implementation by adding additional layers. The goal is to enhance the model's precision in handling edges by encouraging it to learn a richer combination of local and global features. To achieve this, the Pyramid Pooling Module (PPM) from the original PSPNet is used to aggregate contextual features from multiple scales. Given an input feature map, the PPM pools it into a four-level pyramid with different sizes. Each level's depth is reduced using a 1×1 convolution, followed by bilinear interpolation to upsample the features back to the original size. Finally, the upsampled features are fused with the input feature map.

To enhance the PPM module, we integrate an attention mechanism that assigns weights to each channel of the four-level pyramid. This mechanism evaluates the quality of information in each channel based on both spatial and channel dimensions, enabling the model to focus on the most relevant ones. Our attention module (Fig. 4) is inspired by and extends the Squeeze and Excitation Network[14] (SE), which recalibrates channel-wise responses through a two-step process. In the squeeze step, each channel is reduced to a single value by applying global average pooling to generate channel-wise statistics, followed by the excitation step where two fully connected layers is used to capture channel-wise dependencies. This process produces a weight vector that is multiplied with the input feature map. One limitation of the approach is its lack of consideration for spatial information, as it focuses solely on the channel dimension. We address this limitation by extending the attention mechanism to incorporate spatial attention as well.

As described in Algorithm 1, during the sqeeze step, given an input feature map $X \in \mathbb{R}^{w \times h \times c}$, we define a matrix $SA \in \mathbb{R}^{w \times h}$ by computing the average values along the spatial dimension. Similarly, a vector $CA \in \mathbb{R}^{c}$ is obtained by applying global average pooling along the channel dimension.

To further analyze X, we compute a distance features map $D \in \mathbb{R}^{w \times h \times c}$, representing the difference between the input feature map and the spatial average matrix SA. The goal is to evaluate each channel at the space level and identify those that closely match the average values in SA, which is considered as a reference for accurate predictions. layers that are closer to SA are hypothesized to contribute more effectively to the final prediction, as they are more likely to align with the collective representation captured by the network.

The distance feature map D is converted into a vector of weights, reflecting the quality of each channel's alignment with the spatial average SA. This vector is then multiplied by the channel average CA, emphasizing the channels that best align with both spatial and channel contexts while giving less importance to those that are less relevant. A ReLU activation is applied to the resulting vector. The excitation step is then performed by passing this vector through two dense layers, with ReLU and Sigmoid activation functions applied sequentially, as shown in Equation 1.

$$\mathbf{s} = \sigma\left(\mathbf{W}_2 \cdot \delta\left(\mathbf{W}_1 \cdot \mathbf{z} + \mathbf{b}_1\right) + \mathbf{b}_2\right) \tag{1}$$

Where:

- $z \in \mathbb{R}^{C}$: Input vector (squeezed vector).

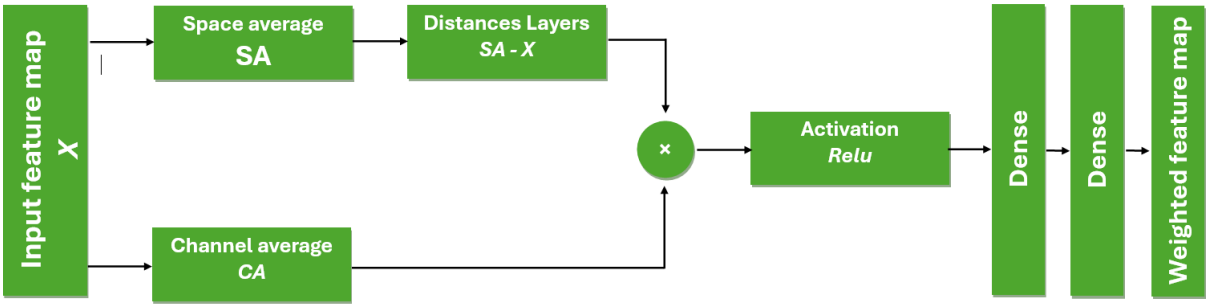

FIGURE 4. Given an input feature map, both spatial and channel-wise average values are computed. The spatial average value is used to compute a vector of distances to highlight layers rich in features. This vector is multiplied by the channel-wise average value and passed through two dense layers to produce the final vector of weights

- $W_1 \in \mathbb{R}^{C \times r}$: Weight matrix of the first dense layer. r present the reduction ratio
- $b_1 \in \mathbb{R}^r$: Bias of the first dense layer.
- $\delta(\cdot)$: ReLU activation function.
- $W_2 \in \mathbb{R}^{r \times C}$: Weight matrix of the second dense layer.
- $b_2 \in \mathbb{R}^C$: Bias of the second dense layer.
- $\sigma(\cdot)$: Sigmoid activation function.

The PPM module, integrated with the attention mechanism, is applied twice within the decoder, followed by the application of a dilated block three times before outputting the final predictions. Each dilated block consists of three convolutional layers. The first layer employs dilated convolution, chosen for its ability to expand the receptive field and capture more global contextual information. The second layer uses a classic 3×3 convolution, and the final layer uses a 1×1 convolution to reduce the channel depth of the input feature map before upsampling it. This design choice enhances the model's ability to detect edges and refine boundaries.

The model output predicts whether each pixel belongs to a text area, a boundary area, or the background. The primary objective is to utilize boundary predictions to distinguish closely positioned text regions, enabling the model to focus on accurately separating them. This approach prevents the detection of multiple text regions as a single one, which would otherwise result in false detections.

Algorithm 1 The proposed space-channnel attention mechanism

Require: Input feature map: `inputs`
Ensure: Weighted feature map
SA ← Average values across space dimension
CA ← GlobalAveragePooling2D()(inputs)
dist ← $(\text{inputs} - \text{SA})^2$
$squeeze$ ← ReLu(dist × CA)
$excitation$ ← DenseLayer($squeeze$)
$excitation$ ← ReLu($excitation$)
$excitation$ ← DenseLayer($excitation$)
$excitation$ ← Sigmoid($excitation$)
return inputs × $excitation$

3.1. **Loss function :** As a loss function, we propose using a combination of Generalized Dice Loss (GDL) and Categorical Cross-Entropy (CCE) Loss. Text within an image is typically represented by a relatively small number of pixels, leading to an imbalanced representation compared to the background. Additionally, since our model predicts the boundaries between text regions as a separate class, this further introduces another source of imbalance. To stabilize the training process and mitigate overfitting, we adopt a balanced combination of these losses. The overall expression of our loss function is formulated as follows:

$$\mathcal{L}_{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c y_i^c \log(p_i^c) \tag{2}$$

$$\mathcal{L}_{GDL} = 1 - \frac{2\sum_{c=1}^{C}\sum_{i=1}^{N} p_i^c y_i^c}{\sum_{c=1}^{C}\sum_{i=1}^{N} p_i^c + \sum_{c=1}^{C}\sum_{i=1}^{N} y_i^c} \tag{3}$$

$$\mathcal{L}_{Total} = \alpha\mathcal{L}_{GDL} + \beta\mathcal{L}_{CCE} \tag{4}$$

Where:

- N is the number of pixels,
- C is the number of classes,
- w_c: a weight assigned to each class to give more focus during the training process.
- y_i^c is the ground truth label for the i-th pixel and class c,
- p_i^c is the predicted probability for the i-th pixel belonging to class c,
- α and β are hyperparameters used to balance the contribution of the two losses. Typically, they are set to 1.

This loss function empirically ensures better boundary refinement because the Dice Loss is particularly effective in scenarios that require optimizing the overlap with both the ground truth and predicted regions. This focus helps improve generalization and capture the global context. On the other hand, the Cross-Entropy Loss ensures stable optimization by emphasizing pixel-level accuracy and capturing fine-grained details.

3.2. **Post-processing.** Semantic segmentation models operate at the pixel level to predict the class of each pixel. However, in some cases, the resulting segmentation map is not always accurate and requires post-processing to address false pixel classifications. In our case, it is crucial that the detected boundaries are continuous and fully enclose the text regions to prevent multiple ones from being merged.

Algorithm 2 outlines the post-processing steps we adopt to address this issue. First, the boundaries and text regions are separated. Then, morphological dilation is applied to increase the size of the detected boundary zones, which helps to correct any gaps in the boundaries. The resulting map is merged with the text region predictions, and a connected components algorithm[15] is used to detect each text region separately. Since the boundary regions are oversized, the size of the predicted text regions is reduced. To restore their original size, we iterate over each detected region and apply morphological dilation before extracting the vertices of the contour surrounding it[16]. Repeating this operation produces a list of polygons, each describing a distinct text region.

4. EXPERIMENTAL RESULTS.

4.1. **Benchmark Datasets.** Detecting text in natural scene images requires addressing various real-world challenges, including diverse text forms (e.g., curved or multi-oriented) and complex backgrounds, which can significantly affect model performance and lead to false detections. To overcome these challenges, it is crucial to use datasets that capture

Algorithm 2 Post-Processing Algorithm

Require: Prediction map *pred* with values representing background, text regions, and boundaries.

Ensure: List of polygons *list_polygons* representing refined text regions.

$BM \leftarrow$ get_boundaries_map(*pred*)
$TM \leftarrow$ get_text_regions_map(*pred*)
$BM \leftarrow$ exclude_boundaries_of_small_region(BM)
$BMD \leftarrow$ morphological_dilation(BM)
Corrected_map $\leftarrow$ merge(TM, BMD)
new_map $\leftarrow$ get_new_text_regions(*Corrected_map*)
text_regions $\leftarrow$ connected_Components(*new_map*)
list_polygons $\leftarrow$ []
for each *tr* in *text_regions* **do**
    tr $\leftarrow$ morphological_dilation(*tr*)
    polygon $\leftarrow$ get_contour_points(*tr*)
    list_polygons.add(*polygon*)
end for
return *list_polygons*

a wide variety of scenarios and accurately represent real-life conditions. In our case, we have selected the following datasets to train and evaluate our model.

- **Total-Text Dataset** [17] : introduced in 2017, consists of natural scene images designed to capture real-world scenarios. It includes a diverse mixture of curved and multi-oriented text. Some images exhibit challenges such as low contrast and other quality issues, increasing the difficulty of detection. The dataset comprises 1,555 images, split into 1,255 for training and 300 for testing. The text is predominantly in English and is annotated at the word level using a series of points that form polygons outlining the text regions.
- **CTW-1500** [18] : The CTW-1500 dataset, introduced in 2017, consists of 1,500 images, with 1,000 allocated for training and 500 for testing. The text in these images features both curved and multi-oriented shapes. The dataset includes approximately 10,751 cropped text instances, of which 3,500 are curved [19]. The text instances are in both English and Chinese and are annotated at the text-line level.
- **MSRA-TD500** [20] : comprises 500 images, with 300 designated for training and 200 for testing. The text is presented in a multi-oriented format and includes a mixture of English and Chinese. Annotations are provided at the text-line level, with ground truth defined using rotated rectangles. Each rectangle is specified by its top-left corner coordinates, width, height, and inclination angle.

4.2. **Implementation Details.** The training and testing of our model were performed using the Nvidia L4 GPU. For the ResNet backbone, we employed transfer learning to accelerate the training process. Additionally, we used data augmentation to diversify the types of images presented to our model during training, especially since the number of images in the used datasets is very low. The augmentation operations focused primarily on rotation and scaling. The Adam optimizer was used with a learning rate of 10^{-4},β_1 and β_2 were set to 0.9 and 0.999, respectively.

4.3. **Evaluation metrics.** The evaluation of our model involves assessing the correspondence between predicted and ground truth polygons. This is achieved through a two-step

process. The first step computes the Intersection over Union (IoU) metric, which measures the degree of overlap between the two sets of polygons. The IoU metric is defined as :

$$IoU = \frac{AREA(G \cap P)}{AREA(G \cup P)} \tag{5}$$

Here, G represents the Ground Truth polygon, and P is the predicted one.

When a predicted polygon matches a ground truth one, this is considered as a true positive (TP) detection; otherwise, it is evaluated as a false positive (FP). Evaluating all predicted polygons allows us to classify them into three categories: true positives (TP), false positives (FP), and false negatives (FN). These classifications enable the computation of the Precision, Recall and F1-score metrics, defined as:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{8}$$

5. **Result.**

5.1. **Quantitative results.** The created model has been evaluated on various datasets. Total-Text and CTW-1500 consist of both curved and multi-oriented text, while MSRA-TD500 is composed solely of multi-oriented text. The evaluation process starts by detecting the text regions, followed by computing the IoU metric to evaluate each prediction and verify its match with a ground truth polygon. The predicted polygons are then classified into True Detection, False Detection, or Not Detected (in case there is no prediction equivalent to a ground truth polygon). These values are used to compute the precision, recall, and F1 score of our model.

Tables 1, 2, and 3 present the results obtained for the Total-Text, CTW-1500, and MSRA-TD500 datasets, respectively. These results demonstrate how our model consistently outperforms existing methods across all three datasets, highlighting its superior ability to detect text regions in both multi-oriented and curved text environments. The use of the attention mechanism, coupled with the proposed post-processing, leads to a noticeable improvement in detection accuracy and robustness across various text orientations. This can be observed in the well-balanced trade-off between precision and recall, which ensures the model's ability to correctly identify positive text regions and maximize the detection of relevant ones. This reflects the model's strong generalization ability across a wide range of text orientation scenarios.

5.2. **Qualitative results.** Figure 5 shows examples of cases where the post-processing successfully separates adjacent text regions. As shown, the dilation of the boundary regions corrects prediction errors made by the model, contributing positively to the detection of text regions in the desired format. This operation improves the model's performance by preventing correctly detected text regions from being evaluated as false detections simply because they do not match the required format.

Figure 6 presents examples of predictions made by our model. In the first three rows, we can observe how the model successfully detects text in both curved and multi-orientation formats. The detected polygons also vary in size, and adjacent text regions are correctly detected as separate instances. However, the last row highlights some examples of incorrect predictions. In the first image on the left, the model mistakenly identifies a ladder as

text due to the similarity in its structural features to certain characters (I or H). In the remaining examples, the post-processing fails to correctly separate adjacent text regions. This issue may be attributed to poor boundary predictions that, even after dilation, are insufficient to fully separate the adjacent regions.

TABLE 1. Detection results on the Total-Text Dataset.

	Precision	***Recall***	***F1-score***
EAST [21]	50.0	36.2	42.0
Liu et al.[22]	74 .0	71.0	73.0
Wang et al.[23]	80.9	76.2	78.5
Xue et al.[24]	83.8	74.8	79.0
Dai et al.[25]	84.6	78.6	81.5
Our method	**88.65**	**80.49**	**84.37**

TABLE 2. Detection results on the CTW-1500 Dataset.

	Precision	***Recall***	***F1-score***
EAST [21]	78.7	49.1	60.4
Liu et al. [26]	81.1	76.0	78.4
Tian et al.[27]	82.7	77.8	80.1
Li et al.[13]	82.5	79.9	81.2
Wang et al. [2]	82.8	80.4	81.6
Our method	**83.57**	**81.47**	**82.47**

TABLE 3. Detection results on the MSRA-TD500 Dataset.

	Precision	***Recall***	***F1-score***
PixelLink [28]	83.0	73.2	77.8
TextSnake [29]	83.2	73.9	78.3
DBNet(ResNet-18) [12]	86.8	78.4	82.3
CRAFT [30]	88.2	78.2	82.9
Our method	**85.23**	**81.90**	**83.53**

5.3. **Ablation study.** To validate the attention mechanism and the post-processing steps proposed in this paper, an ablation study was conducted. A series of experiments were performed using different versions of our model. The evaluation is divided into three parts: in the first, the complete model is used (baseline); in the second, the model is used without the attention mechanism; and in the third, the complete model is used without the post-processing step. Table 4 presents the obtained results.

When the attention module is not used, the F1-score is slightly lower than baseline. A small drop in both precision and recall is observed in all datasets, except in the case of the CTW-1500 dataset, where Recall is slightly higher. These results indicate how the attention mechanism significantly helps in distinguishing text features from noise

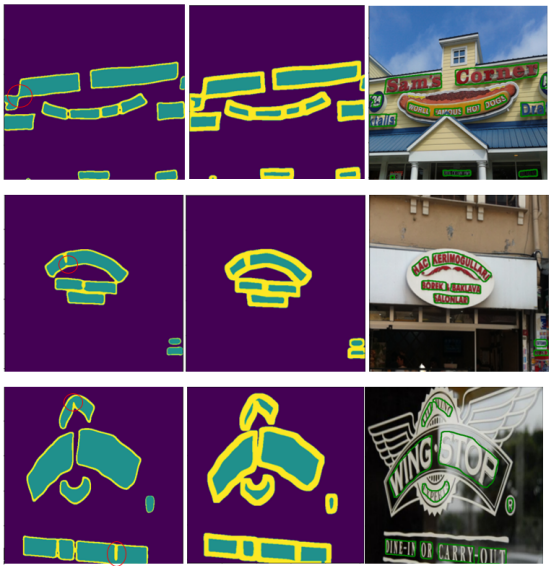

FIGURE 5. In the first column, showing the predictions made by our model, we can distinguish text regions (green color), boundary predictions (yellow color), and background (purple color). The second column presents the predictions after applying the post-processing operations. In the last column, we can find the polygons obtained in the final step.

FIGURE 6. Samples of predictions made by our model for detecting both curved and multi-oriented text. The last row highlights some false predictions.

	CTW-1500			Total-Text			MSRA-TD500		
	P	R	F1	P	R	F1	P	R	F1
Model With attention	83.75	81.47	82.47	88.65	80.49	84.37	85.23	81.90	83.53
Model No attention	79.23	82.44	80.80	87.31	78.99	82.94	82.06	78.82	80.41
Model No Post-Processing	82.60	77.55	79.99	85.94	77.49	82.26	81.40	73.40	77.19

TABLE 4. Ablative Study experiments on CTW1500, Total-Text, and MSRA-TD500. The proposed model is trained under various conditions: baseline, without the attention mechanism, and without post-processing."

and improving detection consistency. In the absence of post-processing, we observe a significant drop in recall, particularly for the CTW-1500 and MSRA-TD500 datasets, with decreases of 3.92% and 8.5%, respectively. This outcome is expected, especially for the MSRA-TD500 dataset, where text regions are typically small, and the spacing between them is often narrow, making boundary refinement crucial for accurate detection. On the other hand, the results for the Total-Text dataset are somewhat surprising. Given that text in this dataset is annotated at the word level, the risk of merging adjacent text regions is higher, yet the drop in recall is less pronounced. This may suggest that the model, even without post-processing, is better at handling the separation of word-level annotations in this particular dataset.

In all cases, it is clear that the results obtained without post-processing are inferior to those of the baseline model across all metrics: precision, recall, and F1-score. This highlights the importance of the attention mechanism and the post-processing in improving the accuracy and reliability of text detection, particularly in challenging scenarios with closely spaced or overlapping text regions.

6. **Conclusion.** In this work, we propose a novel architecture and post-processing pipeline for text detection in multi-oriented and curved formats. Our approach specifically addresses the challenge of adjacent text regions being in close proximity, which often leads to the model merging them, a scenario that negatively impacts performance evaluation, as such predictions are considered false positives.

The proposed model not only detects text regions but also delineates their boundaries, leveraging post-processing to correct potential errors arising during prediction. The results obtained across various datasets are promising and demonstrate the potential of the approach. Future improvements could include refining the attention module, revisiting the overall architecture, or enhancing the post-processing pipeline to further boost performance.

7. **Acknowledgement.** This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco (Alkhawarizmi/2020/01).

REFERENCES

[1] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "Abcnet: Real-time scene text spotting with adaptive bezier-curve network," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9806–9815, 2020.

[2] F. Wang, Y. Chen, F. Wu, and X. Li, "Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection," *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[3] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *AAAI Conference on Artificial Intelligence*, 2016.

[4] A. Semma, S. Lazrak, and Y. Hannad, "Handwriting based writer identification using afragment encoding system," *J. Inf. Hiding Multim. Signal Process.*, vol. 15, pp. 304–318, 2024.

[5] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Real-time text detection with similar mask in traffic, industrial, and natural scenes," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[6] X. Qin, P. Lyu, C. Zhang, Y. Zhou, K. Yao, P. Zhang, H. Lin, and W. Wang, "Towards robust real-time scene text detection: From semantic to instance representation learning," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 2025–2034.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, 2016.

[8] H. Y. Darshan, G. M. T, and M. C. Hanumantharaju, "Text detection and recognition using camera based images," in *International Conference on Frontiers in Intelligent Computing: Theory and Applications*, 2014.

[9] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1241–1248.

[10] P. Keserwani, A. Dhankhar, R. Saini, and P. P. Roy, "Quadbox: Quadrilateral bounding box based scene text detection using vector regression," *IEEE Access*, vol. 9, pp. 36 802–36 818, 2021.

[11] L. Wu, S. Tian, Y. Wang, and P. Xiong, "Cpn: Complementary proposal network for unconstrained text detection," *ArXiv*, vol. abs/2402.11540, 2024.

[12] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 919–931, 2022.

[13] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9328–9337, 2018.

[14] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2017.

[15] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, "Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling," *IEEE Transactions on Image Processing*, vol. 29, pp. 1999–2012, 2020.

[16] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[17] C.-K. Chng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 935–942, 2017.

[18] Y. Liu, L. Jin, S. Zhang, and S. Zhang, "Detecting curve text in the wild: New dataset and new solution," *ArXiv*, vol. abs/1712.02170, 2017.

[19] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1 – 35, 2020.

[20] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, 2012.

[21] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: An efficient and accurate scene text detector," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2642–2651, 2017.

[22] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognit.*, vol. 90, pp. 337–345, 2019.

[23] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6442–6451, 2019.

[24] C. Xue, S. Lu, and W. Zhang, "Msr: Multi-scale shape regression for scene text detection," in *International Joint Conference on Artificial Intelligence*, 2019.

[25] P. Dai, H. Zhang, and X. Cao, "Deep multi-scale context aware feature aggregation for curved scene text detection," *IEEE Transactions on Multimedia*, vol. 22, pp. 1969–1984, 2020.

[26] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, and W. L. Goh, "Towards robust curve text detection with conditional spatial expansion," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7261–7270, 2019.

[27] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4229–4238, 2019.

[28] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *AAAI Conference on Artificial Intelligence*, 2018.

[29] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *European Conference on Computer Vision*, 2018.

[30] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9357–9366, 2019.