Volume XXIX 2026
MBNA Publishing House Constanta 2026



Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Geometric Representations in Scene Text Detection: A Structural Perspective

To cite this article: Milutinovici Sorin, Racuciu Ciprian, Scientific Bulletin of Naval Academy, Vol. XXIX 2026, pg. 376-387.

Submitted: 27.04.2026
Revised: 25.10.2026
Accepted: 03.11.2026

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-26-I1-033
SBNA© 2026. This work is licensed under the CC BY-NC-SA 4.0 License

Geometric Representations in Scene Text Detection: A Structural Perspective

Milutinovici Sorin[1,*], Racuciu Ciprian[2,]
[1]"Titu Maiorescu" University, Faculty of informatics, sorin.milutinovici@prof.utm.ro
[2] Military Technical Academy, Doctoral School
*** sorin.milutinovici@prof.utm.ro**

Abstract. Automatic detection of text regions in natural images, sometimes called scene text detection, is a difficult problem in computer vision due to the high variability of text in the appearance, orientation, and geometric structure. Existing approaches are typically categorized based on architectural design or historical development, while the role of geometric representation is treated implicitly. In this paper, we formulate scene text detection as a structured prediction problem, in which the output consists of geometric objects embedded in the image plane. We propose a unified taxonomy of geometric representations, covering parametric models such as bounding boxes and quadrilaterals, as well as dense representations based on masks, affinity maps, and geometric fields. As methods evolve to address increasingly complex text instances, including rotated, deformed, and partially occluded text, representations shift from compact, explicit parameterizations toward flexible, implicit encodings that require reconstruction from dense predictions. This perspective highlights the trade-offs between representational expressiveness, computational cost, and post-processing complexity. In structured scenarios such as news broadcast text extraction, simpler geometric models provide an effective balance between accuracy and efficiency, while more complex representations are primarily justified in the presence of highly irregular text layouts.
Keywords: Detection Pipelines; Scene Text Detection; Geometric Modelling

1. Introduction

The automatic detection of text placement in natural images has evolved into a specific problem in computer vision, referred to as scene text detection. Unlike classical document analysis, where text is structured, uniform and well aligned, text in natural scenes appears under highly variable conditions: it may be small, elongated, arbitrarily oriented, distorted by perspective, or even arranged along curved trajectories. In addition, it often coexists with complex backgrounds, varying illumination, and visual noise. These characteristics make text fundamentally different from the compact, well-delimited objects typically considered in general object detection (Ye & Doermann, 2015; Zhu et al., 2016).

Modern approaches can be broadly categorized based on their prediction strategy into region-based methods, regression-based methods, dense prediction approaches methods (Naiemi et al., 2022) and query-based transformer frameworks (Zhu et al., 2023). While very different, these approaches have the same goal: to localize text instances by predicting various geometric structures in the image plane.

Existing literature tends to describe these methods either chronologically or in terms of architectural differences or training strategies. In this work, we adopt a different perspective. We argue that the central design choice in scene text detection is not the network architecture itself, but the geometric representation used to encode text instances in the available datasets. Choosing to predict axis-aligned bounding boxes, oriented rectangles, quadrilaterals, pixel-wise masks, or dense geometric fields means

choosing the nature of the prediction problem, the loss function, and the required post-processing steps (Zhou et al., 2017; Long et al., 2018; Baek et al., 2019; Zhu et al., 2021).

From this perspective, scene text detection can be formulated as a problem of reconstructing geometric structures embedded in the image. Different methods correspond to different ways of parameterizing these structures: as global objects described by a small set of parameters, or as spatially distributed signals that must be aggregated into coherent instances. This leads to multiple types of representations, from simple, explicit models (e.g., axis-aligned bounding boxes) to highly flexible, implicit ones (e.g., segmentation masks and geometric fields) (Long et al., 2018; Zhu et al., 2021).

Recent transformer-based networks simplify the detection pipeline by treating text instances as a direct set prediction. Instead of generating thousands of candidate 'proposals' that require manual filtering, these models use object queries to identify and localize text instances directly. This moves the effort of organizing and refining detections from external post-processing scripts into the model attention mechanisms (Carion et al., 2020; Zhang et al., 2022).

2. Scene Text Detection as Geometric Prediction

Scene text detection can therefore be formulated as a problem of predicting structured geometric entities embedded in the image plane. Given an input image, the objective is to identify and localize instances of text through explicit or implicit geometric descriptions. Each detected instance corresponds to a subset of the image domain that satisfies both visual and structural constraints associated with text (Zhou et al., 2017; Long et al., 2018; Baek et al., 2019).

The output of a scene text detector can be viewed as a set of geometric objects:

$$Y = \{G_1, G_2, G_3, \cdots, G_n\} \tag{1}$$

where each G_i represents a text instance encoded via a particular geometric representation. The shape of G_i varies it may be an axis aligned rectangle, a rotated rectangle, a polygon, a pixel region, or a more complex structure (Zhou et al., 2017; Long et al., 2018; Zhu et al., 2021).

This perspective highlights an essential distinction between scene text detection and standard classification problems. In classification, the output space is discrete and low-dimensional, typically consisting of a small number of categories. In contrast, scene text detection involves predicting elements of a hierarchical, high-dimensional, structured output space, where each prediction must capture both the location and the shape of an object. As discussed in the context of object detection, such problems are inherently structured, involving dependencies between multiple predicted components (Ren et al., 2015)

A key consequence of this formulation is that the design of a detection system is largely determined by how the geometric objects G_i are parameterized. Different parameterizations induce different learning problems. Representing text as axis-aligned rectangles implies predicting a small set of continuous parameters per instance, while representing text as a pixel mask moves the problem towards dense classification over the entire image (Zhou et al., 2017; Wang et al., 2019). Quadrilaterals or polygons, introduce additional degrees of freedom, needed for rotated of arbitrary shaped text instances that still retain an explicit geometric structure (Long et al., 2018; Zhu et al., 2021)

These choices directly affect all the fundamental components:

- **Prediction mechanism** — whether the network produces sparse, instance-level outputs (e.g., bounding boxes) or dense predictions (e.g., segmentation maps) (Ren et al., 2015; Wang et al., 2019).
- **Loss formulation** — whether the error is computed over a small set of parameters, over pixel-level classifications, or over more complex structural relationships (Zhou et al., 2017; Deng et al., 2018).
- **Post-processing strategy** — whether the final detections are obtained through simple NMS, connected component analysis, or complex geometric reconstructions (Deng et al., 2018; Wang et al., 2019).

Early approaches favoured compact, low-dimensional representations that were easy to interpret but limited in their ability to model more complex shapes. More recent methods favour dense, spatially distributed representations that offer greater flexibility at the cost of increased computational complexity and significantly more elaborate post-processing.

In the area of scene text detection, this shift is particularly significant. The variability in text appearance—especially in terms of orientation, aspect ratio, and curvature—makes it difficult to capture all relevant cases using simple parametric models. As a result, many modern approaches rely on dense predictions that encode local geometric information, which is subsequently integrated into coherent text instances (Baek et al., 2019; Zhu et al., 2021)

This paper builds on this formulation by analysing the different geometric representations used in practice and by organizing them along the spectrum from explicit, low-dimensional parameterizations to implicit, dense encodings.

3. Design Dimensions of Geometric Representations

The previous section emphasizes that scene text detection consists in predicting geometric structures. However, the ways in which these structures are encoded vary significantly across methods. It is useful to identify a set of fundamental dimensions along which they differ and not look at them as unrelated design choices. These dimensions can present as a common space in which all geometric representations can be positioned and compared.

3.1. Explicit vs Implicit Geometry

A primary distinction concerns whether the geometry of a text instance is represented explicitly or implicitly.

In **explicit representations**, the geometry is parameterized and predicted as a compact set of values. Examples are axis-aligned bounding boxes, oriented rectangles, or quadrilaterals, where the output of the model works toward a compact geometric description of the object. Representative methods include Wang et al. (2012), Jaderberg et al. (2014), DeepText (Zhong et al., 2016), and TextBoxes (Liao et al., 2016) for axis-aligned boxes, as well as EAST (Zhou et al., 2017) and TextBoxes++ (Liao et al., 2018) for oriented or quadrilateral representations. Recent transformer-based approaches follow a similar explicit formulation, predicting instance geometry through set-based outputs without relying on dense spatial reconstruction (Carion et al., 2020; Zhang et al., 2022).

In contrast, **implicit representations** do not encode the geometry in a single prediction. Instead, they produce spatially distributed signals—such as segmentation masks or geometric fields—from which the final shape must be reconstructed. In this case, geometry is not predicted directly but emerges from the aggregation of local predictions across the image domain. Examples of this approach include PixelLink (Deng et al., 2018), CRAFT (Baek et al., 2019), and PSENet (Wang et al., 2019), which rely on pixel-level predictions followed by reconstruction procedures such as thresholding, grouping, or expansion, as well as TextSnake (Long et al., 2018), which encodes geometry through dense geometric fields, and FCENet (Zhu et al., 2021), which predicts dense features that parameterize contours using a fixed set of Fourier coefficients.

This distinction shows different ways of encoding geometric information, which may be combined with either direct parameter estimation or reconstruction from dense predictions. These dimensions are conceptually orthogonal: the form of geometric encoding is independent of whether predictions are produced globally or through dense spatial outputs.

3.2. Global vs Local Encoding

In **global representations**, each text instance is described by a single, compact prediction. The model produces one set of parameters per object, typically corresponding to a bounding text region. This approach yields a low-dimensional output but constrains the shape of the object. Representative

examples include Wang et al. (2012), Jaderberg et al. (2014), DeepText (Zhong et al., 2016), TextBoxes (Liao et al., 2016), as well as oriented extensions such as EAST (Zhou et al., 2017) and TextBoxes++ (Liao et al., 2018. Transformer-based approaches adopt a similar global formulation, in which each text instance is predicted directly, without using dense intermediate representations (Carion et al., 2020; Zhang et al., 2022).

In **local (dense) representations**, geometric information is distributed across the image, with predictions made at the level of pixels or small regions. Each location contributes partial information about the presence or structure of text, and the final instances are obtained by integrating these local signals. Examples include PixelLink (Deng et al., 2018), CRAFT (Baek et al., 2019), and PSENet (Wang et al., 2019), - pixel-level predictions followed by reconstruction, as well as TextSnake (Long et al., 2018), TextField (Xu et al., 2019), and FCENet (Zhu et al., 2021), which encode geometry through dense directional or parametric fields.

3.3. Fixed vs Variable Dimensionality

Fixed-dimensional representations describe each text instance using a fixed number of parameters. Typical examples are rectangular bounding boxes (axis aligned or rotated) and quadrilaterals - the number of predicted values per instance is constant. These representations are well-suited for regression frameworks and are relatively stable during training. Furthermore, these methods often demonstrate superior data efficiency; they can typically achieve competitive performance with a significantly smaller volume of training material compared to more flexible, high-dimensional, or implicit alternatives that must learn geometric constraints from scratch. Representative methods include Wang et al. (2012), Jaderberg et al. (2014), DeepText (Zhong et al., 2016), and TextBoxes (Liao et al., 2016). Transformer-based approaches also adopt fixed-dimensional representations (Carion et al., 2020; Zhang et al., 2022). However, transformer-based approaches often exhibit lower data efficiency than their purely convolutional counterparts. Although they typically leverage **convolutional backbones** for initial feature extraction, the subsequent **transformer-based detection stages** lack the rigid geometric priors—such as anchor boxes or fixed sliding windows—inherent in traditional architectures. Without these 'architectural shortcuts' to guide object grouping, the model must rely on its attention mechanisms to learn spatial relationships and instance boundaries from the data itself. Consequently, these models require a **massive volume of training material** to achieve the same geometric stability that traditional regression-based detectors gain through their structural design.

Variable-dimensional representations, - polygons with an arbitrary number of vertices or contour-based geometries, allow modelling much more complex shapes. But the need to handle sequences of varying length and to define appropriate loss functions increases complexity. Examples include PSENet (Wang et al., 2019), PAN (Wang et al., 2019), and DBNet (Liao et al., 2020), where polygonal contours with a variable number of vertices are obtained through post-processing of segmentation outputs.

3.4. Parametric and Non-Parametric Representations

Parametric representations, describes the instances using a small set of real parameters. They may correspond to positions, dimensions, orientations, or coefficients of a predefined model (e.g., control points of a curve). Parametric representations are compact and easier to interpret but will struggle with highly irregular shapes. Representative methods include TextBoxes (Liao et al., 2016), where geometry is predicted as global bounding box parameters, as well as EAST (Zhou et al., 2017) and FCENet (Zhu et al., 2021), which predict parametric geometry through dense, spatially distributed outputs. Transformer based approaches also follow a parametric description - the geometry of each text instance is directly predicted, rather than reconstructed from dense spatial signals (Carion et al., 2020; Zhang et al., 2022).

In **non-parametric representations**, the geometry is defined implicitly by pixels belonging to a region or vectors describing local properties. These representations do not impose a global parametric form on the instance shape. However, they require additional processing to extract explicit object boundaries and are generally more computationally demanding. Examples include PixelLink (Deng et

al., 2018), CRAFT (Baek et al., 2019), and PSENet (Wang et al., 2019), where geometry is reconstructed from dense predictions through procedures such as thresholding, grouping, or contour extraction.

3.5. Geometry Encoding Strategy

Some methods rely on **direct regression of geometric parameters**, where the model predicts object-level descriptors such as bounding boxes or polygon vertices. Representative examples include EAST (Zhou et al., 2017) and TextBoxes (Liao et al., 2016), where geometry is obtained through regression of bounding box parameters. Other methods use **classification**, often using predefined geometric hypotheses (e.g., anchors), and the process is selecting and refining candidate regions. Examples include DeepText (Zhong et al., 2016) and TextBoxes (Liao et al., 2016), which rely on anchor-based classification and regression, as well as earlier region-proposal-based methods such as Jaderberg et al. (2014).

A third category contains **dense prediction**, where the model outputs spatial maps—such as probability maps, affinity maps, or geometric fields that encode local information about text presence and structure. These maps must then be postprocessed to obtain text instances. Representative methods include PixelLink (Deng et al., 2018), CRAFT (Baek et al., 2019), and PSENet (Wang et al., 2019).

A fourth strategy is introduced by transformer-based approaches, which formulate detection as **set-based prediction**, based on a fixed number of object queries. These queries act as internal search probes that identify and localize each word directly within the network's layers, producing a final list of detections that requires no manual assembly or grouping. (Carion et al., 2020; Zhang et al., 2022).

3.6. Reconstruction Complexity

In some representations, the output of the model corresponds directly to the result (text instances). Bounding box-based methods, for example, need **low reconstruction** - non-maximum suppression and simple filtering. Representative methods include TextBoxes (Liao et al., 2016) and DeepText (Zhong et al., 2016), where the model outputs can be converted to final detections with minimal additional processing. In contrast, methods such as EAST (Zhou et al., 2017) rely on dense predictions that require local aggregation and suppression to produce final instances.

Other representations require **medium reconstruction**, - predicted parameters must be refined, merged, or filtered to produce tet instances. Examples include TextBoxes++ (Liao et al., 2018), where quadrilateral predictions are refined, and CTPN (Tian et al., 2016), where sequences of local proposals are aggregated into word-level detections.

Dense and field-based representations, require **high-level reconstruction**. The final geometry must be reconstructed from distributed predictions, often involving thresholding, connected component analysis, graph-based grouping, or iterative expansion procedures, shifting significant complexity to the post-processing stage. Representative methods include PixelLink (Deng et al., 2018), CRAFT (Baek et al., 2019), PSENet (Wang et al., 2019), and TextSnake (Long et al., 2018).

Transformer-based approaches represent an extreme case in which **no reconstruction** is needed, as text instances are predicted directly through set-based formulations without requiring grouping, clustering, or other post-processing procedures (Carion et al., 2020; Zhang et al., 2022).

4. Geometric Representations

Both datasets and models usually decide on a certain mode of representation of text.

4.1. Axis-Aligned Bounding Boxes

Axis-aligned bounding boxes represent the simplest geometric formulation. Each text instance is encoded using a fixed set of parameters describing position and size. This representation is explicit, instance-level, fixed-dimensional, and parametric. Each box is described by four parameters:

$$R = \{x_0, y_0, w, h\} \tag{2}$$

where x0, y0 is a reference point of the rectangle, the coordinates of the box center or the top left corner. In Fig.1 a) we can see the axis-aligned bounding boxes used to represent the text instances.

Axis-aligned bounding boxes were used extensively in early methods and remain common as intermediate representations or coarse localization stages in modern pipelines, particularly due to their compatibility with region-proposal-based object detection frameworks. Representative methods that directly predict such boxes (explicit representations) include Wang et al. (2012), Jaderberg et al. (2014), DeepText (Zhong et al., 2016), and TextBoxes (Liao et al., 2016). Other approaches, such as CTPN (Tian et al., 2016), WeText (Tian et al., 2017), and WordSup (Hu et al., 2017), also produce bounding boxes as final outputs, but these are obtained through grouping or aggregation of lower-level predictions rather than being directly regressed by the model (implicit representations).

Available datasets were also one of the reasons some of these methods choose the axis-aligned representation, the methods that are based on grouping of lower-level predictions could be converted to produce more complex geometries, as we will see later.

From the perspective of the dimensions defined in Section 3, bounding boxes occupy the most constrained region of the representation space. They assume that text instances can be approximated by compact rectangular regions aligned with the image axes. This assumption simplifies both prediction and post-processing, allowing the use of standard regression losses and non-maximum suppression.

However, this simplicity comes at the cost of limited expressiveness. Scene text often violates these assumptions, appearing rotated, elongated, or fragmented across multiple regions. As a result, axis-aligned representations are insufficient for many real-world scenarios and are primarily associated with early approaches or with intermediate stages in more complex pipelines.

4.2. Oriented Bounding Boxes

Oriented bounding boxes extend the previous representation by introducing an additional parameter that encodes rotation. This preserves the advantages of explicit, global, and fixed-dimensional representations while partially addressing the limitations related to orientation.

Oriented bounding boxes add a supplementary parameter

$$R = \{x_0, y_0, w, h, \theta\} \tag{3}$$

where θ is the rotation angle of the box. In the representation space, oriented boxes remain parametric and compact, but offer increased flexibility compared to axis-aligned boxes. They are particularly effective for text that is approximately linear but rotated with respect to the image axes. Representative methods include instance-level regression detectors that predict rotated rectangles, such as RRPN (Ma et al., 2018) and R2CNN-based adaptations for scene text. In Fig.1 b) we can see the oriented bounding boxes used to represent the text instances.

Despite this improvement, the representation remains fundamentally limited by its rectangular structure. It cannot adequately capture curved text or complex shapes and still imposes a global approximation over potentially non-uniform structure.

4.3. Quadrilateral Representations

Quadrilateral representations further relax geometric constraints by allowing each text instance to be described by four arbitrary corner points. This removes the requirement of orthogonality and allows the approximation of perspective distortions commonly encountered in natural scenes.

$$Q = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\} \tag{4}$$

where x_i, y_i are the coordinates of the four corners of the quadrilateral. Within the design space, quadrilaterals are explicit, global, and fixed-dimensional, but more flexible than rectangular models.

They increase geometric flexibility compared to rectangles while maintaining a fixed number of parameters, maintaining a manageable parameterization while accommodating a broader class of shapes. Representative methods include TextBoxes++ (Liao et al., 2018), CAST (Baek et al., 2020) where text instances are described using four-point polygonal approximations. Other methods such as EAST (Zhou et al., 2017) produce quadrilateral geometry through dense predictions followed by aggregation. In Fig.1 c) we can see the quadrilaterals used to represent the text instances.

However, the representation still assumes that each instance can be approximated by a four-sided polygon. This assumption stops working in case of curved or highly irregular text, where a single quadrilateral cannot capture the underlying structure.



a) b) c)

Figure 1. a) axis-aligned rectangles; b) rotated rectangles; c) quadrilateral representation

4.4. Polygonal Representations

Polygonal representations generalize quadrilaterals by allowing a larger number of vertices, either fixed or variable. This significantly increases geometric flexibility, enabling the approximation of complex contours. A polygon is usually defined by a variable set of points with a certain order (clockwise, counterclockwise)

$$P = \{(x_i, y_i)\}_{i=1}^{N}, \ (x_i, y_i) \in R^2 \tag{5}$$

In terms of the design dimensions, polygonal representations remain explicit and global but are typically obtained through post-processing of dense predictions, such as segmentation maps or geometric fields. Representative methods are PSENet (Wang et al., 2019), PAN (Wang et al., 2019), and DBNet (Liao et al., 2020), where polygon instances contours are computed by a succession of thresholding, grouping, and contour extraction procedures applied to pixel outputs.

While polygons can approximate arbitrary shapes with sufficient resolution, their increased flexibility shifts complexity to the reconstruction stage, where contour extraction, smoothing, and consistency of the recovered boundaries must be carefully handled.

4.5. Curve-Based Representations

Curve-based representations model text using parametric curves that explicitly describe the shape of the text instance. These methods view text regions as continuous curves, such as Bezier or spline functions, defined by a fixed set of control points. This representation allows for a compact yet flexible encoding of complex shapes and are explicit, global, and parametric. Representative methods include ABCNet (Liu et al., 2020), which models text instances using Bezier curves, as well as subsequent spline-based approaches that directly regress control points defining the text contour.

Curve-based representations offer a very efficient description of the text shapes, reducing the number of parameters required especially for curved text instances. However, they require reconstruction of the full region from the predicted curve, and their effectiveness depends on how well the chosen parametric model can approximate the real text geometry.

4.6. Instance-Aware Dense Representations

To address the limitations of simple masks, many methods introduce additional channels that encode relationships between pixels. These include affinity maps indicating if neighbouring pixels belong to the same instance or various linking mechanisms that model connectivity. Representative methods include PixelLink (Deng et al., 2018), which predicts links between adjacent pixels, CRAFT (Baek et al., 2019) and SegLink (Shi et al., 2017), which models both character regions and affinities between them.

These representations remain dense and implicit but incorporate relational information that facilitates instance separation. They can be viewed as extending mask-based approaches with structured signals that encode grouping constraints. From the perspective of the design dimensions, they increase the amount of local information available at each pixel, at the cost of more complex prediction targets and post-processing procedures.

4.7. Geometric Field Representations

Geometric field representations encode local geometric properties at each pixel, such as direction vectors, distances to boundaries, or other spatial cues. The model learns how pixels relate to the underlying geometric structure of text. Representative methods include TextField (Xu et al., 2019), which predicts direction fields pointing away from text boundaries, TextSnake (Long et al., 2018), which encodes local geometry through centreline, radius, and orientation maps, and FCENet (Zhu et al., 2021), which represents text contours using Fourier-based coefficients defined over the image.



Figure 2. Example of Fourier fitting for arbitrary-shaped texts (green contours) with increasing Fourier degree K (from left to right, K=3, 5, 10), where green contours denote ground-truth - FCENet (Zhu et al., 2021)

This approach represents the most flexible end of the representation spectrum. It can model highly irregular and curved text but requires complex reconstruction algorithms and loss functions.

4.8. Hybrid Representations

Some methods combine multiple representations to leverage their respective advantages. For example, a model may predict both bounding boxes and segmentation masks or combine dense fields with explicit geometric parameters. Representative methods include Mask TextSpotter (Lyu et al., 2018), which combines detection and segmentation, DBNet (Liao et al., 2020), which combines segmentation with differentiable binarization, and PAN (Wang et al., 2019), which utilizes multi-scale kernel prediction to isolate text instances, followed by a progressive enlargement algorithm to reach the final contours.

Hybrid approaches achieve strong empirical performance with increased architectural and training complexity. Transformer-based approaches generally maintain simple geometric representation, using a very different prediction paradigm - instances are directly inferred through set-based formulations. They operate over explicit, instance-level representations while eliminating the need for intermediate spatial encodings and reconstruction procedures (Carion et al., 2020; Zhang et al., 2022).

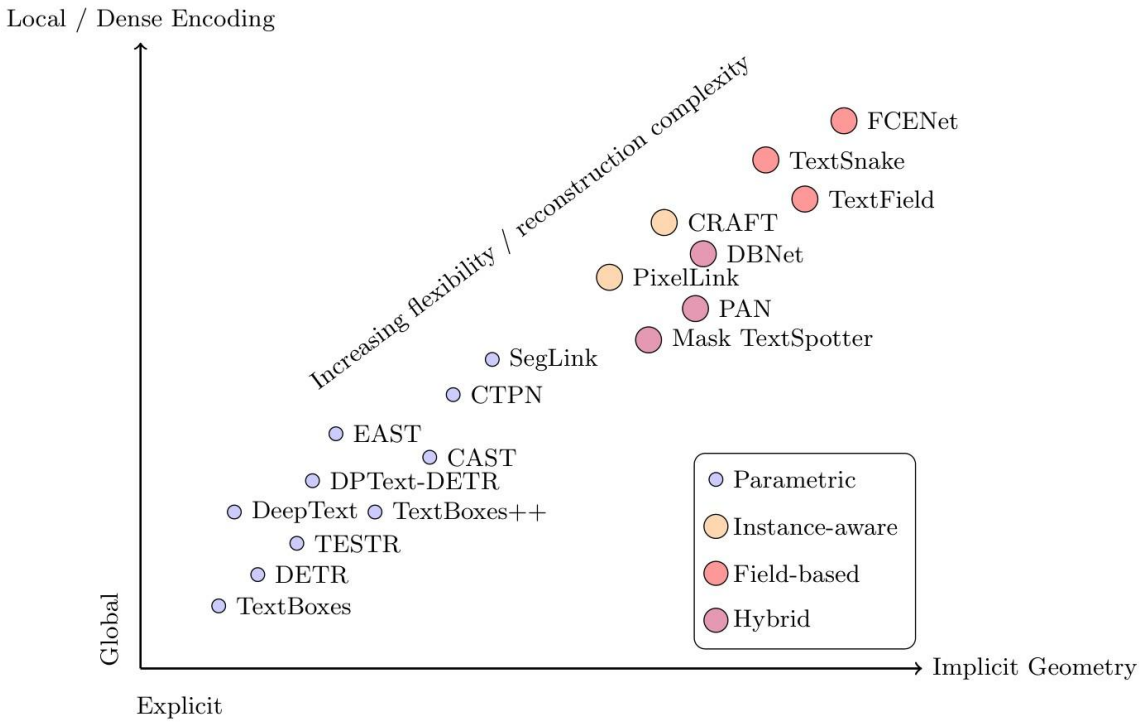


Figure 3. Geometric representation space for scene text detection methods. The horizontal axis reflects the transition from explicit to implicit representations, while the vertical axis captures the shift from global, instance-level predictions to local, dense encoding. Methods are positioned according to their underlying representation, illustrating the progression from compact parametric models to flexible dense and field-based approaches, along with the associated increase in reconstruction complexity.

5. Design Dimensions of Geometric Representations

5.1. Trade-offs Across Representations

The different categories of representations reflect a set of fundamental trade-offs that cannot be avoided, but only balanced.

A first trade-off concerns flexibility versus simplicity. Compact parametric representations are simple to predict and efficient to process but assume a very simple shape of text instances. As the representation increases its flexibility - polygons, masks, or geometric fields more complex shapes can be modelled. This increased flexibility comes at the cost of significantly larger outputs, more complex loss functions, and an increase in post-processing.

A second trade-off is between explicit and implicit geometry. Explicit representations require a small number of parameters, making the prediction problem well-defined and easy interpretable. Implicit representations, require additional steps to reconstruct object boundaries.

A third trade-off concerns global versus local encoding. Global representations produce one prediction per instance, leading to compact outputs but limited adaptability to local variations. Local,

dense representations encode information at the pixel level, capturing fine-grained structure but requiring aggregation mechanisms to form coherent instances.

5.2. Why Dense Representations Dominate

The prevalence of dense representations in recent methods is not accidental but reflects that text in natural images does not typically form compact, convex regions. It appears as elongated structures, composed of straight, rotated, or curved characters distributed along lines of various shapes. The boundaries of these structures are not always well-defined but emerge from local regularities such as alignment, spacing, and continuity. Dense representations address this challenge by modelling text at local level. This allows the detector to adapt to increasingly complex shapes without imposing restrictive global assumptions.

5.3. Representation Suitability Across Scenarios

Different types of instance representations may be better suited to different types of scenes and text distribution.

Parametric representations - bounding boxes and quadrilaterals - are appropriate in scenarios where text is predominantly horizontal or slightly rotated, and where computational efficiency is critical.

More complex representations are better suited for scenes with perspective distortion or moderate curvature of the text lines.

Dense representations, including pixel-level masks and geometric fields, are particularly effective in complex environments where text exhibits arbitrary shapes, strong curvature, or fragmentation, but at the cost of increased computational and algorithmic complexity.

The choice of representation should therefore be guided by the characteristics of the target data and by the constraints of the application, rather than by a single notion of optimality.

5.4. The Role of Post-Processing

An important consequence of the shift toward implicit representations is the increasing role of post-processing in the overall system.

In explicit representations, the output of the model closely matches the desired result. Post-processing is limited to filtering redundant detections, typically through non-maximum suppression.

Implicit and dense representations require reconstruction steps to convert the distributed predictions into discrete text instances. These steps usually involve thresholding, connected component analysis, graph-based clustering, or region expansions. Field-based methods add additional procedures needed to integrate local geometric information into coherent shapes.

We can see that increased flexibility of representations moves complexity from the model to the reconstruction stage. Therefore, the performance of a detection system depends not only on the quality of the learned representation, but also on the effectiveness of the post-processing algorithm.

6. Conclusions

Scene text detection can be understood as a problem of predicting geometric structures embedded in the image plane. The representation of these structures determines the formulation of the prediction task, the complexity of the model, and the post-processing algorithms required to obtain final detections.

Geometric representations range from explicit, low-dimensional parameterizations - bounding boxes and quadrilaterals - to implicit, high-dimensional encodings based on dense masks or geometric fields. This progression increases the ability to work with complex and irregular text but also triggers higher computational cost and more elaborate reconstruction procedures.

In practical scenarios, the choice of representation must match the structure of the data and the operational constraints. A relevant example is the extraction of text from news broadcasts, where text typically appears in well-defined regions of the screen: horizontal captions, scrolling tickers, titles, and overlays. In such cases, the geometry of text is relatively regular, with limited curvature and predictable

alignment therefore, simpler parametric representations may offer a favourable balance between accuracy and efficiency. They are sufficiently expressive to capture the predominant text layouts while avoiding the overhead and occasional instability associated with dense prediction and complex reconstruction. Axis-aligned boxes may be sufficient for strictly horizontal captions, while oriented boxes provide additional robustness to minor variations in alignment.

More flexible representations, such as pixel-wise masks or geometric fields, become advantageous only when the structure of text deviates significantly from these assumptions - for example, in the presence of strong perspective distortion, irregular layouts, or curved text.

This illustrates a broader principle: the effectiveness of a scene text detection method depends not only on its expressive power, but on how well its geometric assumptions align with the target domain. Simpler representations can be more effective when their constraints match the data, while more complex representations are justified only when the geometry of text requires them.

Future developments in scene text detection may benefit from adaptive approaches that adjust the level of geometric complexity to the characteristics of the input, selecting between compact parametric models and more flexible dense representations as needed.

References

1. Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character region awareness for text detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Deng, D., Liu, H., Li, X., & Cai, D. (2018). PixelLink: Detecting scene text via instance segmentation. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).
3. Naiemi, F., Ghods, V., & Khalesi, H. (2022). Scene text detection and recognition: a survey. Multimedia Tools and Applications, 81, 20255 - 20290.
4. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., & Yao, C. (2018). TextSnake: A flexible representation for detecting text of arbitrary shapes. Proceedings of the European Conference on Computer Vision (ECCV).
5. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems (NeurIPS).
6. Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu, G., & Shen, C. (2019). Shape robust text detection with progressive scale expansion network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
7. Ye, Q., & Doermann, D. (2015). Text detection and recognition in imagery: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.
8. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An efficient and accurate scene text detector. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
9. Zhu, Y., Yao, C., & Bai, X. (2016). Scene text detection and recognition: Recent advances and future trends. Frontiers of Computer Science.
10. Zhu, Y., Chen, J., Liang, J., Kuang, Z., & He, W. (2021). Fourier contour embedding for arbitrary-shaped text detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
11. Wang, K., Babenko, B., & Belongie, S. (2011). End-to-end scene text recognition. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1457–1464.
12. Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Reading text in the wild with convolutional neural networks. arXiv preprint arXiv:1412.1842.

13. Zhong, Z., Jin, L., & Huang, S. (2017). DeepText: A new approach for text proposal generation and text detection in natural images. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1208–1212.
14. Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. European Conference on Computer Vision (ECCV), 56–72.
15. Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J., & Ding, E. (2017). Wordsup: Exploiting word annotations for character based text detection. Proceedings of the IEEE International Conference on Computer Vision (ICCV).
16. Tian, Z., Huang, W., He, P., Qiao, Y., & He, X. (2017). Detecting text in natural image with semi-supervised learning. Proceedings of the International Conference on Document Analysis and Recognition (ICDAR).
17. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV).
18. Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017). TextBoxes: A fast text detector with a single deep neural network. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).
19. Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2018). TextBoxes++: A single-shot oriented scene text detector. IEEE Transactions on Image Processing, 27(8), 3676–3690.
20. Shi, B., Bai, X., & Belongie, S. (2017). Detecting oriented text in natural images by linking segments. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
21. Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
22. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., & Yan, J. (2018). FOTS: Fast oriented text spotting with a unified network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
23. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., & Xue, X. (2018). Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia, 20(11), 3111–3122. https://doi.org/10.1109/TMM.2018.2818020
24. Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., & Luo, Z. (2017). R2CNN: Rotational region CNN for orientation robust scene text detection. arXiv preprint arXiv:1706.09579. https://arxiv.org/abs/1706.09579
25. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV).
26. Zhang, Y., Nie, X., Liu, S., He, Y., Zhang, C., & Liu, X. (2022). TESTR: Text spotting transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
27. Ye, J., Chen, J., Liang, J., Liu, X., & Bai, X. (2022). DPText-DETR: Towards better scene text detection with dynamic points in transformer. arXiv preprint arXiv:2207.04491.