KAN-Based Superpixel Segmentation with Boundary Constraint and Semantic Guidance

Xiaohong Jia[1], Fuhai Wang[1], Tong Tong[1], Long Ma[1]* and Guanghui Yan[1]
[1]School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China
jiaxhm@lzjtu.edu.cn, wangfhai0109@163.com, tongt7@mail.lzjtu.cn, mmll529@163.com, ghyan@mail.lzjtu.cn

***Abstract*—Superpixel segmentation serves as a critical computational primitive for downstream vision tasks by clustering perceptually similar pixels into compact regions. Existing methods based on Fully Convolutional Networks often grapple with an inherent trade-off between boundary adherence and semantic consistency, particularly when handling complex scenarios. To resolve this inherent trade-off, we present the KAN-Based Superpixel Network (KBSNet), a tailored framework constructed to enforce geometric boundary constraints and simultaneously establish robust deep semantic representations. Structurally, we construct a hybrid encoder that pioneers the integration of Kolmogorov-Arnold Networks into superpixel generation. To refine feature representation, we design a Geometric Boundary Constraint Module (GBCM) to explicitly model physical contours. We introduce a Semantic-Guided Local Attention (SGLA) mechanism, which leverages deep features as "semantic agents" to suppress texture noise via top-down modulation. Extensive experiments on four benchmark datasets demonstrate that KBSNet outperforms state-of-the-art methods across ASA, BR-BP, UE, and CO metrics.**

***Index Terms*—Superpixel Segmentation, KAN, Semantic Guidance, Boundary Perception**

I. Introduction

As a cornerstone preprocessing technique in computer vision, superpixel segmentation aims to reorganize rigid pixel grids into perceptually meaningful compact regions, superseding raw pixels as the elementary units of image representation [1]. This mid-level visual representation not only drastically reduces the computational overhead of subsequent processing but also effectively preserves local structural integrity, thereby finding extensive applications in semantic segmentation [2], medical image analysis [3], object tracking [4], and salient object detection [5].

The evolution of this field has witnessed a continuous quest for optimal graph-partitioning strategies, ranging from the early Felzenszwalb and Huttenlocher (FH) algorithms [6] utilizing minimum spanning trees to Entropy Rate Superpixel (ERS) [7] balancing objectives via entropy rate functions. Subsequently, Simple Linear Iterative Clustering (SLIC) [8] revolutionized the field with efficient local clustering, inspiring a series of clustering-based methods such as Superpixels Extracted via Energy-Driven Sampling (SEEDS) [9], Linear Spectral Clustering (LSC) [10], and Simple Non-Iterative Clustering (SNIC) [11]. This progression continued until Jampani *et al.* [12] proposed Superpixel Sampling Networks (SSN), bridging the gap between traditional algorithms and deep learning by introducing differentiable modules. In 2020, Yang *et al.* [13] introduced Superpixel Convolutional Networks (SCN), which completely abandoned the traditional clustering paradigm and proposed an end-to-end generative framework based on Fully Convolutional Networks (FCN). By directly predicting the soft association probability between pixels and grids, SCN greatly improved inference efficiency, thus establishing the current mainstream benchmark for deep superpixel segmentation.

To resolve the inherent conflict between boundary adherence and semantic consistency, we propose the KAN-Based Superpixel Segmentation Network (KBSNet). Inspired by the non-linear mapping capabilities of Kolmogorov-Arnold Networks (KAN) [14], we construct a hybrid encoder to establish a robust semantic representation, distinguishing our approach from traditional local autocorrelation mechanisms [15]. We design a Geometric Boundary Constraint Module (GBCM) to explicitly enforce physical contour adherence via multi-scale edge signals. Furthermore, drawing on edge-guidance principles [16], we introduce a Semantic-Guided Local Attention (SGLA) mechanism. This module leverages deep features as "semantic agents" to filter texture noise, effectively unifying geometric precision with semantic integrity. The principal innovations proposed in this work are highlighted below:

- We pioneer the integration of KAN into the end-to-end superpixel generation task. By harnessing the powerful non-linear mapping capability of KAN to directly infer affinity probabilities from pixel-level features, we construct the first-ever KAN-based framework for superpixel generation.
- We devise GBCM and SGLA to resolve the inherent trade-off in superpixel segmentation by addressing the problem from the dual dimensions of explicit boundary constraints and high-level semantic guidance.
- Our quantitative analysis confirms that KBSNet surpasses current cutting-edge techniques on four standard datasets, showing significant improvements in ASA, BR-BP, UE, and CO metrics.

The remainder of this paper is organized as follows. Section II reviews related work on traditional and deep learning-based superpixel segmentation. Section III presents the proposed KBSNet, including the KAN, GBCM, and SGLA modules. Section IV reports experimental results and analyses. Finally, Section V concludes the paper.

*Corresponding author: Long Ma (mmll529@163.com)

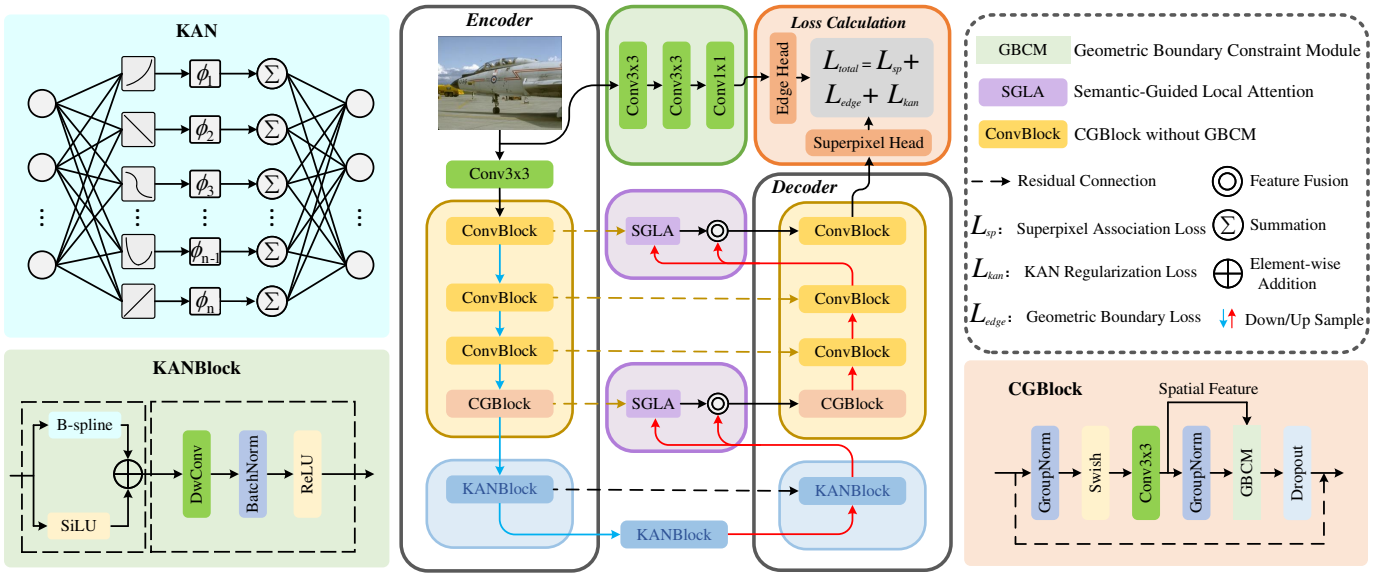


Fig. 1. KBSNet comprises four parts: an encoder for feature extraction, deep layers utilizing KANs for non-linear dependencies, and a decoder integrating SGLA. The edge head leverages shallow features for edge map prediction, while the superpixel head produces association maps from the decoder output. CGBlock denotes the specific deep residual blocks equipped with GBCM.

II. Related Work

A. Traditional Superpixel Segmentation

Existing traditional methods can be broadly divided into graph-based and clustering-based categories. Graph-theoretic methods [17], including FH, ERS, and Random Walks (RW) [18], optimize graph cuts for compactness, with Dynamic Random Walk (DRW) [19] further balancing boundary regularity. Clustering-based methods like SLIC achieved efficient, regular segmentation by restricting clustering to local neighborhoods.

This paradigm inspired various improvements: LSC and Manifold SLIC [20] optimized feature mapping for boundary adherence, while Fast Linear Iterative Clustering (FLIC) [21] and SNIC accelerated convergence via active search and non-iterative updates. Concurrently, methods such as Spatially Constrained Subspace Clustering (SCSC) [22], Local Adaptive Distance (LAD) [23], and Vine Spread for Superpixel Segmentation (VSSS) [24] explored advanced initialization, merging strategies, and adaptive distances. Despite these advancements, the reliance on fixed, handcrafted features and the non-differentiable nature of these methods constitute a fundamental barrier to their integration into end-to-end deep learning frameworks.

B. Deep Superpixel Segmentation

Deep learning has transformed superpixel segmentation techniques by introducing learnable hierarchical features. Early hybrid methods like Segmentation-Aware Affinity Loss with entropy rate superpixel (SEAL) [25] combined CNN with graph algorithms. Jampani *et al.* [12] later proposed Superpixel Sampling Networks (SSN), establishing the first differentiable clustering framework via soft pixel assignments. SCN solidified the FCN paradigm, reformulating segmentation as an efficient association prediction task.

Recent research addresses limitations in geometric fidelity and robustness. Association Implantation Network (AINet) [26] and Efficient Superpixel Network (ESNet) [27] enhance local perception and boundary adherence through implantation strategies and auxiliary edge losses, respectively. For cross-domain scenarios, Content Disentangle Superpixel (CDS) [28] extracts style-invariant features to bolster generalization. Furthermore, Biologically Inspired Network (BINet) [29] and Edge Guided Local-Global Attention Network (ELGANet) [16] improve boundary alignment via bio-inspired labels and multi-source edge fusion. Regarding KAN, while U-KAN [30] validated its efficacy in medical segmentation, existing adaptations like AEKAN [31] remain constrained to processing pre-generated patches. We aim to bridge this gap by establishing the first end-to-end KAN-based superpixel generation framework.

III. Proposed Method

A. Overview

As illustrated in Fig. 1, KBSNet adopts an asymmetric hybrid encoder-decoder architecture to synergize CNN's local perception with KAN's non-linear semantic modeling. The encoder incorporates a GBCM to enforce early contour adherence via multi-scale edge signals. Instead of using standard concatenation in the decoding stage, we integrate the SGLA module for feature fusion. This mechanism employs deep KAN features as "semantic agents" to modulate feature reconstruction, ensuring both geometric precision and semantic consistency.

B. KAN Module

To transcend the limited expressivity of CNN in approximating complex pixel affinities, we construct a tokenized

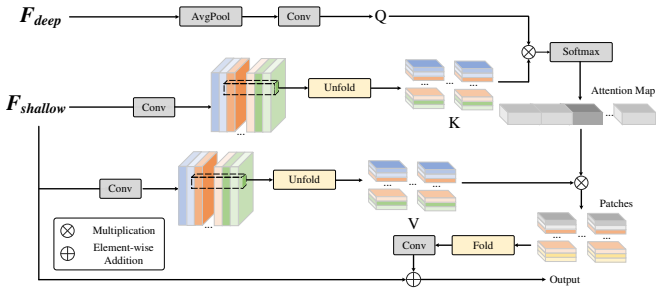


Fig. 2. In SGLA, shallow features $F_{shallow}$ generate local key-value pairs preserving high-resolution spatial structure, while deep features F_{deep} encode KAN-based semantics to produce "semantic agents" for high-level guidance.

KAN within the deep encoder. As depicted in the KAN block of Fig. 1, we adopt the core KAN philosophy by relocating learnable activation functions from nodes to edges (denoted as ϕ). This paradigm shift enables the efficient capture of intricate semantic dependencies with superior parameter efficiency. We extend KAN to the 2D visual tasks by converting feature maps into a sequence of visual tokens through overlapping patch embedding. This tokenization preserves local context while liberating the model from rigid convolutional grids. Consequently, the computation replaces simple linear weighting with the aggregation of non-linear transformations:

$$y_j = \sum_{i=1}^{n} \phi_{i,j}(x_i), \tag{1}$$

where y_j is the j-th component of the output feature, x_i denotes the i-th component of the input feature, $\phi_{i,j}$ the learnable univariate function connecting input i to output j, and n represents the input dimension. We formulate the univariate function ϕ by aggregating a set of B-spline basis functions $B_r(x)$:

$$\phi_{i,j}(x) = \sum_{r=1}^{G+k} c_{i,j,r} B_r(x), \tag{2}$$

where $c_{i,j,r}$ represents the learnable spline coefficients. The hyperparameters G and k determine the grid granularity and the polynomial order, respectively. We set the spline order $k = 3$ and the grid granularity $G = 5$. To balance convergence speed and stability, we employ a hybrid strategy parallelizing SiLU activation with B-spline transformations:

$$y_j = \sum_{i=1}^{n} \left(w_{i,j} \sigma_{base}(x_i) + \phi_{i,j}(x_i) \right), \tag{3}$$

where $w_{i,j}$ denotes the linear weight and σ_{base} is the SiLU function.

C. Semantic-Guided Local Attention Module

SGLA accepts two inputs: the non-linear semantic features F_{deep} extracted by KAN, and the shallow features $F_{shallow}$ rich in spatial detail. We leverage F_{deep} as the guidance for local aggregation. By employing adaptive average pooling, we align F_{deep} with the resolution of the target superpixel grid. As illustrated in Fig. 2, for the m-th attention head, the semantic agent Q_{agent} is generated via a linear projection. These agent vectors encapsulate robust class-discriminative information, facilitating the effective delineation of distinct object regions. The formulation of Q_{agent} is given by:

$$Q_{agent}^{(m)} = W_q \left(Pool \left(F_{deep}^{(m)} \right) \right), \tag{4}$$

where W_q denotes a 1×1 convolution, $Pool$ refers to average pooling.

We treat the shallow features $F_{shallow}$ as the source of fine-level spatial context. Instead of global dense computation, we employ a sliding window operation to unfold the local neighborhood features surrounding each spatial position. This mechanism allows the model to efficiently encode fine-level features and contexts into tokens, ensuring that each semantic agent preserves access to high-resolution texture details within a $k \times k$ receptive field, where k denotes the local window size. This effectively addresses the problem of traditional Transformers neglecting local details. The formulations for local keys $K_{local}^{(m)}$ and values $V_{local}^{(m)}$ are expressed as:

$$\begin{cases} K_{local}^{(m)} = Unfold \left(W_k \left(F_{shallow}^{(m)} \right) \right) \\ V_{local}^{(m)} = Unfold \left(W_v \left(F_{shallow}^{(m)} \right) \right) \end{cases}, \tag{5}$$

where $Unfold$ refers to the sliding window unfolding operation, and W_k, W_v denote learnable linear projections.

SGLA introduces a semantic-driven bias. We calculate the attention weights across M parallel heads by measuring the matching degree between the global "semantic agents" Q_{agent} and the fine-grained local spatial features K_{local}. This acts as a semantic filter, selecting the most relevant boundary details from the local window. The attention map A_{attn} is formulated as:

$$A_{attn}^{(m)} = Softmax \left(\frac{Q_{agent}^{(m)} \cdot (K_{local}^{(m)})^T}{\sqrt{\frac{C}{M}}} \right), \tag{6}$$

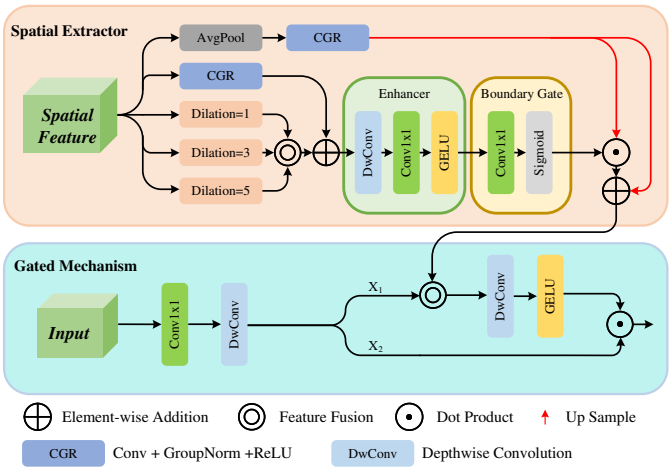


Fig. 3. In GBCM, the spatial extractor fuses multi-scale boundary contexts with spatial input. The gated mechanism splits features into gating (X_1) and content (X_2) streams; X_1 is concatenated with enhanced spatial features and modulated by X_2 to generate the output.

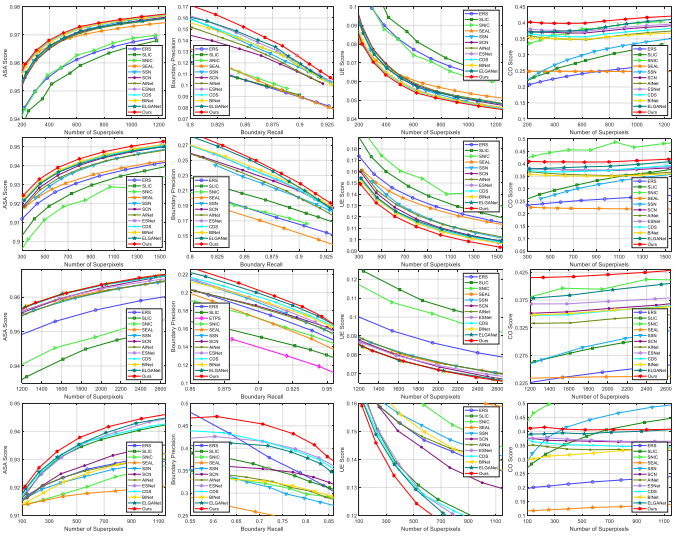


Fig. 4. Quantitative comparison of metrics on four datasets from different domains. From top to bottom: BSDS500, NYUv2, KITTI, and DRIVE datasets. From left to right: ASA, BR-BP, UE, and CO metrics.

where C represents the total channel dimension of the features.

The generated attention map is subsequently applied to the local values to synthesize reconstructed features via weighted aggregation. These features are then restored to their original resolution through folding and bilinear upsampling operations, followed by integration with the original input features via a residual connection:

$$F_{out} = Conv_{1\times1}\Big(Up\Big(Fold\Big(Concat_{m=1}^{M}\big(A_{attn}^{(m)} \cdot V_{local}^{(m)}\big)\Big)\Big)\Big) + F_{shallow}, \tag{7}$$

where $Fold$ denotes the spatial reshaping operation, $Conv_{1\times1}$ represents the 1×1 convolution, Up represents the upsampling operation, and $Concat$ denotes the channel concatenation operation.

D. Geometric Boundary Constraint Module

As illustrated in Fig. 3, GBCM consists of a Spatial Extractor for capturing high-frequency geometric priors and a Gated Mechanism for dynamic feature modulation. Our Spatial Extractor is designed to explicitly recover geometric boundary signals that may be diluted in deep layers. Taking the spatial feature map, we employ a multi-scale strategy to capture both fine edges and contextual contours. These multi-scale features $F_{ms\text{-}edge,l}$ are concatenated and subsequently fused via a 1×1 convolution to yield an integrated multi-scale edge feature map $F_{ms\text{-}edge}$:

$$F_{ms\text{-}edge} = Conv_{1\times1}\Big(Concat_{l=1}^{L}\big(F_{ms\text{-}edge,l}\big)\Big), \tag{8}$$

where L indicates the total number of feature scales.

The outputs from the two branches are fused via element-wise addition. Specifically, we combine the multi-scale edge map $F_{ms\text{-}edge}$ with the local edge features E_{local} extracted from the parallel shallow branch, yielding an enriched boundary map E_{fused}. This map also serves as an auxiliary output of the module, facilitating additional boundary supervision. The formulation of E_{fused} is given by:

$$E_{fused} = E_{local} + F_{ms\text{-}edge}. \tag{9}$$

The module leverages the generated E_{fused} to augment a parallel spatial feature stream. Specifically, E_{fused} is refined within the boundary attention gating mechanism to yield $E_{enhanced}$. Subsequently, $E_{enhanced}$ is fed into a boundary gating unit (comprising a 1×1 convolution and a sigmoid activation function) to generate the spatial attention map $A_{boundary}$:

$$A_{boundary} = \sigma\left(Conv_{1\times1}\left(E_{enhanced}\right)\right), \tag{10}$$

where σ denotes the sigmoid activation function.

Within the spatial feature stream modulation, the spatial information map S is processed via an independent branch to extract general spatial context features S_{feat}. These features are subsequently modulated by the previously generated boundary attention map $A_{boundary}$, yielding the boundary-enhanced spatial features $S_{enhanced}$:

$$S_{enhanced} = S_{feat} \odot \left(1 + A_{boundary}\right), \tag{11}$$

where $\odot$ denotes element-wise multiplication, and the term $(1 + A_{boundary})$ serves as a form of residual enhancement.

The input features are expanded and decomposed into a gating stream (X_1) and a content stream (X_2) via depthwise convolutions. This decomposition allows the model to separate feature transformation from spatial selection. To inject geometric constraints, the boundary-enhanced spatial feature $S_{enhanced}$ is concatenated with the gating stream X_1. This fused branch then undergoes a non-linear activation to form a "boundary-aware gate," which dynamically selects relevant information from the content stream X_2.

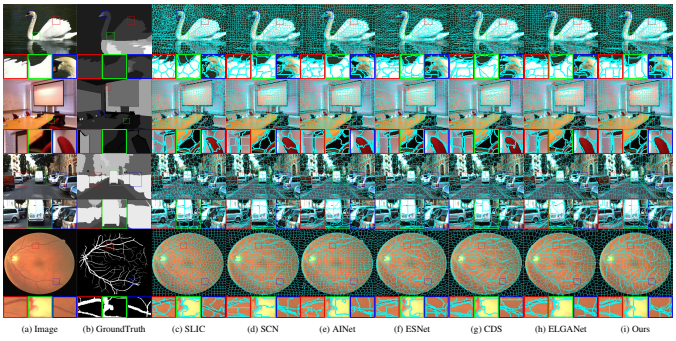


Fig. 5. Visual comparison of different algorithms. From top to bottom, the images are from the BSDS500, NYUv2, KITTI, and DRIVE datasets. Alternating rows display the segmentation details for each image.

E. Loss Function

Our loss function consists of three components: the superpixel association loss L_{sp}, the auxiliary geometric boundary loss L_{edge}, and the KAN regularization loss L_{kan}. The total loss function L_{total} is defined as follows:

$$L_{total} = L_{sp} + \lambda_1 L_{edge} + \lambda_2 L_{kan}, \tag{12}$$

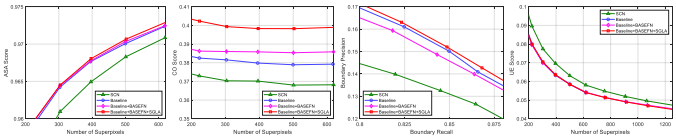


Fig. 6. Quantitative ablation results on the BSDS500 dataset in terms of four metrics.

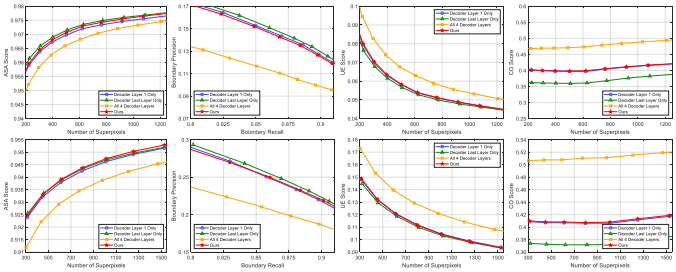


Fig. 7. From top to bottom, the metrics in the figure correspond to the BSDS500 and NYUv2 datasets.

where λ_1 and λ_2 are weighting coefficients designed to balance the auxiliary edge loss and KAN regularization, respectively. We set $\lambda_1 = 1.0$ and $\lambda_2 = 5 \times 10^{-4}$.

The superpixel association loss (L_{sp}) minimizes the reconstruction error of pixel features; it is calculated as the weighted sum of the distances between original and reconstructed color/spatial features, controlled by a compactness weight m and grid interval s [13]. The auxiliary geometric boundary loss (L_{edge}) employs a weighted binary cross-entropy term to supervise the edge probability predictions against groundtruth boundaries. The KAN regularization loss (L_{kan}) induces global sparsity by summing the mean norm of spline coefficients and the entropy of their distribution across all layers.

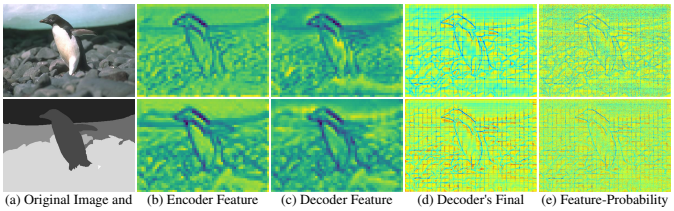
IV. Experiment

A. Experimental Settings

Following standard protocols [32], our model is trained on the BSDS500 dataset and evaluated across four diverse benchmarks. These include the standard BSDS500 [33] for natural scenes, the indoor RGB-D dataset NYUv2 [34] employing a subset of 400 test images, the KITTI [35] street-view dataset providing 200 high-resolution images for autonomous driving, and the medical DRIVE [36] dataset comprising 40 fundus images to assess cross-domain capability.

To comprehensively assess performance, we adopt four standard metrics. Achievable Segmentation Accuracy (ASA) evaluates the highest degree of semantic segmentation achievable. Boundary Recall and Precision (BR-BP) jointly evaluate boundary adherence and ground truth recovery. Furthermore, Under Segmentation Error (UE) assesses the leakage across ground truth boundaries, while Compactness (CO) quantifies shape regularity.

Implemented in PyTorch on a single NVIDIA RTX 3090, our model is optimized using AdamW with a batch size of 4, an initial global learning rate of 5×10^{-5}, a weight decay



(a) Original Image and GroundTruth (b) Encoder Feature Map (c) Decoder Feature Map (d) Decoder's Final Feature Map (e) Feature-Probability Overlay Map

Fig. 8. Visual comparison of intermediate feature responses. The top row shows the complete KBSNet, while the bottom row displays the baseline model.

of 1×10^{-5}, and cosine annealing. Inputs undergo random horizontal and vertical flips, and cropping to 208×208.

B. Comparative Experiments

We compare KBSNet against representative traditional and state-of-the-art methods, deriving results from public literature or official implementations. Quantitative metrics in Fig. 4 demonstrate our model's superior performance and robust generalization across four datasets. Qualitatively, Fig. 5 shows that our model generates regular and compact superpixel shapes even in large weak-texture regions, yielding highly satisfactory segmentation results.

C. Ablation Study

We conduct ablation studies on the BSDS500 dataset. As illustrated in Fig. 6, we evaluate four configurations:

SCN: The original FCN-based model. Baseline: Replaces bottleneck convolutions with KAN blocks. Baseline + GBCM: Substitutes specific residual blocks in the encoder and decoder with GBCM. Baseline + GBCM + SGLA (Ours): The complete model, introducing SGLA during decoder feature fusion.

Fig. 6 demonstrates that our Baseline significantly outperforms SCN across all metrics. Adding GBCM boosts Compactness via geometric constraints, while the subsequent integration of SGLA acts as a semantic filter to attenuate texture noise. This combination effectively balances geometric rigidity with semantic guidance, achieving a synergy between segmentation accuracy and shape regularity.

Regarding SGLA deployment, as shown in Fig. 7, we evaluated four strategies across BSDS500 and NYUv2. Deploying on all layers induces excessive smoothing, while shallow-only or deep-only insertion suffers from shape irregularity or signal dilution, respectively. In contrast, our dual-stream strategy optimally balances contour adherence and shape regularity.

D. Feature Visualization Analysis

Fig. 8 visualizes feature evolution. The baseline (bottom), lacking our modules, is sensitive to texture noise and blurred boundaries. In contrast, our model with SGLA and GBCM (top) suppresses noise while preserving principal contours, resulting in smooth interiors and sharp edges. The overlay further confirms that these enhanced features align precisely with the generated superpixel boundaries.

V. Conclusion

This paper proposes a synergistic boundary-aware and semantic-guided dual-enhancement network, effectively resolving the inherent trade-off between geometric accuracy and semantic consistency in superpixel segmentation. We innovatively incorporate KAN to overcome current bottlenecks in superpixel segmentation by establishing robust non-linear deep semantic representations. Building upon this, the designed GBCM imposes shallow geometric constraints via edge modeling, while the SGLA module leverages KAN features to achieve top-down semantic guidance. Extensive experiments validate the superiority of our method across four benchmark datasets. Addressing the associated increase in computational overhead, future work will focus on lightweight designs and efficient attention mechanisms to further enhance real-time processing capabilities while maintaining high performance.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 62366029 and 62466032.

References

[1] X. Jia, Y. Li, J. Jiao, Y. Zhao, and Z. Xia, "SSMamba: Superpixel segmentation with Mamba," *IEEE Signal Process. Lett.*, vol. 32, pp. 1715–1719, 2025.

[2] S. Yi, H. Ma, X. Wang, T. Hu, X. Li, and Y. Wang, "Weakly-supervised semantic segmentation with superpixel guided local and global consistency," *Pattern Recognit.*, vol. 124, pp. 108504, Apr. 2022.

[3] X. Jia, Y. Zhao, B. Zhang, X. Zhang, and G. Yan, "Fuzzy C-means clustering with region constraints for superpixel generation," *Int. J. Fuzzy Syst.*, Apr. 2025, doi: 10.1007/s40815-025-02017-w.

[4] J. Walther, R. Giraud, and M. Clément, "Superpixel anything: A general object-based framework for accurate yet regular superpixel segmentation," *arXiv preprint arXiv:2509.12791*, 2025.

[5] J. Chen, H. Zhang, M. Gong, and Z. Gao, "Collaborative compensative transformer network for salient object detection," *Pattern Recognit.*, vol. 154, pp. 110600, Oct. 2024.

[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[7] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2097–2104.

[8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[9] M. Van den Bergh, X. Boix, G. Roig, B. De Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 13–26.

[10] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1356–1363.

[11] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4651–4660.

[12] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel sampling networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 352–368.

[13] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13964–13973.

[14] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold networks," *arXiv preprint arXiv:2404.19756*, Apr. 2024.

[15] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, "VOLO: Vision outlooker for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6575–6586, May 2023.

[16] M. Xu, Z. Sun, Y. Hu, H. Tang, Y. Hu, X. Song, and L. Nie, "Superpixel segmentation with edge guided local-global attention network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 12, pp. 11922–11934, Dec. 2025.

[17] F. M. de Assunção, R. H. L. e Silva, A. M. C. Machado, P. S. S. Rodrigues, and Z. K. G. Patrocínio Jr., "A graph-based superpixel segmentation method for measuring pressure ulcers," *IEEE Latin America Trans.*, vol. 21, no. 7, pp. 797–805, Jul. 2023.

[18] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[19] X. Kang, L. Zhu, and A. Ming, "Dynamic random walk for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 29, pp. 3871–3884, 2020.

[20] Y.-J. Liu, C.-C. Yu, M.-J. Yu, and Y. He, "Manifold SLIC: A fast method to compute content-sensitive superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 651–659.

[21] J. Zhao, B. Ren, Q. Hou, M.-M. Cheng, and P. Rosin, "FLIC: Fast linear iterative clustering with active search," *Comput. Vis. Media*, vol. 4, no. 4, pp. 333–348, Dec. 2018.

[22] H. Li, Y. Jia, R. Cong, W. Wu, S. T. W. Kwong, and C. Chen, "Superpixel segmentation based on spatially constrained subspace clustering," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7501–7512, Nov. 2021.

[23] L. Sun, D. Ma, X. Pan, and Y. Zhou, "Weak-boundary sensitive superpixel segmentation based on local adaptive distance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2302–2316, May 2023.

[24] P. Zhou, X. Kang, and A. Ming, "Vine spread for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 32, pp. 878–891, 2023.

[25] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and J. Kautz, "Learning superpixels with segmentation-aware affinity loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 568–576.

[26] Y. Wang, Y. Wei, X. Qian, L. Zhu, and Y. Yang, "AINet: Association implantation for superpixel segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7078–7087.

[27] S. Xu, S. Wei, T. Ruan, and Y. Zhao, "ESNet: An efficient framework for superpixel segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 7, pp. 5389–5399, Jul. 2024.

[28] S. Xu, S. Wei, T. Ruan, and L. Liao, "Learning invariant inter-pixel correlations for superpixel generation," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2024, pp. 6351–6359.

[29] T. Zhao, B. Peng, Y. Sun, D. Yang, Z. Zhang, and X. Wu, "Rethinking superpixel segmentation from biologically inspired mechanisms," *Appl. Soft Comput.*, vol. 156, pp. 111467, May 2024.

[30] C. Li, X. Liu, W. Li, C. Wang, H. Liu, Y. Liu, Z. Chen, and Y. Yuan, "U-KAN makes strong backbone for medical image segmentation and generation," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2025, pp. 4652–4660.

[31] T. Liu, J. Xu, T. Lei, Y. Wang, X. Du, W. Zhang, Z. Lv, and M. Gong, "AEKAN: Exploring superpixel-based AutoEncoder Kolmogorov-Arnold network for unsupervised multimodal change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 5601114, 2025.

[32] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," *Comput. Vis. Image Understand.*, vol. 166, pp. 1–27, Jan. 2018.

[33] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGB-D images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 746–760.

[35] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: efficient data generation for urban driving scenes," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 961–972, Sep. 2018.

[36] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.