Differentiable Laplacian Matrix Guided Superpixel Segmentation

Jeremy Juybari, Josh Hamilton, Shuvra Das, Chaofan Chen, Andre Khalil, Yifeng Zhu
University of Maine
{jeremy.juybari,josh.hamilton,shuvra.das,chaofan.chen,andre.khalil,yifeng.zhu}@maine.edu

Abstract

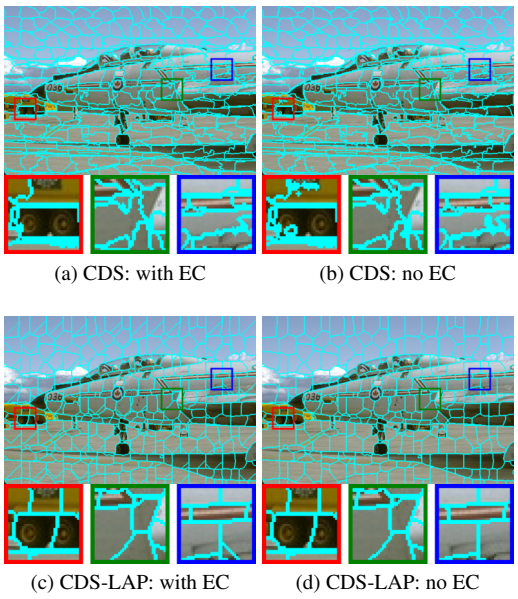
Superpixels partition an image into perceptually coherent regions, reducing the cost of downstream vision tasks. Modern deep learning methods excel at superpixel generation but often yield irregular boundaries and isolated pixels, necessitating non-differentiable post-processing to enforce connectivity. This undermines the end-to-end learning capabilities. We propose a simple, fully differentiable graph-Laplacian loss that encourages spatial regularity and connectivity during training. The loss is model-agnostic and can be seamlessly integrated into the training of existing architectures to improve the quality of superpixels. In addition, we introduce two novel metrics, the average stray pixel count and excess component count, to measure the quality of superpixels. We demonstrate both qualitative and quantitative improvements over state-of-the-art methods with and without enforced connectivity. Our approach represents a significant step toward eliminating non-differentiable post-processing. Project code: https://github.com/jeremyJJB/Differentiable-Laplacian-Matrix-Guided-Superpixel-Segmentation.

1. Introduction

Superpixels are clusters of pixels that exhibit visual similarity and continuity, functioning as an effective means of data compression in computer vision. This technique significantly reduces computational complexity for downstream tasks such as object segmentation and detection [7, 11, 15, 20, 22, 24, 30, 49, 50, 57, 59, 60], optical flow [27], stereo matching [51], tracking and medical image applications [3, 25, 29, 31, 34, 43, 61, 62]. By aggregating pixels into coherent regions characterized by similar features while adhering to image boundaries, superpixels facilitate a transition from pixel-level processing to a less computationally intensive region-level analysis, often achieving reductions in complexity by several orders of magnitude [49]. Traditional methods for superpixel generation, such as Simple Linear Iterative Clustering (SLIC), fulfill the criteria for high-quality superpixels, which include boundary adherence, compactness, and, most critically, spatial connectivity [2]. In such algorithms, each superpixel label corresponds to a single connected component, and isolated pixels are naturally avoided by design.

Deep learning has dramatically improved superpixel quality (boundary adherence and segmentation accuracy) with models such as SCN [51], AINet [42], CDS [47], and SSM [17]. However, a major limitation is that these models often produce irregular boundaries, disconnected superpixels (same label in multiple fragments), and isolated pixels as seen in the first row of Fig. 1. In practice, a non-differentiable post-processing step borrowed from SLIC, **Enforced Connectivity (EC)**, is applied after inference to reassign stray pixels so that each superpixel forms a single connected component. This compromises differentiability, breaks end-to-end learning, preventing joint optimization with downstream tasks [2, 16, 17, 42, 47, 51]. In fact, one model [54] is an exception: it enforces connectivity through alternating multi-layered horizontal and vertical interpolation steps within a dedicated architecture (*i.e.*, it is not a model-agnostic, drop-in loss), yet lags in overall accuracy.

More specifically, enforced connectivity (EC) introduces several fundamental challenges. In traditional methods, connectivity emerges naturally from the design because isolated or irregular superpixels are assigned distinct identifiers. In contrast, deep learning models frequently produce fragmented superpixels, *i.e.*, disjoint regions that share the same identifier. These disjointed fragments, or *excess components*, violate spatial continuity. EC addresses this by reassigning these excess components to new or neighboring superpixels, forcibly restoring connectivity. However, this correction is a non-differentiable operation that breaks the end-to-end learning pipeline [54]. As a result, models cannot be trained jointly with downstream tasks, limiting their integration into fully differential architectures. Consequently, most deep learning applications that rely on superpixels continue to depend on traditional algorithms for superpixel generation [3, 7, 22, 25, 27, 29, 31, 34, 43, 49, 50, 57, 61, 62]. Essentially, EC serves as a workaround that masks a great limitation: deep learning models have yet to *learn* to produce spatially connected superpixels.



(a) CDS: with EC (b) CDS: no EC

(c) CDS-LAP: with EC (d) CDS-LAP: no EC

Figure 1. Comparison of superpixels from a deep learning model (CDS) under Enforced Connectivity (EC) post-processing and the same model with graph-Laplacian (LAP) regularization. Our proposed LAP yields more compact superpixels, more precise boundaries, and fewer excess components without requiring EC.

To address this limitation, we propose a graph-Laplacian (LAP) loss that encourages spatial regularity and connectivity of superpixels during training. We represent each superpixel as a graph of its individual pixel assignment probabilities, wherein neighboring nodes (pixels) are strongly connected when there is a high likelihood that they belong to the same superpixel. Each node in the graph can have up to eight edges, *i.e.*, a maximum degree of 8. In graph theory, the multiplicity of the null eigenvalue of the Laplacian matrix equals the number of connected components. By maximizing the Laplacian's trace (sum of degrees), we implicitly reduce disconnected components as seen in the second row of Fig. 1. Crucially, this loss is fully differentiable and can be added to any superpixel model.

To evaluate the quality of connectivity, we define two new metrics: *average stray pixel count* and *excess component count*. The excess component count refers to the total number of isolated regions of a superpixel excluding the largest region, while the stray pixel count represents the sum of pixels in isolated components. A perfect superpixel has 0 stray pixels and 0 excess components (a property of traditional methods after EC). We show that our Laplacian loss drives these counts toward zero.

In summary, our contributions include:

- **Graph-Laplacian loss:** A novel loss term that significantly reduces excess components and stray pixels, enhancing connectivity with a minimal accuracy trade-off.
- **Novel weighted reconstruction and minimum semantic distance loss terms:** These losses when combined with the Laplacian loss achieve 15% better compactness for equivalent ASA while improving boundary recall, at a modest cost in boundary precision.
- **New evaluation metrics:** We introduce stray-pixel and excess-component counts as quantitative measures of superpixel fragmentation before EC.
- **Toward end-to-end superpixels:** By embedding connectivity into the loss, we begin to eliminate the need for non-differentiable post-processing, paving the way for seamless integration with downstream vision networks.

2. Related Work

Traditional approaches. Classic superpixel algorithms optimize color–spatial homogeneity subject to boundary regularity, and are commonly grouped into clustering-, graph- and energy-based categories. SLIC dominates practice by adapting k-means clustering to a 5D color-spatial space and restricting updates to local regions surrounding cluster centers, yielding regular, connected regions [2]. Other clustering approaches include LSC, which achieves spectral clustering benefits without expensive eigenvalue decomposition [23]. Graph-based methods treat images as weighted graphs [9, 26, 28, 32, 37], while energy-based methods perform hill-climbing optimization over color histogram similarity and boundary smoothness [39].

Deep superpixels. Deep learning transformed superpixel segmentation around 2018 by making algorithms differentiable and trainable end-to-end. SSN replaces SLIC's hard assignments with soft pixel-superpixel associations, enabling task-specific optimization [16]. SEAL introduces segmentation-aware affinity loss that emphasizes boundary pixels by incorporating over-segmentation errors [38]. SCN removes iterative clustering entirely via an encoder-decoder architecture that directly predicts pixel-to-superpixel probabilities [51]. AINet explicitly models pixel-superpixel relations with boundary-aware loss [42]. CDS separates unchanging inter-pixel correlations from statistical properties of the underlying dataset to improve assignment robustness [47]. SSM leverages the vision Mamba transformer with global-local fusion for broader semantic awareness [17]. SIN enforces connectivity through alternating horizontal and vertical interpolation within a dedicated architecture, eliminating post-processing but sacrificing accuracy and generalizability as a drop-in component [54]. A common assumption throughout the literature is the existence of a hard trade-off between ASA and compactness [1, 2, 13, 14, 21, 35, 41, 46, 47, 55, 58], where strong boundary adherence has come at the cost of fragmented label maps, motivating losses and architectures that promote connectivity without sacrificing end-to-end differentiability.

Laplacian regularization. Laplacian regularization has

been widely adopted in deep learning to enforce smoothness, manifold structure, or cluster separation in learned representations. Early graph-based semi-supervised methods penalize quadratic forms to ensure nearby samples in an affinity graph have similar embeddings, preserving local geometry [5]. This principle has been incorporated into deep architectures through graph embedding losses, Laplacian eigenmap objectives, and manifold regularization layers that constrain intermediate features to vary smoothly along data graphs [19, 44, 53]. Related graph convolutional approaches interpret repeated Laplacian smoothing as low- and high-pass filtering, denoising graph signals and encouraging feature homogeneity within neighborhoods [10, 19, 45]. Yang et al. [52] proposed a graph-regularized deep network for unsupervised image representation learning in which a Laplacian term preserves neighborhood relationships, echoed by subsequent work leveraging graph Laplacian regularization for smoothness and structural consistency [6, 12, 56]. More broadly, trace-based Laplacian terms are generally used to regularize features rather than enforce explicit topological properties of discrete spatial assignments.

3. Method

We train a generic superpixel assignment network $f_\theta = h \circ g$, where g is the pixel embedding network, h is a convolutional layer (with softmax normalization) to produce superpixel assignment probabilities and θ denotes all trainable parameters. To promote spatial connectivity, we optimize f_θ with three proposed loss terms: a graph-Laplacian loss, a minimum semantic distance loss, and a weighted reconstruction loss. For a given image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ with height H and width W, the network outputs a superpixel assignment matrix $Q = f_\theta(\mathbf{x}) \in [0,1]^{N\times M}$, where $N = H \times W$ and M is the number of superpixels. Q is a sparse matrix where each element $q_{i,s}$ specifies the probability that the i-th pixel is assigned to the s-th superpixel.

Following prior work [16, 17, 42, 47, 51], the image is partitioned into a regular grid of square blocks induced by a 16 x 16 stride, and each superpixel s is centered on one such block, $\mathcal{B}$. We restrict the assignment of each pixel i to only the 9 superpixel candidates in its local neighborhood (e.g., 3×3 blocks), denoted $\mathcal{S}_i$, for computational efficiency. Conversely, for each superpixel $s \in \{1, \ldots, M\}$, we define $\mathcal{P}_s$ as the set of pixels that may be assigned to it; this set spans the 3×3 blocks around s. We denote by $\mathbf{y}_i$ the ground-truth semantic property (e.g., class label vector) at pixel i. Note that Q is row-normalized with $\sum_{s\in\mathcal{S}_i} q_{i,s} = 1$.

3.1. Graph-Laplacian Loss

To encourage pixels within each superpixel to be connected, we introduce a graph-Laplacian loss, which is applied to a set of graphs constructed based on Q. In particular, for each superpixel $s \in \{1, ..., M\}$, we construct an undirected graph $\mathcal{G}_s$ whose vertices are the pixels in $\mathcal{P}_s$ and whose edges connect each pixel i to its 8 neighboring pixels. For each pair of neighboring pixels i and j, we define the edge weight to be $q_{i,s}q_{j,s}$.

It follows that the degree $d_{i,s}$ of pixel i in the graph $\mathcal{G}_s$, which is the sum of all edge weights involving pixel i, is given by:

$$d_{i,s} = \sum_{j\in\mathcal{N}_i} q_{i,s}q_{j,s}, \tag{1}$$

where $\mathcal{N}_i$ is the 8 immediate neighbors of pixel i.

For each superpixel s, let $N_s = |\mathcal{P}_s|$ be the number of pixels in s. The graph-Laplacian of $\mathcal{G}_s$ associated with the superpixel s is defined as $L_s = D_s - A_s$, where $D_s \in \mathbb{R}^{N_s\times N_s} = \text{diag}(d_{1,s}, ..., d_{N_s,s})$ is the diagonal degree matrix and $A_s \in \mathbb{R}^{N_s\times N_s}$ is the adjacency matrix, where each (i,j)-th entry is the edge connection between pixel i and pixel j and has a weight of $q_{i,s}q_{j,s}$. Since the graph-Laplacian matrix of an undirected graph is symmetric and positive semi-definite, all eigenvalues of L_s are nonnegative. The eigenvalues of L_s capture structural properties of the graph $\mathcal{G}_s$ corresponding to superpixel s. In particular, the multiplicity of the zero eigenvalue equals the number of connected components in $\mathcal{G}_s$ [40]. Hence, reducing the multiplicity of the zero eigenvalue encourages spatial connectivity among pixels within the same superpixel. In practice, maximizing the sum of eigenvalues of L_s, which equals the trace of L_s, $\text{tr}(L_s) = \sum_{i\in\mathcal{P}_s} d_{i,s}$, serves as a convenient proxy for minimizing this multiplicity (see supplement).

Therefore, we define the graph-Laplacian loss $\mathcal{L}_{\text{LAP}}$ across all superpixels in a training image as

$$\mathcal{L}_{\text{LAP}}(\theta;\mathbf{x}) = 1 - \frac{1}{M}\sum_{s=1}^{M}\frac{\text{tr}(L_s)}{8N_s} \tag{2}$$

where $8N_s$ is the maximum trace for a graph with the same structure as $\mathcal{G}_s$ with all edge weights 1.

3.2. Minimum Semantic Distance Loss

A key challenge in superpixel generation is to ensure that superpixels respect object boundaries and do not span multiple semantic classes. To address this, we introduce a minimum semantic distance loss that explicitly maximizes inter-class distances in the learned latent space. By encouraging greater separation between pixels of different semantic categories, the model learns representations where class boundaries are more pronounced, facilitating the generation of superpixels that naturally align with semantic structure without requiring explicit boundary detection.

To encourage such inter-class separation, we seek to maximize the Euclidean distance between pixel embeddings across all pixel pairs of different semantic classes. For

pixel i, let $\mathbf{z}_i = g(\mathbf{x}_i)$ be the i-th pixel embedding. Ideally, we would like to maximize the minimum embedding difference $\min \|\mathbf{z}_i - \mathbf{z}_j\|_2$ over all pixel pairs $i, j \in \mathbf{x}$ with $\mathbf{y}_i \neq \mathbf{y}_j$. However, the computational cost of evaluating all inter-class pixel embedding distances (in order to find the minimum) is proportional to both the number of pixels per class and the number of class pairs, which is prohibitively expensive for a typical multi-class image. To reduce computational cost, we sample 100 pixels per semantic class – with replacement if a class contains less than 100 pixels, and maximize $\rho = \min_{\substack{i,j \in \mathcal{V} \\ \mathbf{y}_i \neq \mathbf{y}_j}} \|\mathbf{z}_i - \mathbf{z}_j\|_2$, where $\mathcal{V}$ denotes the set of randomly sampled pixels across all semantic classes.

Mathematically, we define our minimum semantic distance (MSD) loss as

$$\mathcal{L}_{\text{MSD}}(\theta; \mathbf{x}, \mathbf{y}) = \left(\max(0, m - \rho)\right)^2. \tag{3}$$

In Eq. (3), we incorporate a margin m so that no MSD loss is incurred once the minimum inter-class pixel embedding distance ρ between the sampled pixels exceeds this threshold. This margin helps the MSD loss focus on difficult training examples where pixel embeddings of different semantic classes are more alike, thereby improving the accuracy of the network. In our experiments, we used $m = 1.5$.

3.3. Weighted Reconstruction Loss

Following prior superpixel methods, we adopt the standard reconstruction loss, which encourages the pixel-superpixel assignment matrix Q to preserve per-pixel semantic properties by penalizing discrepancies between ground-truth and reconstructed per-pixel properties. Mathematically, as in [16, 17, 42, 47, 51], the reconstruction loss with cross-entropy $E(\cdot, \cdot)$ is defined as

$$\mathcal{L}_R(\theta; \mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} E\big(\mathbf{y}_i, \mathbf{y}'_i\big), \tag{4}$$

where $\mathbf{y}_i$ denotes the ground-truth semantic property (e.g., class label vector) of pixel i and $\mathbf{y}'_i$ denotes the corresponding property reconstructed from the superpixel representation via Q (detailed construction in supplement).

The standard reconstruction loss, defined by Eq. (4) and used in prior literature, implicitly treats all pixels as equally informative. However, in practice, most image blocks in a regular grid partition of an image are *homogeneous*, i.e., contain pixels from a single semantic class. Only a minority of blocks are *mixed*, i.e., contain multiple classes, and thus intersect object boundaries where accurate superpixel assignments are critical. To bias the learning process toward these challenging regions, we introduce a weighted reconstruction loss that prioritizes pixels within mixed blocks while de-emphasizing those in single-class blocks.

Formally, for each block $\mathcal{B}$ in a regular grid partition of an input image, we assign a per-pixel weight w_i to all pixels $i \in \mathcal{B}$:

$$w_i = \begin{cases} 0.1 & \text{if } \mathcal{B} \text{ contains exactly one class,} \\ 1.0 & \text{if } \mathcal{B} \text{ contains } \geq 2 \text{ classes.} \end{cases} \tag{5}$$

where pixels in mixed blocks are assigned ten times the weight of pixels in single-class blocks. In the BSDS500 dataset we found approximately 70% of blocks are of a single class (see supplement). The 10:1 weighting compensates for this imbalance so that the mixed blocks will exert an influence comparable to the single-class blocks on the weighted reconstruction loss.

However, directly using w_i from Eq. (5) to weight a pixel in the reconstruction loss would make the average per-pixel weighting of an image dependent on its layout – pixels in an image with more mixed blocks would be weighted more heavily on average than those in an image with fewer. This would inadvertently vary the effective learning rate across the training images by entangling the localized emphasis on mixed blocks with the average per-pixel weight of each image. To ensure more equalized contributions to the loss gradient from pixels across different training images regardless of their block layout, we normalize the per-pixel weights w_i so that their sum is equal to the number of pixels N:

$$W_i = \frac{N \cdot w_i}{\sum_{j=1}^{N} w_j}. \tag{6}$$

By construction, we have $\sum_{i=1}^{N} W_i = N$, which means that the *average* per-pixel weight remains 1 regardless of the number of mixed and single-class blocks in an image. Intuitively, the normalized weights redistribute the contributions from pixels in easier single-class blocks to pixels in harder mixed blocks without changing the overall scale of the reconstruction loss across training images. The resulting novel weighted reconstruction loss is:

$$\mathcal{L}_{\text{WR}}(\theta; \mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} W_i E\big(\mathbf{y}_i, \mathbf{y}'_i\big). \tag{7}$$

Crucially, our weighted reconstruction loss maintains a total loss magnitude consistent with the unweighted loss, but redistributes the gradient signal from single-class regions to mixed blocks that straddle true semantic boundaries.

3.4. Final Network Loss Design

For each baseline model (SCN, AINet, CDS, and SSM), we incorporate the Laplacian loss and minimum semantic distance loss into their respective loss functions, and replace the standard reconstruction loss with the weighted one. Our final loss formulation is:

$$\begin{aligned} \mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) = \mathcal{L}_{\text{base}}(\theta; \mathbf{x}, \mathbf{y}) + \lambda_{\text{LAP}} \mathcal{L}_{\text{LAP}}(\theta; \mathbf{x}) \\ + \lambda_{\text{MSD}} \mathcal{L}_{\text{MSD}}(\theta; \mathbf{x}; \mathbf{y}) + \lambda_{\text{WR}} \mathcal{L}_{\text{WR}}(\theta; \mathbf{x}, \mathbf{y}) \end{aligned} \tag{8}$$

where $\mathcal{L}_{\text{base}}$ denotes the loss function of the corresponding baseline model without the reconstruction loss Eq. (4), and λ_{LAP}, λ_{MSD} and λ_{WR} are hyperparameters. We used $\lambda_{\text{MSD}} = 10^{-3}$, $\lambda_{\text{LAP}} = 360$ and $\lambda_{\text{WR}} = 1$. For details of $\mathcal{L}_{\text{base}}(\theta; \mathbf{x}, \mathbf{y})$ for each baseline model see [17, 42, 47, 51].

4. Experiments

4.1. Datasets

We apply our three novel losses to train four deep superpixel models (SCN, AINet, CDS, and SSM) and evaluate them on three common datasets (BSDS500, NYUv, PascalVOC 2012). BSDS500 contains 200 training, 100 validation, and 200 test images [4]. Class labels in BSDS500 refer to objects rather than semantic classes. Each image has up to five human-drawn object contours; the resulting annotations vary in scale and granularity (e.g., one annotator labels a cat as a single region, while another separates eyes/head/paws/body). Following prior work [16, 18, 42, 48, 51], we treat each annotation as a valid ground truth and restrict to 50 classes. All models were trained only on BSDS500. For inference-only evaluation, we additionally use **NYUv2** indoor scenes [33] and the **Pascal VOC 2012** validation set [8] (results in supplement).

4.2. Evaluation Metrics

In addition to reporting the standard superpixel metrics used in [16, 36, 42, 48, 51], namely Achievable Segmentation Accuracy (ASA), Compactness (CO), and Boundary Recall/Precision (BR/BP), we introduce two novel fragmentation metrics and motivate area-under-curve (AUC) summaries. Currently, there are no metrics that directly quantify how often a single predicted superpixel is broken into multiple disconnected pieces before enforced connectivity (EC) is applied. Our goal in this section is to formalize this notion of pre-EC superpixel fragmentation.

For a given image, let $S = \{s_1, \ldots, s_M\}$ denote the set of hard superpixel assignments partitioning the image, such that every pixel belongs to exactly one $s \in S$. We decompose each superpixel assignment s into connected sub-components. In an ideal superpixel partition, there is only a single connected sub-component per each s, as is the case post EC. However, before EC where individual superpixel connectivity is not enforced, each $s \in S$ may have multiple sub-components. For each superpixel s, let T_s denote the total number of sub-components c_t^s where t is the sub-component index. The complete set of sub-components for superpixel s is denoted by $\{c_t^s\}_{t=1}^{T_s}$, where we have:

$$\bigcup_{t=1}^{T_s} c_t^s = s \quad \text{and} \quad c_t^s \cap c_{t'}^s = \emptyset \text{ for all } t \neq t'. \tag{9}$$

Let the largest connected sub-component of s be $c_{\max}^s = \arg\max_t |c_t^s|$.

In a high quality superpixel segmentation, each superpixel should have minimal number of connected sub-components (ideally 1) and minimal number of pixels outside the largest connected sub-component. To quantify such quality, we introduce two metrics: the **number of excess components** and the **number of stray pixels**. For a given superpixel s, the number of excess components counts the number of sub-components beyond the ideal case and is given by $T_s - 1$, and for a given image with the superpixel segmentation S, the total number of excess components is:

$$XC(S) = \sum_{s \in S} (T_s - 1). \tag{10}$$

However, XC alone is an insufficient quality measure because it ignores the size of individual sub-components $|c_t^s|$ – a superpixel with 2 sub-components has only 1 excess component, but its second-largest component could be nearly as large as the largest component $c_{\max}^s$. To achieve a more granular assessment of superpixel fragmentation, for each superpixel s, we also count the number of stray pixels, defined by the number of pixels outside the largest component $c_{\max}^s$, and for a given image with the (hard) superpixel segmentation S, the total number of stray pixels is:

$$\text{Stray}(S) = \sum_{s \in S} \left| s \setminus c_{\max}^s \right|. \tag{11}$$

Intuitively, while $XC(S)$ counts the number of unwanted additional fragments, $\text{Stray}(S)$ accounts for their sizes. In the ideal case, both measures should be 0. We use XC_{avg} and $\text{Stray}_{\text{avg}}$ to denote the dataset-wide averages. Note that since EC merges all disconnected fragments, thereby making both metrics 0 by construction, these metrics are designed to evaluate fragmentation *before* EC.

AUC summaries. ASA, CO, and BR/BP are typically plotted as functions of the superpixel count n for inference [16, 18, 42, 48, 51]. However, different papers often use different ranges of superpixel counts, which makes comparisons rely solely on visual inspections. To obtain scalar summary scores that are comparable across methods, we fix a common superpixel-count range $[n_{\min}, n_{\max}]$ and compute a normalized area-under-curve (AUC) for each metric curve over this range. For example, the normalized AUC for ASA is:

$$\text{ASA}_{\text{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \text{ASA}(n)\, dn. \tag{12}$$

Normalized AUCs for other metrics are defined similarly. Note $\text{B}_{\text{AUC}} = \text{BR}_{\text{AUC}} \times \text{BP}_{\text{AUC}}$.

4.3. Implementation Details

We follow each model's original training recipe [16, 18, 42, 48, 51], but extend training to ensure convergence

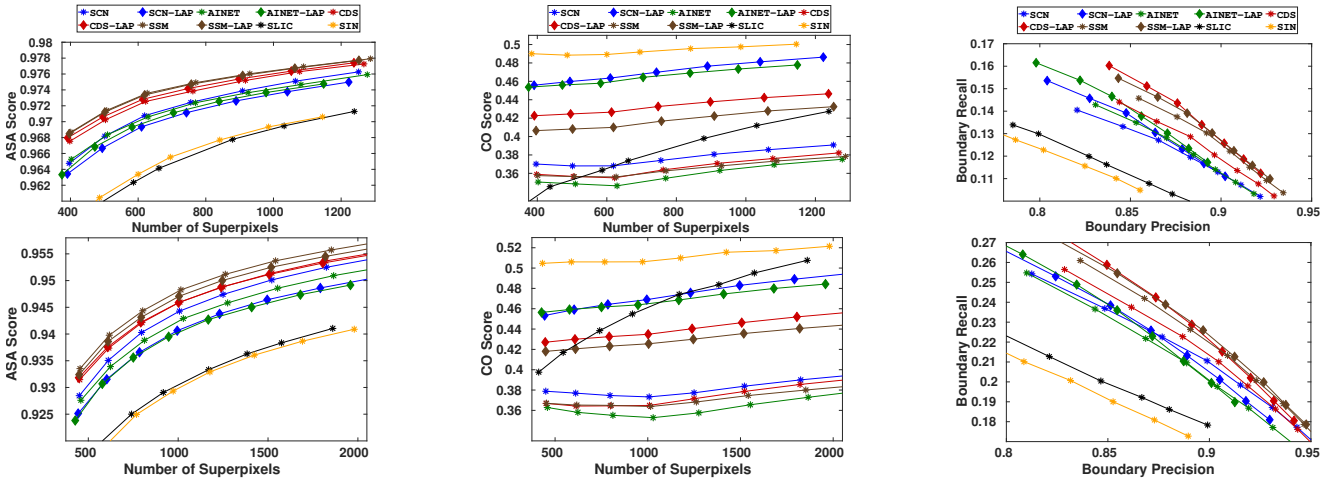


Figure 2. Standard metrics *with enforced connectivity (EC)*. Top row: BSDS500 test set; bottom row: NYUv2. Columns (left→right): **ASA**, **CO**, and **BR–BP**. Baselines are plotted with star markers—SCN (blue), AINet (green), CDS (red), SSM (brown), SIN(orange) and SLIC (black). Laplacian variants use the same color with a diamond (⋄) marker (not applicable to SLIC and SIN). Laplacian variant models out perform their counterparts on CO and BR with minimal impact on ASA and BP.

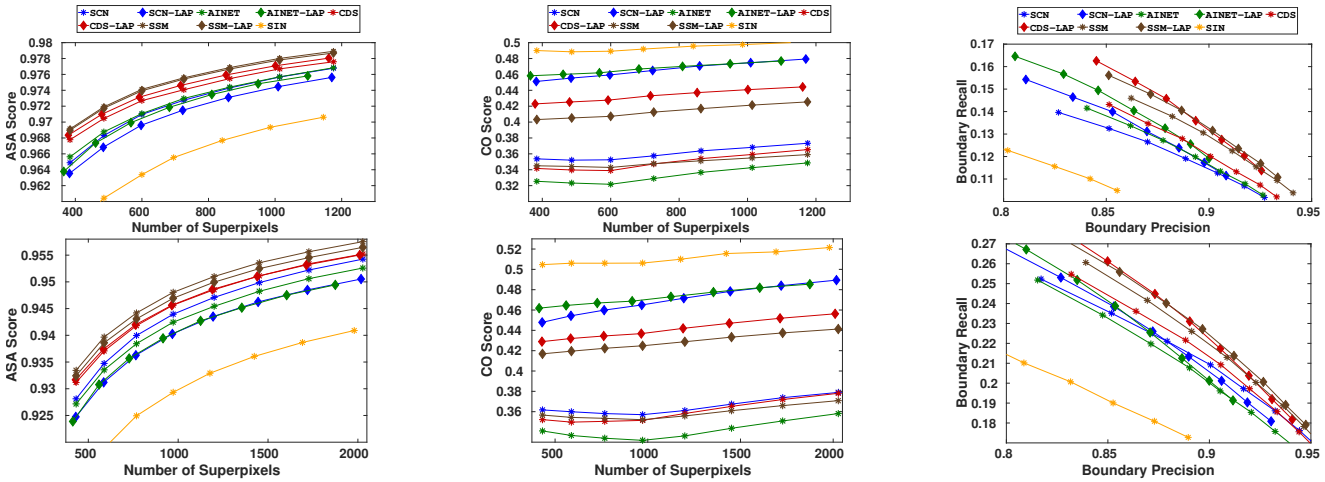


Figure 3. Standard metrics *without enforced connectivity (EC)*. Top row: BSDS500 test set; bottom row: NYUv2. Columns (left→right): **ASA**, **CO**, and **BR–BP**. Baselines are plotted with star markers—SCN (blue), AINet (green), CDS (red), SSM (brown), SIN (orange); Laplacian variants use the same color with a diamond (⋄) marker (not applicable to SIN). Laplacian variant models without EC out perform their counterparts on CO and BR with minimal impact on ASA and BP.

of the graph-Laplacian loss. For AINet and SSM, the Laplacian loss is applied only in the second stage of their standard multi-stage training schedules. Unless otherwise noted, the Laplacian weight is set to $\mathcal{L}_{\text{LAP}}$=360, selected from a sweep that balances connectivity against ASA/CO. All models are evaluated following the common protocol [16, 18, 42, 48, 51], with EC applied to only "with EC" variants. All reported numbers are averages over three runs.

4.4. Results

With and without enforced connectivity. Aggregated results on BSDS500 and NYUv2 are shown in Figs. 2 and 3. As expected, EC slightly lowers ASA but guarantees contiguity. With our Laplacian term, state-of-the-art models generally *improve compactness (CO)* while maintaining competitive ASA. SCN and AINet with Laplacian exhibit small ASA drops but outperform their non-Laplacian counterparts on CO. Boundary recall (BR) typically increases with a minor decrease in boundary precision (BP). Without EC, models *without* the Laplacian suffer notable CO degradation due to fragmented labels, whereas models *with* the Laplacian retain CO comparable to their EC results. An exception is SIN which has the best CO across all datasets, but suffers greatly on ASA, boundary recall, and boundary precision. SIN was

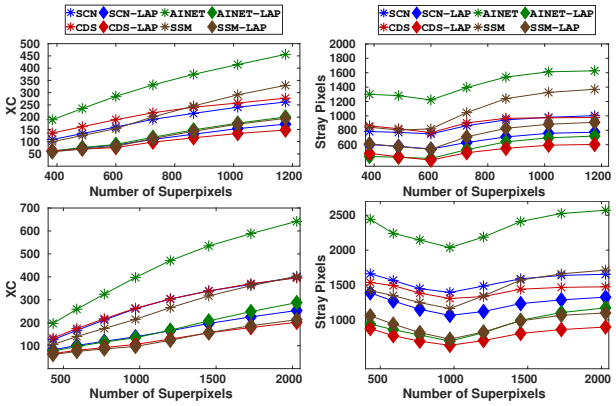


Figure 4. Fragmentation metrics *without enforced connectivity (EC)*. Top row: BSDS500; bottom row: NYUv2. Columns (left→right): average excess components counts ($\mathrm{XC}_{\mathrm{avg}}$) and average stray pixels counts ($\mathrm{Stray}_{\mathrm{avg}}$). Baselines are plotted with star markers—SCN (blue), AINet (green), CDS (red), SSM (brown). Laplacian variants use the same color with a diamond ($\diamond$) marker Lower metric values are less fragmented superpixels and all Laplacian variants are lower.

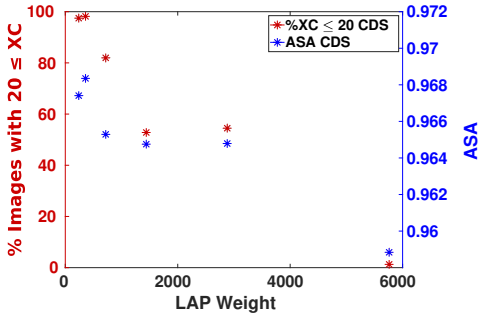


Figure 5. A CDS-LAP model was trained on the BSDS500 dataset for each λ_{LAP} which is shown on the x-axis using an inference superpixel count of 384. The left y-axis shows the percent of images that have $XC \leq 15$ and the right y-axis is the ASA. There is a clear relationship between the weight on the graph-Laplacian loss, XC, and ASA.

Table 1. Laplacian Weight Study. Increasing λ_{LAP} improves all measurements of superpixel quality at a minor expense to ASA.

λ_{LAP}	$\mathrm{ASA}_{\mathrm{AUC}}(\uparrow)$	$\mathrm{CO}_{\mathrm{AUC}}(\uparrow)$	$\mathrm{B}_{\mathrm{AUC}}(\uparrow)$	$XC_{\mathrm{AUC}}(\downarrow)$
240	0.974	0.433	0.116	99.4
360	**0.975**	0.434	0.122	99.7
720	0.972	0.472	0.119	94.9
1440	0.971	0.508	0.128	81.7
1800	0.971	0.510	0.131	82.5
2880	0.971	0.521	**0.136**	71.6
5760	0.966	**0.571**	0.128	**10.1**

included on both plots for comparison, but inherently does not utilize EC. SLIC is omitted from the no-EC plots because it relies on EC to control the number of superpixels and otherwise produces far more segments.

Table 2. Quantitative performance on BSDS500. All results use 384–1200 superpixels. The best value in each EC setting is bolded. Models trained with the graph-Laplacian (LAP) loss produce higher-quality superpixels with minimal impact on ASA.

EC	Model	$\mathrm{ASA}_{\mathrm{AUC}}\uparrow$	$\mathrm{CO}_{\mathrm{AUC}}\uparrow$	$\mathrm{B}_{\mathrm{AUC}}\uparrow$	$XC_{\mathrm{AUC}}\downarrow$	$\mathrm{ST}_{\mathrm{AUC}}\downarrow$
Yes	SCN	0.9724	**0.3775**	0.1052	0.0000	0.0000
	AINET	0.9722	0.3592	0.1068	0.0000	0.0000
	CDS	0.9739	0.3670	0.1082	0.0000	0.0000
	SSM	**0.9748**	0.3655	**0.1107**	0.0000	0.0000
	SCN-LAP	0.9710	**0.4720**	0.1125	0.0000	0.0000
	AINET-LAP	0.9710	0.4657	0.1176	0.0000	0.0000
	CDS-LAP	0.9741	0.4344	**0.1192**	0.0000	0.0000
	SSM-LAP	**0.9746**	0.4189	0.1165	0.0000	0.0000
No	SCN	0.9726	**0.3606**	0.1055	**194.5483**	**880.4743**
	AINET	0.9728	0.3330	0.1071	339.2057	1439.7918
	CDS	0.9741	0.3501	0.1084	218.6546	901.6156
	SSM	**0.9755**	0.3495	**0.1118**	215.9656	1089.0388
	SCN-LAP	0.9714	0.4663	0.1138	116.7020	661.0723
	AINET-LAP	0.9726	**0.4687**	0.1189	129.4661	562.9952
	CDS-LAP	0.9746	0.4339	**0.1215**	**103.8117**	**507.4504**
	SSM-LAP	**0.9753**	0.4140	0.1185	124.0733	734.4290

Fragmentation metrics. Fig. 4 quantifies continuity of superpixels (no EC). Ideal superpixels should have XC=Stray=0, as generated by traditional algorithms after EC. Across all architectures, adding the Laplacian loss substantially reduces both excess components and stray pixels relative to their non-Laplacian baselines.

Effects of the Laplacian weight. Fig. 5 varies λ_{LAP} for CDS while keeping other losses fixed. Smaller λ_{LAP} increases ASA but leaves many excess components that would require post-hoc relabeling. Increasing λ_{LAP} progressively suppresses fragmentation; e.g., at λ_{LAP}=2880, roughly half of test images have near-zero excess components while ASA at superpixel count n=384 remains 0.964. Additionally, Tab. 1 shows the effect of increasing λ_{LAP} using aggregated metrics for the superpixel count range from 384 to 1200. While increasing λ_{LAP} leads to a noticeable decline in ASA, it yields improvements across all other superpixel quality metrics.

Qualitative & quantitative results. Fig. 6 visualizes outputs for AINet, CDS, and SSM with/without EC and with/without the Laplacian loss. Incorporating the Laplacian loss yields visibly more compact regions and cleaner boundaries. When EC is applied, label changes are minimal if the Laplacian loss is used – this is consistent with the reductions in XC and Stray. Tab. 2 summarizes the results on the BSDS500 dataset using the AUC summary scores for easy quantitative comparison with a superpixel range of 384-1200, the same range as that used for the visual result curves. Tables for NYUv2 and PASCAL VOC 2012 are in the supplement. These quantitative results clearly confirm the visual results that models trained with our novel losses have higher compactness, better boundary statistics, lower excess components, and lower stray pixels while maintaining state-of-the-art ASA.

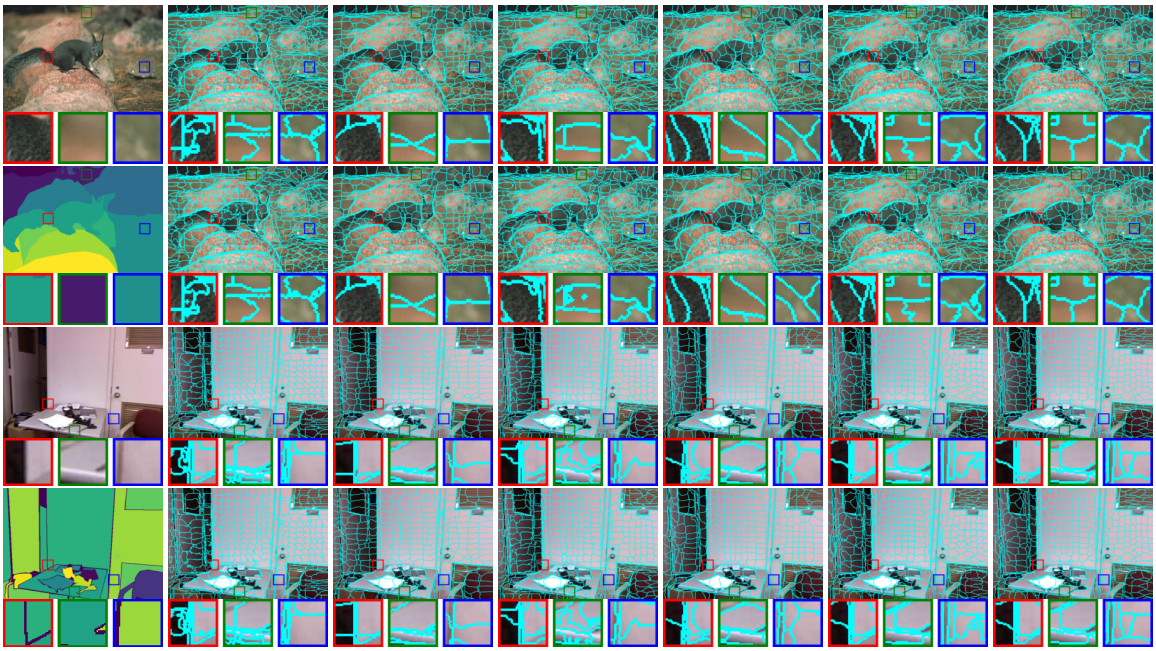


Figure 6. Qualitative comparison on **BSDS500** (top two rows) and **NYUv2** (bottom two rows). For each dataset, the first row shows the *input image* followed by outputs *with enforced connectivity (EC)* for: AINet, AINet-LAP, CDS, CDS-LAP, SSM, SSM-LAP. The second row shows the *ground-truth labels* followed by the corresponding outputs *without EC*. Colored boxes (red/green/blue) highlight regions where EC relabels fragmented components and where boundary irregularities are reduced; training with the proposed graph-Laplacian (LAP) loss yields more compact, connected superpixels and fewer label changes under EC.

Table 3. Ablation study. Adding $\mathcal{L}_{\text{LAP}}$ improves CO_{AUC}, B_{AUC}, and XC_{AUC} with minimal effect on ASA.

Config				AUC metrics			
MSD	LAP	WR	EC	$\text{ASA}^{\uparrow}_{\text{AUC}}$	$\text{CO}^{\uparrow}_{\text{AUC}}$	$\text{B}^{\uparrow}_{\text{AUC}}$	$XC^{\downarrow}_{\text{AUC}}$
				0.974	0.350	0.108	218.7
✓				0.973	0.368	0.107	204.0
		✓		**0.975**	0.319	0.110	316.6
✓		✓		**0.975**	0.312	0.110	329.1
			✓	0.974	0.367	0.108	0
✓			✓	0.973	0.386	0.107	0
		✓	✓	0.974	0.342	0.109	0
✓		✓	✓	0.974	0.335	0.109	0
	✓			0.973	0.468	0.118	61.6
✓	✓			0.973	0.479	0.116	**49.9**
	✓	✓		0.974	0.443	0.120	94.6
✓	✓	✓		**0.975**	0.434	**0.122**	103.8
	✓		✓	0.973	0.469	0.117	0
✓	✓		✓	0.973	**0.483**	0.116	0
	✓	✓	✓	0.974	0.445	0.118	0
✓	✓	✓	✓	0.974	0.434	0.119	0

4.5. Ablation Study

We train all combinations of the proposed losses on CDS to assess their contributions: **LAP** (graph-Laplacian), **MSD** (minimum semantic distance), and **WR** (weighted reconstruction), with and without EC. Tab. 3 reports ASA_{AUC}, CO_{AUC}, B_{AUC} and XC. The LAP term is the primary driver of connectivity: it dramatically cuts fragmentation with minor impact on ASA and typically improves CO. MSD and WR offer complementary gains (e.g., boundary emphasis) but do not, on their own, eliminate fragmentation.

5. Conclusion

We have introduced a novel graph-Laplacian loss to promote spatial connectivity in deep superpixel networks. This loss is simple, model-agnostic, and fully differentiable. Experiments demonstrate that it drastically reduces disconnected fragments and stray pixels in raw superpixel outputs, while preserving or even slightly improving segmentation accuracy (ASA) and recall. Our minimum semantic distance and weighted reconstruction losses help balance the trade-off between connectivity and fidelity. Our proposed excess component and stray pixel counts quantitatively capture superpixel continuity pre-EC; after training with our losses, these metrics approach zero even without EC.

By significantly reducing the need for enforced connectivity, our work moves superpixel generation closer to being fully end-to-end differentiable. We also advocate reporting area-under-curve metrics (ASA_{AUC}, CO_{AUC}, B_{AUC}) for clearer quantitative comparisons. In future work, we aim to integrate our approach into joint vision pipelines (e.g., segmentation networks) to fully leverage end-to-end differentiability of learned superpixels.

Acknowledgments

This work was supported by NSF grants OIA-2218063, IIS-2442039, and NIH grant R15CA246335. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NSF and NIH.

References

[1] Radhakrishna Achanta and Sabine Susstrunk. Superpixels and polygons using simple non-iterative clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4660, 2017. 2

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 1, 2

[3] Abdulkadir Albayrak and Gokhan Bilgin. A hybrid method of superpixel segmentation algorithm and deep learning method in histopathological image segmentation. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–5. IEEE, 2018. 1

[4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 5

[5] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11), 2006. 3

[6] Myriam Bontonou, Carlos Lassance, Ghouthi Boukli Hacene, Vincent Gripon, Jian Tang, and Antonio Ortega. Introducing graph smoothness loss for training deep learning architectures. In *2019 IEEE Data Science Workshop (DSW)*, pages 160–164. IEEE, 2019. 3

[7] Guangpu Dang, Zhongan Mao, Tingyu Zhang, Tao Liu, Tao Wang, Liangzhi Li, Yu Gao, Runqing Tian, Kun Wang, and Ling Han. Joint superpixel and transformer for high resolution remote sensing image classification. *Scientific Reports*, 14(1):5054, 2024. 1

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 5

[9] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 2

[10] Guoji Fu, Peilin Zhao, and Yatao Bian. p-laplacian based graph neural networks. In *International conference on machine learning*, pages 6878–6917. PMLR, 2022. 3

[11] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *European conference on computer vision*, pages 597–613. Springer, 2016. 1

[12] George Gartzonikas, Evaggelia Tsiligianni, Nikos Deligiannis, and Lisimachos P Kondi. A graph laplacian regularizer from deep features for depth map super-resolution. *Information*, 16(6):501, 2025. 3

[13] Rémi Giraud and Michaël Clément. Superpixel segmentation: A long-lasting ill-posed problem. *arXiv preprint arXiv:2411.06478*, 2024. 2

[14] Rémi Giraud, Vinh-Thong Ta, and Nicolas Papadakis. Scalp: Superpixels with contour adherence using linear path. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2374–2379. IEEE, 2016. 2

[15] Shengfeng He, Rynson WH Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *International journal of computer vision*, 115(3):330–344, 2015. 1

[16] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018. 1, 2, 3, 4, 5, 6

[17] Xiaohong Jia, Yonghui Li, Jianjun Jiao, Yao Zhao, and Zhiwei Xia. Ssmamba: Superpixel segmentation with mamba. *IEEE Signal Processing Letters*, 2025. 1, 2, 3, 4, 5

[18] Xiaohong Jia, Yonghui Li, Jianjun Jiao, Yao Zhao, and Zhiwei Xia. Ssmamba: Superpixel segmentation with mamba. *IEEE Signal Processing Letters*, 32:1715–1719, 2025. 5, 6

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

[20] Suha Kwak, Seunghoon Hong, and Bohyung Han. Weakly supervised semantic segmentation using superpixel pooling network. In *Proceedings of the AAAI conference on artificial intelligence*, 2017. 1

[21] Cheng Li, Baolong Guo, Nannan Liao, Jianglei Gong, Xiaodong Han, Shuwei Hou, Zhijie Chen, and Wangpeng He. Conic: contour optimized non-iterative clustering superpixel segmentation. *Remote Sensing*, 13(6):1061, 2021. 2

[22] Yao Li, Liyi Zhang, Lei Chen, and Yunpeng Ma. Superpixel guided spectral-spatial feature extraction and weighted feature fusion for hyperspectral image classification with limited training samples. *Scientific Reports*, 15(1):3473, 2025. 1

[23] Zhengqin Li and Jiansheng Chen. Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1356–1363, 2015. 2

[24] Yun Liang, Yuqing Zhang, Yihan Wu, Shuqin Tu, and Caixing Liu. Robust video object segmentation via propagating seams and matching superpixels. *IEEE Access*, 8:53766–53776, 2020. 1

[25] Hui Liu, Haiou Wang, Yan Wu, and Lei Xing. Superpixel region merging based on deep network for medical image segmentation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(4):1–22, 2020. 1

[26] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *CVPR 2011*, pages 2097–2104. IEEE, 2011. 2

[27] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 4571–4580, 2019. 1

[28] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001. 2

[29] Wenjian Qin, Jia Wu, Fei Han, Yixuan Yuan, Wei Zhao, Bulat Ibragimov, Jia Gu, and Lei Xing. Superpixel-based and boundary-sensitive convolutional neural network for automated liver segmentation. *Physics in Medicine & Biology*, 63(9):095017, 2018. 1

[30] Philip Sellars, Angelica I Aviles-Rivero, and Carola-Bibiane Schönlieb. Superpixel contracted graph-based learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(6):4180–4193, 2020. 1

[31] Yiwei Shen, Junchen Guo, Yan Liu, Chang Xu, Qingwu Li, and Fei Qi. Smanet: Superpixel-guided multi-scale attention network for medical image segmentation. *Biomedical Signal Processing and Control*, 100:107062, 2025. 1

[32] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 2

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 5

[34] Sudhir Sornapudi, Ronald Joe Stanley, William V Stoecker, Haidar Almubarak, Rodney Long, Sameer Antani, George Thoma, Rosemary Zuna, and Shelliane R Frazier. Deep learning nuclei detection in digitized histology images by superpixels. *Journal of pathology informatics*, 9(1):5, 2018. 1

[35] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018. 2

[36] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018. 5

[37] David A Tolliver and Gary L Miller. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1053–1060. IEEE, 2006. 2

[38] Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, and Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 568–576, 2018. 2

[39] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin De Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *European conference on computer vision*, pages 13–26. Springer, 2012. 2

[40] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 3

[41] Hai Wang, Xiongyou Peng, Xue Xiao, and Yan Liu. Bslic: Slic superpixels based on boundary term. *Symmetry*, 9(3): 31, 2017. 2

[42] Yaxiong Wang, Yunchao Wei, Xueming Qian, Li Zhu, and Yi Yang. Ainet: Association implantation for superpixel segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7078–7087, 2021. 1, 2, 3, 4, 5, 6

[43] Zhonghua Wang, Junyan Lyu, and Xiaoying Tang. Set: Superpixel embedded transformer for skin lesion segmentation. *Medical Image Analysis*, page 103738, 2025. 1

[44] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*. Springer, 2012. 3

[45] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6861–6871. PMLR, 2019. 3

[46] Sen Xu, Shikui Wei, Tao Ruan, and Yao Zhao. Esnet: An efficient framework for superpixel segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7): 5389–5399, 2023. 2

[47] Sen Xu, Shikui Wei, Tao Ruan, and Lixin Liao. Learning invariant inter-pixel correlations for superpixel generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6351–6359, 2024. 1, 2, 3, 4, 5

[48] Sen Xu, Shikui Wei, Tao Ruan, and Lixin Liao. Learning invariant inter-pixel correlations for superpixel generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6351–6359, 2024. 5, 6

[49] Junjie Yan, Yinan Yu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. Object detection by labeling superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5107–5116, 2015. 1

[50] Qi Yan, Shuzhen Zhang, Xiang Chen, and Ziyou Zheng. Multiscale superpixel depth feature extraction for hyperspectral image classification. *Scientific Reports*, 15(1):13529, 2025. 1

[51] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13964–13973, 2020. 1, 2, 3, 4, 5, 6

[52] Shijie Yang, Liang Li, Shuhui Wang, Weigang Zhang, and Qingming Huang. A graph regularized deep neural network for unsupervised image representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1203–1211, 2017. 3

[53] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 40–48, New York, New York, USA, 2016. PMLR. 3

[54] Qing Yuan, Songfeng Lu, Yan Huang, and Wuxin Sha. Sin: Superpixel interpolation network. In *Pacific Rim International Conference on Artificial Intelligence*, pages 293–307. Springer, 2021. 1, 2

[55] Ye Yuan, Zhiliang Zhu, Hai Yu, and Wei Zhang. Watershed-based superpixels with global and local boundary march-

ing. *IEEE Transactions on Image Processing*, 29:7375–7388, 2020. 2

[56] Jin Zeng, Jiahao Pang, Wenxiu Sun, and Gene Cheung. Deep graph laplacian regularization for robust denoising of real images. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 3

[57] Wei Zhao, Yi Fu, Xiaosong Wei, and Hai Wang. An improved image semantic segmentation method based on superpixels and conditional random fields. *Applied Sciences*, 8 (5):837, 2018. 1

[58] Dan Zhong, Tiehu Li, and Yuxuan Dong. An efficient hybrid linear clustering superpixel decomposition framework for traffic scene semantic segmentation. *Sensors*, 23(2):1002, 2023. 2

[59] Xianen Zhou, Yaonan Wang, Qing Zhu, Changyan Xiao, and Xiao Lu. Ssg: superpixel segmentation and grabcut-based salient object segmentation. *The Visual Computer*, 35(3): 385–398, 2019. 1

[60] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2814–2821, 2014. 1

[61] Konstantinos Zormpas-Petridis, Henrik Failmezger, Shan E Ahmed Raza, Ioannis Roxanis, Yann Jamin, and Yinyin Yuan. Superpixel-based conditional random fields (supercrf): incorporating global and local context for enhanced deep learning in melanoma histopathology. *Frontiers in oncology*, 9:1045, 2019. 1

[62] Konstantinos Zormpas-Petridis, Rosa Noguera, Daniela Kolarevic Ivankovic, Ioannis Roxanis, Yann Jamin, and Yinyin Yuan. Superhistopath: a deep learning pipeline for mapping tumor heterogeneity on low-resolution whole-slide digital histopathology images. *Frontiers in oncology*, 10:586292, 2021. 1