ADAPTIVE ZONE MERGING: GRAPH-BASED ALGORITHM FOR HIERARCHICAL OVER-SEGMENTATION

Petar Kotsev[⋆†△] *João F. C. Mota*[†] *Robert L. Nicol* [△] *Alex Serb*[⋆]

[⋆] University of Edinburgh, School of Engineering, Edinburgh, United Kingdom
[†] Heriot-Watt University, School of Engineering & Physical Sciences, Edinburgh, United Kingdom
[△] STMicroelectronics, Imaging Sub-group, Edinburgh, EH3 5DA, United Kingdom.

ABSTRACT

Over-segmentation is a widely used operation in computer vision pipelines, often employed as an intermediate spatial representation for efficient image perception, including superpixel-based tokenisation and interpretable region-based analysis. As these systems are increasingly deployed in real-time safety-critical applications such as medical imaging, industrial inspection and security, algorithms that are boundary-adherent, deterministic, and hierarchically-consistent become increasingly important. Existing methods typically trade off these properties, making it difficult to satisfy all four simultaneously. We propose Adaptive Zone Merging (AZM), a deterministic graph-based bottom-up over-segmentation algorithm designed to combine these properties while remaining amenable to parallel execution through local merge proposals. AZM exposes a single colour-distance threshold that induces a nested hierarchy obtained by monotonic region merging. This enables efficient multi-scale analysis with minimal re-computation overhead, providing stable region primitives for downstream learning and symbolic aggregation. We evaluate AZM on the BSDS500 dataset using boundary-focused metrics, demonstrating improved boundary F_1-score compared to widely used baselines. Source code is available at: `https://github.com/PetarKotsev/azm-segmentation`.

Index Terms— Over-segmentation, Superpixel segmentation, Graph-based segmentation, Hierarchical segmentation

1. INTRODUCTION

The field of image processing encompasses a wide range of methods, among which image segmentation plays a central role. Segmentation decomposes an image into coherent regions and is commonly divided into semantic segmentation and over-segmentation. Semantic segmentation assigns pixels to high-level, human-interpretable categories, whereas over-segmentation partitions an image into small homogeneous regions based on heuristics such as colour and texture similarity. While both paradigms are important, this work focuses on over-segmentation, often referred to as superpixel segmentation.

Over-segmentation has long been used as a preprocessing step to improve computational efficiency by reducing the number of primitives processed in subsequent stages of an image processing pipeline. More recently, superpixels have also been employed to support robustness, interpretability, and structured region-level representations in combination with modern deep learning models [1–5]. In these settings, over-segmentation is no longer merely a compression tool, but an explicit structural prior that influences downstream decision making.

Beyond natural-image benchmarks, computer vision systems are increasingly deployed in scientific, industrial, and medical applications, where image data is standardised and boundaries often correspond to physically meaningful structures. In such settings, segmentation errors directly impact downstream decisions — for example, in defect localisation, anatomical measurement, or safety-critical inspection — and even small structural errors can bias quantitative measurements and clinical interpretation [6, 7]. These domains typically place a higher cost on false positive boundaries than on missed fine details, since spurious edges fragment coherent regions and propagate uncertainty into subsequent learning or reasoning stages [8]. Over-segmentation methods intended for these applications must therefore prioritise boundary adherence, determinism, and parametric stability.

An additional and often under-emphasised requirement in these applications is the ability support consistent multi-scale spatial analysis. Many widely used over-segmentation methods expose a scale parameter, but changing this parameter typically requires recomputing the segmentation from scratch and does not guarantee that coarser segmentations remain structurally consistent with finer ones. While hierarchical segmentation frameworks exist, they are rarely designed to provide deterministic, controllable, and computationally efficient over-segmentation primitives for modern learning and reasoning pipelines. In contrast, a hierarchically consistent over-segmentation allows regions obtained at a coarser scale to be expressed as unions of regions obtained at finer scales, providing a natural link between fine-grained boundary detection and higher-level region aggregation while avoiding redundant computation.

These considerations directly motivate the proposed Adaptive Zone Merging (AZM) algorithm. We demonstrate that AZM achieves strong boundary adherence and hierarchical consistency on BSDS500 [9], positioning it as a practical primitive for safety-critical and multi-scale vision systems.

2. RELATED WORK

Over-segmentation methods divide into two branches. Classical methods are governed by hand-designed rules [10–13]; learning-based methods derive their behaviour from training data [14–16]. Learning-based methods achieve the higher boundary accuracy, but depend on labelled data, lose determinism through stochastic optimisation, and remain difficult to audit — limiting their use in safety-critical deployment. AZM targets this regime: deterministic, training-free, controlled by a single interpretable parameter, and producing a strictly nested hierarchy as a structural guarantee. We therefore benchmark against classical baselines.

Felzenszwalb and Huttenlocher (FH) [10] formulate segmentation as graph-based region merging. The image is represented as a weighted graph, where edge weights encode feature dissimilarity (often seen as the ℓ_2-norm in RGB space) and nodes represent neighbourhoods of pixels. Two nodes C_1 and C_2 are merged if the minimum inner component difference $\text{Diff}(C_1, C_2)$ is smaller than their internal variation,

$$\text{Diff}(C_1, C_2) \leq \min\big(\text{Int}(C_1) + \delta(C_1),\ \text{Int}(C_2) + \delta(C_2)\big),$$

where $\text{Int}(C)$ is the maximum edge weight inside the component and $\delta(C) = s/|C|$ controls the segmentation scale, where $s \in \mathbb{R}$ is the scale parameter. The FH algorithm uses global edge sorting and union-find merging, giving a computational complexity of $O(N \log N)$.

Simple Linear Iterative Clustering (SLIC) [13] produces superpixels by performing localised k-means clustering in a joint spatial–colour space. Each pixel is assigned to the nearest cluster centre by minimising the distance

$$D = \sqrt{d_c^2 + \left(\frac{m}{S}\right)^2 d_s^2},$$

where d_c is the colour distance, d_s is the spatial distance, S is the grid interval, and m controls the compactness–boundary trade-off. As SLIC is based on k-means, its computational complexity runs in $O(TN)$, where T is the number of refinement iterations.

The Watershed algorithm [11] interprets the image gradient magnitude as a topographic surface. Regions are formed by flooding from minima, with boundaries occurring along high-gradient ridges. The segmentation corresponds to catchment basins associated with local minima of the surface. Common implementation of the Watershed algorithm runs at near $O(N)$.

Quickshift [12] is a density-based mode-seeking method. It embeds each pixel as a 5D vector where the first two values are image coordinates, and the latter three are the colour values. It estimates a kernel density $\hat{p}(x)$ in that embedding space and links each point to its nearest neighbour with higher density:

$$y(x) = \arg \min_{y:\, \hat{p}(y) > \hat{p}(x)} \|x - y\|,$$

forming segments as connected components of these ascent paths. This yields adaptive region sizes but increases computational cost to $O(N^2)$.

We now turn to segmentation evaluation metrics. Let $\Omega \subset \mathbb{Z}^2$ denote the set of pixel coordinates of an image $I : \Omega \to \mathbb{R}^C$, where $C \in \mathbb{Z}$ is the number of colour channels. Consider a superpixel segmentation $S = \{S_j\}_{j=1}^J$, where $J \in \mathbb{N}$, forming a disjoint partition of Ω, and a set of ground-truth regions $G = \{G_r\}_{r=1}^R$, where $R \in \mathbb{N}$. The image domain may be indexed as $\Omega = \{x_n\}_{n=1}^N$, where N is the number of pixels in the image.

Boundary adherence: To quantify how well the predicted segment boundaries adhere to the human-annotated contours, one can use precision, recall, or the F_1-score. Let predicted and ground-truth boundaries are represented as sets of boundary pixel coordinates, S_C and G_C. Following the evaluation protocol of the Berkeley Segmentation Dataset and Benchmark (BSDS) [9], boundary pixels and predicted pixels are not matched by exact position, but instead associated using a tolerance threshold and a minimum-cost bipartite matching procedure. This produces a one-to-one correspondence between boundary pixels in the two maps, ensuring that each pixel may be matched at most once. In our experiments, we adopt the standard BSDS tolerance of 0.75% of the image diagonal. Let $M(S_C, G_C) \in \mathbb{Z}$ denote the number of boundary pixels S_C matched to the boundary pixels in G_C under this tolerance. Therefore, if we have S_C be the map of the predicted boundaries and G_C be a ground truth boundary map, then $M(S_C, G_C)$ will give us the number of matching predicted pixels, and $M(G_C, S_C)$ will give us the number of matching ground truth pixels to the predicted boundary pixels. From this, we can get the following expressions for the precision and recall of the boundary predictions:

$$BA_{\text{Recall}} = \frac{M(G_C, S_C)}{|G_C|}, \quad BA_{\text{Precision}} = \frac{M(S_C, G_C)}{|S_C|}.$$

Finally, the boundary adherence F_1-score is computed as:

$$BA_{F_1} = \frac{2 \times BA_{\text{Precision}} \times BA_{\text{Recall}}}{BA_{\text{Precision}} + BA_{\text{Recall}}}.$$

AZM is designed to optimise adherence to image gradients while being constrained to segment merging. As a result, we expect AZM to confidently and accurately predict contrasting boundaries (precision-first) but to combine objects of low contrast together.

Input : Image $I \in [0, 255]^3$; tile size $t \in \mathbb{Z}^+$; merge threshold $\tau \in \mathbb{R}^+$
Output: Segmentation S
$S \leftarrow$ partition I into $t \times t$ tiles (unique labels)
Build region adjacency graph G on S
Initialize min-heap $\mathcal{H}$ with all edges (a, b) keyed by $\|\mu(a) - \mu(b)\|_2$
while *$\mathcal{H}$ not empty* **do**
    $(d, a, b) \leftarrow$ pop minimum from $\mathcal{H}$
    if $d > \tau$ **then**
        break
    end
    if *a or b stale (version changed)* **then**
        continue
    end
    $c \leftarrow$ Merge(a, b)
    Update mean colour $\mu(c)$ and size
    Update G: replace a, b with c and relink neighbours
    foreach *neighbour u of c in G* **do**
        $d' \leftarrow \|\mu(c) - \mu(u)\|_2$
        push (d', c, u) into $\mathcal{H}$ with current versions
    end
end
return S

Algorithm 1: Core function of the Adaptive Zone Merging algorithm

Explained Variation: This metric quantifies the quality of the segmentation on the basis of colour variation, removing the need for ground truth annotations [17]:

$$EV(S) = \frac{\sum_{S_j} |S_j|(\mu(S_j) - \mu(\Omega))^2}{\sum_{x_n}(I(x_n) - \mu(\Omega))^2},$$

where μ denotes the mean of colour values in a segment as defined in (1). Although AZM does not directly optimise explained variation, its similarity-driven merging criterion encourages low intra-region colour variance, suggesting performance comparable to existing baseline methods.

Computational efficiency: Computational efficiency of an algorithm depends on factors such as implementation, computer language, and memory access management. To enable a hardware-agnostic comparison, we report worst-case computational complexity (number of floating-point operations) using standard Big-O notation.

3. METHODOLOGY

AZM constructs a nested family of segmentations through bottom-up agglomerative merging, enabling efficient multi-scale reasoning in downstream learning. It is a bottom-up region merging method operating on a Region Adjacency Graph (RAG). Starting from an initial fine tiling, adjacent regions are iteratively merged based on appearance similarity. The segmentation granularity is controlled by a single threshold parameter τ, which induces a consistent hierarchy of partitions across scales. The full procedure is summarised in Algorithm 1 and is further detailed in this section.

To decrease computational cost and over-adherence to gradients, AZM begins by partitioning the image into non-overlapping square tiles of size t (usually between 2 and 4), each assigned a unique initial label. This results in initialising S as a set of $N/(t^2)$ items. For each region S_j, AZM maintains a mean colour descriptor

$$\mu(S_j) = \frac{1}{|S_j|} \sum_{x_n \in S_j} I(x_n), \tag{1}$$

which provides an efficient appearance-based representation for merge decisions.

From the initial partition, AZM constructs a RAG G_S, where nodes correspond to regions and edges connect spatially adjacent superpixels. Each edge $(S_a, S_b) \in E$ is assigned a merge priority based on the Euclidean distance between region means: $d(S_a, S_b) = \|\mu(S_a) - \mu(S_b)\|_2$. All adjacency edges are inserted into a global min-heap $\mathcal{H}$, ensuring that merges are performed in order of increasing appearance similarity.

At each iteration, AZM extracts the most similar adjacent pair

$$(d, S_a, S_b) \leftarrow \text{popmin}(\mathcal{H}),$$

and merges the corresponding regions where d is smaller then the merge threshold τ. The merged region is defined as $S_c = S_a \cup S_b$, with updated mean descriptor computed by size-weighted averaging:

$$\mu(S_c) = \frac{|S_a|\mu(S_a) + |S_b|\mu(S_b)}{|S_a| + |S_b|}.$$

After merging, the adjacency graph is updated locally, and new candidate edges are inserted into $\mathcal{H}$.

A key property of AZM is that it induces a nested segmentation hierarchy. For two thresholds $\tau_1 < \tau_2$, the segmentation at the coarser scale is obtained by continuing the same merge process beyond the stopping point of τ_1. Therefore, $S(\tau_2) \preceq S(\tau_1)$, meaning that each region at scale τ_2 can be expressed as a union of regions obtained at scale τ_1. Consequently, boundaries at coarser thresholds form a subset of those at finer thresholds, enabling consistent multi-scale analysis without re-computation. While FH also performs bottom-up region merging, this property does not strictly hold, since its merge criterion depends not only on colour dissimilarity but also on the segment size and internal variation. As a result, changing the scale parameter s can lead to different merge orders and thus different boundary configurations. Fig. 1 shows the boundary lines that are merged at different merge thresholds.

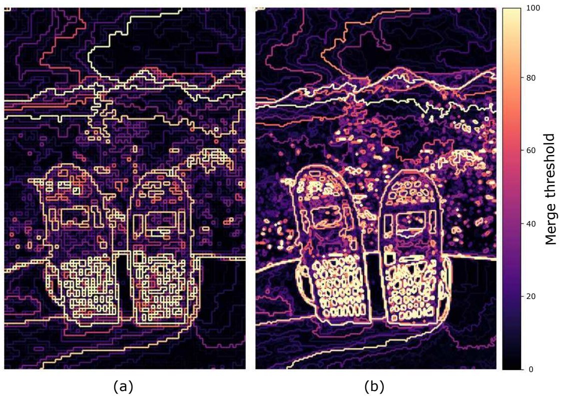

Fig. 1. Ultrametric Contour Map for AZM. The two images represent different tile sizes: (a) 4x4 pixels, (b) 2x2 pixels.

We note that AZM follows a graph-based agglomerative segmentation strategy. In the current CPU implementation, candidate merges are maintained in a min-heap priority queue, yielding complexity comparable to edge-ordered methods such as FH. Unlike FH, however, AZM operates on t^2 pixel tiles, reducing the effective graph size to $\frac{N}{t^2}$ nodes and giving an overall cost of $O\left(\frac{N}{t^2}\log\frac{N}{t^2}\right)$. While heap operations are inherently sequential, their core function - minimum-contrast candidate selection—can be expressed as a parallel $\arg\min$ reduction. On dedicated hardware, tournament networks (analogous to adder trees for multi-term summation) can compute this operation with $O(\log n)$ parallel depth (and $O(n)$ total comparisons), enabling pseudo-parallel acceleration even under globally ordered scheduling. Moreover, AZM's hierarchical merging suggests further speedups via local-first selection in early stages before global resolution. Fully parallel, hierarchy-aware scheduling strategies are left to future work.

4. RESULTS AND DISCUSSION

4.1. Qualitative

Fig. 2 presents a qualitative comparison between Adaptive Zone Merging (AZM) and several baseline over-segmentation methods at a fixed granularity of approximately 1200 segments (±1) on an image from the BSDS500 test set. The scene includes regions with both strong local contrast and more gradual transitions, providing representative structures for visual assessment. AZM produces fine partitions in areas with clear intensity changes, while maintaining larger, coherent regions in visually homogeneous parts of the image. In addition, AZM groups together areas of high colour similarity without introducing excessive fragmentation, yielding segmentations that remain well-aligned with perceptually meaningful boundaries across the scene. The figure also shows the impact of tile sizing - the 4-by-4 tile size provides a better differentiation of closely collared segments as well as faster computation, where as the 2-by-2 pixel tile size provides more precise borders and broader and a more course colour separation.

Quantitatively, we evaluate the AZM algorithm as a standalone over-segmentation method, focusing on boundary adherence and explained segment variation. We compare AZM against widely used over-segmentation algorithms on the BSDS500 dataset [9]. To stabilise all algorithms to small pixel variation we apply a Gaussian blur with spread $\sigma = 0.5$ as a pre-processing step to all presented experiments.

Fig. 2. Qualitative assessment on BSDS500 [9] with segments represented by average colour of the segment. All segmentations were produced with images having undergone Gaussian blurring of $\sigma = 0.5$. The images above show: (a) the original image; (b) AZM with a tile size of $t = 4$ and threshold of $\tau = 25$; (c) AZM with a tile size of $t = 2$ and threshold of $\tau = 60$; (d) FH with scale threshold of 18.7; (e) SLIC with a specified number of segments 1300 and compactness of 23.1; (f) Quickshift with kernel-size of 2 and maximum distance of 10.85.

4.2. Quantitative

Fig. 3 reports boundary precision-recall for AZM and standard over-segmentation baselines varying segmentation parameters. Dashed iso-contours denote constant boundary

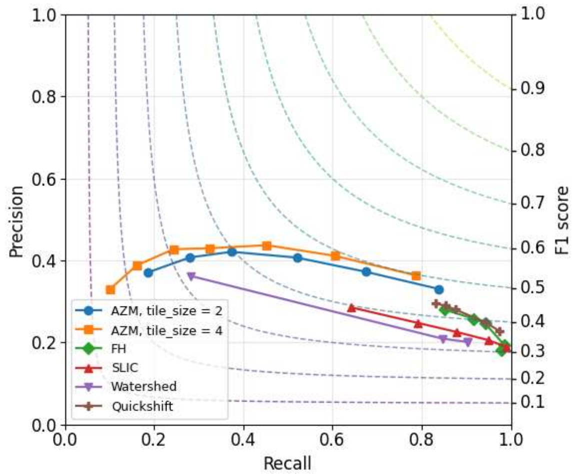

Fig. 3. Boundary precision–recall (F_1-score iso-contours) curves on 200 test images in BSDS500.

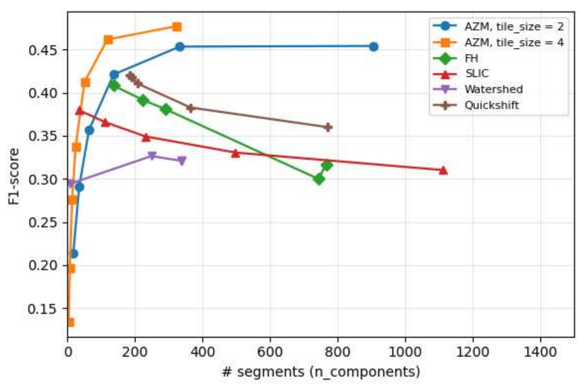

Fig. 4. Boundary F_1-score curves against average number of segments. Run on 200 test images in BSDS500.

F_1-score. AZM (at tile sizes $t = 2$ and $t = 4$) operates in a higher-precision regime than SLIC, Watershed, and Quick-Shift, indicating fewer spurious boundaries, and achieves the strongest boundary F_1 across a broad recall range. In contrast, FH and Quickshift favour higher recall at substantially reduced precision, reflecting noisier boundary predictions. These results highlight AZM's favourable boundary-adherence trade-off in settings where false positive edges are costly.

Fig. 4 shows the boundary F_1-score as a function of the number of segments produced by each method. AZM achieves the strongest boundary adherence across a broad operating range, reaching high F_1 values at moderate segment counts and maintaining performance as the segmentation is coarsened. In contrast, compactness-driven baselines such as SLIC and Watershed exhibit lower peak boundary accuracy, while FH degrades more noticeably at coarser scales. These results confirm that AZM provides a favourable precision–recall balance and robust boundary localisation without requiring excessive over-segmentation.

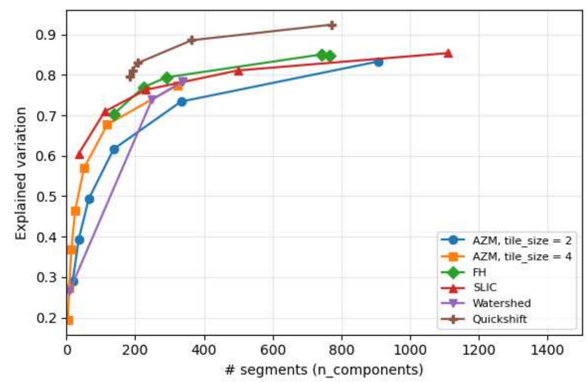

Fig. 5. Explained Variation curves on 200 test images in BSDS500.

Fig. 5 reports explained variation (EV) as a function of the number of segments produced by each method. While EV is not a metric that AZM directly optimises, the algorithm achieves competitive results across a broad range of segment counts, approaching the performance of established baselines such as SLIC and FH.

5. CONCLUSIONS

We introduced AZM, a deterministic and hierarchically consistent over-segmentation algorithm designed to prioritise boundary adherence in safety-critical vision settings, such as medical diagnosis and industrial defect detection. AZM performs bottom-up region aggregation through appearance-based merging controlled by a single threshold parameter, yielding a nested family of segmentations suitable for progressive multi-scale analysis. Experiments on BSDS500 [9] demonstrate that AZM achieves strong boundary precision and competitive explained variation across a broad range of segment counts, providing a favourable trade-off between reliable boundary localisation and region-level signal preservation. The monotonic relationship between the merge threshold and segmentation granularity further highlights AZM's suitability as a controllable intermediate representation for downstream inference and recognition tasks.

6. ACKNOWLEDGEMENTS

This work was collaboratively supported by STMicroelectronics, the Engineering and Physical Sciences Research Council and the Ministry of Defence Centre for Doctoral Training in Sensing, Processing and AI for Defence and Security, [EP/Y013859/1].

7. REFERENCES

[1] Jaihyun Lew, Soohyuk Jang, Jaehoon Lee, Seungryong Yoo, Eunji Kim, Saehyung Lee, Jisoo Mok, Siwon Kim, and Sungroh Yoon, "Superpixel Tokenization for Vision Transformers: Preserving Semantic Integrity in Visual Tokens," Mar. 2025, arXiv:2412.04680 [cs].

[2] Tsung-Wei Ke, Sangwoo Mo, and Stella X. Yu, "Learning Hierarchical Image Segmentation For Recognition and By Recognition," May 2024, arXiv:2210.00314 [cs].

[3] Alex Zihao Zhu, Jieru Mei, Siyuan Qiao, Hang Yan, Yukun Zhu, Liang-Chieh Chen, and Henrik Kretzschmar, "Superpixel Transformers for Efficient Semantic Segmentation," Oct. 2023, arXiv:2309.16889 [cs].

[4] Ángel Víctor Juanco-Müller, João F. C. Mota, Keith Goatman, and Corné Hoogendoorn, "Deep Supervoxel Segmentation for Survival Analysis in Head and Neck Cancer Patients," in *Head and Neck Tumor Segmentation and Outcome Prediction*, Vincent Andrearczyk, Valentin Oreiller, Mathieu Hatt, and Adrien Depeursinge, Eds., vol. 13209, pp. 257–265. Springer International Publishing, Cham, 2022, Series Title: Lecture Notes in Computer Science.

[5] Alex Yang, Charlie T. Veal, Derek T. Anderson, and Grant J. Scott, "Recognizing Image Objects by Relational Analysis Using Heterogeneous Superpixels and Deep Convolutional Features," Aug. 2019, arXiv:1908.00669 [cs].

[6] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez, "A Survey on Deep Learning in Medical Image Analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 2017, arXiv:1702.05747 [cs].

[7] Matthias Bruhns, Jan T. Schleicher, Maximilian Wirth, Marcello Zago, Sepideh Babaei, and Manfred Claassen, "Effects of segmentation errors on downstream-analysis in highly-multiplexed tissue imaging," *PLOS Computational Biology*, vol. 21, no. 9, pp. e1013350, Sept. 2025.

[8] Jo Schlemper, Ozan Oktay, Michiel Schaap, Mattias Heinrich, Bernhard Kainz, Ben Glocker, and Daniel Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images," *Medical Image Analysis*, vol. 53, pp. 197–207, Apr. 2019.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th int'l conf. Computer vision*, July 2001, vol. 2, pp. 416–423.

[10] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sept. 2004.

[11] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, June 1991.

[12] Andrea Vedaldi and Stefano Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in *Computer Vision – ECCV 2008*, vol. 5305, pp. 705–718. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, Series Title: Lecture Notes in Computer Science.

[13] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk, "SLIC superpixels," *Technical report, EPFL*, June 2010.

[14] 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*, 2020, pp. 13964–13973.

[15] Hankui Peng, Angelica I. Avilés-Rivero, and Carola-Bibiane Schoenlieb, "HERS superpixels: Deep affinity learning for hierarchical entropy rate segmentation," *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 72–81, 2021.

[16] V. Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz, "Superpixel sampling networks," in *European conference on computer vision*, 2018.

[17] Alastair Moore, Simon Prince, Jonathan H. Warrell, Umar Mohammed, and Graham Jones, "Superpixel lattices," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.