PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Graduate Program in Informatics

Isabela Borlido Barcelos

A SURVEY ON THE STATE-OF-THE-ART SUPERPIXEL SEGMENTATION

Belo Horizonte
2022

Isabela Borlido Barcelos

A SURVEY ON THE STATE-OF-THE-ART SUPERPIXEL SEGMENTATION

Dissertation presented to the Graduate Program in Informatics of the Pontifícia Universidade Católica de Minas Gerais, as a partial requirement for obtaining the title of Master in Informatics.

Advisor: Dr. Silvio Jamil Ferzoli Guimarães
Co-advisor: Dr. Alexandre Xavier Falcão

Research areas: Image segmentation and Digital Image Processing

Belo Horizonte
2022

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

Barcelos, Isabela Borlido
B242s A survey on the state-of-the-art superpixel segmentation / Isabela Borlido Barcelos. Belo Horizonte, 2022.
89 f. : il.

Orientador: Silvio Jamil Ferzoli Guimarães
Coorientador: Alexandre Xavier Falcão
Dissertação (Mestrado) - Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Informática

1. Processamento de imagens - Técnicas digitais. 2. Algoritmos computacionais. 3. Computação gráfica. 4. Teoria dos grafos. 5. Algoritmos. 6. Análise por agrupamento. I. Guimarães, Silvio Jamil Ferzoli. II. Falcão, Alexandre Xavier. III. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Informática. IV. Título.

CDU: 681.3.093

Ficha catalográfica elaborada por Fabiana Marques de Souza e Silva - CRB 6/2086

Isabela Borlido Barcelos

A SURVEY ON THE STATE-OF-THE-ART SUPERPIXEL SEGMENTATION

Dissertation presented to the Graduate Program in Informatics at the Pontifícia Universidade Católica de Minas Gerais, as a partial requirement for obtaining the title of Master in Informatics.

Research area: Image segmentation and Digital Image Processing

Dr. Silvio Jamil Ferzoli Guimarães (Advisor) – PUC Minas

Dr. Alexandre Xavier Falcão (Co-advisor) – UNICAMP

Dr. Zenilton Kleber Goncalves do Patrocinio Junior – PUC Minas

Dr. Paulo André Vechiatto de Miranda – USP

Belo Horizonte, 31 de agosto de 2022

ABSTRACT

Superpixel segmentation aims to divide images into homogeneous regions such that the union of one or more regions consists of the image object. It has several benefits, such as reducing the workload, reducing hundreds of thousands of pixels to hundreds of superpixels, and providing higher-level information than pixels. Consequently, their methods have been used in several applications. Superpixel segmentation has a vast literature covering several techniques. Due to this, some benchmarks were proposed to evaluate existing methods, verifying their state-of-the-art. Some of these works also proposed categorizations for superpixel methods. However, the existing categorizations do not cover several superpixel strategies. Furthermore, in contrast to the rapid progress in proposing new superpixel strategies, the proposed methods are often evaluated against classical methods, with few comparisons between recent proposals. Therefore, reviewing the most recent works and a new categorization for its methods becomes essential. Given the wide variety of superpixel segmentation strategies, a taxonomy to provide a classification based on different aspects of the superpixel approaches seems more appropriate. In this work, we provide a new taxonomy for superpixel segmentation according to their processing steps and the processing level of their features. To compose the categories of each processing step, we analyze 45 recent superpixel segmentation methods and present a review of these methods. Although these properties of superpixels are not a consensus in the literature, the inner color similarity usually underlies their methods. Among several existing measures, Explained Variation (EV) and Intra-cluster Variation (IV) seem to be the only ones focusing on color homogeneity. However, EV presents a high sensitivity, penalizing perceptually homogenous variations, while IV reduces penalization by averaging those differences. In this work, we argue that a small set of representative colors, not very different from each other, should describe the superpixel's colors. Such a set of colors must be minimal and able to represent a perceptually homogeneous texture. Therefore, we propose a new color homogeneity measure, named Similarity between Image and Reconstruction from Superpixels (SIRS), that appropriately penalizes superpixels with heterogeneous colors while maintaining high scores for perceptually homogeneous ones. The proposed measure uses a novel color descriptor, RGB Bucket Descriptor (RBD), representing the superpixel as a small set of its most relevant colors. Experiments on three datasets show that SIRS can better distinguish segmentation algorithms according to color homogeneity than EV (the most popular measure). The results also show that SIRS is more robust to slight color variations due to luminosity than EV. Using SIRS and the most used metrics in the literature, we evaluated 19 state-of-the-art superpixel segmentation methods in terms of their average performance and stability. Our evaluation intends to provide insights into the different approaches and support identifying the most suitable superpixel methods for each application. The evaluation results demonstrate the performance and limitations of state-of-the-art algorithms.

Keywords: Superpixel segmentation. Survey. Color homogeneity measure. Image segmentation.

RESUMO

A segmentação de superpixel visa dividir imagens em regiões homogêneas de tal forma que a união de uma ou mais regiões consiste no objeto da imagem. Ela possui vários benefícios, como redução da carga de trabalho, redução de centenas de milhares de pixels para centenas de superpixels e extração de informações de alto nível. Consequentemente, seus métodos têm sido usados em diversas aplicações. A segmentação de superpixels possui uma vasta literatura que abrange diversas técnicas. Devido a isso, alguns *benchmarks* foram propostos para avaliar os métodos existentes, verificando seu estado da arte. Alguns desses trabalhos também propuseram categorizações para métodos de superpixels. No entanto, as categorizações existentes não se aplicam a várias estratégias de superpixels. Além disso, em contraste com o rápido progresso de novas estratégias, os métodos propostos são frequentemente avaliados em relação aos métodos clássicos, com poucas comparações entre propostas recentes. Portanto, revisar os trabalhos mais recentes e uma nova categorização para seus métodos torna-se essencial. Dada a grande variedade de estratégias de superpixels, uma taxonomia que forneça uma classificação baseada em diferentes aspectos parece mais apropriada. Neste trabalho, propomos uma nova taxonomia para segmentação de superpixels de acordo com suas etapas de processamento e o nível de processamento de suas *features*. Para compor as categorias de cada etapa de processamento, analisamos 45 métodos recentes de segmentação de superpixels e apresentamos uma revisão desses métodos. Embora as propriedades desejadas na segmentação de superpixels não sejam um consenso na literatura, a homogeneidade de cores geralmente fundamenta seus métodos. Dentre as várias medidas existentes, a *Explained Variation* (EV) e a *Intra-cluster Variation* (IV) parecem ser as únicas com foco na homogeneidade de cores. No entanto, EV apresenta alta sensibilidade, penalizando variações perceptualmente homogêneas, enquanto IV reduz a penalização ao calcular a média dessas diferenças. Neste trabalho, defendemos que um pequeno conjunto de cores representativas, não muito diferentes entre si, deve descrever as cores do superpixel. Tal conjunto de cores deve ser mínimo e capaz de representar uma textura perceptualmente homogênea. Portanto, propomos uma nova medida de homogeneidade de cores, denominada *Similarity between Image and Reconstruction from Superpixels* (SIRS), que penaliza adequadamente os superpixels com cores heterogêneas, mantendo pontuações altas para os perceptualmente homogêneos. A medida proposta usa um novo descritor de cores, *RGB Bucket Descriptor* (RBD), representando o superpixel como um pequeno conjunto de suas cores mais relevantes. Experimentos em três conjuntos de dados mostram que o SIRS pode distinguir melhor os algoritmos de segmentação de acordo com a homogeneidade de cores do que o EV (a medida mais popular). Os resultados também mostram que o SIRS é mais robusto a pequenas variações de cor devido à luminosidade do que o EV. Usando SIRS e as métricas mais utilizadas na literatura, avaliamos 19 métodos de segmentação superpixels em termos de desempenho e estabilidade médios. Nossa avaliação pretende fornecer informações sobre as diferentes abordagens e apoiar a identificação dos métodos de superpixel mais adequados para cada aplicação. Os resultados da avaliação demonstram o desempenho e as limitações desses algoritmos.

Palavras-chave: Segmentação de superpixels. Revisão da literatura. Medida de homogeneidade de cor. Segmentação de imagem.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ANRW – Adaptive Nonlocal Random Walk
AWkS – Adaptative W-k-means-based Superpixels
BR – Boundary Recall
CFBS – Coarse-to-Fine Boundary Shift
CO – Compactness index
CONIC – Contour Optimized Non-Iterative Clustering
DAFnet – Dual-Attention Fusion Network for superpixel segmentation
DAL – Deep Affinity Learning network
DISF – Dynamic and Iterative Spanning Forest
DMMSS – Deep Merging Model for superpixel-based segmentation
DMMSS-FCN – Deep Merging Model for Superpixel Segmentation by Fully Convolutional Networks
DSR – Dynamic Spectral Residual
DPS – Density Peaks Superpixel
DRW – Dynamic Random Walk
DSC – Deep Superpixel Cut
EAM – Extract and Merging
EV – Explained Variation
FCSS – Fine-to-Coarse Superpixel Segmentation
ECCPD – Edge-Constrained Centroidal Power Diagram
EW-RIM – Edge-Aware RIM
GL$l_{1/2}$RSC – Graph laplacian $l_{1/2}$ regularized subspace clustering
HMLI-SLIC – Hierarchical and Multi-Level LI-SLIC
IBIS – Iterative Boundaries implicit Identification for superpixels
ICV – Intra-cluster Variation
MEE – Mean Exponential Error
MFGS – Multi-feature Fusion Graph for superpixels
ODISF – Object-based DISF
OISF – Object-based ISF
PGDPC – Peak-Graph-based fast Density Peak Clustering
RBD – RGB Bucket Descriptor
RISF – Recursive Iterative Spanning Forest
RSS – Root Spanning Superpixels
SCAC – Superpixel segmentation with Context-Adaptive Criteria
SCBP – Superpixel Based on Color and Boundary Probability
SCSC – Spatially Constrained Subspace Clustering
SEN – Superpixel Embedding Network
SENSS – Squeeze-and-Excitation Network for superpixel segmentation
SIN – Superpixel Interpolation Network
SIRS – Similarity between Image and Reconstruction from Superpixels
SLIC – Simple Linear and Iterative Clustering
SSFCN – Superpixel Segmentation for Fully Convolutional Network
SuperAE – Superpixel-wise Autoencoder
TASP – Texture-Aware and Structure-Preserving
UOIFT – Unsupervised OIFT

SUMMARY

1 INTRODUCTION

Digital image processing is an important area of research that encompasses several tasks. In particular, image segmentation aims to divide images into perceptually distinct regions according to some characteristic (*e.g.*, color) to simulate visual human perception (WANG et al., 2020; CUEVAS-VELASQUEZ; GALLEGO; FISHER, 2020). A common image partition approach is generating several disjoint groups of connected pixels, named superpixels, according to a predetermined criterion (*e.g.*, color similarity). Such procedure may have several benefits: (i) workload magnitude reduction (*i.e.*, pixels to superpixels); (ii) high-level semantic information by the superpixels; and (iii) accurate object delineation by its compounding superpixels. Consequently, superpixel segmentation methods have been used in several applications, such as segmentation (LIANG et al., 2020; SHENG et al., 2018), semantic segmentation (ZHAO et al., 2018), object detection (REN; ZHAO; WANG, 2019; SHU; DEHGHAN; SHAH, 2013), saliency detection (ZHANG et al., 2019; ZHOU et al., 2019), and image classification (FANG et al., 2015; SELLARS; AVILES-RIVERO; SCHÖNLIEB, 2020).

Figure 1 presents three superpixel segmentation examples, in which the superpixels' borders are shown in red. In the superpixel literature, several authors identify its desired properties. Although there is no consensus in the literature, most authors agree that superpixels must be composed of connected pixels, adhere to the objects' borders, and have a compact and regular shape (STUTZ; HERMANS; LEIBE, 2018; WANG et al., 2017). Moreover, the methods must be computationally efficient and generate a controllable number of superpixels. Although, several superpixel methods do not meet all criteria. In general, this occurs when one property's improvement leads to another's worsening. For example, while some approaches improve delineation, others maintain compactness at the expense of its delineation (SCHICK; FISCHER; STIEFELHAGEN, 2012; SCHICK; FISCHER; STIEFELHAGEN, 2014). Although the segmentation in Figure 1(a) have maximum compacity, its delineation is worse. Similarly, Figure 1(c) focuses on delineation but has low compacity. Some superpixel approaches try to manage this trade-off, which may lead to low delineation (Figure 1(b)). Since superpixel methods can attend to different properties, the evaluation measures used may vary depending on the property optimized.

Figure 1 – Superpixel segmentation examples

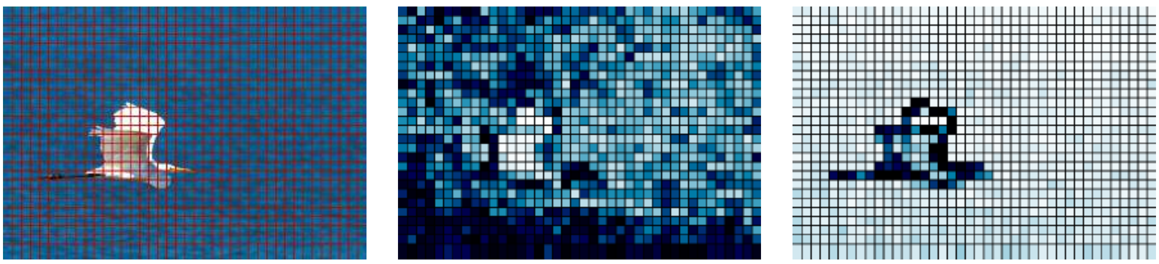
(a) (b) (c)

Source: Author.

Due to the lack of ground truth for superpixel segmentation, the ground-truth-dependent measures generally use the object's ground truth to evaluate the segmentation's quality concerning the object's borders. This evaluation usually ignores the internal partitioning of the object and the background. However, the same image can have different objects depending on the task performed. Therefore, the evaluation of the object delineation may be insufficient. On the other hand, most measures independent of ground-truth for quantitative evaluation assess superpixel compactness and regularity.

Regarding the independent ground truth measures for superpixel evaluation, only *Intra-cluster Variation* (ICV) (BENESOVA; KOTTMAN, 2014) and *Explained Variation* (EV) (MOORE et al., 2008) assess color homogeneity. However, both ICV and EV measures have several limitations. ICV computes the standard deviation of the pixel colors in each superpixel and averages them. It then reduces penalization for superpixels with subtle color variations. However, by not being a normalized measure, IV is not comparable between images nor with other measures (STUTZ; HERMANS; LEIBE, 2018). On the other hand, EV ignores color differences inside superpixels. It computes the differences between each superpixel's mean color and the mean color of the image. Despite its popularity (GIRAUD; TA; PAPADAKIS, 2017; STUTZ; HERMANS; LEIBE, 2018), EV cannot describe perceptually homogeneous regions in some situations. Figure 2(a) shows an image with slight color variations due to luminosity. In Figure 2(b-c), whiter values indicate higher scores (homogeneous regions). EV cannot capture the color homogeneity of a simple grid segmentation (Figure 2(b)). The color homogeneity measured may have improved results (*i.e.*, closer to the perceptual homogeneity) when representing the superpixels by its most frequent colors and measuring its color homogeneity as the image reconstruction error using the *Similarity between Image and Reconstruction from Superpixels* (SIRS) (Figure 2(c)), the color homogeneity measure proposed in this work.

Figure 2 – Difference between color homogeneity measures in a grid segmentation with 1000 superpixels



(a) Grid segmentation (b) EV (c) SIRS

Source: Author.

The literature on superpixel segmentation has significantly expanded in recent years, from improvements in classical methods to entirely new approaches. However, the papers generally compare their proposals with classical methods. Furthermore, the existing benchmarks do not cover several recent works (STUTZ; HERMANS; LEIBE, 2018; WANG et al., 2017), which makes it very difficult to identify the contribution of the recent superpixels methods. Therefore, a new extensive evaluation is necessary to establish the current state-of-the-art.

1.1 Research questions and hypotheses

To explore the mentioned superpixel segmentation problems, we reduced them to two research questions.

Question 1: *Is there a current categorization for superpixel methods?*

In general, papers related to superpixels categorize their methods into clustering-based, graph-based, and deep learning-based methods. Although it is the most predominant categorization, the definition of each category varies according to the focus of each work. Stutz, Hermans and Leibe (2018) presented the most comprehensive categorization, but their categories have ambiguous characteristics and are not based on well-defined criteria. For example, the watershed-based and clustering-based categories refer to methods based on the watershed and k-means algorithms, respectively. In contrast, the energy optimization and graph-based categories refer to the modeling of the methods. Therefore, the reference for categorization is not established. In addition, there is some ambiguity between categories, such as graph-based and path-based, as path-based methods are usually modeled on graphs, thus fitting into both categories. According to Stutz, Hermans and Leibe (2018), their categorization does not cover several approaches. In particular, methods based on deep learning do not fit into any of the categories. At the same time, other works generally limit themselves to placing all methods based on deep learning in a single category, despite the wide range of deep-based approaches.

Hypothesis 1.1: *Every superpixel method has an algorithm related to its implementation, and each algorithm can be divided into processing steps.*

Due to the varied approaches to superpixel segmentation, the definition of a single and definitive set of categories for the methods is a temporary and ineffective solution. Temporary because the development of new strategies can either put old approaches into disuse or compose new approaches that do not fit into pre-existing categories. And it is not very effective because the current categorization does not provide knowledge to explore the categorized methods for developing new methodologies. To provide a taxonomy that allows a comprehensive comparison between methods, not only by their general process, we define the partitioning of the methods in up to three steps: pre-processing, main processing, and post-processing. Despite restricting the taxonomy, the definition of a finite and pre-established set of steps allows comparing and joining methodologies more naturally. However, one may note that multi-step methods may have a set of processes related to one of the established steps, while methods with fewer steps may not contain pre- or post-processing.

Hypothesis 1.2: *Suppose that each processing step can be divided into processes (or sub-steps) and that the processes of each step are related only to the method algorithm — i.e., its high-level implementation — for that process. In this case, one can observe the similarity between methods with similar steps, even if they are not based on the same method.*

Despite starting in 2003 (REN; MALIK, 2003), the superpixel literature is vast and contains very different strategies. However, superpixel methods that are not directly related can perform similar processes, such as sampling seeds in the pre-processing stage or merging regions in the post-processing stage. By reflecting on such similarities in the taxonomy, one may establish connections between different methods, favoring the analysis of strategies and providing both the improvement of their processes and the study of the effect of such sub-steps on the final segmentation.

Question 2: *Is the current quantitative assessment of superpixels adequate to assess their quality?*

Firstly, it is necessary to establish the ideal superpixel segmentation characteristics as its desired properties, as mentioned in several works in the literature. Although the properties may differ among the papers, Stutz, Hermans and Leibe (2018) establish those most cited by the authors: partitioning, connectivity, adherence to contours, compactness, efficiency, and a controllable number of superpixels. In addition, color homogeneity is deeply related to the superpixels' definition. For ground-truth-based assessment, datasets for object segmentation, object saliency, and semantic segmentation are generally used due to the lack of ground-truth for superpixel segmentation.

Superpixels are ideally composed of connected pixels, homogeneous in color, and with limits that adhere to the image boundaries. According to these properties, one can ideally partition a homogeneous image region into various equally correct superpixels. In this case, the segmentation application may define what partitioning best suits its requirement. Therefore, establishing the ideal ground-truth for superpixel segmentation is difficult for general-purpose segmentation — *i.e.*, when the application is unknown. Ground-truth-based superpixel evaluation measures assess their delineation against one or more objects but not against all image boundaries. As delineation is only one of the desired properties and there is no ground-truth for superpixels, further measures are necessary for its evaluation. Most of the independent ground-truth measures for superpixel evaluation assess compactness and regularity. In addition, only two assess color homogeneity. Among the measures that evaluate color homogeneity, only one is widely used in the literature. However, its high sensitivity to color variations, among other limitations, makes the measured variation little corresponding to visual color homogeneity.

Hypothesis 2.1: *A superpixel's color variation does not directly correspond to its visual color homogeneity. Therefore, by including robustness to low color variations, the measured variation becomes more similar to the visually perceived one.*

Superpixels are non-overlapping image regions, ideally homogeneous in color. However, color and lighting variations in natural images are common, especially in textured areas. In superpixel color homogeneity evaluation, the result must resemble the visual homogeneity. Nevertheless, the color homogeneity measures assess the color variation with their variance, leading to high penalties in barely perceptible color variations.

Hypothesis 2.2: *The mean color of a superpixel is insufficient to represent its information.*

Although superpixels are ideally homogeneous, obtaining regions with visible color variations in natural images is common. Simpler textures may be present in areas whose variation is limited to a set of similar colors. Visually, such textures do not represent significant heterogeneity; therefore, the homogeneity assessment measure must reflect this information. The widely used Explained Variation measure assesses color homogeneity and represents the superpixel's content by its mean color. However, the mean color can be unrepresentative, missing important color information. One can argue that homogeneous superpixels can have some color variation, visually low representative. Therefore, instead of representing superpixels by their mean color, one can describe their content with a small set of colors (*e.g.*, its most frequent colors).

Hypothesis 2.3: *Assessing the homogeneity of superpixels as independent regions provide a result closer to their visual quality than assessing their homogeneity in relation to the image.*

One may have different assessments for visually similar superpixels in different images by conditioning the superpixel color homogeneity to global image characteristics. Since a homogeneous region does not depend on its context, such an assessment may cause a discrepancy between visual and measured homogeneity.

1.2 Contributions

The contributions in this work are:

- Provide a comprehensive overview of the recent superpixel approaches;
- Provide a taxonomy of the superpixel methods to better represent them with a less restrictive representation;
- Propose a new color homogeneity measure for quantitative superpixel evaluation;
- Performs an extensive qualitative and quantitative assessment on various datasets, providing a guideline to future methodologies.

1.3 Overview

Chapter 2 presents the concepts of image segmentation, the previous reviews on superpixel literature, the mathematical image modeling used in this work, the concepts and limitations of color-based measures for superpixel segmentation, and a brief relation between superpixel descriptor and image reconstruction. Then, Chapters 3 and 4 present the preliminary results of the proposed taxonomy and review recent methods for superpixel segmentation. Chapter 5 presents the proposed measure, named SIRS, for color homogeneity in superpixel segmentation. Chapters 6 and 7 present the evaluation of proposed SIRS and an extensive evaluation of the state-of-the-art superpixel methods, respectively. Finally, Chapter 8 presents the conclusions and future works.

2 BACKGROUND

This Chapter presents an overview of image segmentation and its various categories of approaches according to the supervision of methods, types of regions produced, and problem modeling (Section 2.1). Then, the literature papers that perform an overview of superpixel segmentation methods and evaluation benchmarks are presented in Section 2.2. Next, we present the definitions for image and segmentation modeling used in the color homogeneity measure proposed (Section 2.3) and the other measures for superpixel segmentation assessment (Section 2.4). Finally, Section 2.5 presents a modeling of image reconstruction based on superpixel descriptor.

2.1 Image segmentation

Image segmentation aims to divide images into perceptually distinct regions according to some characteristic. Due to the poor definition of human perception, image segmentation remains an open problem. However, several works have proposed solutions approaching the problem from different perspectives (for example, as a problem of super segmentation or object delineation), which may contain different solutions depending on the purpose of the application used (PARIHAR; BOVEIRI, 2018; ZHU et al., 2016; RAMADAN; LACHQAR; TAIRI, 2020)..

According to Zhu et al. (2016), the segmentation strategies can be broadly classified as unsupervised (or automatic), semi-supervised (or weakly supervised), and fully supervised. While unsupervised methods use only image features to form significant regions without user intervention, semi-supervised methods receive user information, which can be scribbles on the image (in interactive segmentation) or sets of images containing similar objects. (in co-segmentation) (ZHU et al., 2016; RAMADAN; LACHQAR; TAIRI, 2020). Finally, fully supervised methods use annotated images to train the segmentation model. After training, these methods perform segmentation from unannotated images (ZHU et al., 2016).

In segmentation, regions usually represent objects and backgrounds, multiple objects, or superpixels. Object segmentation aims to partition images into two (or more) non-overlapping regions, labeled as object or background. On the other hand, superpixel segmentation aims to obtain homogeneous regions with high adherence to the image borders, such that the union of one or more regions consists of the image object (REN; MALIK, 2003). The desired properties in the superpixel segmentation are to create a controllable number of homogeneous, compact, disjoint regions with high adherence to the image borders (STUTZ; HERMANS; LEIBE, 2018).

Segmentation approaches can model the problem in discrete or continuous spaces. According to Zhu et al. (2016), both clustering methods and graph-based methods model the problem in discrete spaces, in which the former model the problem in the space of local characteristics, and the latter have the advantage of not committing discretization errors due to its operators acting in a space provided by graph theory (PENG; ZHANG; ZHANG, 2013). On the other hand, methods that model in continuous spaces treat the image as a continuous surface, whose segmentation aims to minimize or maximize a functional of this surface.

2.2 Suveys on superpixel segmentation

The first benchmark for superpixel evaluation, proposed by Neubert and Protzel (2012), is composed of eight algorithms and evaluates the design and robustness of the methods to affine transformations. In their work, the authors argue that the undersegmentation measure causes a biased penalty for the superpixel size. Due to this, to evaluate the proposed benchmark, the authors proposed a modified under-segmentation error measure to consider the smallest part of the superpixel leakage. The evaluation showed that the segmentation approaches present similar results in all assessments, demonstrating that the most appropriate methods for each task depend on the crucial characteristics. In addition, according to the evaluation performed, algorithms less focused on compactness showed greater robustness to image transformations, being more appropriate for superpixel segmentation.

Achanta et al. (2012) also perform an assessment, comparing five superpixel methods to determine their benefits and limitations regarding their boundary adherence and efficiency. Unlike Neubert and Protzel (2012), Achanta et al. (2012) characterized the superpixel methods as graph-based and gradient-ascent-based. The former contains methods that model the segmentation problem based on graph theory generating superpixels by minimizing a cost function defined on the graph. The second iteratively refines its initial clusters until reaching a convergence criterion. Achanta et al. (2012) also extensively evaluated the *Simple Linear and Iterative Clustering* (SLIC), a method for segmenting superpixels based on k-means for efficient segmentation and improved delineation.

SLIC is more efficient than k-means because it reduces the segmentation complexity to linear concerning the number of pixels by limiting the search space to a region proportional to the superpixel size. And its best delineation comes from a distance measurement that gives it better control over the size and compactness of the superpixels. SLIC starts with a grid sampling of the superpixel centers. Then, it assigns the most similar pixels to each superpixel based on a distance measure limited to the region around the superpixel center. At the end of each iteration, the centers are updated to the pixel with the most similar color to the mean superpixel color. The iterations stop when the residual error is less than a threshold or the number of iterations reaches a maximum value. As post-processing, SLIC ensures connectivity by assigning unconnected superpixels to their nearest neighbors. Compared with five other state-of-the-art algorithms, SLIC showed better performance in 2D images, and Achanta et al. (2012) also demonstrated its application in 3D biomedical images.

Although Achanta et al. (2012) and Neubert and Protzel (2012) settled that methods that control superpixel compactness are more appropriate, their quantitative evaluation does not cover the segmentation compactness. Schick, Fischer and Stiefelhagen (2012) identified this drawback and proposed a compactness measure based on the isoperimetric coefficient (POLYA, 2020). The authors demonstrated that there is a trade-off between compactness and boundary recall. They also present a new algorithm that controls this trade-off, overcoming the state-of-the-art superpixel methods. The proposed method improves SLIC to control the superpixels' compactness with a new distance measure. It also enforces the superpixels' connectivity by updating only the pixels on superpixels' borders. Schick, Fischer and Stiefelhagen (2012) evaluated the association between compactness and delineation metrics. The results showed an inverse and non-linear association between compactness and boundary recall. Due to this, the authors argue that non-compactness is similar to overfitting the image boundaries, capturing many low-importance borders. Therefore, a more accurate segmentation would not imply better overall performance. Thus, the authors claim that compact superpixels better capture spatially coherent information allowing an easier information extraction from their boundaries. In addition, the proposed method showed better object boundary adherence than the other evaluated methods. In (SCHICK; FISCHER; STIEFELHAGEN, 2014), the authors extend their work to present a new algorithm modification to reinforce the superpixels' regularity while maintaining the lattice structure. By adding restrictions to preserve lattice, the proposed segmentation produces superpixels slightly more compact and less adherent to the object boundaries. However, the trade-off between compactness and delineation surpasses the other evaluated methods. In addition, the proposal also presents greater stability regarding its superpixels positioning.

Stutz (2015) perform a more exhaustive evaluation and present a new benchmark evaluation in two image datasets and fifteen superpixel segmentation methods, including algorithms and datasets that use depth information. To obtain the best performance of each approach, the authors optimized their parameters for the delineation measures with a grid search. According to the evaluation, several methods present excellent performance with low execution time. However, the inclusion of depth in the segmentation may not represent an improvement in the results. Regarding visual quality, the authors settled that the high quantitative results in the delineation assessment do not necessarily reflect the segmentations' visual quality.

Mathieu, Crouzil and Puel (2017) argue that the two datasets used by Stutz (2015) may be insufficient for an exhaustive evaluation. They overcome this with a new dataset, called the *Heterogeneous Size Image Dataset* (HSID). The HSID mainly contains large images (with millions of pixels) and allows evaluating the superpixel methods according to the image size. Using the HSID, the authors carefully analyzed the five best superpixel methods in (STUTZ, 2015) and the Waterpixels (MACHAIRAS et al., 2015) method. However, to evaluate superpixel segmentation using *Boundary Recall* (BR) (LEVINSHTEIN et al., 2009), the tolerance error in HSID images using the image diagonal can be very high due to the large proportions of some images. To solve this problem, the authors used the theory of fuzzy sets to define a new boundary adherence measure, the *Fuzzy Boundary Recall* (FBR). According to Mathieu, Crouzil and Puel (2017), the evaluated methods do not achieve a satisfactory trade-off between adherence to contours, conciseness (smallest possible number of superpixels), and efficiency. Therefore, the authors argue that the superpixel method must be chosen according to the necessary superpixels' characteristics for the desired task.

Despite regularity being indicated as a desired property in superpixel segmentation, its evaluation was restricted to qualitative analysis. Wang et al. (2017) proposed a regularity measure for superpixels, allowing the quantitative regularity analysis. The authors also provided an overview of the superpixel methods and a benchmark with fifteen state-of-the-art methods and thirteen evaluation metrics, including the proposed one. Wang et al. (2017) categorize superpixel methods as clustering-based (or gradient-based) and graph-based, following the characterization of Achanta et al. (2012). According to Wang et al. (2017) , methods based on clustering showed greater efficiency, while those based on graphs presented an improved delineation. However, the running time was not considered satisfactory, and the authors settled that the evaluated algorithms are hardly applicable in scenarios requiring real-time responses.

Stutz, Hermans and Leibe (2018) present a more comprehensive evaluation in a benchmark with 28 state-of-the-art superpixel algorithms in 5 datasets that include indoor, outdoor, and people images. In addition to the benchmark, the authors also propose evaluation measures independent of the number of superpixels and based on existing delineation metrics: *Average Miss Rate* (AMR), *Average Under-segmentation Error* (AUE), and *Average Unexplained Variation* (AUV). The authors evaluated the stability of the methods, considering the minimum, maximum, and standard deviation of each metric;

and its robustness to noise, blur, and affine transformations. Based on the categorization by Achanta et al. (2012), they categorize superpixel methods by their high-level approach, allowing them to relate their categories to experimental results. The categories defined by the authors were: Watershed-based, Density-based, Graph-based, Contour evolution, Path-based, Clustering-based, Energy optimization, and Wavelet-based. Among these, Watershed-based and Clustering-based contain methods with algorithms based on watershed and k-means, respectively; Graph-based and Path-based are related to image modeling for creating superpixels, where both contain methods that model the image as a graph, but the former corresponds to undirected graphs that are partitioned according to some criterion, and the latter aims to compute paths in graphs according to some criterion; Energy optimization contains methods whose pixel clustering is performed according to the iterative optimization of an energy function; Contour evolution contains methods whose superpixels evolve from their contours; Finally, Wavelet-based corresponds to the Edge-Avoiding Wavelets (STRASSBURG et al., 2015), which does not fit into the other categories. Despite the broad categorization performed by Stutz, Hermans and Leibe (2018), the authors settled that some methods in the literature are not included in their categorization.

Stutz, Hermans and Leibe (2018) evaluated the correlation between the evaluation metrics. They settled that the Undersegmentation Error (UE) of Neubert and Protzel (2012) has a high correlation with *Achievable Segmentation Accuracy* (ASA) (LIU et al., 2011) and that the same does not occur with the UE of Levinshtein et al. (2009). They hypothesize that this might be related to Levinshtein's UE unfairly penalizing some superpixels, as suggested by Achanta et al. (2012). Furthermore, they point out that the measures *Mean Distance to Edge* (MDE) (BENESOVA; KOTTMAN, 2014) and Boundary Recall (BR) (MARTIN; FOWLKES; MALIK, 2004) are strongly correlated. Therefore, their assessment does not use ASA and MDE. In relation to the Intra-cluster variation (ICV), although it is not correlated with the Explained variation (EV), the ICV was not used in this work because it is not normalized. In both qualitative and quantitative evaluation, the path-based and density-based methods performed worse, while some iterative energy optimization, clustering-based, and graph-based methods performed better. The authors also settled that good contour adherence generally requires less compactness, regularity, and/or smoothness, and these characteristics are not necessarily linked and can be evaluated separately. The authors conclude that the proposed metrics accurately reflect the performance of the methods and that the affine transformations did not influence the performance of the algorithms. Based on the proposed evaluation, they create a ranking of the evaluated methods, and they recommend six of them: ETPS (YAO et al., 2015), SEEDS (BERGH et al., 2012), ERS (LIU et al., 2011), CRS (CONRAD; MERTZ; MESTER, 2013), ERGC (BUYSSENS; GARDIN; RUAN, 2014), and SLIC (ACHANTA et al., 2012).

2.3 Image modeling

Let an *image* $\mathcal{I}$ be defined as a pair $(\mathbf{I}, I)$ in which $\mathbf{I} \subset \mathbb{Z}^2$ is the set of *picture elements* (*i.e.*, *pixels*) whose *colors* is a vector mapped by $I(p) \in \mathbb{R}^m$, given $m \in \mathbb{N}^*$. Note that, when $m = 1$, $\mathcal{I}$ is *grayscale* and it is *colored* otherwise. We may compute the *ℓ-norm* of $I(p) = \langle I_1(p), \ldots, I_m(p)\rangle$ of the colors of the pixel p by $\|I(p)\|_\ell = \left(\sum_{j=1}^{m} |I_j(p)|^\ell\right)^{1/\ell}$, given $\ell \in \mathbb{N}^*$. By setting $\ell = 1$ and $\ell = 2$, the ℓ-norm is equivalent to the *Manhattan* and *Euclidean* distances, respectively.

If a set $X \subseteq \mathbf{I}$ of pixels is provided, one may calculate its *mean color* $\mu(X) \in \mathbb{R}^m$

by $\mu(X) = \frac{\sum_{x \in X} I(x)}{|X|}$, where $|X|$ denotes its size. Furthermore, we may *segment* X into $k \in \mathbb{N}^*$ subsets by a function $\mathbf{S}(X, k) \in \mathbb{P} \backslash \emptyset$, being $\mathbb{P}$ the *power set*, resulting in a *partition* (or *grouping*) $\{X_1, \ldots, X_K\}$ such that $\bigcup_{i=1}^{k} X_i = X$, $\bigcap_{i=1}^{k} X_i = \emptyset$, and $k \leq |X|$. We may extend such concepts for describing the *segmentation* $S \in \mathbf{S}(\mathbf{I}, k)$ of an image $\mathcal{I}$, in which every S_i is a *region* or *superpixel*.

2.4 Evaluation measures for superpixel segmentation

Ideally, superpixels should be homogeneous in color, composed of connected pixels, adhere to the object's borders, and have a compact, regular shape. Furthermore, its methods must be computationally efficient and generate a controllable number of superpixels. Several evaluation measures were proposed to assess the quality of the literature methods, each evaluating a specific segmentation's characteristic. In general, the measures for superpixel evaluation can be divided into measures that evaluate: (i) superpixel delineation; (ii) its shape; or (iii) its color homogeneity.

2.4.1 Delineation measures

In superpixel segmentation, delineation metrics evaluate the overlap of the superpixel boundaries with the image object. The delineation-based evaluation is widespread in superpixel segmentation since the oversegmentation of the object and background regions is not penalized. On the other hand, the quality of the superpixels inside these regions is also not evaluated (STUTZ; HERMANS; LEIBE, 2018).

Boundary Recall (BR) (MARTIN; FOWLKES; MALIK, 2004) is a widely used measure for superpixel evaluation. It measures the fraction of ground-truth boundary pixels correctly detected, as presented in Equation 2.1, where TP is the number boundary pixels that match in a segmentation S and a ground-truth G, and FN is the number of boundary pixels in G that does not match with S. The boundary pixels are matched within a local neighborhood of size $(2r+1)^2$, in which r is 0.0025 times the image diagonal.

$$\mathrm{BR}(S, G) = \frac{\mathrm{TP}(G, S)}{\mathrm{TP}(G, S) + \mathrm{FN}(G, S)} \tag{2.1}$$

In the superpixel literature, BR is commonly evaluated regarding the number of superpixels, but its relation to other metrics has also been explored (SCHICK; FISCHER; STIEFELHAGEN, 2012; MARTIN; FOWLKES; MALIK, 2004; GIRAUD; TA; PAPADAKIS, 2018).

Similar to BR, *Boundary Precision* (BP) estimates the percentage of superpixel contours that overlap the ground-truth contours at a minimum distance r, according to Equation 2.2, where FP is the number of boundary pixels in S that does not match with G.

$$\mathrm{BP}(S, G) = \frac{\mathrm{TP}(G, S)}{\mathrm{TP}(G, S) + \mathrm{FP}(G, S)} \tag{2.2}$$

Benesova and Kottman (2014) argue that one of the disadvantages of BR is to consider the same recall for all boundary pixels in S at a minimum distance of G. To overcome this limitation, the authors proposed the *Mean Distance to Edge* (MDE) (Equation 2.3), where $\mathcal{B}(A)$ is the set of boundary pixels in a segmentation A. The MDE evaluates the average distance of all boundary pixels in G to the closest boundary pixels in S. However, Stutz, Hermans and Leibe (2018) evaluated a high correlation between BR and MDE.

$$\mathrm{MDE}(S, G) = \frac{1}{|G|} \sum_{p \in \mathcal{B}(G)} \min_{q \in \mathcal{B}(S)} \{ \, \|p - q\|_2 \, \} \tag{2.3}$$

Another metric widely used to assess the quality of superpixel segmentation delineation is the *Undersegmentation Error* (UE). Introduced by Levinshtein et al. (2009), the UE measures the adherence of the boundary pixels in S to the G contours based on the area between S and G regions. UE has different versions (STUTZ; HERMANS; LEIBE, 2018). The most recommended was proposed by Neubert and Protzel (2012) that evaluated the adherence to contours based on the minimum area of overlap between S and G, as presented in Equation 2.4, where N is the number of pixels G and k is the number of regions in G.

$$\mathrm{UE}(S, G) = \frac{1}{N} \sum_{i}^{k} \sum_{S_j \cap G_i \neq \varnothing} \min\{|S_j \cap G_i|, |S_j - G_i|\} \tag{2.4}$$

The delineation quality in superpixel segmentation can also be evaluated from its accuracy using the *Achievable Segmentation Accuracy* (ASA) (LIU et al., 2011). The ASA measures the segmentation accuracy from the maximum overlap between each superpixel and the regions of G, as shown in Equation 2.5. Despite being a widely used metric, ASA strongly correlates with UE (STUTZ; HERMANS; LEIBE, 2018; GIRAUD; TA; PAPADAKIS, 2017).

$$\mathrm{ASA}(S, G) = \frac{1}{N} \sum_{S_j} \max_{G_i} \{|S_i \cap G_j|\} \tag{2.5}$$

In this work, we used the BR, BP, and UE measures to evaluate design due to their high correlation with MDE and ASA and because they are more used in the superpixel literature.

2.4.2 Shape-based measures

Shape-based evaluation metrics assess the superpixels' compactness and regularity. These metrics evaluate whether the superpixels have compact shapes with smooth contours and are arranged regularly — *i.e.*, in a grid. Although these properties have an inverse relationship to the delineation, an improved boundary recall does not necessarily imply better segmentation (SCHICK; FISCHER; STIEFELHAGEN, 2012; SCHICK; FISCHER; STIEFELHAGEN, 2014). Due to this, the superpixels' methods quality has been evaluated in previous benchmarks according to the trade-off between its shape quality and delineation (STUTZ, 2015; WANG et al., 2017).

The *Compactness index* (CO) (SCHICK; FISCHER; STIEFELHAGEN, 2012) measure uses the isoperimetric quotient to measure the similarity between the shape of a superpixel and a circle, which constitutes the most compact geometric shape. The CO measure is presented in Equation 2.6, in which $A(S_j)$ and $P(S_j)$ are the superpixel area and perimeter, respectively.

$$\mathrm{CO}(S) = \frac{1}{N} \sum_{S_j} |S_j| \frac{4\pi A(S_j)}{P(S_j)} \tag{2.6}$$

In the superpixel literature, regularity and compactness are often used to name the same properties of convexity and smooth contours in superpixels. Machairas et al. (2015)

and others proposed to evaluate only the contours of the superpixels rather than their shape. For this, the authors proposed the *Contour Density* (CD), which measures the relative number of contours in the segmentation. However, CD showed a high correlation with CO (STUTZ; HERMANS; LEIBE, 2018).

$$\mathrm{CD}(S) = \frac{\mathcal{B}(S)}{N} \tag{2.7}$$

Giraud, Ta and Papadakis (2017) argue that the superpixel compactness should be independent of the regular convex geometry. The authors proposed the *Global Regularity* (GR), which assesses the regularity and smoothness of superpixel contours without a pre-established geometry. In GR, the *Convexity Criterion* (CC) (Equation 2.8) is measured according to the relationship between each region's perimeter and area. The *Regularity Criterion* (RC) (Equation 2.9) is then defined as the relationship between the convexity of the convex hull of S_j ($H(S_j)$) and the convexity of the region S_j.

$$\mathrm{CC}(S_j) = \frac{|P(S_j)|}{|S_j|} \tag{2.8}$$

$$\mathrm{RC}(S_j) = \frac{\mathrm{CC}(H(S_j))}{\mathrm{CC}(S_j)} \tag{2.9}$$

To evaluate the balance of the distribution of superpixels in the image, the variance term Vxy was defined according to Equation 2.10, where $\sigma_{S_{jx}}$ and $\sigma_{S_{jy}}$ are the standard deviations of the x and y positions of the pixels in S_j. The *Shape Regularity Criteria* (SRC) (Equation 2.11) is defined by the pixel's regularity, smoothness, and balance.

$$\mathrm{V}_{xy}(S_j) = \sqrt{\frac{\min(\sigma_{S_{jx}}, \sigma_{S_{jy}})}{\max(\sigma_{S_{jx}}, \sigma_{S_{jy}})}} \tag{2.10}$$

$$\mathrm{SRC}(S) = \sum_{S_j} \frac{|S_j|}{N} \mathrm{RC}(S_j)\mathrm{V}_{xy}(S_j) \tag{2.11}$$

Despite measuring the compactness independent of geometry, the SRC does not consider the superpixel size variation. For this, the authors defined the *Smooth Matching Factor* (SMF), which evaluates the spatial distribution of the average size of superpixels. The SMF is defined in Equation 2.12, where S_j^* is the centered version of S_j and S^* is the average of the centered shapes of the superpixels in S.

$$\mathrm{SMF}(S) = 1 - \sum_{S_j} \frac{|S_j|}{N} \left\| \frac{\overline{S}}{|\overline{S}|} - \frac{S_j^*}{|S_j^*|} \right\|_1 \tag{2.12}$$

$$\mathrm{GR}(S) = \mathrm{SRC}(S)\mathrm{SMF}(S) \tag{2.13}$$

Due to the high correlation between CO and CD, we use the CO and GR metrics to evaluate the quality of the superpixel shapes in this work.

2.4.3 Color-based measures

Although the desired properties of superpixels are not a consensus in the literature, the inner color similarity usually underlies their methods. Among the existent measures for superpixel assessment, only the *Intra-cluster Variation* (ICV) (BENESOVA; KOTTMAN, 2014) and the *Explained Variation* (EV) (MOORE et al., 2008) assess color homogeneity. However, both measures have some drawbacks. Followed by the intuition that uniformity exhibits low color variability towards the mean, the Intra-Cluster Variation (ICV) (BENESOVA; KOTTMAN, 2014) measures homogeneity of a superpixel S_i by its standard color deviation. Consequently, as shown in Equation 2.14. the homogeneity of an image $\mathcal{I}$ is defined as the mean color homogeneity of the segmentation S:

$$ICV(S) = \frac{1}{|S|} \sum_{S_i \in S} \frac{\sqrt{\sum_{p \in S_i} \|I(p) - \mu(S_i)\|_1^2}}{|S_i|} \tag{2.14}$$

One major drawback of ICV is not presenting normalized values, being not comparable across images and datasets (STUTZ; HERMANS; LEIBE, 2018). Moreover, it penalizes all superpixels equally within the computation (GIRAUD; TA; PAPADAKIS, 2017). That is, the importance of each region, and thus its deviation is equivalent irrespective of its size. Finally, the mean color amortizes the color variations within the superpixel, possibly resulting in an inaccurate color when the composing ones are significantly discrepant.

In contrast to ICV, the *Explained Variation* (MOORE et al., 2008) defines homogeneity by comparing the variance of the superpixels' mean color $\mu(S_i)$ and the variance of the pixels' color $I(p)$ towards the image's mean color $\mu(\mathbf{I})$, resulting in a normalized measure (Equation 2.15). This measure is maximum when $|S| = |\mathbf{I}|$ or when $I(p) = \mu(S_i)$ for all $p \in S_i$ and for every $S_i \in S$. However, similarly to ICV, EV considers the superpixels' mean color, which is insufficient for describing perceptually homogeneous textures (MOORE et al., 2008).

$$EV(S) = \frac{\sum_{S_i \in S} |S_i| \, \|\mu(S_i) - \mu(\mathbf{I})\|_1^2}{\sum_{p \in \mathbf{I}} \|I(p) - \mu(\mathbf{I})\|_1^2} \tag{2.15}$$

2.5 Superpixel descriptor to image reconstruction

In superpixel segmentation, one may interpret the superpixels' description as an image reconstruction. By representing (or describing) each superpixel by its mean color, the reconstructed image may have regions whose color does not express its actual content. On the other hand, when using a set of colors for this representation, the recovered image is more similar to the original image. From this, we argue that the image's representation by the mean color of its superpixels can lead to a significant loss of information. On the other hand, the purpose of the description here is not an ideal image reconstruction but a description of its superpixels. Thus, limiting this description to a small set of relevant colors for each superpixel should result in minimal information loss for those with low variation. Ideally, suppose the relevant colors are not very distinct from each other, and their loss of information in the reconstructed image is minimal. In this case, the superpixel is composed of visually homogeneous colors.

Considering that the average color used in the ICV and EV is insufficient to describe the colors of a superpixel, a new description is necessary. Supposing that a small set of colors can represent a visually homogeneous superpixel, we can assume that a heterogeneous superpixel cannot be well represented in the same way. Therefore, the heterogeneity of a superpixel may be related to the quality of its description from a few colors. Thus, this descriptor's error in recovering the superpixel content in the original image may be related to its color homogeneity. Based on this idea, in this work, we assess the quality of the superpixel segmentation by its ability to reconstruct the original image. More formally, let $\mathcal{R} = (\mathbf{I}, R)$ be a *reconstructed image* of $\mathcal{I}$ in which every pixel $p \in \mathbf{I}$ has its reconstructed (or predicted) color $R(p) \in \mathbb{R}^m$. Such reconstruction is ideal when $R \equiv I$. If a segmentation S is provided, the popular approach is to assign $R(p) = \mu(S_i)$ for all $p \in S_i$ and every $S_i \in S$.

3 TAXONOMY FOR SUPERPIXEL SEGMENTATION

In recent years, several works have contributed to the development of new strategies for superpixel segmentation (PENG; AVILES-RIVERO; SCHÖNLIEB, 2022; ZHU et al., 2021; WU; LIU; LI, 2021; BELÉM et al., 2021). Due to this, benchmarks were proposed to evaluate existing methods, verifying the state-of-the-art and presenting improvements in the evaluation (NEUBERT; PROTZEL, 2012; ACHANTA et al., 2012; SCHICK; FISCHER; STIEFELHAGEN, 2014; WANG et al., 2017; STUTZ, 2015; STUTZ; HERMANS; LEIBE, 2018). Some of these works also categorized superpixel methods, making it possible to relate their categories with the experimental results (STUTZ; HERMANS; LEIBE, 2018; ACHANTA et al., 2012), as presented in Chapter 2.

The first categorization of superpixel methods, proposed by Achanta et al. (2012), divides the superpixel methods into graph-based and gradient ascending-based methods. While the former minimizes a cost function defined over the image graph, the latter optimizes clusters by iteratively updating them until reaching a convergence criterion. Later, Stutz, Hermans and Leibe (2018) provides a more comprehensive categorization based on high-level approaches to each method. Using the categorization proposed in their work, Stutz, Hermans and Leibe (2018) identifies common characteristics in the results of methods of the same category. However, they settled that their categorization does not cover several existing strategies.

With the advancement in superpixel segmentation, many strategies were proposed, diverging from the categories established by Stutz, Hermans and Leibe (2018). Among these, several approaches based on deep learning have emerged with different strategies to overcome the limitations of convolutional networks (YANG et al., 2020; PENG; AVILES-RIVERO; SCHÖNLIEB, 2022). Therefore, the existing classifications are insufficient to cover the wide variety of approaches to superpixels, and the rapid advance in the superpixel literature makes it difficult to establish a better categorization for their methods.

In this work, we establish that a good classification of methods can satisfy the following statements: (i) the characteristics (or definitions) of the categories of methods need to be abstract enough to encompass (*i.e.*, be valid for) all the methods in them included; and (ii) the definitions of the categories need to be distinct from each other so that their boundaries are clear. Given the wide variety in the strategies used for superpixel segmentation, a taxonomy that provides a classification based on different aspects of the methods seems more appropriate, as it allows the establishment of better-defined categories, requiring a lower level of abstraction in its definition. In addition, classification based on different aspects can make it possible to identify trends caused by specific aspects common to a set of methods.

3.1 Processing steps

To provide a comprehensive taxonomy with a more natural representation of the superpixel approaches, we identified that superpixel segmentation methods generally have three steps: (i) initial processing, (ii) main processing, and (iii) final processing. In the initial processing step, methods usually perform seed sampling, initial segmentation, or feature extraction. The main processing usually contains the clustering process, with several strategies for generating superpixels. Finally, in the final processing step, cluster refinement operations are generally performed to ensure their connectivity or to fine-tune the segmentation. Among the established steps, the main step is the most challenging because it contains the superpixel generation process and has a greater variety of strategies.

3.2 Processing level features

In addition to the processing steps, the characteristics used are important to consider in the segmentation methods. In the superpixel literature, several features are used, such as edge maps, semantic features, affinity maps, and object saliency maps. To categorize a superpixel method based on the processing level of the image features, we assign the highest level used considering the following categories:

- **Pixel-level features:** Raw data resources in images — *e.g.*, pixel color, position, and depth;

- **Mid-level features:** features that can be computed based on a set of pixels, smaller than the entire image — *e.g.*, patch-based feature, path-based feature, gradient, or boundary;

- **High-level features:** features that combine pixel properties and high-level information. The high-level information cannot be extracted from a small set of pixels. They are given directly by the user or predicted by other models — *e.g.*, saliency map, semantic features, texture, or a desired object geometry;

3.3 Categories in recent superpixel literature

To compose the categories of each of the aforementioned processing steps, we analyzed 45 recent superpixel segmentation methods. This analysis identified that some neural architectures used do not consist of pre-existing models in the literature, while others contain simple architectures of convolutional networks. To find broad categories that identify the process performed by the methods that use convolutional networks, we categorized the network architecture and objective learning — *e.g.*, for a network with encoder-decoder architecture to learn affinity maps, its architecture may be identified as encoder-decoder, and its objective learning as affinity maps. While the method may have specific loss functions, making it impossible to create more comprehensive categories, the network learning objective is related to the optimization task.

The categories shown in Table 1 identify the main processing of the evaluated methods that do not use convolutional networks in this step. These categories were defined based on the main processing performed to obtain superpixels. Although we indicate at most one category for each processing step in this work, it is possible to have more than one category in the same processing step for more complex methods that mix different approaches. It is important to emphasize that the categories defined in this work do not intend to cover all the superpixel literature. On the other hand, the taxonomy structure developed, in which the methods are represented by up to three processing steps and the level of the processed features, can be used in other superpixel methods.

Table 2 presents the proposed taxonomy applied to the analyzed methods, the color space, and the inspiration method. In the processing steps whose solutions contain convolutional networks in Table 2, the architecture (*arch*) of the network and its training objective (*train*) are informed. In Table 2, processing steps that use another superpixel method are indicated with the category *Clustering method*. Similarly, the other categories of the initial and final processing stages were established according to their high-level objective.

According to our analysis, most superpixel methods do not have convolutional networks in any of their processing steps and perform seed sampling in their initial processing.

Table 1 – Main processing categories excluding methods based on neural networks

Categories	Explanation
Neighborhood-based clustering	Performs clustering based on the similarity between pixels restricted to a maximum spatial distance from some reference point in the image.
Boundary evolution clustering	These algorithms iteratively update the superpixels' boundaries to improve their superpixels, usually using a coarse-to-fine image block strategy.
Dynamic-center-update clustering	The dynamic-center-update algorithms perform clustering with a distance function based on the features of the clusters, dynamically updating their centers.
Path-based clustering	Path-based approaches generate superpixels by creating paths in the image graph based on some criteria. Usually, its clustering criterion is a path-based function to optimize during clustering.
Hierarchical clustering	These algorithms create regions in the image that form a hierarchical structure, obeying the criteria of locality and causality (GUIGUES; COCQUEREZ; MEN, 2006).
Density-based clustering	The superpixel methods rely on an optimization function to find the cluster centers, modeling the problem of finding superpixels in a problem of finding density peaks.
Sparse linear system clustering	Model the segmentation problem with a sparse matrix and use its properties to find superpixels.
Data distribution-based clustering	The approach assumes that the image pixels follow a specific distribution and perform the clustering based on this conjecture.
Regional feature extraction	Iteratively extracts regional features to perform clustering based on these features.
Polygonal decomposition clustering	Segmentation in these methods consists of decomposing the image into non-overlapping polygons.
Graph-based clustering	Perform superpixel segmentation based on graph topology

Source: Author.

Furthermore, most of the analyzed methods perform region-based clustering on their main processing step, and the most frequent final processing consists of merging neighboring regions (Merging step), usually to enforce connectivity.

* Partially † With post-processing

Table 2 – Recent methods for superpixel segmentation

Method	Year	Iterative	#Iter.	#Superp.	Connec.	Compact.	Superv.	Color	Initial processing	Main processing	Final processing	Features: Pix.	Features: Mid.	Features: High.	Inspired
K-SLIC	2021	✓	✓	✓		✓		CIELAB	Compute optimum K	Clustering with SLIC		✓			SLIC (2012)
TASP	2021	✓	✓	✓				CIELAB	Seed sampling	Neighborhood-based clustering			✓		SLIC (2012)
MFGS	2020			✓*	✓	✓		CIELAB	Seed sampling	Neighborhood-based clustering	Merging step		✓		SLICO (2012)
DSR	2021	✓		✓		✓		CIELAB	Seed sampling	Neighborhood-based clustering	Merging step			✓	dSLIC (2018)
Semasuperpixel	2021	✓	✓	✓	✓†		✓	CIELAB	*arch:* Encoder-decoder *train:* Semantic map	Neighborhood-based clustering	Merging step			✓	SLIC(2012)
AWkS	2021	✓	✓	✓				CIELAB	Seed sampling	Neighborhood-based clustering	Merging step	✓			W-k-means (2005)
IBIS, IBIScuda	2021	✓		✓	✓†	✓		CIELAB	Grid segmentation	Boundary evolution clustering	Merging step	✓			SLIC (2012)
CFBS	2020	✓		✓	✓	✓		CIELAB	Grid segmentation	Boundary evolution clustering		✓			SLIC (2012)
SCAC	2021b			✓*	✓	✓		CIELAB	Grid segmentation	Boundary evolution clustering	Boundary evolution clustering		✓		WSBM (2020)
LSC-Manhattan	2022	✓		✓	✓	✓			Classification	Boundary evolution clustering				✓	LSC (2017)
CONIC	2021			✓	✓	✓		CIELAB	Seed sampling	Dynamic-center-update clustering			✓		SNIC (2017), SCALP (2018)
DRW	2020			✓	✓				Seed sampling	Dynamic-center-update clustering	Label propagation		✓		RW (2006)
FCSS	2021	✓	✓*	✓	✓†	✓		CIELAB		Dynamic-center-update clustering		✓			SNIC (2017)
F-DBSCAN	2021			✓	✓			CIELAB		Dynamic-center-update clustering		✓			RT-DBSCAN (2018)
SCBP	2021			✓	✓	✓		RGB		Dynamic-center-update clustering	Merging step		✓		DBSCAN (2016)
A-DBSCAN	2021			✓	✓	✓		RGB	Compute features	Dynamic-center-update clustering	Merging step			✓	DBSCAN (2016)
RSS	2020			✓	✓	✓			Seed sampling	Path-based clustering			✓		IFT (2004)
DISF	2020	✓		✓	✓			CIELAB	Seed oversampling	Path-based clustering			✓		ISF 2019
ODISF	2021	✓		✓	✓			CIELAB	Seed oversampling	Path-based clustering				✓	DISF (2020), OISF (2018)
UOIFT	2020			✓	✓			CIELAB	Clustering method	Hierarchical clustering			✓		IFT (2004), OIFT (2013)
HMLI-SLIC	2021	✓	✓	✓*	✓	✓		CIELAB	Clustering method	Hierarchical clustering	Merging step	✓			SLIC (2012)
RISF	2018	✓	✓	✓	✓	✓		CIELAB		Hierarchical clustering	Hierarchical region merging		✓		ISF (2019)
DAL-HERS	2022			✓	✓		✓	RGB	*arch:* Multi-scale Residual CNN *train:* Affinity map	Hierarchical clustering				✓	SEAL (2018), ERS (2011)
PGDPC, SLIC-PGDPC	2021			✓	✓			CIELAB	Seed sampling	Density-based clustering			✓		DPC (2018)
DPS	2021			✓*				CIELAB	Compute features	Density-based clustering	Clustering method		✓		DP (2014)
ANRW	2020			✓	✓			CIELAB	Seed sampling	Sparse linear system clustering				✓	NRW (2015)
GL$l_{1/2}$RSC	2022	✓		✓					Clustering method	Sparse linear system clustering	Encoding procedure		✓		CAWR (2017)
SCSC	2020	✓	✓	✓				RGB	Clustering method	Sparse linear system clustering	Clustering method			✓	
EAM	2020			✓*	✓			RGB	Noise remotion	Regional attributes extraction	Merging step		✓		
ECCPD	2020	✓	✓	✓	✓			RGB	Seed sampling	Polygonal decomposition clustering	Boundary evolution clustering		✓		
gGMMSP	2020	✓	✓	✓*	✓†	✓		CIELAB		Data distribution-based clustering	Merging step		✓		GMMSP (2018)
E2E-SIS	2020			✓	✓†		✓	CIELAB		*arch:* FCN *train:* Superpixels	Superpixel pooling layer and merging step			✓	DEL (2018), SSN (2018)
ss-RIM	2020			✓*				RGB		*arch:* FCN *train:* Image reconstruction and Superpixels				✓	DIP (2018), RIM (2010)
EW-RIM	2021			✓	✓	✓		RBG		*arch:* FCN *out:* Image reconstruction and Superpixels				✓	ss-RIM (2020), DIP (2018)
SEN	2020			✓				RGB	*arch:* Encoder-Decoder *train:* Deep embeddings	Clustering method				✓	RPEIG (2018)
DMMSS-FCN	2020				✓		✓	RGB		*arch:* Encoder-Decoder *train:* Edge map decision				✓	
UDAG	2021				✓			CIELAB	Clustering method	*arch:* Encoder-Decoder *train:* Inpainting	Clustering method			✓	GL Graph (2015)
SuperAE-DSC	2021	✓	✓	✓	✓			RGB	*arch:* Autoencoder CNN *train:* Image reconstruction and pixel labeling	Clustering method	Differentiable clustering			✓	
SSFCN	2020			✓*			✓	CIELAB		*arch:* Encoder-Decoder *train:* Superpixels	Merging step			✓	SSN (2018)
SENSS	2022			✓*	✓	✓	✓	CIELAB		*arch:* Encoder-Decoder *train:* Superpixels				✓	SSFCN (2020)
DAFnet	2021			✓	✓		✓	CIELAB		*arch:* Weight-shared CNN *train:* Superpixels				✓	SSFCN (2020)
LNSNet	2021			✓		✓		LAB/RGB		*arch:* FCN *train:* Image reconstruction and Superpixels	Merging step				
DMMSS	2021	✓			✓				Clustering method	*arch:* FCN *train:* Binary classification	*arch:* FCN *train:* Binary classification			✓	
SIN	2021a			✓*	✓		✓			*arch:* Interpolation Network *train:* Superpixels				✓	
BP-net	2021			✓			✓	RGB-D	Seed sampling	*arch:* Multi-scale CNN and FCN *train:* Boundary map and superpixels	Merging step			✓	

Source: Author.

4 STATE-OF-THE-ART ON SUPERPIXEL SEMENTATION

Superpixel segmentation has a vast literature covering several techniques. Stutz, Hermans and Leibe (2018) provides a benchmark for superpixels with an extensive evaluation. Nevertheless, due to the rapid progress in developing new strategies for superpixel segmentation, an analysis of the most recent proposals becomes essential. Therefore, this section reviews the recent literature on superpixel segmentation.

4.1 Neighborhood-based clustering

Neighborhood-based methods for superpixel segmentation perform clustering of image pixels based on the similarity between pixels restricted to a maximum spatial distance from some reference point in the image. For example, in Simple Linear and Iterative Cluster (SLIC) (ACHANTA et al., 2012), the maximum distance is a fixed-size image patch, and the reference point is the cluster center. In general, neighborhood-based methods perform multiple iterations until reaching a point of convergence or a maximum number of iterations and do not guarantee the spatial connectivity of their superpixels.

K-SLIC. The SLIC segmentation method allows controlling the number of desired superpixels, but it can be a challenging parameter to set. The authors in (ULLAH; BHATTI; ZIA, 2021) propose a granulometric approach and a quality metric method to surpass this drawback. In the granulometric approach, the number of superpixels K_g is given by the weighted average of the image pattern spectrum PS_w computed for each color channel in an arithmetic summation, presented in Equation (4.1), in which I_r, I_g, and I_b are the RGB planes of an image I, and the PS_w value represents the relative importance of the image components.

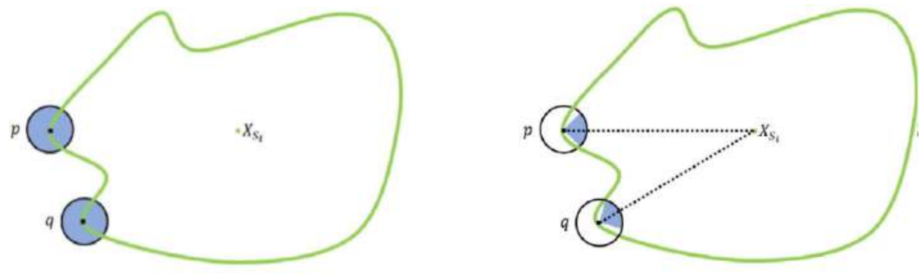
$$K_g = \sum PS_w(I_r) + \sum PS_w(I_g) + \sum PS_w(I_b) \tag{4.1}$$

The second approach proposed uses several metrics based on entropy, texture, and ground-truth independent quality metrics to choose by the majority vote. The SLIC segmentation of both proposals visually presents good choices to K_g. In bad-lighted conditions, the quality metric method is less affected and provides a large number of superpixels as compared to the granulometric method. The quality measure performs better with different spatial resolutions, unaffected by spatial resolution changes. Despite the improved results with the quality metric method, it is computationally expensive, while the granulometric method is faster to compute but has worse performance.

TASP. To solve the problem of handling weak gradient structures and strong gradient textures, a *Texture-Aware and Structure-Preserving* (TASP) superpixel segmentation algorithm is proposed in (WU; LIU; LI, 2021). The proposal's pipeline is based on SLIC (ACHANTA et al., 2012) with an integrated structure-avoiding clustering distance. based on a centroid-oriented quarter-circular mask and a hybrid gradient. The TASP's proposed distance is based on a centroid-oriented quarter-circular mask and a hybrid gradient.

Figure 3 shows the difference between simple circular masks (Figure 3(a)) and the proposed mask (Figure 3(b)). In Figure 3, the masks are centered at p and q pixels, the region with a green border is a superpixel with centroid X_{S_i}, the blue part represents the truth area of the mask, and the dashed lines in (b) represent the direction of the masks. The hybrid gradient is based on the product of the Sobel and interval (LEE et al., 2017) gradients, and the proposed mask (Figure 3(b)) prevents inconsistent texture pixels from

Figure 3 – Centroid-oriented quarter-circular mask of TASP algorithm



(a) Circular mask (b) Centroid-oriented quarter-circular mask

Source: Wu, Liu and Li, 2021.

being sampled from the local image patch.

Based on the centroid-oriented quarter-circular mask and the hybrid gradient, the proposed distance function catches thin structure edges and prevents the superpixels from crossing the object boundaries, due to its tradeoff between texture suppression and structure edge detection. The Equation 4.2 presents the proposed distance function $D(p, S_i)$, where d_G allows catching thin structure edges using the maximum hybrid gradient magnitude on the linear path, d_F measures the tradeoff between weak gradient structures and strong gradient textures, d_S is the spatial distance, and N_C and N_S are the size and compactness factors, respectively.

$$D(p, S_i) = d_G\left(\left(\frac{d_F}{N_C}\right)^2 + \left(\frac{d_S}{N_S}\right)^2\right) \tag{4.2}$$

TASP has an effective structure-preserving and texture-suppression procedure, outperforming state-of-the-art methods, especially in images with strong texture and weak boundaries structure. However, TASP is highly time-consuming, and it does not produce more superpixels in regions with finer details, missing some structure boundaries.

MFGS. In (LIU; DUAN, 2020), the authors proposed a two-stage method for superpixels segmentation for RGB-D images, named *Multi-feature Fusion Graph for superpixels* (MFGS). In the first stage, the MFGS uses color, and 2D and 3D spatial positions (with depth) to perform an iterative clustering based on SLICO (ACHANTA et al., 2012). In the second stage, it performs a merging multi-feature step, where the proposal uses the euclidean distance, covariance matrix distance, and boundary distance to estimate the similarity between superpixels and uses the label cost proposed in (DELONG et al., 2012) to remove redundant labels. The MFGS method is faster, produces compact and regular superpixels, and has a higher segmentation accuracy. However, the proposal's merging stage does not allow control of the number of final superpixels.

DSR. Inspired by the dynamic region range of dSLIC (MAIERHOFER et al., 2018), a SLIC-based method called *Dynamic Spectral Residual* (DSR) superpixels is proposed in (ZHANG et al., 2021) to improve seed initialization and clustering range, by incorporating saliency information. The proposed method computes the saliency map based on Fourier analysis, proposed by Hou et al. (HOU; ZHANG, 2007). The DSR uses a structure measure $\mathcal{G}$ (Equation 4.3) based on the saliency map $\mathcal{SR}$ and it's mean $\overline{\mathcal{SR}}$ to define the search range for clustering and the seed initialization.

$$\mathcal{G}(x) = \exp(\mathcal{SR} - \bar{\mathcal{SR}}) \tag{4.3}$$

The DSR initializes selecting the superpixel centers on the structure measure minima. After, the structures of its neighbor's pixels are set as maximum and increase the structure measure of pixels in the initialization range. Next, DSR performs clustering similar to SLIC (ACHANTA et al., 2012), but instead of using a fixed grid-based patch, it uses a search range based on the structure measure. The search range can connect uniform regions, avoiding unnecessary small superpixels in large regions.

DSR creates more seeds in heterogeneous areas but avoids creating redundant seeds. Also, it produces larger (and few) superpixels on homogenous regions, by connecting pixels in a range search based on saliency. Compared to SLIC, the proposed method provides a consistent performance improvement by increasing a low computational load, producing superpixels that capture more details, and reducing the redundancy of the represented information. However, it creates less regular and compact superpixels.

Semasuperpixel. The authors in (WANG et al., 2021) propose a superpixel segmentation algorithm that improves SLIC clustering with a new distance measure function considering SLIC-based distance and semantic information with a dynamic confidence value. The authors used a DeepLab v3+ (CHEN et al., 2018) network without any retraining to obtain semantic knowledge, and the rest of the algorithm follows the SLIC pipeline. The proposal clusters pixels based on semantic information and uses color and spatial information as refinement factors, achieving excellent boundary adherence and substantially reducing leakage. Therefore, the proposal improves SLIC performance and achieves competitive results with state-of-the-art superpixel methods.

AWkS. The proposed method in (GUPTA et al., 2021) adopts dynamic weighted distances based on weighted k-means clustering (W-k-means) (HUANG et al., 2005) and proposes an adaptative term for each variable in its distance formulation. The proposed *Adaptative W-k-means-based Superpixels* (AWks) extend SLIC (ACHANTA et al., 2012) to explore the degree of feature relevances during objective function minimization.

In general, AWks adopts a pipeline similar to SLIC, but its distance function considers the weighted distances for each feature or for each feature and cluster. The function is presented in Equation 4.4, where β is a user-defined parameter, F is the feature set, and w is an adaptative normalization weight vector. The adaptative weights w_A initialize with 0, updating at the end of each iteration.

$$D(p, q) = \sum_{A \in F} w_A^{\beta} d(p_A, q_A) \tag{4.4}$$

The authors also evaluated the proposal with color and spatial position features in a two-channel or five-channel way. AWks outperforms SLIC in boundary adherence and produces visually better segmentations, with more compact superpixels and fewer small ones close to the image boundaries. However, the proposal has a high running time compared to SLIC.

4.2 Boundary evolution clustering

In boundary evolution clustering, the algorithm iteratively updates the superpixels' boundaries to improve delineation, usually using a coarse-to-fine image block strategy. SEEDS (BERGH et al., 2012) and ETPS (YAO et al., 2015) are examples of superpixel methods using the boundary evolution strategy for clustering.

IBIS, IBIScuda. The *Iterative Boundaries implicit Identification for Superpixels* (IBIS) (BOBBIA et al., 2021) produces fast superpixel segmentation by implicitly identifying the boundaries between superpixels and using only a fraction of the input image pixels to perform segmentation. The paper also presents a GPU variant aimed at real-time use cases, the IBIScuda. The proposed method starts the segmentation with a grid segmentation and, according to the SLIC's distance measure (ACHANTA et al., 2012), compares the pixels located on the edge of the blocks, subdividing in 4 those blocks assigned to another superpixel. At each iteration, pixels in non-homogeneous blocks are assigned to the nearest superpixel according to the SLIC's distance measure. After the clustering step, IBIS performs the same merging stage as SLIC. The IBIS is much faster than other methods and achieves similar results as SLIC. Also, its Cuda version can even improve its efficiency, reducing computational time. However, similar to SLIC, the IBIS's boundary adherence, and accuracy are not competitive with other methods in the literature.

CFBS. The *Coarse-to-Fine Boundary Shift* (CFBS) (WU et al., 2020) algorithm aims to overcome the two major limitations of many methods based on k-means: redundancy and the need for post-processing. The post-processing procedure increases the execution time of the algorithm and reduces its accuracy since its major goal in most cases is to ensure connectivity. And redundancy is related to the iterative assignment of all pixels to superpixels. Since most superpixels undergo few changes over iterations close to their center, the iterative reassignment of all pixels consists of redundancy, reducing the algorithm's efficiency.

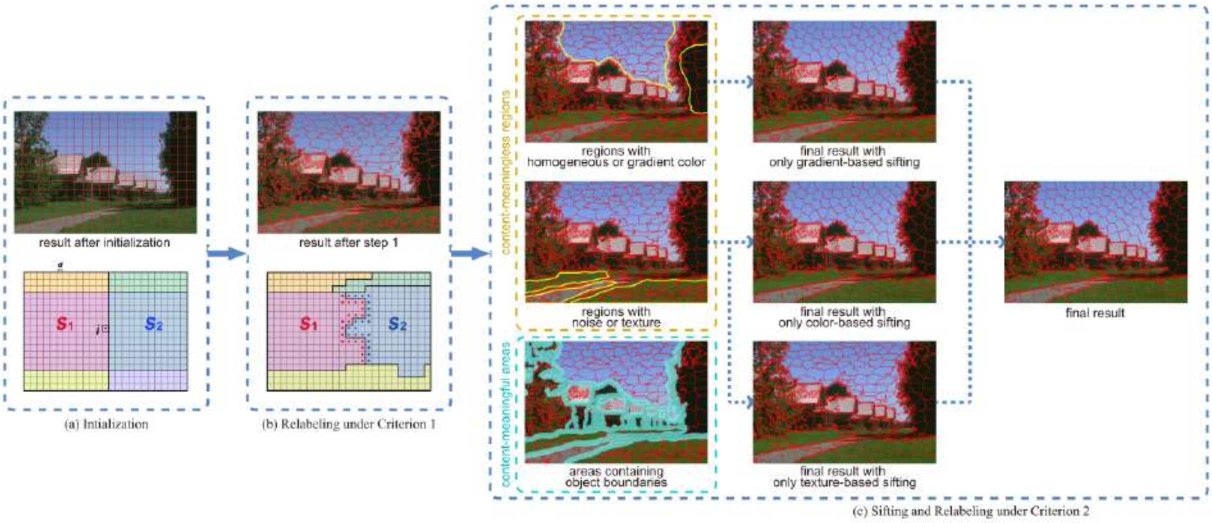
The CFBS proposal has the same pipeline as many iterative k-means-based methods. Similar to SEEDS (BERGH et al., 2012) and ETPS (SHEN et al., 2016), CFBS performs pixel block optimization in a coarse-to-fine manner using an optimization function similar to SLIC (ACHANTA et al., 2012).

The CFBS updates all pixel blocks in the superpixels' boundary, while the centers are updated dynamically. The number of iterations is defined by the maximum split operations of the initial block pixels. Through many experiments, the authors demonstrated the proposal's ability to increase the performance of k-means-based methods while reducing its running time, not only for superpixel segmentation but also for different applications. However, the CFBS segmentation does not capture finer details in more complex image regions, leading to a worse adherence to the image borders in these regions.

LSC-Manhattan. To improve LSC performance in segmentation accuracy and efficiency, the work in (QIAO; DI, 2022) proposes an adaptive subsampling method that improves the LSC (CHEN; LI; HUANG, 2017) performance with a distance measurement based on non-convex image features and Manhattan distance. The proposed LSC-Manhattan performs a subsampling strategy to label pixels according to texture complexity, such that different subsampling ratios are applied based on the image texture complexity levels.

Firstly, the proposal classifies the input image according to its texture complexity to determine the subsampling. For non-convex image structures, LSC has low accuracy. Therefore, the proposed method employs Manhattan distance instead of Euclidean to improve the accuracy and speed of computation. The authors perform a semantic segmentation using DeepLabV3+ (CHEN et al., 2018) to classify whether a pixel is part of some convex region. The LSC-Manhattan achieves competitive segmentation according to other superpixel methods, and it also produces better segmentation than LSC, with a reduced running time. However, the proposed distance measure is based on a specific dataset, which can lead to generalization issues.

Figure 4 – Diagram of SCAC



Source: Yuan et al., 2021b.

SCAC. In (YUAN et al., 2021b), the authors proposed a *Superpixel segmentation with Context-Adaptive Criteria* (SCAC) to differentiate regions with meaningful and meaningless content and, at the same time, avoid the tradeoff between compactness and accuracy. The SCAC identifies meaningless-content regions, produces more compact superpixels, and prioritizes accuracy in meaningful regions.

Figure 4 presents the SCAC diagram. From an initial grid pattern segmentation, SCAC performs an accuracy step followed by a compactness step. The accuracy step relabels the superpixel boundaries to maximize the adherence to the object contours according to balanced color weighted and spatial distances. Then, the compactness step performs a second relabelling based on color, gradient, and texture filters to detect content-meaningless regions. The gradient, color, and texture filters identify homogenous, noised, and similar texture pattern regions. For gradient and texture computation, the authors used the Sobel operator and WLD (CHEN et al., 2010) descriptors, respectively.

The SCAC is very competitive in delineation, with low degrading in compactness. Due to its high boundary adherence in meaningful regions, SCAC's compactness is median. Additionally, the proposal can run in real-time, but its runtime increases for a high number of superpixels. Also, SCAC provides limited control over the number of superpixels, producing a number similar to the desired.

4.3 Dynamic-center-update clustering

The dynamic-center-update algorithms perform clustering with a distance function based on the features of the clusters, dynamically updating its centers. Unlike neighborhood-based clustering, this approach does not perform a limited regional search to calculate distances.

FCSS. The cluster-based *Fine-to-Coarse Superpixel Segmentation* (FCSS) (LI et al., 2021) algorithm improves the segmentation by adding depth information to a SNIC-based algorithm (ACHANTA; SUSSTRUNK, 2017). Also, the FCSS uses a seed relocation process to solve the miss segmentation problem caused by the initial seed position and uses a priority queue to speed up the clustering process.

Considering a pixel as a six-dimensional vector of color, position, and depth, the FCSS starts including K cluster centers with distance zero in a priority queue. Then, iteratively remove from the queue pixel vectors based on their minimum distance, including them in a superpixel, updating its cluster center, and including in the queue the unlabeled neighboring pixels based on depth and color distances thresholds. When the queue is empty, it can perform a relocation to avoid unlabeled pixels by including uniformly distributed unlabeled pixels in the queue. At each relocation, new cluster centers are pushed to the queue, and the color threshold is updated. Finally, after the clustering step, the proposal merges the unconnected pixels to ensure connectivity. The FCSS is relatively fast, even with the addition of time complexity due to the seed relocation processing. Also, it achieves competitive results with a visually balanced segmentation between compactness and boundary adherence. However, the FCSS segmentation does not capture finer details in structure-rich regions, even reducing the compactness factor.

CONIC. A novel *Contour Optimized Non-Iterative Clustering* (CONIC) based on SNIC (ACHANTA; SUSSTRUNK, 2017) and SCALP (GIRAUD; TA; PAPADAKIS, 2018) is proposed in (GONG et al., 2021). The CONIC incorporates contour prior in a new distance measure, named joint color-spatial-contour measurement, which prevents the boundary pixels from being assigned prematurely.

The proposed distance function $D'(p, b_k)$ is presented in Equation 4.5, where p is a pixel, b_k is the barycenter of superpixel k, p_c and p_s are the color and spatial values of p, $C(b_k)$ and $P(b_k)$ are the color and spatial values of b_k, N_c and N_s are constants that represent the maximum color and spatial difference within the cluster k, and the contour weight N_b (Equation 4.6) adjust the feature distance from p to b_k using a constant ϵ that balances the influence of the contour prior on the feature distance.

$$D'(p, b_k) = N_b \left(\|p_c - C(b_k)\|_2^2 + \|p_s - P(b_k)\|_2^2 \cdot \left(\frac{N_c}{N_s} \right)^2 \right) \tag{4.5}$$

$$N_b = \exp(\epsilon \times c(p)) \tag{4.6}$$

The proposal achieves competitive performance against SNIC and SCALP, with moderate compactness and an improved F-measure and boundary precision. The CONIC also produces superpixels less sensitive to the gradient variation in textured regions with less boundary degradation. Compared to SNIC, CONIC avoids redundant feature distance computations having a faster execution than SNIC and SCALP. However, the contour prior fails to identify some weak image boundaries.

SCBP. A *Superpixel Based on Color and Boundary Probability* (SCBP) is proposed in (ZHANG; GUO; ZHANG, 2021). The SCBP is a two-stage non-iterative method based on DBSCAN (SHEN et al., 2016), which maintains DBSCAN's efficiency with more accurate superpixels. In the first stage, the proposal clusters the pixels in the conventional image order with an adaptative distance measure, processing each pixel only once and dynamically updating the cluster centers.

The adaptative distance measure (Equation 4.7) weights the spatial and color distances (d_s and d_c, respectively), balanced by a boundary probability term α computed with the Sobel operator. In Equation 4.7, $\alpha + \beta = 1$. The second stage merges superpixels based on a second distance measure 4.8 combining their size, $s(k_1, k_2)$ measures the relative size of the combined superpixel according to the expected size, and λ is a constant.

$$D_c(i, j, \overline{C}_k) = \beta \times d_c(i, j) + \alpha \times d_c(i, \overline{C}_k) \tag{4.7}$$

$$D(k_1, k_2) = s(k_1, k_2) \times d_c(k_1, k_2) + \lambda \times d_s(k_1, k_2) \tag{4.8}$$

The proposed method produces compact and regular superpixels in homogenous image regions and superpixels closer to the boundaries in complex regions. Therefore, SCBP has visually better segmentations than the compared methods and competitive results in boundary adherence evaluation but underperforms LSC (CHEN; LI; HUANG, 2017). In addition, the proposal has $O(n)$ time complexity, with a running time close to DBSCAN and faster than the others.

A-DBSCAN. A DBSCAN-based algorithm for low contour density superpixels and faster computation is proposed in (WANG; ZHANG, 2021). The proposed Adaptative DBSCAN (A-DBSCAN) adopts an adaptative threshold and uses a new distance measurement that constrains superpixel shapes based on the linear path from a pixel to a seed. The proposal also uses a local binary pattern operator (KE-CHEN et al., 2013) to compute texture and balance regularity and boundary adherence. After the clustering step, the A-DBSCAN performs a merging stage to produce final superpixels with regular size.

During the clustering stage, the distance $D(i, j, C_k)$ (Equation 4.9 measures the pixel-superpixel distance based on spatial distance $d_s(i, c_k)$ to the kth cluster center c_k, α and β parameters to balance the function terms, and an intensity term $C(i, j, C_k)$ 4.10 to constraint the homogeneity within the superpixels. The intensity term $C(i, j, C_k)$ is based on the color distance D_c, the cumulative color distance along the linear path $D_{LC}(i, c_k)$, and use λ and ψ as balancing parameters, where $\lambda + \psi = 1$.

$$D(i, j, C_k) = \alpha \times C(i, j, C_k) + \beta \times d_s(i, c_k) \tag{4.9}$$

$$C(i, j, C_k) = \lambda \times \frac{D_c(i, j) + D_{LC}(i, c_k)}{2} + \psi \times D_c i, c_k \tag{4.10}$$

The proposed method is faster than DBSCAN (SHEN et al., 2016) and produces fewer regular superpixels in textured regions, even with weak edges, achieving a more accurate delineation. It also has a competitive performance compared with the other superpixel methods.

F-DBSCAN. To improve the time complexity of the DBSCAN algorithm, the authors in (LOKE et al., 2021) proposed the Fast DBSCAN (F-DBSCAN). The proposal surpasses many drawbacks of the previous Real-Time DBSCAN (RT-DBSCAN) (GONG; SINNOTT; RIMBA, 2018) and parallelization issues. Instead of limiting the search range, the F-DBSCAN defines a limited number of points to assign for each superpixel. This minimizes the overlap and enables parallelization. The performance is also improved by maximizing the memory hints with large memory buffers, which eliminates fragmentation. After the clustering step, the F-DBSCAN merges small clusters using a watershed transformation (BEUCHER, 1992).

The proposal's segmentation presents similar qualitative results to RT-DBSCAN with much faster computation. The processing time for F-DBSCAN dropped as the degree of parallelism was increased without increasing leakage. However, F-DBSCAN presents a poor performance in images with blue-white boundaries and low contrast due to the CIELAB colorspace used. For GPU processing, the F-DBSCAN presents much slower results, owing to its regional parallelization instead of parallelizing a whole image.

DRW. In (KANG; ZHU; MING, 2020), the authors proposed a random walk-based superpixel method with improved adherence to image borders and low time complexity. The proposal, named *Dynamic Random Walk* (DRW), can be efficiently computed by a greedy strategy, and the proposed Weighted Random Walk Entropy (WRWE). The DRW model uses dynamic nodes, which reduces the redundant calculation by limiting the walking range. The superpixel segmentation algorithm proposed performs a new seed initialization strategy that creates a seed set with regular distribution in both 2D and 3D and can combine boundary prior information, such as gradient information or boundary probability (MARTIN; FOWLKES; MALIK, 2004).

The proposed model can be efficiently computed in linear time and allows control of the distribution of superpixels in the complex and homogenous image regions, such that adjusting for fewer superpixels in the complex regions, capture more fine details in these regions while producing bigger superpixels in the homogenous ones. The proposed segmentation method has a competitive performance with the state-of-the-art superpixel segmentation algorithms and it is faster than existing RW models. However, DRW segmentation does not produce compact superpixels.

4.4 Path-based clustering

Path-based approaches generate superpixels by creating paths in the image graph based on some criteria. Usually, its clustering criteria are a path-based function to optimize during clustering. The ISF (VARGAS-MUÑOZ et al., 2019) is an example of a path-based method that calculates a forest of optimal paths based on a path cost function.

RSS. In (CHAI, 2020), the authors propose a rooted spanning forest algorithm named *Root Spanning Superpixels* (RSS) that can be extended to supervoxels. To measure color similarity and spatial closeness, they also proposed two path-based cost functions, that have proven to be more robust than the geodesic distance. The Equations 4.11 and 4.12 present the proposed maximal difference and maximal range functions, respectively, over a path $\pi = \langle v_1, v_2, ..., v_p \rangle$.

$$f^d_\infty(\pi) = \max_{1 \leq i \leq p} \{\|I(v_1) - I(v_i)\|_\infty\} \tag{4.11}$$

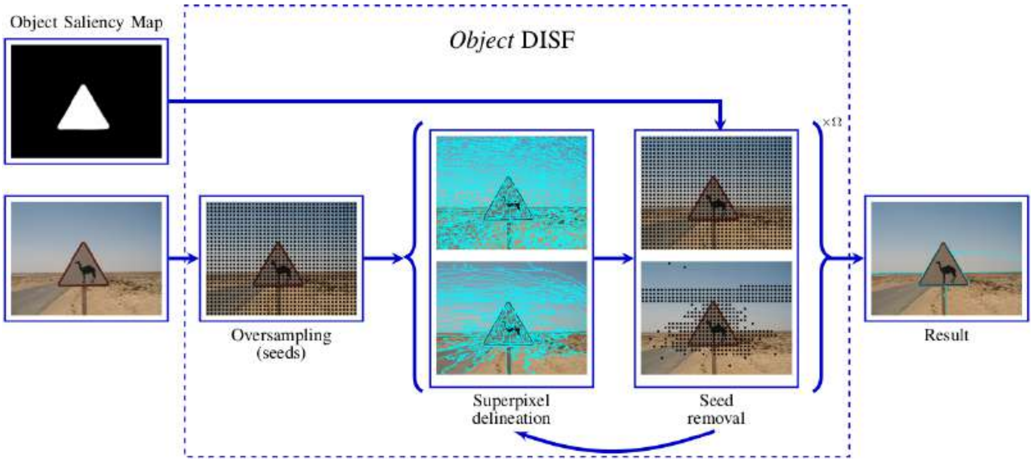
$$f^r_\infty(\pi) = \| \max_{1 \leq i \leq p} \{I(v_1)\} - \min_{1 \leq i \leq p} \{I(v_i)\} \|_\infty \tag{4.12}$$

The RSS method follows the IFT (FALCÃO; STOLFI; LOTUFO, 2004) algorithm and can form a forest with optimal costs. Inspired by counting sort and bucket sort, the RSS computes optimal forest with buckets of queues and groups of seeds in an IFT (FALCÃO; STOLFI; LOTUFO, 2004)-based algorithm. Due to the sorting strategy, the proposal has $O(N)$ complexity.

The RSS algorithm is fast and its performance is competitive to the compared superpixel methods. The main strengths of this proposal are the low computational complexity, great boundary adherence with stable performance, and adjustable compactness. However, besides the proposal extends to supervoxel segmentation, it performs poorly compared with the evaluated methods. Also, due to the initial seed sampling in a regular grid (ACHANTA et al., 2012), the RSS generates more superpixels in homogenous regions, which leads to a degrading in boundary adherence in complex regions.

DISF. Most superpixel methods have their performance rapidly degraded for reduced numbers of superpixels. Based on (VARGAS-MUÑOZ et al., 2019), the *Dynamic and Iterative Spanning Forest* (DISF) (BELÉM; GUIMARÃES; FALCÃO, 2020) is a three-

Figure 5 – ODISF segmentation diagram



Source: Belém et al., 2021.

step framework for superpixel segmentation that improves its delineation even for fewer superpixels. DISF initializes with a seed over-segmentation that performs grid sampling (ACHANTA et al., 2012) for a high number of seeds. Then, iteratively compute a forest rooted at the seeds with an IFT (FALCÃO; STOLFI; LOTUFO, 2004) execution with a further reduction of the seed set by choosing the most relevant ones. The IFT computation and seed set reduction are repeated until achieves the desired number of superpixels.

DISF outperforms all evaluated methods, especially for a few numbers of superpixels. Therefore, the proposal's segmentation is able to correctly selects relevant seeds, reducing its boundary adherence degradation when decreasing the number of final superpixels. Despite its iterative process increasing the running time, the DISF performs a reduced and limited number of iterations. However, ISF segmentation can reduce its running time by employing a differential (CONDORI et al., 2020) computation, which is impracticable for DISF due to its dynamic path-based cost function.

ODISF. Motivated by the *Object-based ISF* (OISF) (BELÉM; GUIMARÃES; FALCÃO, 2018) performance, the proposal in (BELÉM et al., 2021) extends the DISF (BELÉM; GUIMARÃES; FALCÃO, 2020) for an object-based proposal to improve the superpixel performance using object saliency maps. The proposal, named *Object-based DISF* (ODISF), performs a three-step pipeline (Figure 5), similar to DISF. First, the ODISF performs a seed oversampling step with some strategy (*e.g.*, GRID or random). Then, it performs a path-based superpixel generation followed by an object-based seed removal. In the superpixel generation step, the ODISF executes an IFT (FALCÃO; STOLFI; LOTUFO, 2004) algorithm with a dynamic path-based cost function (BRAGANTINI et al., 2018). In the remotion step, the algorithm maintains seeds closer to the object saliency boundaries or with higher saliency. To generate the object saliency maps, the authors used an U2-net (QIN et al., 2020). Finally, ODISF iteratively performs the generation and removal steps until reaching the desired number of superpixels.

The proposed method demonstrates a generalization ability by performing an effective superpixel segmentation in datasets with different object properties. The proposal also demonstrates robustness to saliency map errors in comparison with OISF. Despite the ODISF delineation step being saliency-independent, its object-based removal strategy can circumvent the saliency errors. On the other hand, the ODISF does not allow controlling the number of iterations. Also, despite its computational complexity, it has a high running time, being faster only than the OISF.

4.5 Hierarchical clustering

Hierarchical segmentation methods are generally not mentioned in the literature as superpixel methods. However, they fit most definitions for superpixels. Although hierarchical methods do not obtain a compact or regular segmentation, the regions produced are generally homogeneous. Furthermore, from the generated hierarchy, it is possible to control the desired number of regions without increasing the execution time.

HMLI-SLIC. A *Hierarchical and Multi-Level LI-SLIC* (HMLI-SLIC) algorithm is proposed in (DI et al., 2021) to improve segmentation accuracy and robustness to noise. The HMLI-SLIC consists of three steps: (i) initial segmentation, (ii) hierarchical multi-level segmentation, and (iii) superpixel merging. In the first step, the proposal produces a controlled number of superpixels with SLIC (ACHANTA et al., 2012) segmentation. Then, it performs coarse to fine segmentation with the SLIC algorithm to ensure that each superpixel does not contain multiple object regions, producing a hierarchical segmentation. Finally, it performs a merging step with the most similar superpixels.

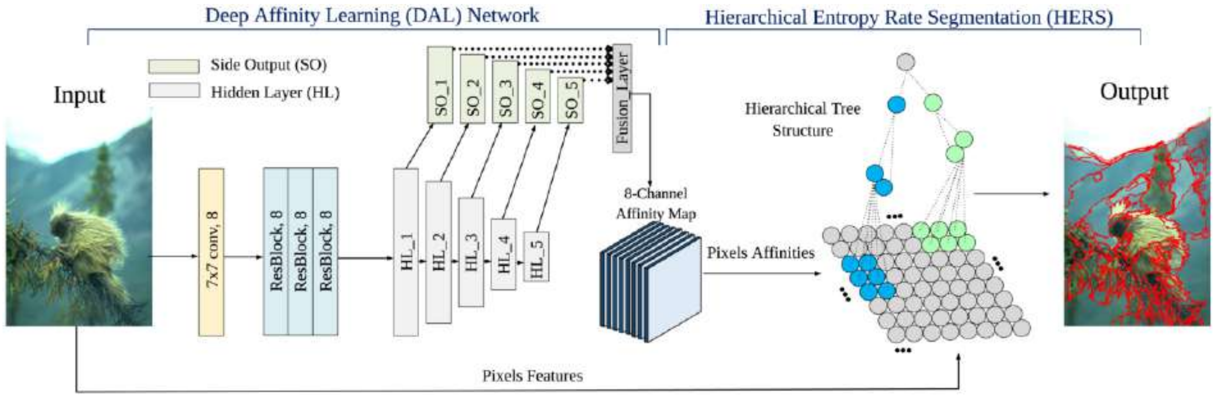
The proposal is robust to noise and can fit image boundaries since it produces more superpixels in heterogeneous regions and less in homogenous ones. In addition, the HMLI-SLIC does not perform under- or over-segmentation, setting the number of seeds and superpixels automatically. Therefore, it does not allow controlling the number of superpixels. However, compared with other superpixels methods, the proposal has high time-consuming and does not produce regular or compact superpixels.

RISF. In a previous work (GALVÃO; FALCÃO; CHOWDHURY, 2018), the authors present a hierarchical segmentation algorithm based on ISF. The proposal in (GALVÃO; FALCÃO; CHOWDHURY, 2018) is limited to two observation scales, being incapable of producing an entire hierarchy. In (GALVÃO; GUIMARÃES; FALCÃO, 2020), the authors surpass this drawback by proposing a *Recursive Iterative Spanning Forest* (RISF) to hierarchical segmentation using a region merging algorithm as post-processing. RISF produces a sparse hierarchy by computing a multi-scale superpixel segmentation using ISF over the Region Adjacency Graph (RAG) resulting from the previous scale. The region merging algorithm produces a dense hierarchy from a mid-level superpixel segmentation for more accurate segmentation in coarser scales.

The proposal produces more irregular superpixels than ISF, but the quantitative and qualitative results demonstrate that RISF has better segmentation than the state-of-the-art superpixel methods. Compared to dense hierarchical superpixel methods, RISF's performance is competitive. The RISF can produce a hierarchy from any superpixel segmentation method and has a low complexity of $O(n \log n)$, being faster than most of the evaluated methods due to its computation over RAGs. However, due to the hierarchy construction, errors in coarser scales are propagated to the finer ones.

UOIFT. The proposal in (BEJAR; GUIMARÃES; MIRANDA, 2020) extends (BEJAR; MANSILLA; MIRANDA, 2018) to propose a hierarchical and unsupervised image segmentation method that exploits non-monotonic-incremental cost functions in directed graphs to incorporate high-level priors of the objects as boundary polarity. The proposal, named *Unsupervised OIFT* (UOIFT), computes an initial forest over the image pixels and partitions the graph with multiple executions of the OIFT (MANSILLA; MIRANDA, 2013;

Figure 6 – The DAL-HERS framework



Source: Peng, Aviles-Rivero and Schönlieb, 2022.

MIRANDA; MANSILLA, 2013) computed over the Region Adjacency Graph (RAG) of the previous forest.

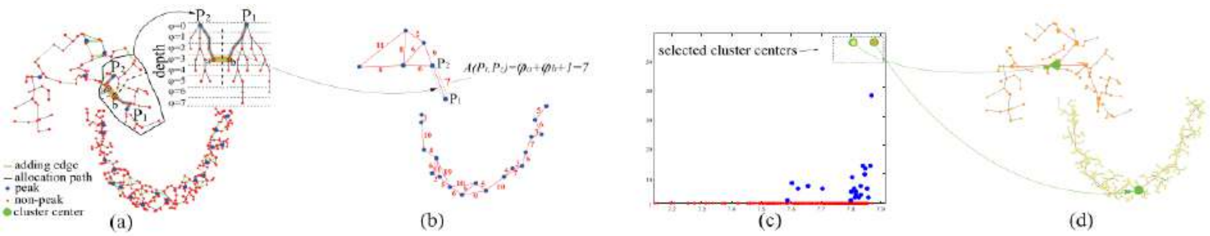
The UOIFT is fast and demonstrated its ability to accurately segment medical images and colored images with different lighting conditions, outperforming all hierarchical image segmentation methods compared. Besides its boundary polarity allowing improved segmentation for a specific color (or texture or local contrast) transition, setting this parameter can be challenging, for more generic applications.

DAL-HERS. The DAL-HERS, a two-stage superpixel framework is proposed in (PENG; AVILES-RIVERO; SCHÖNLIEB, 2022), which consists of a *Deep Affinity Learning* (DAL) neural network architecture and a *Hierarchical Entropy Rate Segmentation* (HERS) method. As shown in Figure 6, the DAL network learns an 8-channel pixel affinity map (left), used by the HERS algorithm to construct a hierarchical tree structure.

The DAL network aggregates multi-scale information to learn pairwise pixel affinities, and the HERS builds a hierarchical tree structure by maximizing the graph's entropy rate. The proposed DAL network consists of two parts: (i) a convolutional residual model based on (HE et al., 2016) to learn intermediate pixels affinities without too much additional computational cost and (ii) a HED (XIE; TU, 2015) model to capture neighborhood information from the intermediate pixels affinities at varying scales with increasing receptive field sizes. Using the DAL's affinity map, the proposed HERS algorithm constructs a hierarchy with Borůvka's algorithm (WEI et al., 2018) based on the entropy rate of the graph.

The DAL-HERS method preserves fine details on the objects by focusing on rich-structure parts rather than uniform regions, producing large superpixels in homogeneous regions and an over-segmentation in texture-rich regions. Therefore, it captures semantically homogenous regions and highly adheres to the object boundaries. Also, compared with deep-based learning methods, the DAL-HERS running time is competitive, and it requires the same $O(N)$ time to produce any number of superpixels. Due to the highly adaptive nature of the produced superpixels, delineating fine details, the proposal has a low ASA score. The authors mitigate this problem by incorporating edge information, but the variant suffers from BR and EV degradation.

Figure 7 – PGDPC segmentation procedure



Source: Guan et al., 2021.

4.6 Density-based clustering

In the density-based clustering approach, the superpixel methods rely on an optimization function to find the cluster centers, calling them density peaks. The clustering of the non-peak pixels is performed according to the centers, generally assigning a pixel to the superpixels with the spatially closest density peak. Therefore, such methods model the problem of finding superpixels in a problem of finding density peaks.

PGDPC. In (GUAN et al., 2021), authors proposed a fast density peak clustering method, the *Peak-Graph-based fast Density Peak Clustering* (PGDPC), based on DPC (WANG; WEI; TSE, 2018) to improve its time complexity and accuracy. The proposal performs a two-step strategy, firstly dividing data points into peaks and non-peaks and computing a graph using DPC-based allocation. Then, the peaks candidates are assigned along geodesic paths in a peak graph.

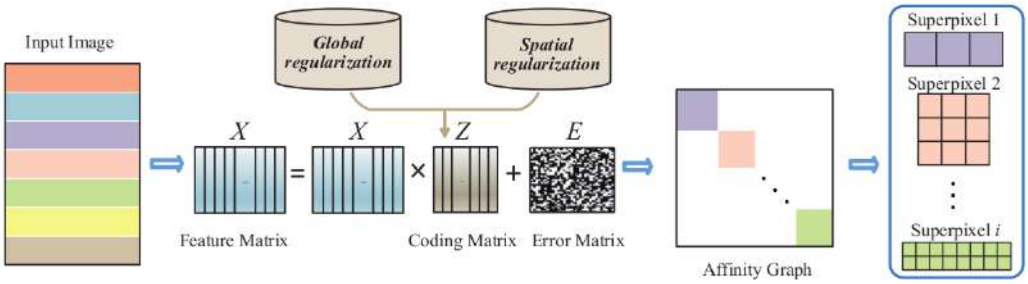
The PGDPC initializes computing the KNN density (XIE et al., 2016) for each pixel and classifies them as peak candidates and non-peaks. Then, compute a graph based on a DPC allocation, and initializes a peak graph with peak candidates as roots (Figure 7 (a)). The non-peak nodes are assigned to the closest root cluster, forming trees whose edges' weights are the minimum path distance between pixels, called a geodesic path (Figure 7 (b)). The trees are connected by adding edges between neighbors' pixels from different trees whose edge's weight is an association distance, as the sum of the geodesic distance of these nodes. Finally, by the centers of the clusters as candidate peaks with higher density than their neighbors and high geodesic distance for the highest density peaks (Figure 7(c)), the clustering step is complete (Figure 7(d)).

The proposal is faster than the compared methods, having an $O(n \log n)$ time complexity. In synthetic datasets, PGDPC demonstrates its ability to cluster complex structures using the proposed peak graph methodology, achieving an improved performance compared with DPC. To evaluate the proposal, the authors combined PGDPC and SLIC (ACHANTA et al., 2012) to reduce the computational cost for natural images, and they used image compressing for medical images. The PGDPC achieves competitive performance in natural and medical datasets. However, using SLIC as pre-processing, the SLIC errors can be propagated to PGDPC, reducing its performance.

DPS. Density peak-based algorithms often achieve excellent performance, but they can have a high computational cost (RODRIGUEZ; LAIO, 2014), due to their high search range for density peaks. In (SHAH et al., 2021), the authors propose a *Density Peaks Superpixel* (DPS) algorithm to perform an efficient non-iterative density peak segmentation in a limited search region.

The DPS initializes computing the pixels' density ρ (Equation 4.13) and finds the density of peaks δ (Equation 4.14), searching in a limited region Ω_i, where d_c is a threshold

Figure 8 – SCSC diagram



Source: Li et al., 2020.

distance and d_{ij} measures the distance between pixel i and j. Then, the superpixels' centers are founded based on pixel density threshold T_1 and peak density T_2 threshold. Finally, it assigns the remaining pixels to their nearest superpixel with a higher density. Due to the regional search, its time complexity is $O(m^2)$, where $m \times m$ is the region size.

$$\rho_i = \sum_{j \in \Omega_i} e^{-(\frac{d_{ij}}{d_c})^2} \tag{4.13}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{4.14}$$

The proposed DPS is faster than Density Peak (RODRIGUEZ; LAIO, 2014) and has a competitive segmentation, even using only color and spatial distances in a single iteration to compute the superpixels. However, the DPS does not produce regular and compact superpixels. Also, its control over the number of superpixels is indirect and based on two parameters.

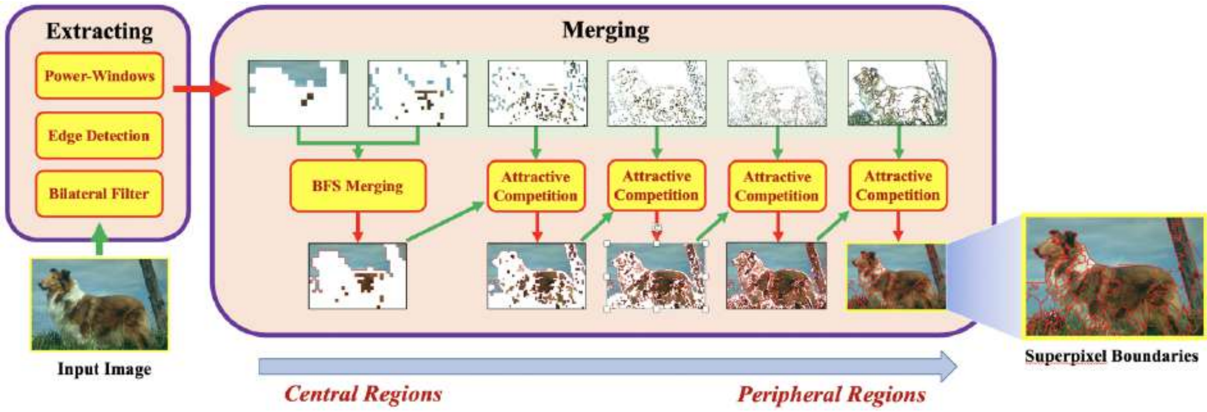
4.7 Sparse linear system clustering

ANRW. Random walk-based superpixel methods (GRADY, 2006) have improved performance in textured images and weak borders. However, these methods usually are sensitive to their initial seed points. To overcome this issue, (WANG et al., 2020) propose the *Adaptive Nonlocal Random Walk* (ANRW), which performs seed initialization based on the regional minima with a trade-off between local contrast and spatial distance.

The ANRW is based on the nonlocal random walk (NRW) (YUAN et al., 2015), which uses K-Nearest Neighbor (KNN) linking points to weight pixel features exploring local relationships. From the generated seed set, the ANRW computes the weight matrix of the pixels according to an adaptative gaussian function and the KNN features. The authors propose an adaptative method to choose the number of KNN-linking points according to the size of the superpixel. Also, the proposal computes the KNN weights with the Fast Library for Approximate Nearest Neighbors in (VEDALDI; FULKERSON, 2010).

From the adaptative gaussian and KNN features, the proposed method computes a Laplacian matrix and solves the Dirichlet problem to achieve the pixels' probabilities and assign the labels according to it. The KNN link may produce many small parts in the superpixel segmentation. Therefore, after the clustering step, the ANRW performs a coarse-to-fine merging strategy. The ANRW can deal with textured images, outperforming the compared methods in boundary recall, under-segmentation error, and accuracy, but the proposal has high computational complexity. Although the ANRW doesn't produce compact superpixels in complex regions, it does in homogeneous ones.

Figure 9 – The EAM diagram



Source: An et al., 2020.

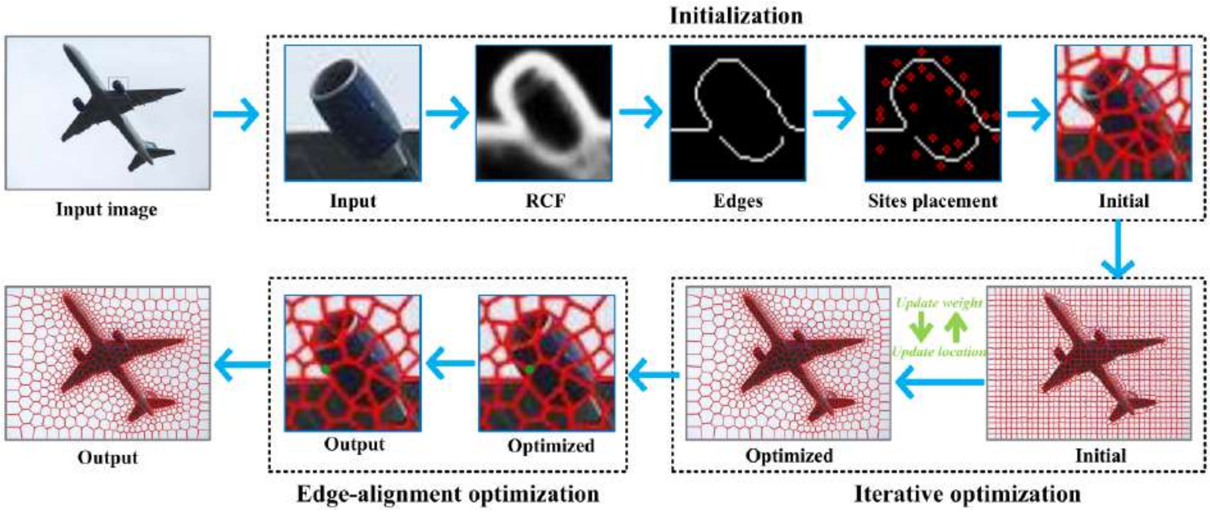
GL$l_{1/2}$RSC. The authors in (FRANCIS; BABURAJ; GEORGE, 2022) propose a noise-robust image segmentation algorithm based on subspace clustering with enhanced segmentation capability using Laplacian and $l_{1/2}$ regularization techniques. The proposed *Graph laplacian $l_{1/2}$ regularized subspace clustering* (GL$l_{1/2}$RSC) method addresses the challenge of obtaining an improved sparse solution or a sparse representation matrix under the circumstances of noise-corrupted feature data vectors.

The $\text{GLl}_{1/2}$RSC starts with an initial superpixel segmentation (LU et al., 2012) and computes its Local Spectral Histogram (LSH) features to obtain a feature data matrix. Then, perform a spectral clustering on the matrix to obtain clustered data points and execute an encoding procedure to map the clustered superpixels into optimal regions (WANG; WU, 2017) (ZOHRIZADEH; KHEIRANDISHFARD; KAMANGAR, 2018). The $\text{GLl}_{1/2}$RSC segmentation preserves the image structures, producing better results for images with a large number of small dominant regions. However, similar to other subspace clustering methods, the proposal has a high running time due to the LSH feature vector generation.

SCSC. The authors in (LI et al., 2020) proposed a *Spatially Constrained Subspace Clustering* (SCSC) algorithm capable of capturing detailed regions without significantly increasing the number of superpixels. The SCSC formulates the superpixels problem as a subspace clustering problem. The SCSC diagram is shown in Figure 8. The proposed method first performs a K-means clustering. Then, it constructs a coding matrix using the superpixel-based feature vectors and solves the matrix with an algorithm based on the alternating direction method of multipliers (ADMM) (BOYD et al., 2011). The superpixel-based feature vectors chosen are mean gradient and color, the superpixel's center, and texture (SILVA; BOUWMANS; FRÉLICOT, 2015). Finally, the SCSC computes the affinity graph and performs an NCut segmentation (SHI; MALIK, 2000) with a further merging step to guarantee connectivity.

The method can achieve good performance even with a low number of superpixels. The SCSC outperforms the classic methods evaluated and is comparable to the deep learning ones. Its quantitative results produce more flat plots, meaning that its performance doesn't decrease so fast as the other methods when the number of superpixels decreases. Also, the superpixels generated are capable of capturing finer boundary details. However, the proposal may not produce regular or compact superpixels. Also, its execution can take many seconds to generate hundreds of superpixels.

Figure 10 – The ECCPD diagram



Source: Ma et al., 2020.

4.8 Regional attributes extraction

EAM. The *Extract and Merging* (EAM) (AN et al., 2020) algorithm computes superpixels based on regional attributes. The proposal extracts multi-level attributes using squared windows of various sizes, called power-windows. Then, it performs clustering based on the regional attributes.

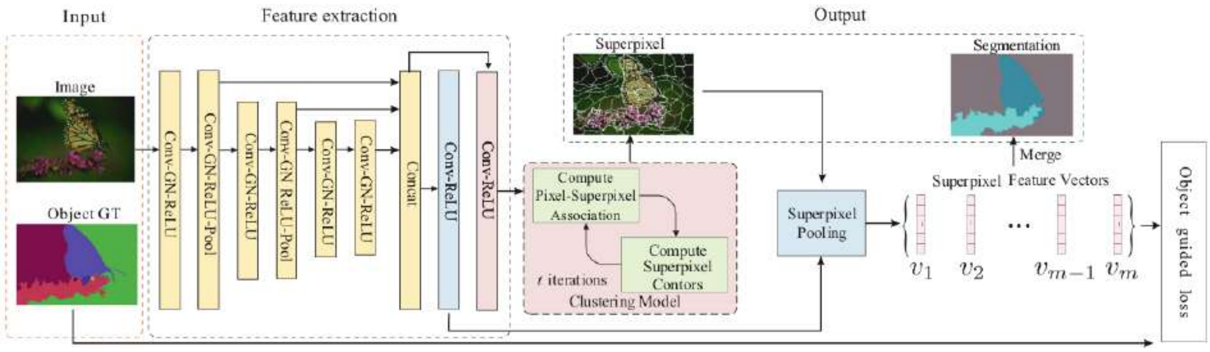
The EAM pipeline is shown in Figure 9, where the green arrows in the figure indicate inputs and the red arrow indicates outputs. Firstly, the EAM removes noise from the input image with a bilateral filtering (TOMASI; MANDUCHI, 1998) and extracts regional attributes using power-windows by computing its boundary clearness, which determines whether a power-window contains a single object (Figure 9 on left). The power-windows with more than one object are iteratively split into four until achieving a minimum size. Next, the EAM performs a merging step (Figure 9 on right) computing a Dijkstra (DIJKSTRA et al., 1959) algorithm to merge similar power-windows, followed by a binary search algorithm to merge them with unreached windows. To reach a more appropriate number of superpixels, the authors use the cluster diameter threshold to control the degree of detail of segmentation.

The EAM is relatively fast and generates larger and fewer superpixels in homogenous regions, capturing more details in complex regions. The proposal's ability to capture more homogenous superpixels, with a superior adherence to the image boundaries, with superior performance in Explained Variation. Also, it achieves a highly competitive accuracy with results closer to deep-based approaches. However, the EAM's superpixels were neither compact nor regular. Also, its running time is not competitive with other unsupervised methods with linear time complexity.

4.9 Polygonal decomposition clustering

ECCPD. In (MA et al., 2020), the authors propose a polygonal decomposition method to generate compact and convex superpixels while adhering to the image boundaries. The proposed *Edge-Constrained Centroidal Power Diagram* (ECCPD) algorithm formulates the superpixel generation problem into a Centroidal Power Diagram (CPD) (AUREN-

Figure 11 – E2E-SIS diagram



Source: Wang, Li and Zhang, 2020.

HAMMER, 1987) problem.

The proposed method shown in Figure 10, initializes superpixels with fixed cluster centers according to the CPD and random centers equally spaced. Then, iteratively adapt the power cell sizes according to the superpixel's distance from the edges, and the similarity between adjacent superpixels. The power cell centers (cluster centers) are also iteratively updated according to their centroid. After performing a number of iterations or achieving a certain threshold, run post-processing to align some boundaries.

Compared with other polygonal superpixel methods, the ECCPD can capture better boundaries in more complex regions. Also, the generated superpixels capture more information in these regions, and the proposal is faster than other strategies to compute the CPD with capacity constraints in geometry. However, ECCPD is very slow compared with some evaluated methods, and it has no competitive performance in boundary recall, under segmentation error or accuracy.

4.10 Data distribution-based clustering

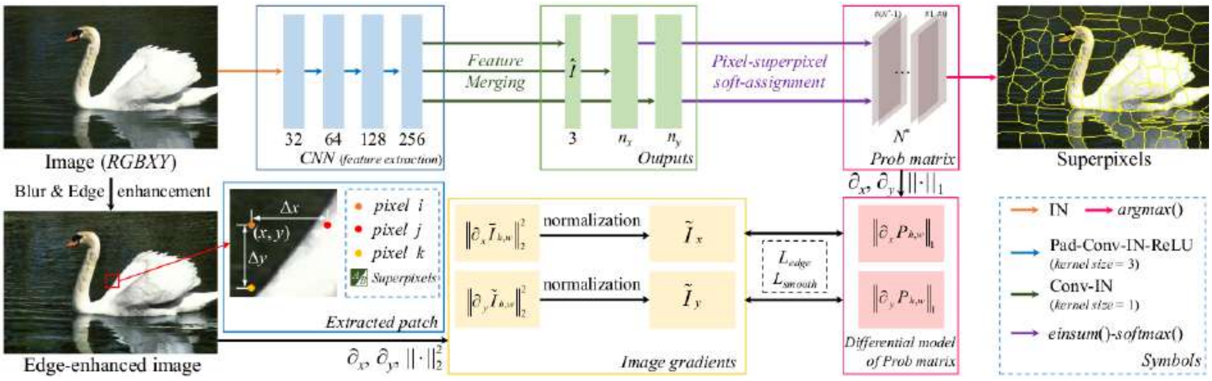
In superpixel segmentation, we name data distribution-based methods the approaches that assume that the image pixels follow a specific distribution. From this initial conjecture, the clustering step is performed. As far as we know, the distribution-based methods that perform superpixel segmentation are based on the gaussian mixture model and assume that the image pixels follow a Gaussian distribution.

gGMMSP. To explore the parallelism in GMMSP (BAN; LIU; CAO, 2018), a real-time solution without the loss of segmentation consistency is proposed in (BAN; LIU; FOURIAUX, 2020). The proposed gGMMSP is implemented on CUDA for GPU processing and gives very similar segmentation results as GMMSP with much faster computation. The proposal maintains the core of the GMMSP algorithm, adapting its data structures and arithmetic computations to perform GPU processing. The GMMSP and gGMMSP require post-processing to ensure connectivity. However, this step has data dependencies that prevent parallel computing, reducing the speedup's proposal. Even with post-processing, the gGMMSP is faster than the serial and openMP versions of GMMSP, achieving speedups of 92.6 and 27.5, respectively.

4.11 Deep learning-based methods with a simple FCN architecture

E2E-SIS. In (WANG; LI; ZHANG, 2020) authors proposed a deep learning-based framework for superpixel and image segmentation. The proposal uses an end-to-end trainable

Figure 12 – Edge-Aware RIM (EW-RIM) diagram



Source: Yu, Yang and Liu, 2021.

CNN that learns deep features with two final layers, one for superpixels and the other for image segmentation. The diagram for E2E-SIS is presented in Figue 11.

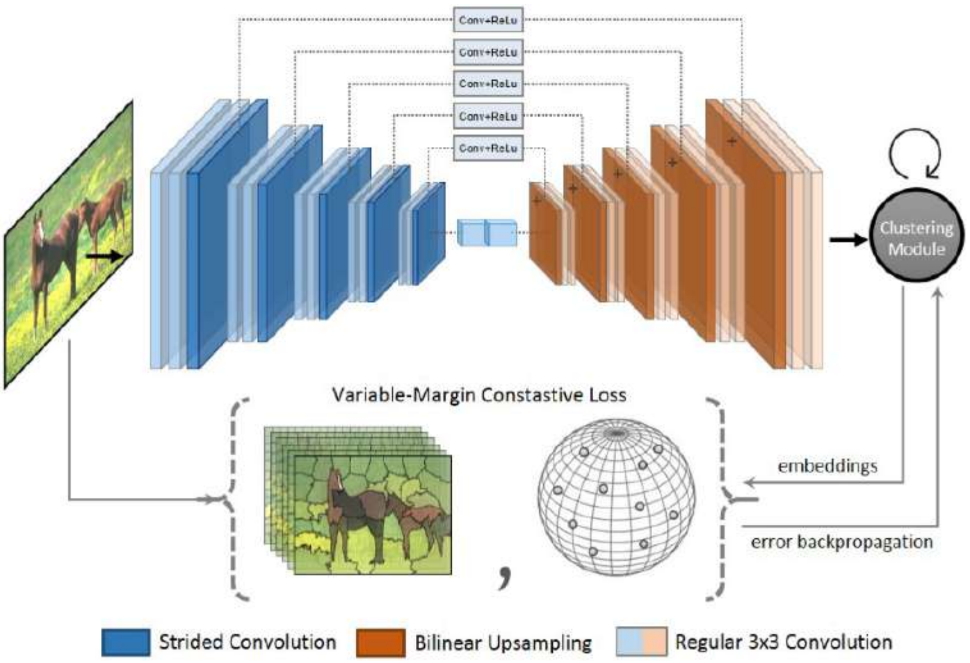
For superpixel segmentation, the deep features from the final layer of the CNN fed a differentiable clustering algorithm module (JAMPANI et al., 2018), generating the final superpixel segmentation. The superpixel results and the deep features from the second final CNN layer are used by a superpixel pooling (KWAK; HONG; HAN, 2017) to learn semantic similarities between superpixels. The final image segmentation is achieved by merging superpixels with high similarities. The E2E-SIS has a high ability to segment superpixels and to perform image segmentation, both with competitive results. Since the proposal is end-to-end trainable, it can be integrated into other deep learning-based methods. However, compared to other superpixel segmentation methods, the proposed framework has a high computational time.

ss-RIM. Based on the idea that low-level features are insufficient to improve segmentation with few superpixels, the authors in (SUZUKI, 2020) induce non-local properties into an unsupervised CNN-based method. The proposal uses the Deep Image Prior (DIP) (LEMPITSKY; VEDALDI; ULYANOV, 2018) procedure to generate task-agnostic superpixels with a new loss function based on clustering, spatial smoothness, and reconstruction.

The clustering term is similar to the mutual information term of RIM (KRAUSE; PERONA; GOMES, 2010), and the spatial smoothness cost is the same as proposed in (GODARD; AODHA; BROSTOW, 2017). Finally, the reconstruction cost helps the loss function fit the superpixels at the components' boundaries. The proposal can generate regular superpixels in homogenous regions and also outperforms the other superpixel methods in ASA, especially for a few superpixels. For the Boundary Recall metric, the proposal also outperforms the other methods, but only for a low number of superpixels. According to the results, the proposed method is able to generate superpixels more attached to the image boundaries, especially in heterogeneous regions. However, the ss-RIM only allows control of the upper bound number of superpixels and does not ensure connectivity.

EW-RIM. An edge-aware term for a deep learning-based superpixel algorithm based on SS-RIM (SUZUKI, 2020) and DIP (LEMPITSKY; VEDALDI; ULYANOV, 2018) is proposed in (YU; YANG; LIU, 2021) to improve boundary adherence using image gradient. The proposal, named *Edge-Aware RIM* (EW-RIM), encompasses a loss function composed of four terms based on clustering (KRAUSE; PERONA; GOMES, 2010),

Figure 13 – Superpixel Embedding Network (SEN) diagram



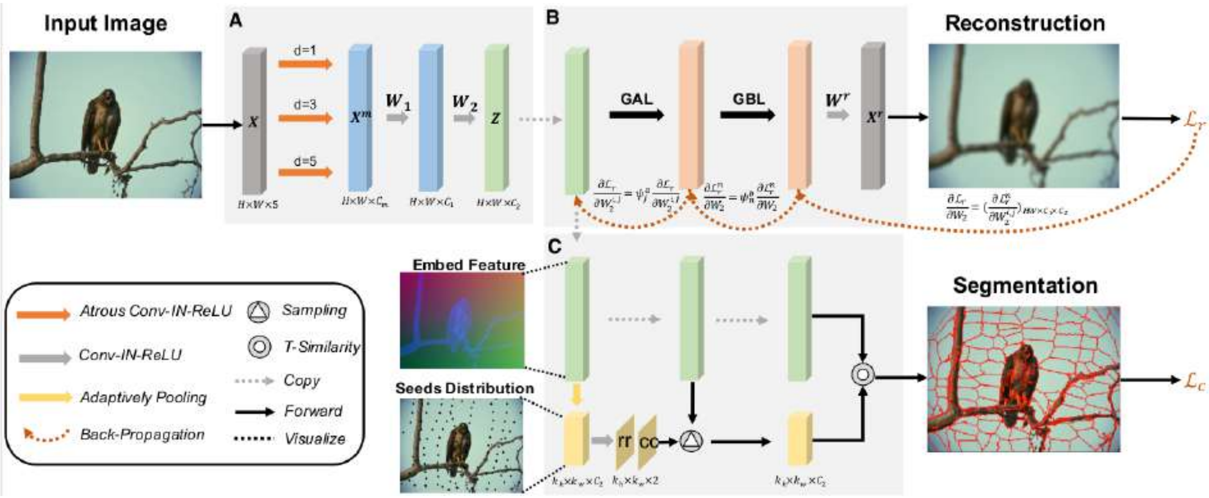
Source: Gaur and Manjunath, 2020.

smooth (SUZUKI, 2020), reconstruction (GODARD; AODHA; BROSTOW, 2017), and edge-aware. The edge-aware accomplishes a differential approximation to the distribution of image gradients. The proposed model enhances the image edges with filters provided in (YIN; GONG; QIU, 2019) and the Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve the smooth and edge-aware loss terms.

Figure 12 shows the EW-RIM diagram. Using RGB color and spatial information as input, the EW-RIM extracts feature information from a CNN architecture (blue layers in Figure 12) with a feature merging step (green layers), to obtain the association probability maps (purple layers in Figure 12). For loss computation, an edge-enhanced image is applied to obtain the edge loss L_{edge} and smooth loss L_{smooth}. The proposed edge-aware term improves the boundary adherence of the proposed EW-RIM compared with its baseline and outperforms all compared methods, also producing compact superpixels. However, since the proposal's segmentation generates more similar superpixels in size, it does not preserve finer details in complex regions.

LNSNet. In (ZHU et al., 2021), the authors proposed an unsupervised CNN-based superpixel segmentation method, called LNSNet, to learn superpixels in a lifelong manner. The LNSNet (Figure 14) is composed of three major modules: a feature embedder module (FEM) (Figure 14(A)), a gradient rescaling module (GRM) (Figure 14(B)), and a non-iterative clustering module (NCM) (Figure 14(C)). The FEM embeds the original feature into a cluster-friendly space. The NCM uses the embedded features in a seed estimation layer (SEL) to estimate the optimal cluster centers and assigns labels to each pixel based on similarity using the clustering layer (CL). Finally, the GRM solves the forgetting caused by lifelong learning during the backward step using a Gradient Adaptive Layer (GAL) and a Gradient Bi-direction Layer (GBL). While the GAL manages the importance of different feature channels to avoid overfitting, The GBL generates confrontation based on the spatial context to improve generalization.

Figure 14 – LNSNet diagram



Source: Zhu et al., 2021.

The experiments showed that the proposed method has a high generalization capacity. In addition, it was able to obtain better quantitative results on the BSDS500 and DME datasets and presented competitive results on the DRIVE dataset. Therefore, the LNSNet generates improved or more competitive superpixels using less complex and computationally faster architecture and without ground truth data. However, the proposal has some drawbacks. Firstly, the proposed model cannot reach a complete convergence, due to the sequential training strategy, requiring post-processing to remove trivial regions. Secondly, GBL's boundary map may contain noises and lead to irregular superpixels when facing a background with a complex texture. Finally, the clustering step requires a distance matrix with an NxK dimension, which is inefficient when calculated by a CPU with a large K.

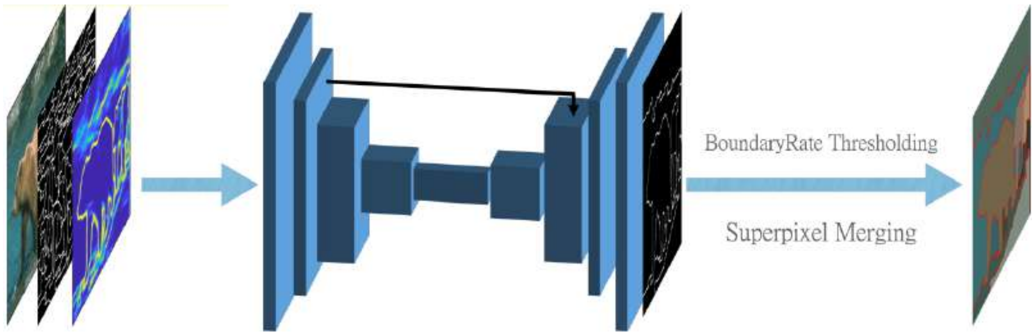
4.12 Deep learning-based methods with an encoder-decoder architecture

SEN. Inspired by (KONG; FOWLKES, 2018), the authors in (GAUR; MANJUNATH, 2020) proposed the *Superpixel Embedding Network* (SEN), an unsupervised deep network method that learns deep embeddings for superpixel segmentation. The SEN diagram is shown in Figure 13. To learn the pixel embeddings, they used the U-net (RONNEBERGER; FISCHER; BROX, 2015) architecture with a differentiable Mean-Shift recurrent clustering based on (KONG; FOWLKES, 2018) for density estimation.

The differentiable clustering module considers the global context, preventing embeddings from being labeled to optimize local distances. To train the SEN architecture, the authors proposed a variable-margin contrastive loss that compares the embedding distances with a superpixel segmentation generated with a randomly selected scale and detail attribute. The proposal is end-to-end trainable and uses superpixel segmentation maps as a pseudo-ground-truth label to learn a new manifold whose feature distances act as a proxy for semantic similarity. The superpixel segmentations for the proposed loss function are generated by the SNIC (ACHANTA; SUSSTRUNK, 2017).

SEN network significantly outperforms the other superpixel segmentation methods in boundary F1 score, but its UE and ASA results are barely competitive with the other methods. Therefore, SEN presents a better boundary adherence, but with larger leakage

Figure 15 – DMMSS-FCN diagram



Source: Huang and Ding, 2020.

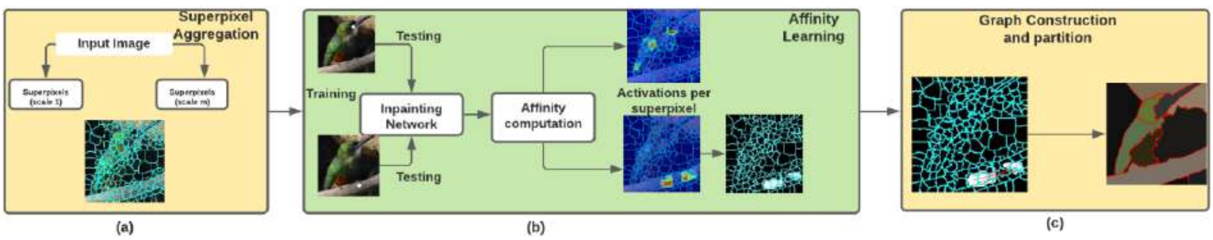
regions. The proposal also produces more superpixels in homogenous image regions, missing some image boundaries in complex regions.

DMMSS-FCN. In (HUANG; DING, 2020), the authors propose a superpixel segmentation framework that treats the segmentation problem as multiple merging decision problems. The proposal, named *Deep Merging Model for Superpixel Segmentation by Fully Convolutional Networks* (DMMSS-FCN), contains a fully convolutional network and a superpixel merging algorithm (Figure 15).

From an RGB image, a superpixel boundary map, and an edge-detection map, the proposal's network decides whether remove each image edge and outputs an edge removal map. The authors compute the edge-detection map using Refine Contour Net (RCN) (KELM; RAO; ZÖLZER, 2019) and the superpixel boundary map from the SEAL (TU et al., 2018) and SSN (JAMPANI et al., 2018) segmentations. For the network architecture, the proposal uses a DeepLabV3+ (CHEN et al., 2017) with Inception-ResNetV2 (SZEGEDY et al., 2017) as the encoder. The superpixel merging algorithm dynamically thresholds the edge removal map and uses majority voting for a superpixel pair. The algorithm merges based on the boundary removal rate until it achieves a final threshold.

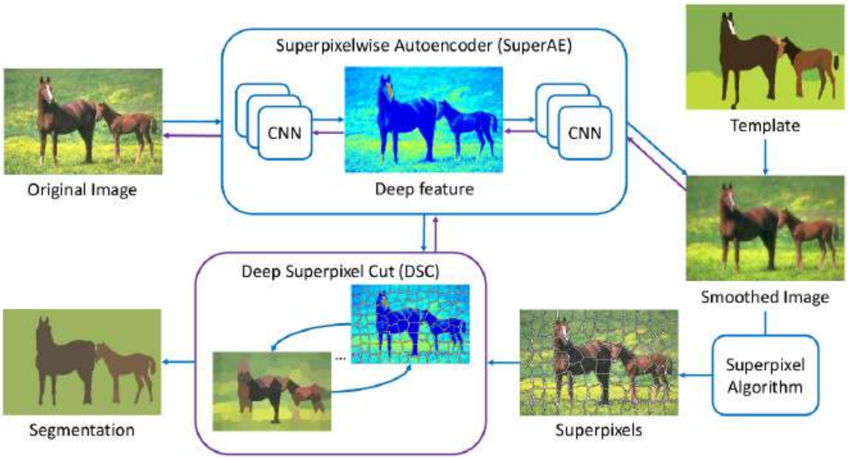
The DMMSS-FCN captures complex structures, performing a more accurate segmentation with fewer small regions. Also, its running time is faster than some evaluated methods. However, for real-time computation, the DMMSS-FCN requires a GPU for the network processing with further CPU processing for the merging algorithm. In addition, the DMMSS-FCN segmentation results fail to capture finer image details, and the final number of superpixels is not controllable.

Figure 16 – UDAG diagram



Source: Bhugra et al., 2021.

Figure 17 – SuperAE-DSC algorithm



Source: Lin, Zhong and Lu, 2021.

UDAG. Motivated by the GL graph (WANG et al., 2015), the authors in (BHUGRA et al., 2021) proposes using unsupervised Deep Learning Affinities in a *Graph-based segmentation* (UDAG) to perform segmentation based on contextual information. Based on the idea that inpainting networks can learn pixel association, the proposal uses the contextual layer (YU et al., 2018) information learned by an inpainting network (PATHAK et al., 2016) to generate an affinity matrix.

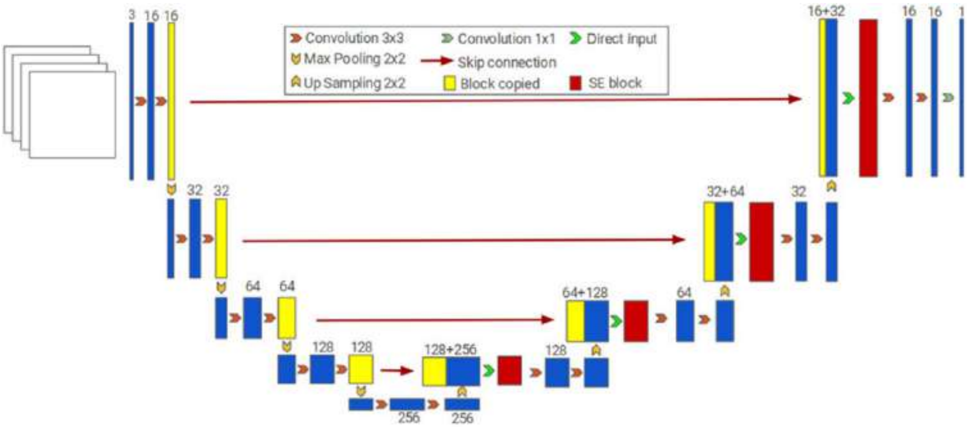
In the first step, the authors perform a superpixel aggregation, following the strategy in (LI; WU; CHANG, 2012), generating superpixels using different methods (FELZENSZWALB; HUTTENLOCHER, 2004) (COMANICIU; MEER, 2002) (Figure 16 (a)). Then, for every superpixel obtained in the decomposition, compute inpainting results (PATHAK et al., 2016) and utilize the features in the contextual layer (YU et al., 2018) to extract similar regions for every pixel in a superpixel (Figure 16 (b)). Using the similarity score, they construct a bipartite graph structure and generate the final superpixels with SAS (LI; WU; CHANG, 2012) algorithm (Figure 16 (c)).

The proposed method demonstrates that the affinity graph learned from the contextual layer (YU et al., 2018) in an inpainting network (PATHAK et al., 2016) provides useful information for superpixel segmentation. Also, the UDAG algorithm can produce fewer superpixels compared to other unsupervised methods, and their superpixel boundaries are similar to semi-supervised approaches. However, the proposal's results do not outperform most of the evaluated methods in the qualitative assessment and the number of superpixels is not controllable.

SuperAE-DSC. The SuperAE-DSC (LIN; ZHONG; LU, 2021) is a deep learning method for superpixel segmentation which consists of a *Superpixel-wise Autoencoder* (SuperAE) and a *Deep Superpixel Cut* (DSC) algorithm. The SuperAE takes the original image and a high-quality over-segment template (ARBELÁEZ et al., 2011) and outputs a smoothed image. The SuperAE encoder learns deep embeddings guided by the high-quality template. On the other hand, the DSC measures the deep similarity between superpixels and partitions them into perceptual regions by soft association, which is differentiable and can be optimized by backpropagation.

Figure 17 presents the SuperAE-DSC diagram, where the purple arrows indicate the gradient flow. The proposed SuperAE-DSC firstly trains the autoencoder for image re-

Figure 18 – SENSS architecture



Source: Wang et al., 2022.

construction using an over-segment template (ARBELÁEZ et al., 2011) as a guideline. By using the deep features embeddings from the SuperAE's encoder and superpixels generated by an existing algorithm like (FELZENSZWALB; HUTTENLOCHER, 2004) with the reconstructed image, the DSC iteratively minimizes its loss function based on the superpixel similarity and deep embeddings to output the final segmentation. The proposed method visually preserves the main structures of the image and its results are competitive with the evaluated algorithms. The SuperAE-DSC segmentation does not produce regular or compact superpixels but preserves fewer finer details of the image.

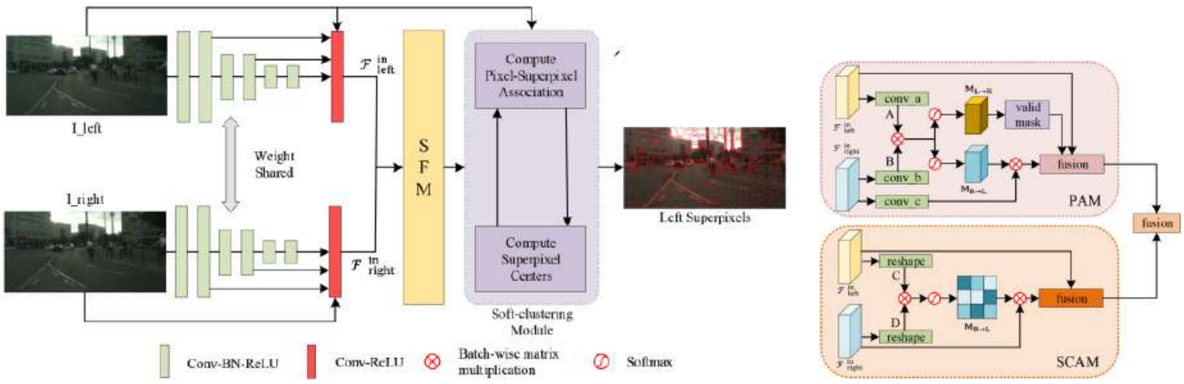
SSFCN. A challenge faced by deep learning-based superpixel methods is how to incorporate superpixels into standard convolution operations. Inspired by grid sampling (ACHANTA et al., 2012), an initialization strategy commonly adopted by traditional superpixel methods, (YANG et al., 2020) proposes an end-to-end trained deep learning model to predict superpixels on a regular image grid.

The proposal generates superpixels competitive with the state-of-the-art, with an easily integrated pre-processing in deep learning models with other applications. The proposed *superpixel segmentation for Fully Convolutional Network* (SSFCN) uses a standard encoder-decoder design with skip connections to predict superpixel association scores between pixels and regular grid cells.

To make the objective function differentiable, they replace the hard assignment with a soft association map. The loss function computes the distance between each reconstructed pixel's value and its superpixel's center value. In this work, the authors proposed two loss functions: one, similar to SLIC, uses an L_2 norm as feature distance; and the other follows SSN (JAMPANI et al., 2018) using cross-entropy with a one-hot encoding vector of semantic labels. Using the predicted map, SSFCN computes superpixels by assigning each pixel to the grid cell with the highest probability.

Compared to the other deep-based methods, the SSFCN generates more compact superpixels, is faster, and is competitive in ASA and BP-BR. It also outperforms in these metrics all non-deep learning methods evaluated. As the main drawbacks, the number of superpixels is indirectly controlled by input image resizing, and a post-processing step is required to enforce connectivity. The authors also demonstrate the proposal's efficacy by modifying a network architecture for stereo matching (CHANG; CHEN, 2018) to simultaneously predict superpixels and disparities.

Figure 19 – DAFnet diagram



(a) DAFnet diagram. (b) SFM module.

Source: Wu et al., 2021.

SENSS. Although the SSFCN (YANG et al., 2020) does not outperform existing methods on some metrics, it generates superpixels in a regular grid, and it's easy to integrate with other deep learning-based tasks. The work in (WANG et al., 2022) presents an improved end-to-end trainable deep learning-based superpixel method. The proposed *Squeeze-and-Excitation Network for superpixel segmentation* (SENSS) incorporates Squeeze-and-Excitation (SE) modules (HU; SHEN; SUN, 2018) into an SSFCN architecture (YANG et al., 2020) (Figure 18).

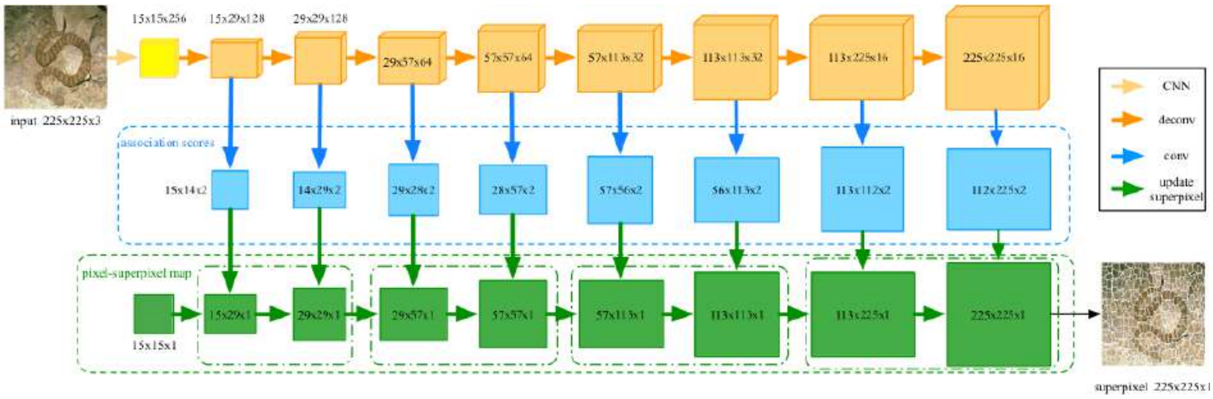
The SE block explicitly models the interdependencies between channels, improving the representation power of the network. Therefore, the proposal has an encoder-decoder architecture with an attention module at each decoder block. The encoder produces high-level feature maps, and the decoder gradually upsamples the feature maps while modeling the channel-wise relationship. For training, the proposal uses the SSFCN's differentiable loss function. The proposed network outperforms the SSFCN performance, improving its learning ability with the SE blocks and achieving competitive results. However, the SE blocks have an additional computational cost. Also, SENSS has the same drawbacks as SSFCN, with limited control of the superpixels' number, and needs post-processing to guarantee connectivity.

4.13 Deep learning-based methods with other architectures

DAFnet. The depth information captured by stereo image pairs and the correspondence of the two views can improve superpixel segmentation to capture information difficult to be distinguished. To exploit stereo images, (WU et al., 2021) proposes an end-to-end *Dual-Attention Fusion Network for superpixel segmentation* (DAFnet), which integrates mutual information from both image views.

The proposal first extracts deep features from both image views with a weight-shared convolution network (Figure 19(a)). Then, the features from both views are integrated with a Stereo Fusion Module (SFM), composed of a Parallax Attention Module (PAM) and a Stereo Channel Attention Module (SCAM). Figure 19(b) presents the SFM architecture. The PAM module models the relationship between the stereo image pair to capture its spatial level correspondence, generating an attention map through a parallax-attention mechanism (WANG et al., 2019). On the other hand, the SCAM module adaptively enhances the important information's channel (HU; SHEN; SUN, 2018). Finally, inspired by SSN (JAMPANI et al., 2018), a soft clustering module uses deep features and pixel-

Figure 20 – Superpixel Interpolation Network (SIN) architecture



Source: Yuan et al., 2021a.

level information to generate the superpixels.

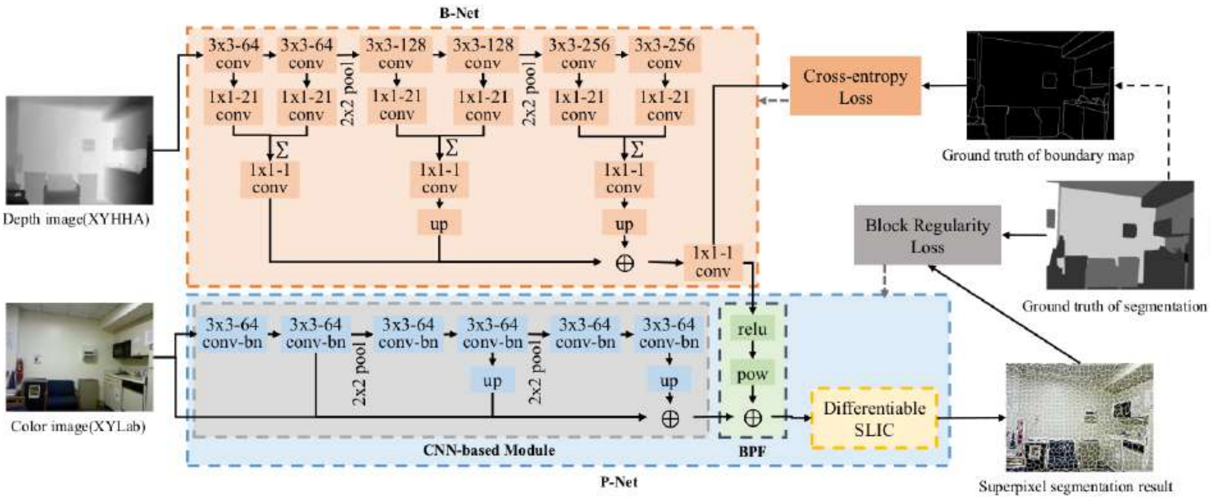
The DAFnet is the first superpixel segmentation method that extracts deep features from stereo image pairs and its proposed PAM and SCAM modules are demonstrated to improve the results. In addition, the DAFnet achieves visually better border adherence and competitive quantitative results without fine-tuning. However, it does not produce compact superpixels.

DMMSS. A *Deep Merging Model for superpixel-based segmentation* (DMMSS) is proposed in (HUANG; DING; HUANG, 2021) to manage superpixels with irregular shapes and non-fixed sizes. The proposal transforms the clustering problem into a two-stage decision problem using two deep networks to decide whether merge a pair of superpixels. Instead of receiving the whole image as the input directly, the DMMSS receives a square image patch with only two neighboring superpixels and outputs a label to indicate whether they will be merged.

The DMMSS initializes with an initial superpixel segmentation. Then, a pair of neighboring superpixels is trimmed in a bounding box. The regions in the bounding box that do not belong to the chosen superpixels' pair are removed using the inpainting technique with the Dirichlet boundary condition. The final bounding box is fed into a learning model to decide whether they will merge. The proposal uses two sequentially connected models, each with a ResNet (HE et al., 2016) architecture. For edge-preserving, the proposal applies some merging criteria after each model. The first merging model uses fixed thresholds with object boundary detection, superpixels' pair neighborhood strength, and area. And the second model uses adaptative thresholds with boundary detection, superpixels' pair neighborhood strength, texture, area, and color difference. The contour map for object boundary detection and the texture features uses the RefineContourNet (RCN) algorithm (KELM; RAO; ZÖLZER, 2019) and the Log-Gabor filter (FIELD, 1987), respectively.

For the initial clustering of DMMSS, the authors use the mean-shift algorithm (COMANICIU; MEER, 2002), SEAL (TU et al., 2018), and SSN (JAMPANI et al., 2018), and it outperforms all other segmentation methods whether the initial clustering algorithm. By transforming the segmentation problem into a decision problem and incorporating mid-level information to improve the merging decision, the DMMSS surpasses the irregular shape and annotated data problems. Since the input of the DMMSS is a superpixel pair, it can acquire a huge amount of data to train the network. In addition, the pro-

Figure 21 – BP-net diagram



Source: Zhang, Kang and Ming, 2021.

posal can merge complex background regions and preserve object regions, but with a high running time and no control over the number of superpixels.

SIN. The authors in (YUAN et al., 2021a) propose a *Superpixel Interpolation Network* (SIN), a deep learning-based method and integrable end-to-end in downstream tasks. The SIN's architecture utilizes multi-layer outputs to predict association scores using interpolations. The proposed architecture is presented in Figure 20, where *deconv* (orange arrows) operations reduce the feature channels by half to extracts multi-layer features with outputs to *conv* operations. Then, the *conv* (blue arrows in Figure 20) operation is a convolutional neural network, which transforms the multi-layer features to 2-dimensional association scores. Finally, a pixel-superpixel map procedure uses multiple interpolations with the association scores to expand the pixel-superpixel association matrix while enforcing spatial connectivity. The initial pixel-superpixel map has a reduced size and initializes with regular sampling.

The proposed method produce connected components without post-processing, being able to integrate them into downstream tasks in an end-to-end way. The SIN is faster than other deep learning-based superpixel methods, and it produces more compact and regular superpixels. Nevertheless, it underperforms some compared methods, not achieving competitive quantitative results for superpixel evaluation.

BP-net. A deep learning-based superpixel method for RGB-D images composed of a boundary detection network (B-net) and pixel labeling network (P-net) in (ZHANG; KANG; MING, 2021). The proposed BP-net combines the geometry edge information extracted by the B-net using multiscale information from depth images with the pixel features extracted by P-net. The BP-net consists of two networks: B-net and P-net. While B-net learns boundaries in different scales to detect the geometry edges for depth information, the P-net extracts k-dimensional features from color information. The features extracted from P-net incorporate the geometry edge information from B-net by using a proposed boundary pass filter. The final feature map feds a differentiable SLIC (JAMPANI et al., 2018) to produce the final segmentation with a merging procedure to enforce superpixel connectivity as post-processing. The B-net is training with a cross-entropy loss and the P-net is training with the proposed block regularity loss, which combines an accuracy term with an additional regularity term.

The BP-net generates visually more regular pixels than the other evaluated deep learning-based algorithms, achieving a generally reasonable regularity and capturing structured-rich regions. At the same time, its accuracy and boundary adherence outperform all the other algorithms. However, it has low compactness compared with unsupervised learning methods.

5 A COLOR HOMOGENEITY MEASURE FOR SUPERPIXELS

This Chapter introduces a new measure to assess color homogeneity in superpixel segmentation. As discussed earlier, the ICV and the EV measures to assess color homogeneity have severe limitations. The mean superpixel color used to calculate its color variation may be unrepresentative. Conversely, one could argue that a small set of representative colors, not very different from each other, should describe the superpixel's colors. Such a set of colors must be able to represent a perceptually homogeneous texture. Ideally, such quantity should be minimal for an acceptable description, being only one when it is monochromatic.

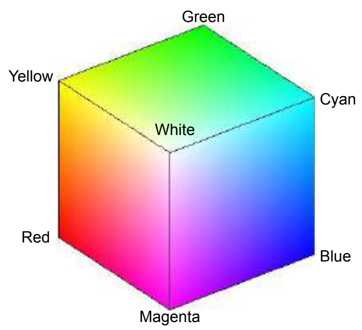
To overcome the mean color drawback, we propose a novel color descriptor, the *RGB Bucket Descriptor* (RBD), representing the superpixel as a small set of its most relevant colors. The RBD exploits the well-behaved RGB space, grouping colors based on their similarities to the cube's edges. Then, it divides each group into several subgroups, selecting the most relevant colors to describe the superpixel. Finally, using RBD, a new superpixel-reconstruction quality assessment function named *Similarity between Image and Reconstruction from Superpixels* (SIRS) measures the segmentation quality.

Using an RBD descriptor for each superpixel, SIRS measures the segmentation quality based on the exponential error of the reconstruction. For image reconstruction, SIRS selects the RBD color most similar to each pixel. Then, a variation of the Mean-Squared Error, the *Mean Exponential Error* (MEE), expresses the reconstruction error between the original and reconstructed image. The MEE increases the error weight of heterogeneous colors based on the maximum distance between the colors of the RBD. The MEE's exponent interval varies between one and two (the absolute or the mean error). Finally, SIRS defines segmentation quality as the Gaussian weighted error of reconstruction using MEE. By doing so, SIRS provides values normalized between zero and one with adequate spread to differentiate between segmentation qualities easily. SIRS appropriately penalizes superpixels with heterogeneous colors while maintaining high scores for perceptually homogeneous ones. Also, it adequately expresses discrepancies between different segmentation qualities.

5.1 RGB Bucket Descriptor

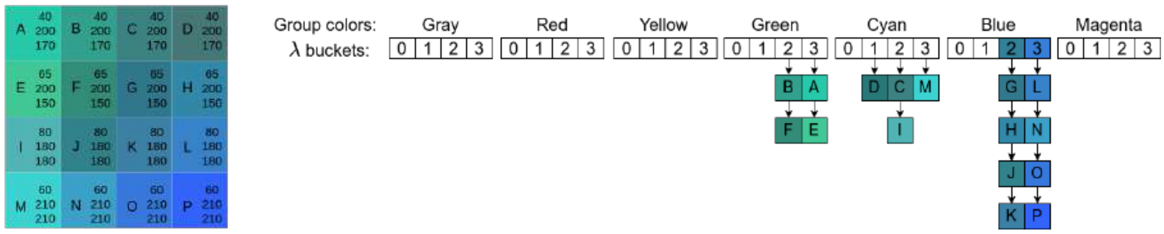
We argue that the color information of any superpixel can be represented by a minimal set of colors due to its homogeneity property. In order to build the palette of the most relevant colors in each superpixel $S_i \in S$, we exploit the RGB space, represented as a

Figure 22 – RGB cube



Source: Author.

Figure 23 – Toy example of RBD execution



(a) Image (b) RBD example

Source: Author.

cube in $[0,1]^3$ (Figure 22). By merging the white and black color vertices, the vertices correspond to the colors with maximum intensity in some channel. Therefore, considering an image $\mathcal{I} = (\mathbf{I}, I)$ and the reconstructed image $\mathcal{R} = (\mathbf{I}, R)$, I and R map to normalized RGB colors.

The Figure 23 presents a toy example of RBD. First, let $G^{S_i} \in \mathbf{S}(S_i, 7)$ represent the set of 7 disjoint groups related to each of the cube's vertices, whose colors are $V = \{c_1, ..., c_8\}$, in which $c_l \in [0,1]^3$ for $1 \leq l \leq 7$ (the 7 group colors in Figure 23(b)). RBD divides the RGB space according to the vertices of its cube representation and merges the white and black vertices to represent gray levels. Therefore, V correspond to all possible combinations of RGB color channels. Let $x = \langle x_i \rangle_{i=1}^{m}$ a vector that indicates the color channels with maximum intensity in $I(p)$ such that $x_i = \mathbb{1}(I_i(p) = \|I(p)\|_\infty)$. We populate each $G_l^{S_i} \in G^{S_i}$ by assigning every $p \in S_i$ to its most similar group using a mapping function $M(p)$ (Equation 5.1).

$$M(p) = \underset{c_i \in V}{\operatorname{argmin}} \{\|x - c_i\|_1\} \tag{5.1}$$

Although $G_l^{S_i}$ contains pixels similar to c_l, they may present significantly distinct luminosities (*i.e.*, color shades), which can be suppressed if the mean color is desired. Thus, we split it into $\lambda \in \mathbb{N}^*$ subgroups (or *buckets*), denoted by $\widehat{G}_l^{S_i} \in \mathbf{S}(G_l^{S_i}, \lambda)$. Without abuse of notation, we insert every $p \in G_l^{S_i}$ into its respective group $\widehat{G}_{l,b}^{S_i}$ given $b = \lfloor \|I(p)\|_\infty \lambda \rfloor$ ($\lambda = 4$ in Figure 23).

We name *RGB Bucket Descriptor* (RBD) the descriptor $\text{RBD}(S_i) = \{c_1, ..., c_\alpha\}$, in which $c_i \in [0,1]^3$, resultant from the selection of the $\alpha \in \mathbb{N}^*$ most relevant colors within G^{S_i} by some predetermined criterion. In this work, $\text{RBD}(S_i)$ selects the average color $\mu(G_{l,b}^{S_i})$ of the most populated buckets, irrespective of l (*i.e.*, its vertex-based group). In Figure 23(b), $\alpha = 2$. Although inaccurate for heterogeneous sets of pixels, the refinement for generating $G_{l,b}^{S_i}$ leads to a better approximation of the most predominant colors by the mean operator. On the other hand, by promoting such grouping, colors with visually indistinguishable differences are assigned to the same bucket, reducing the probability of selecting slight variations of the most frequent color.

5.2 Similarity between Image and Reconstruction from Superpixels

Given $\text{RBD}(S_i) = \{c_1, \ldots, c_\alpha\}$, one could generate a proper approximation of the original texture by the correct ordering, but such task is challenging. Conversely, we propose evaluating the best reconstruction possible from the most relevant colors for measuring the color variation description of S_i. Thus, we build $\mathcal{R}$ such that $R(p) = \operatorname{argmin}_{c_j \in \text{RBD}(S_i)} \{\|I(p) - c_j\|_1\}$.

After generating $\mathcal{R}$ from S, we may compute the *Mean Exponential Error* (MEE), shown in Equation 5.2 between it and the original image $\mathcal{I}$ for weighting each error accordingly:

$$\mathrm{MEE}(S) = \frac{1}{|\mathbf{I}|} \sum_{S_i \in S} \sum_{p \in S_i} \|R(p) - I(p)\|_1^{2-\psi} \tag{5.2}$$

in which $\psi = \max\left\{\|c_l - c_j\|_1\right\}$ and $c_l, c_j \in \mathrm{RBD}(S_i)$. If a superpixel requires a palette of highly discrepant colors, the error impact should be greater since it is describing a complex pattern. Conversely, if the relevant colors are similar and, thus, are representing a more uniform texture, such impact must be small. Finally, we may define the *Similarity between Image and Reconstruction from Superpixels* (SIRS), in Equation 5.3, by a Gaussian distribution centered at $\mathrm{MEE}(S)$:

$$\mathrm{SIRS}(S) = \exp^{-\frac{\mathrm{MEE}(S)}{\sigma^2}} \tag{5.3}$$

in wich σ^2 is a parameter that controls the importance of small error variations. In SIRS, higher the value, better is the color homogeneity of the superpixels in S, represented within $[0, 1]$.

6 COLOR HOMOGENEITY MEASURE EVALUATION

In this Chapter, we describe the experimental setup for validating our proposed measure, SIRS. First, we discuss the impacts of the parameter selection (Section 6.1). Then, in Sections 6.2 and 6.3, we compare our proposal to the EV in quantitative and qualitative evaluations, respectively, on five superpixel segmentation methods with varying segmentation qualities. The implementation of SIRS is available online ‡.

We selected three different datasets which impose different challenges in assessing segmentation. The *Birds* (MANSILLA; MIRANDA, 2016) consists of 150 images of Birds whose thin elongated legs are difficult to segment and, thus, may compromise the color description. The *Sky* (ALEXANDRE et al., 2015) has 60 images with large homogeneous regions with subtle luminosity variations. Finally, the *Extended Complex Scene Saliency Dataset* (ECSSD) (SHI et al., 2015) is composed of 1000 images with objects and backgrounds whose textures are complex.

Moreover, we select five superpixel methods with different properties to evaluate SIRS' expressiveness, leading to distinct color variation descriptions. Specifically, DISF (BELÉM; GUIMARÃES; FALCÃO, 2020) and SH (WEI et al., 2018) are state-of-the-art methods in object delineation, while IBIS (BOBBIA et al., 2021) and SLIC (ACHANTA et al., 2012) present more compact superpixels with fair delineation. Finally, we consider a grid-based segmentation (GRID), representing a segmentation with maximum compactness but poor delineation.

6.1 Parameter Analysis

For evaluating the impact of RBD's α and λ in SIRS, we performed a grid-search for a varying $\alpha \in [1, 2, 4, 8]$ and $\lambda \in [8, 16, 32, 64]$ on a random selection of 30% of the Birds images. From Figure 24(a), it is possible to infer that α and λ are highly correlated. By selecting $\alpha = 2$ and $\lambda = 32$, the reconstruction is compromised due to the reduced number of relevant colors selected in contrast to the low discretization of the color space (*i.e*, small color intervals are grouped on RBD). On the other hand, $\alpha = 8$ and $\lambda = 8$ offers a small penalization for few superpixels, which often present low-quality color variation description. Therefore, we opt for $\alpha = 4$ and $\lambda = 16$ since it severely penalizes for few superpixels, while selecting a fair quantity of relevant colors for reconstruction. Figure 25 illustrates the impacts on such selection: by increasing α and λ, RBD is capable of improving the set of relevant colors, leading to a more accurate reconstruction. It is important to note that the reconstruction may have no errors for α values equal to the number of populated buckets. Therefore, the λ and α values are crucial for our proposal's performance.

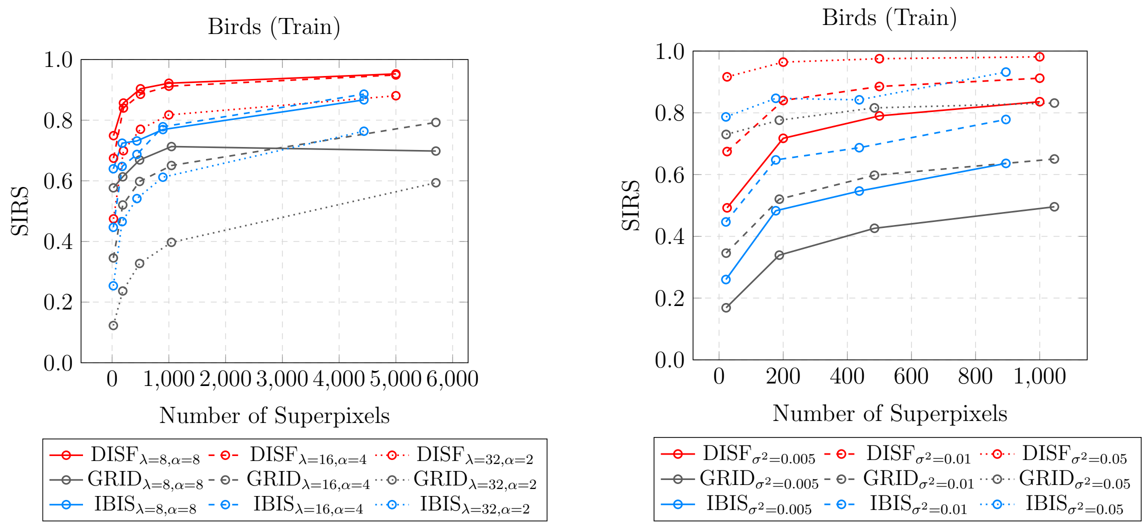
Similarly, to evaluate the impact of the Gaussian variance σ^2, we evaluated varying it between $[0.005, 0.05]$ with step of 0.005. From Figure 24(b), we infer that σ^2 influences on the steepness of the curves, indicating lighter penalizations as σ^2 increases and, finally, reducing expressiveness. Therefore, for a fair error influence and a better spread of the curves, we opted for $\sigma^2 = 0.01$.

6.2 Quantitative Results

As one can see in Figure 26, both SIRS and EV distinguish methods which maximize delineation (*i.e.*, DISF and SH) with those opting for more compact superpixels (*i.e.*, GRID, SLIC, and IBIS). However, EV presents a lesser spread than SIRS, as exemplified in the distance between IBIS' and SH's curves. Moreover, EV tends to result in

‡ ⟨https://github.com/IsabelaBB/SIRS-superpixels⟩

Figure 24 – Impact of different λ, α, and σ^2 for varying superpixel numbers on the train images of Birds dataset

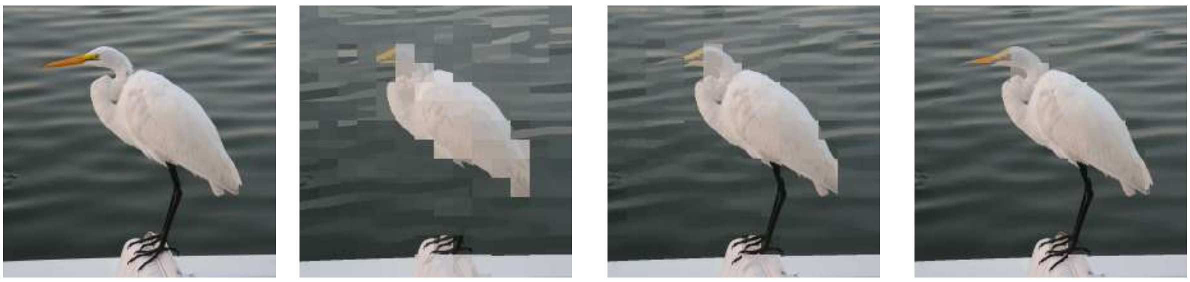


(a) Impact of different λ and α for varying superpixel numbers on the train images of Birds dataset with GRID (gray), IBIS (blue), and DISF (red) segmentations.

(b) Impact of different σ^2 for varying superpixel numbers on the train images of Birds dataset with GRID (gray), IBIS (blue), and DISF (red) segmentations.

Source: Author.

Figure 25 – Impact of different λ and α for image reconstruction using RBD in GRID segmentation with 200 superpixels



(a) Original Image

(b) Reconstruction with $\alpha = 2$ and $\lambda = 8$

(c) Reconstruction with $\alpha = 4$ and $\lambda = 16$

(c) Reconstruction with $\alpha = 8$ and $\lambda = 32$

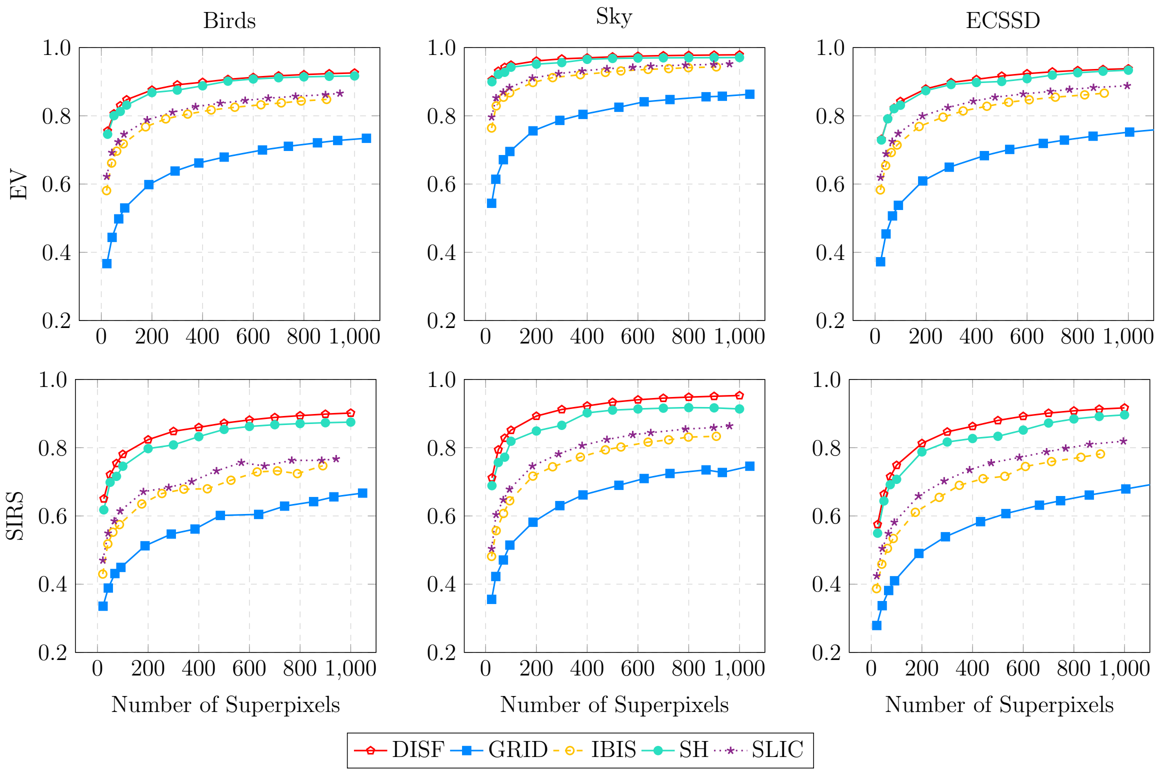
Source: Author.

significantly higher values, especially in contexts where superpixels are increasingly heterogeneous. For example, GRID obtains a score over 0.5 on the Sky dataset with only 25 superpixels. Conversely, SIRS offers a more meticulous discrepancy even with methods with similar performance, like DISF and SH. Also, due to its penalization, SIRS exhibits a more coherent range of values when few superpixels are generated — *i.e.*, in a more heterogeneous segmentation. In the same example, GRID scored less than 0.4 on the same dataset.

6.3 Qualitative Results

Figure 27 presents a visual comparison between the evaluations obtained with SIRS and EV in segmentations of images with large homogeneous (first three columns of Figure 6) or texturized last three columns of Figure 6) regions. The first and fourth column consist on the respective segmentation with 25 and 500 superpixels. The second and fifth column present the EV evaluation representation (whiter values indicate higher scores),

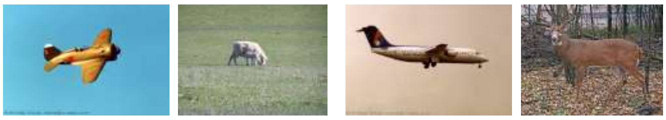
Figure 26 – Results obtained for Birds, Sky and ECSSD for EV and SIRS.



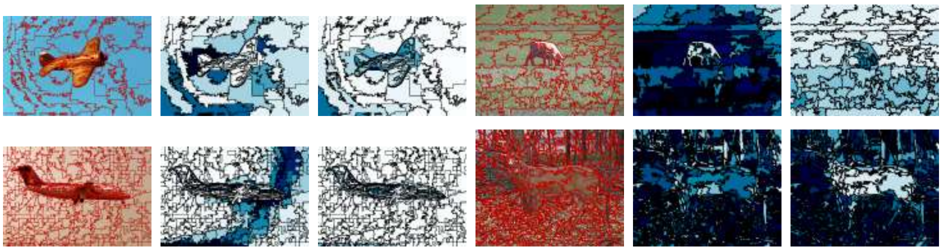
and it is analogous for the third and sixth columns for SIRS. As shown in the first three columns of Figure 6, while the EV has a higher penalty even in smooth color transitions, the SIRS is robust to such changes. For example, in the gray plane segmentation (first three columns of the second row of Figures 6(a)-(e)), superpixels with low color variation in the sky present low scores in EV. On the same superpixels, SIRS can evaluate its homogeneity accordingly. In addition, SIRS also obtains a coherent penalty in more significant color variations. For example, in yellow plane segmentation (first three columns of the first row of Figures 6(a)-(e)), the visually more relevant variations between distinct shades of yellow and between yellow and red present worse evaluations since they are more different from each other.

Concerning more textured backgrounds, SIRS also demonstrates robustness in simpler textures, as can be seen in the three columns to the right of the first line of Figures 6(a)-(e). However, more complex textures, as shown in the three columns to the right of the second row of Figures 6(a)-(e), can receive a significant penalty but are generally softer than EV. In contrast, SIRS consistently perceives homogeneous regions as low-variant ones, independently from the number of superpixels, in both Sky and ECSSD segmentations (first and last three columns of Figure 6, respectivelyy). Moreover, it is interesting to notice that our measure tends to be more correlated to delineation than EV, given the penalizations in regions with high color variance — often at the object borders. As the delineation performance decreases, the color variation captured tends to be more heterogeneous (*i.e.* RBD generates a more diverse palette), leading to a more drastic penalization. We argue that such robustness is directly linked to the accurate selection of colors from RBD, properly describing superpixel homogeneity. Finally, it is worth noticing that, although SIRS may penalize more heterogeneous regions (*e.g.*, those with complex textures), it tends to be lighter than those from EV.

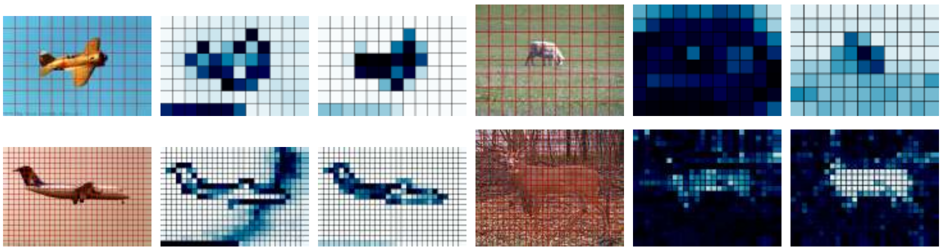
Figure 27 – Segmentation comparison with images from Sky and ECSSD with 100 and 500 superpixels with EV (second column) and SIRS (fifth column) evaluations.



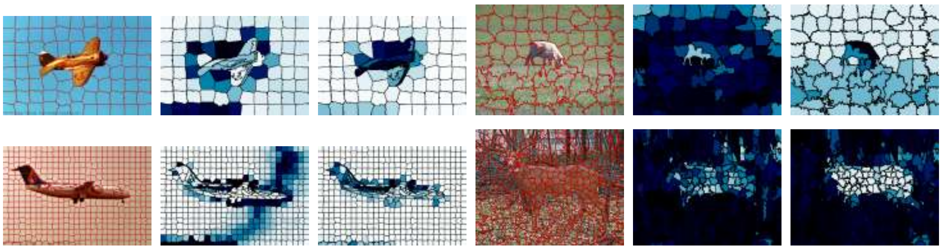
(a) Original images



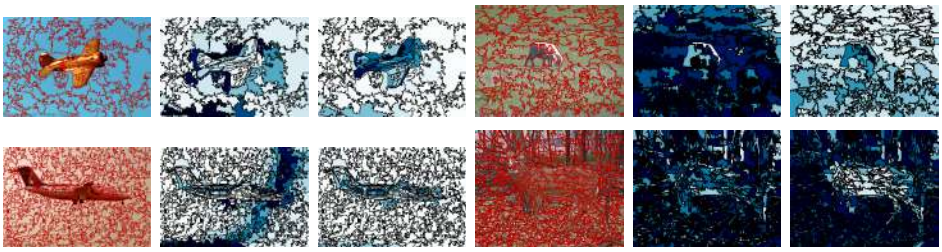
(a) DISF



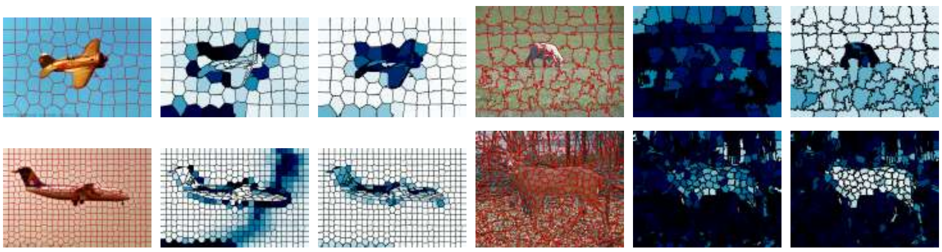
(b) GRID



(c) IBIS



(d) SH



(e) SLIC

7 AN EVALUATION OF THE STATE-OF-THE ART SUPERPIXEL SEGMENTATION

This Chapter presents the experimental evaluation of state-of-the-art superpixel segmentation methods. From the analyzed methods in Chapter 3, we identified 18 as open source, of which 12 show no execution errors. However, we discarded five: ss-RIM (SUZUKI, 2020) and ew-RIM (YU; YANG; LIU, 2021), for demanding execution time, SSFCN (YANG et al., 2020) and SIN (YUAN et al., 2021a), for offering an indirect and limited control of the number of superpixels, and the PGDPC (GUAN et al., 2021) for not offering an automatic strategy for cutting the generated density graph. Therefore, from the analyzed methods in Chapter 3, DISF (BELÉM; GUIMARÃES; FALCÃO, 2020), RSS (CHAI, 2020), ODISF (BELÉM et al., 2021), IBIS (BOBBIA et al., 2021), DRW (KANG; ZHU; MING, 2020), DAL-HERS (PENG; AVILES-RIVERO; SCHÖNLIEB, 2022), and LNSNet (ZHU et al., 2021) were evaluated. In addition to these seven methods, we include the ISF (VARGAS-MUÑOZ et al., 2019), SNIC (ACHANTA; SUSSTRUNK, 2017), SH (WEI et al., 2018), GMMSP (BAN; LIU; CAO, 2018), LSC (LI; CHEN, 2015), and SCALP (GIRAUD; TA; PAPADAKIS, 2018) in this evaluation. We also selected the six methods assessed as state-of-the-art in Stutz, Hermans and Leibe (2018): SLIC (ACHANTA et al., 2012), SEEDS (BERGH et al., 2012), ERS (LIU et al., 2011), ETPS (YAO et al., 2015), CRS (CONRAD; MERTZ; MESTER, 2013), and ERGC (BUYSSENS; GARDIN; RUAN, 2014). Finally, as in Chapter 6, a grid segmentation (GRID) was used as a baseline.

7.1 An overview of the evaluated methods

The Table presented in Appendix A contains an overview of all superpixel methods in this work, including their respective codes. The strategies ERGC, RSS, ISF, DISF, and ODISF , perform clustering based on paths. While RSS provides a non-iterative method that guarantees the optimality of the generated forest, ISF, DISF, and ODISF use iterative strategies. The ISF recalculates the position of the seeds at the end of each iteration. At the same time, the DISF and ODISF perform an iterative removal of the seeds generated by an initial oversampling. While DISF only uses pixel and path-based characteristics, ODISF includes saliency information in its removal step. Unlike the others, ERGC formulates the segmentation with the Eikonal equation, solving it with the *Fast Marching Algorithm* (SETHIAN, 1999), which calculates the minimum geodesic paths of the graph.

SLIC, SCALP, and LSC perform clustering based on a distance function limited to a region concerning a reference point in the image. In these three methods, the reference point consists of the center of the cluster, and the search region size depends on the expected superpixels size. In SCALP, the distance function from a center to a pixel is weighted according to the linear path between these two points using a boundary map. On the other hand, LSC explores features at the pixel level, mapping them into 10-dimensional points. Based on SLIC, the SNIC uses a dynamic center update strategy that guarantees the connectivity of its superpixels during clustering and does not require multiple iterations. The DRW performs a similar clustering step, which formulates the clustering problem based on the Random Walk algorithm (GRADY, 2006) and adds dynamic nodes to the graph to reduce redundant computation and capture features at the region level.

Unlike previous approaches, CRS, SEEDS, ETPS, and IBIS start with a grid segmentation and update the superpixel contours according to an energy function. While IBIS has a reduced execution time and an optimization function similar to SLIC, SEEDS, and ETPS evaluate the superpixels boundaries with a coarse-to-fine strategy. In its iterations, SEEDS uses an approach based on the hill-climbing algorithm and an optimization function with characteristics based on the color histogram. On the other hand, ETPS orders the superpixels boundaries evaluation using a priority queue. Its optimization function uses features at the pixel level to optimize homogeneity, compactness, size, and smoothness.

The SH and DAL-HERS produce a superpixel hierarchy. While SH is based on Boruvka's algorithm (WEST et al., 2001), the DAL-HERS generates affinity maps with a residual convolutional network and uses these maps to create a superpixel hierarchy. In addition to DAL-HERS, LNSNet is also a deep learning-based approach. It uses a clustering module with training based on the loss function of a Lifelong learning reconstruction module. Finally, the GMMSP models the segmentation as a weighted sum of Gaussians, where each gaussian is associated with a superpixel. On the other hand, ERS models the segmentation problem based on the Random Walk (GRADY, 2006) and generates superpixels from the cut in the image graph that optimizes its function.

7.2 Experiments settings

As the SIRS assessment in Chapter 6, we selected Birds, Sky, and ECSSD datasets as they contain different challenging aspects. We also chose the Insects dataset (MANSILLA; MIRANDA, 2016), composed of 130 images of spiders, insects, and other invertebrates, whose images have more homogeneous backgrounds than in Birds. We evaluate the methods in these four datasets according to the object delineation, compactness, and color homogeneity. In addition to the proposed SIRS measurement, we used Explained Variation (EV) (MOORE et al., 2008) to assess color homogeneity. We evaluate delineation using Boundary Recall (BR) (MARTIN; FOWLKES; MALIK, 2004) and Undersegmentation Error (UE) (NEUBERT; PROTZEL, 2012). Finally, we assess superpixels' compactness using the Compactness index (CO) (SCHICK; FISCHER; STIEFELHAGEN, 2012).

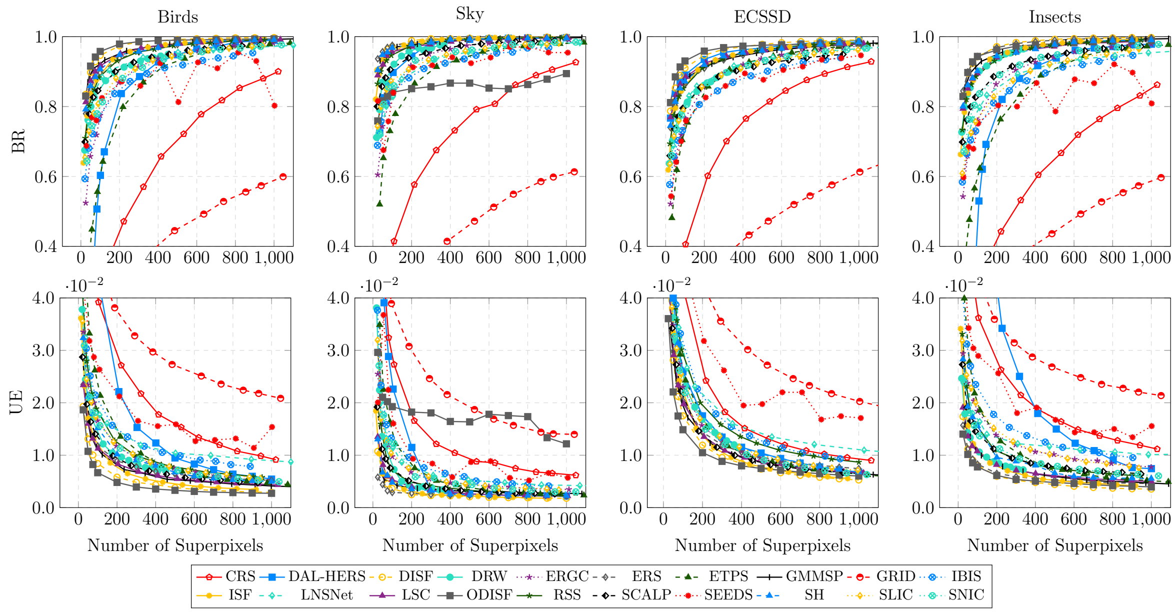
As Stutz, Hermans and Leibe (2018), we also evaluated the stability of segmentations using each evaluation metric's minimum (min), maximum (max), and standard deviation (std). While the minimum and maximum indicate the evaluation limits reached by the segmentations of each method, the standard deviation indicates its stability. Finally, we performed a qualitative evaluation concerning the delineation, compactness, and color homogeneity.

7.3 Quantitative evaluation

As shown in Figure 28, the quantitative differences between the best methods in BR and UE delineation to the other methods are minor. Concerning the delineation with BR and UE, GRID, CRS, and SEEDS achieve the worst results in all datasets. According to the evaluation with UE, most methods have low leakage. Similarly, the delineation measured by BR is generally high. In both BR and UE, ODISF, DISF, LSC, ISF, GMMSP, SH, and ERS achieved best scores. However, followed by GRID, ODISF obtained the lowest delineation according to BR in Sky dataset. Also, RSS have a competitive BR delineation with more leakage measured with UE. Among the other methods, IBIS, ETPS, SLIC, LNSNet, and ERGC obtained a low delineation, only superior to the GRID, SEEDS, and CRS. Their results are followed by ERGC, SNIC, SCALP, and DRW.

One may see in Figure 28 that DAL-HERS obtains low delineation for numbers of superpixels smaller than 400, approximately, in the Birds, ECSSD, and Insects datasets.

Figure 28 – Results for Birds, Sky, ECSSD and Insects for BR and UE.



However, DAL-HERS presents a competitive delineation after 400 superpixels. These low results occur because the method may generate very small regions, resulting in segmentations with low delineation and low color homogeneity.

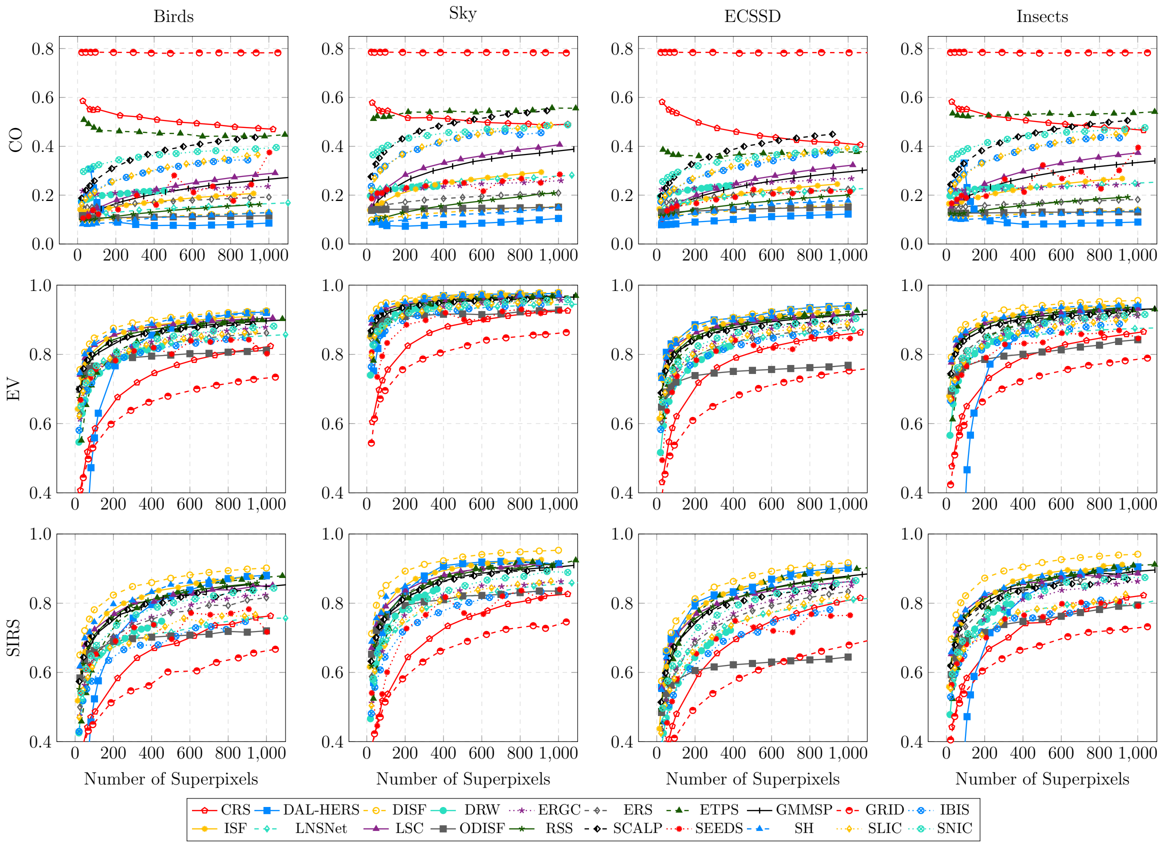
Concerning Compactness (CO) (Figure 29), GRID obtains the most compact segmentations. Aside GRID, CRS and ETPS obtained the highest compactness, followed by SCALP and SNIC. SLIC and IBIS achieve similar compactness, usually lower than SCALP and SNIC. All these methods have a parameter to determine the compactness. While CRS and ETPS produce superpixels by optimizing the contours of a grid segmentation, the others use different approaches based on SLIC. On the other hand, LSC and GMMSP present a similar and moderate compactness. Among the evaluated methods, only SEEDS had greater variability in its compactness. More delineation-focused methods, such as ODISF, DISF, SH, and DAL-HERS produced less compact segmentations.

When evaluating the homogeneity of the segmentations (Figure 29) with EV and SIRS, the results of the first metric were generally very high and closer to each other compared to the second. However, their results show some similarities. In both, GRID and CRS had the worst results in all datasets. In addition, ODISF has low color homogeneity in both measures, being the second worst in ECSSD dataset. In all datasets, it is not easy to define the best methods in the EV evaluation, while DISF obtains the best results in the SIRS evaluation. In both measures, DISF, SH, ISF, LSC, RSS, GMMSP, and SCALP achieve competitive results.

In our analysis, most path-based clustering methods had similar performance in object delineation, compactness, and homogeneity. Among these methods, DISF had better delineation and color homogeneity. On the other hand, ODISF obtained a similar delineation in most datasets but with low color homogeneity. The significant performance reduction of ODISF in the Sky dataset is due to the saliency maps identifying wrong objects. Although path-based methods had optimal delineation, their superpixels have low compactness. With a similar clustering approach, ERS performs clustering based on graphs and obtains excellent delineation in Sky and Insects datasets.

Neighborhood-based clustering approaches had more variate results while LSC achieved

Figure 29 – Results for Birds, Sky and ECSSD for EV, SIRS, and CO.



better delineation and more homogeneous superpixels. SLIC had superpixels with moderate compactness and worse delineation. On the other hand, SCALP obtained a competitive delineation with homogeneous and more compact superpixels than in SLIC.

Methods that perform clustering based on contour optimization also reached different results due to the distinction between their optimization functions. Among these, IBIS achieved better object delineation and color homogeneity, with results similar to SLIC in all evaluation measures. On the other hand, CRS and SEEDS had the worst delineation and homogeneity but greater compactness among all the methods evaluated. Therefore, among the main processing approaches, clustering based on contour evolution produced the worst results in object delineation and color homogeneity but with higher compactness.

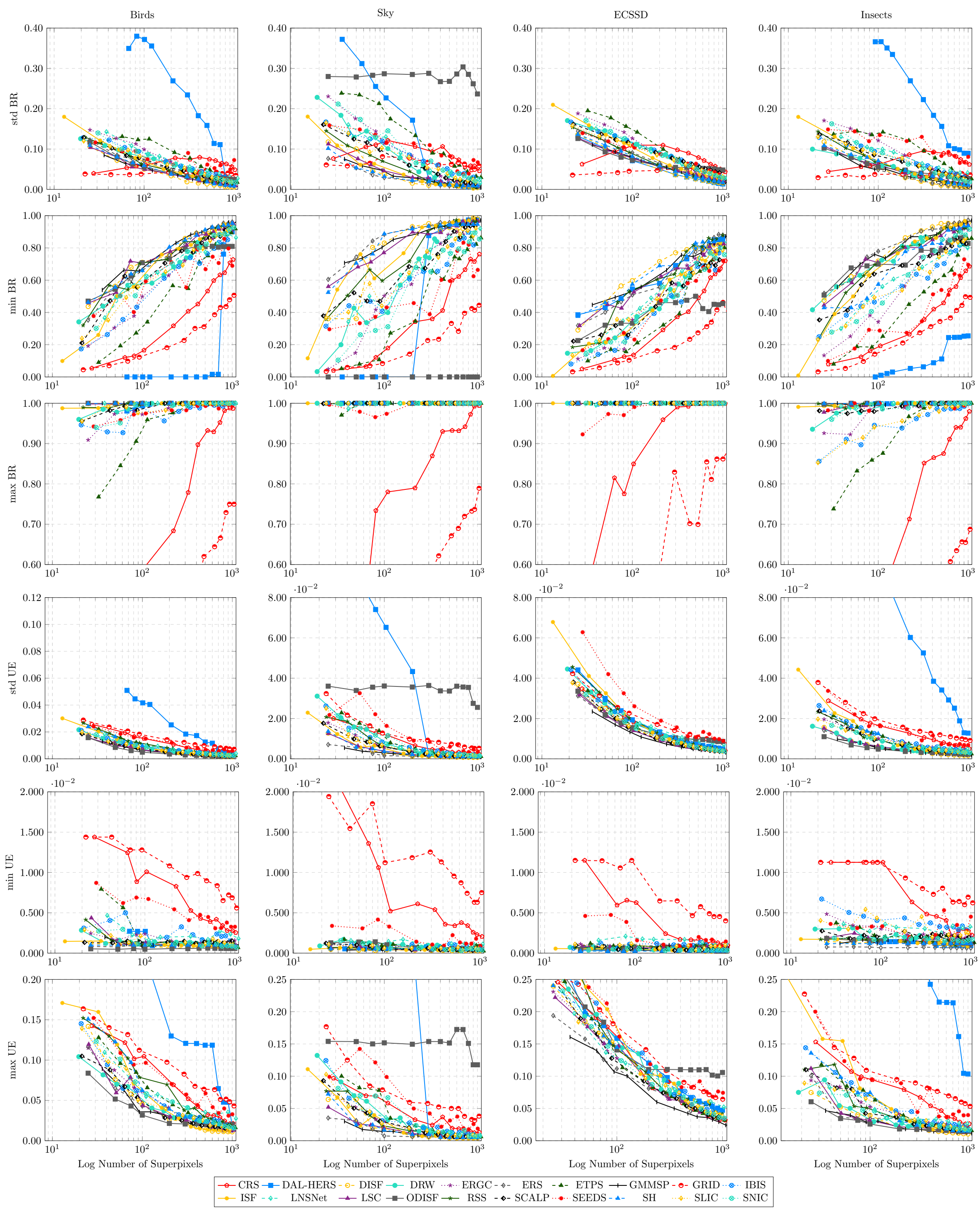
Regarding clustering with a dynamic center update, DRW, and SNIC use strategies to adapt the number of generated superpixels to the image content. Despite their similarities, DRW and SNIC use different features and optimization functions, which explains their different results. While DRW has better delineation and fewer superpixels, SNIC generates more compact and homogeneous superpixels. The lower color homogeneity of DRW compared to SNIC is due to the smaller number of superpixels generated by the DRW than in the other methods.

Concerning hierarchical approaches, SH and DAL-HERS, have low compacity and high color homogeneity. However, SH had competitive delineation in contrast with worse results with DAL-HERS. Finally, GMMSP and LNSNet, unique in their clustering category, presented excellent delineation with BR. Concerning UE and color homogeneity, LNSNet had heterogeneous superpixels with moderate compactness and more leakage. On the other hand, GMMSP achieved competitive results in all evaluated measures.

7.4 Evaluating stability

As one may see in Figure 30, most methods showed high stability regarding object delineation, with low standard deviation, increasing minimum and maximum values in BR, and decreasing in UE. While DAL-HERS, SEEDS, ETPS, and CRS showed lower BR stability in Birds and ECSSD datasets, ODISF showed high instability in the Sky dataset, with very high and almost constant standard deviations. Despite being a delineation-

Figure 30 – Results for Birds, Sky and ECSSD for minimum, maximum and standard deviation of BR and UE.



focused method, the ODISF leakage deviation (UE) in the Sky dataset achieve worse results than GRID for superpixel number greater than 200. The ODISF obtained a similar instability in the UE evaluation in the same dataset, which contrasts with the stability presented in the other datasets. The high standard deviation with BR with the ODISF in the Sky dataset indicates that the low performance shown in the mean evaluation is due to a more significant number of segmentations with low delineation. On the other hand, DAL-HERS was the method that presented greater instability due to the small regions generated mentioned above. As can be seen in Figure 30, the min BR indicates that the DAL-HERS remains with very bad segmentations for almost all the amounts of superpixels evaluated. Based on the results obtained for this method by Peng, Aviles-Rivero and Schönlieb (2022), we consider that the low performance of DAL-HERS evaluated in this work results from some code or execution error.

As shown in Figure 28, in BR and UE evaluation, the DISF, GMMSP, LSC, SH, and ERS methods showed high stability. These methods presented more varied values in the evaluation of minimum BR, while their maximums were concentrated in very high values. On the other hand, ISF, RSS present stable and low std BR and std UE, but with some instability in max UE and min BR values. GRID, CRS, and SEEDS had the worst results among the minimum BR values, while SH, ISF, RSS, GMMSP, DISF, LSC, and ERS had the highest minimums. In the delineation evaluation, while the assessment with BR obtained slightly more varied values, the evaluation with minimum UE resulted in small and similar values between the methods.

When evaluating homogeneity with EV and SIRS (Figure 31), both resulted in very high maximum values, especially SIRS in the ECSSD dataset. On the other hand, the minimum assessments of the EV and SIRS showed more significant variation between the methods. In both the maximum EV and SIRS, the GRID and CRS presented worse evaluations — *i.e.*, lower maximums. However, in the ECSSD dataset, the maximum SIRS of the DRW obtained worse results but was close to the other methods. In the evaluation with SIRS and minimum EV, while the results with EV had increasing values, the results with SIRS had more rigorous minimum scores, with increasing values only in the Sky dataset. In both minimum metrics, the methods evaluated with the highest minimum differ, except for DISF, which presents higher results in all datasets, followed by SH. Among the evaluations with minimum EV, the ODISF presents almost constant and worse results than GRID in the Sky and ECSSD datasets.

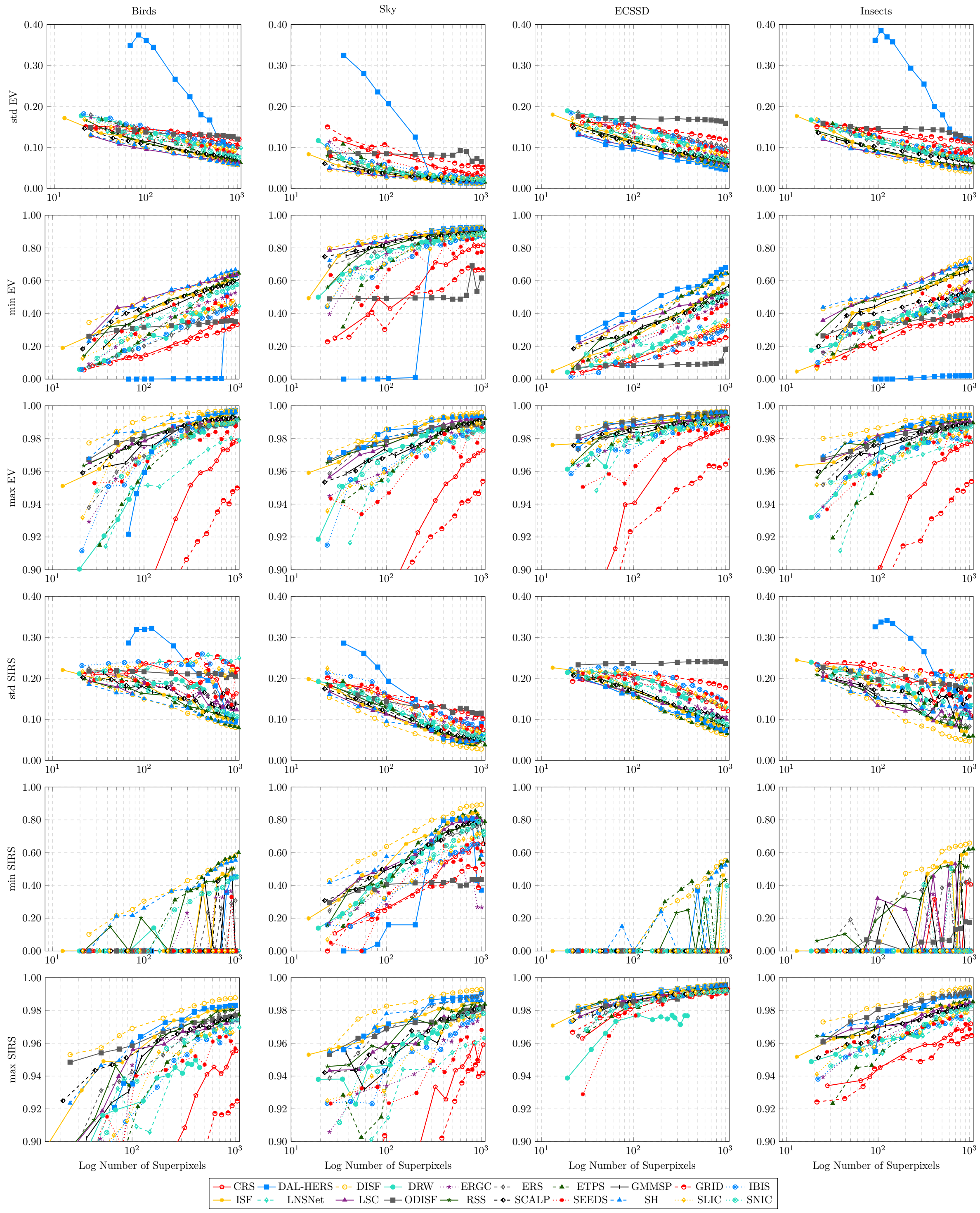
The standard deviation of SIRS and EV also showed distinct variations. While the standard deviation of the EV evaluation obtained less stable results, the standard deviation of the SIRS evaluation presented more increasing results, indicating greater instability in some methods. For the EV's standard deviation assessment, the DISF, SH, and LSC methods showed high stability in all datasets. In addition, the ISF, RSS, and SCALP also showed high stability in at least one dataset. Unlike the EV, in the SIRS standard deviation evaluation, the LNSNet, GRID, IBIS, ODISF, and SLIC methods showed less stability in Birds and Insects. On the other hand, the DISF method showed high stability in SIRS, followed by the SH and ETPS methods.

7.5 Qualitative evaluation

Considering that the image object can vary according to the application, we evaluated the segmentation delineation concerning the image boundaries, regardless of its ground truth. We also evaluated the superpixels' quality according to their contours' smoothness and compactness. In this work, we defined that superpixels have smooth contours when their shapes are close to convex shapes. And we determine that a superpixel is compact when its shape is close to a regular polygon.

Figures 32 and 33 shows superpixel segmentation in Birds, Sky, ECSSD, and Insects datasets, where the superpixels boundaries are shown in red. Relative to path-based clustering methods, the superpixels produced by RSS are not compact. In addition, for a high number of superpixels, RSS tends to generate elongated and thin superpixels

Figure 31 – Results for Birds, Sky and ECSSD for minimum, maximum and standard deviation of EV and SIRS.



at the most evident image boundaries, leading to an optimal delineation. However, by reducing the number of superpixels, the delineation quality decreases dramatically for smooth boundaries. As one may see in Figure 33, unlike RSS, ISF produces compact superpixels in homogeneous regions. However, its high sensitivity to color variations generates non-smooth superpixels in less homogeneous regions with high size variations. For a higher number of superpixels, ISF has excellent delineation. However, reducing the number of superpixels implies a noticeably worse delineation.

An improved delineation may be seen in DISF segmentation, in which its superpixels are neither compact nor smooth, but its segmentation presents a high adherence to the image boundaries. As one may see, DISF maintains good adherence to the image boundaries and generates larger superpixels in more homogeneous regions even with a smaller number of superpixels. Based on DISF, ODISF presents very different results from the previous ones. ODISF produces more superpixels in the area identified by the saliency map. This can improve the delineation of this region, but the superpixels generated are neither compact nor smooth. Due to this, there is a low number of superpixels in regions not identified by the saliency map, leading to a worse delineation. Similar to the previous ones, the segmentation with ERGC has good adherence to the image boundaries. In addition, its superpixels do not have significant variations in size, and their contours are smooth. However, for a smaller number of superpixels, the boundary adherence of ERGC segmentation reduces significantly.

Regarding the neighborhood-based methods, one may see that SLIC produces very compact superpixels with good adherence to the image boundaries. In less homogeneous regions, SLIC generates superpixels with slightly non-smooth contours. By reducing the number of superpixels, the compactness is slightly reduced, even in complex areas of the image. On the other hand, the delineation is more affected. In contrast, SCALP produces very compact superpixels with excellent delineation. The compactness of SCALP segmentation is reduced for a reduced number of superpixels, but the contours remain smooth, and the delineation is reduced slightly. Unlike SLIC and SCALP, LSC produces compact superpixels in more homogeneous regions. However, its high sensitivity to minor color variations results in superpixels with less smooth contours in regions with simpler textures. Furthermore, the LSC generates more elongated superpixels in the most evident image boundaries, obtaining a great delineation but without compactness. By reducing the number of superpixels, the delineation quality suffers a small reduction, and its superpixels have significantly less smooth contours in regions with textures.

With a segmentation visually very similar to SLIC, SNIC also produces superpixels with high compactness and better delineation. On the other hand, in DRW, the superpixels are not compact, and the number of superpixels is noticeably smaller than expected. Despite this, the DRW generates superpixels with good adherence and a smaller number of superpixels in more homogeneous regions. Similar to DRW, the superpixels in SEEDS are not compact and have non-smooth boundaries. The segmentation with a higher number of superpixels in SEEDS has moderate delineation with small leakage regions. By reducing the number of superpixels, the compactness and smoothness do not increase in SEEDS, and there is a noticeable reduction in delineation.

In contrast to SEEDS, CRS generates very compact superpixels but with low adherence to the image boundaries. In a segmentation with 100 superpixels, the image boundaries seem to be almost completely ignored. Similarly, ETPS produces very smooth and compact superpixels, with some elongated and non-smoothness in image boundaries. For a higher number of superpixels, the segmentation generated with ETPS has high adherence to the boundaries. The compactness is maintained by reducing the number of superpixels, but the delineation suffers drastically. IBIS also generates significantly compact pixels, whose compactness and smoothness vary depending on the region's homogeneity. For a larger number of superpixels, their compactness in homogeneous regions is very high, and IBIS has good adherence to the image contours, even in more complex regions. However, its sensitivity to color variations reduces compactness and smoothness in less homogeneous areas. Also, by reducing the number of superpixels, its adherence to contours is significantly reduced.

Regarding the hierarchical methods, the segmentation with SH has an excellent delineation, but its superpixels are not compact and have non-smooth contours in more textured regions. In addition, the method generates elongated and thin superpixels at some of the prominent image boundaries. DAL-HERS also has superb delineation but generates rough superpixels and some tiny ones, resulting in visibly poor segmentation. LNSNet produces a significantly higher number of superpixels than desired. Like ISF, LNSNet produces compact superpixels in homogeneous regions, but its sensitivity to color variations implies very rough superpixels. It has good delineation when the number of superpixels is higher, but the non-smooth contours of the superpixels do not have a high delineation even in regions with a more pronounced boundary, causing small leaks.

In ERS, for a larger number of superpixels, they do not vary much in size and have low smoothness but good boundary adherence. By reducing the number of superpixels, its boundary adherence reduces, but not drastically. In comparison, the GMMSP produces significantly more compact superpixels in more homogeneous regions and less compact, but generally smooth contours, superpixels in less homogeneous regions. By reducing the number of superpixels, the compactness is maintained in the homogeneous areas of the image, but in the less homogeneous regions, the compactness and smoothness are drastically reduced.

Among the evaluated methods, CRS, ERGC, ERS, ETPS, SCALP, SLIC, SNIC, and GMMSP produced visibly compact superpixels. CRS, SCALP, and ETPS showed greater compactness with smoother contours from these methods. Among the less compact segmentations, those of DAL-HERS, SH, and LNSNet presented less smoothness in their superpixels' contours. Regarding the delineation, the CRS presented the worst result, while DISF, GMMSP, ERS, LSC, RSS, and SH presented excellent delineation. DISF and GMMSP generated the segmentations with better adherence to the image boundaries. In addition, ODISF also showed exceptional adherence to boundaries belonging to a specific image region, indicated as an object in the saliency map. As observed in the quantitative evaluation, when the saliency map corresponds to the desired object in the image, the ODISF delineation outperforms the other methods. However, as shown in Figure 32, the high ODISF performance is due to greater competition between superpixels for the object borders indicated in the salience map. This competition results in a smaller amount of superpixels in the other regions of the image. Therefore, it obtains a low delineation in areas not identified as object.

Regarding the main processing, methods with more compact superpixels generally use clustering based on contour evolution followed by those with clustering based on region. One may also observe the same in SNIC, ETPS, and ERGC. On the other hand, hierarchical methods, paths-based clustering methods, and LNSNet showed less compactness concerning the others. These methods, GMMSP and ERS, generally have excellent delineation.

Figure 32 – Segmentation comparison with images from Birds, Sky, ECSSD, and Insects with 100 and 700 superpixels.

Figure 33 – Segmentation comparison with images from Birds, Sky, ECSSD, and Insects with 100 and 700 superpixels.

8 CONCLUSIONS

In this work, we extensively review the recent literature on superpixel segmentation and propose a taxonomy to relate its methods. Our review aims to inform about the most recent approaches, considering the rapid advances in superpixel literature. In this work, we identified that superpixel methods can be divided into up to three processing steps. When dividing each algorithm, we assign to it high-level categories for each step according to its main task. We also identify that the superpixel algorithms incorporate different features, being able to extract them during their execution or using other algorithms for this purpose. Therefore, the taxonomy also includes a categorization according to the processing level of the features used.

The third proposal of this work consists of a new evaluation measure to assess color homogeneity in superpixel segmentation. The proposed measure models the problem of assessing homogeneity in segmentation as an image reconstruction problem. To reconstruct the original image based on segmentation, we developed a new superpixel descriptor, the RGB Bucket Descriptor (RBD). The RBD divides the RGB color space into color sets and describes a superpixel by its most frequent average color intensities. By describing the colors of each superpixel with RBD, the proposed measure, named Similarity between Image and Reconstruction from Superpixels (SIRS), assess the homogeneity of each superpixel based on the error of its reconstruction. Compared to Explained Variation, SIRS is more robust to less perceptual color variations.

Finally, we performed a state-of-the-art superpixel segmentation evaluation to assess the contribution of the recent methods to those widely used in the literature. We identified that from the six recommended methods, the CRS, SEEDS, and ETPS have a significantly lower object delineation than most of the other evaluated methods. From the recommended methods, only the ERS obtained competitive results. Our evaluation demonstrates that methods with path-based and hierarchical clustering on their main processing usually achieve the best delineation with low compactness. In contrast, boundary evolution clustering methods usually have the highest compactness and the worse delineation. Also, neighborhood-based and dynamic center update clustering usually produces compact superpixels with moderate delineation. Concerning other main process categories, the graph-based and the data distribution-based clustering obtain moderate compactness. While the former has a moderate delineation, the second has excellent delineation.

In addition, we assess the superpixel methods' stability based on their standard deviation, minimum, and maximum delineation, and homogeneity. Our results show that DAL-HERS, SEEDS, CRS, ODISF, ETPS, ISF, RSS, and IBIS have instability. The results show that DISF, SH, GMMSP, ERS, and LSC achieved better delineation and stability. Also, LSC, ERS, and GMMSP generate compact superpixels, especially in homogeneous regions. In all datasets, DISF presents the best delineation and color homogeneity. For little more compactness with minimal delineation loss, we recommend GMMSP. However, for applications that require greater compactness and good object delineation, we recommend SCALP. For future works, we intend to explore the optimal selection for λ and α values and improve SIRS to highly correlate it with accurate object delineation. We also intend to include more recent methods in our analysis and perform time and robustness evaluations.

REFERENCES

ACHANTA, R. et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 34, n. 11, p. 2274–2282, 2012. 19, 20, 21, 27, 30, 31, 32, 33, 34, 38, 39, 40, 42, 52, 60, 64, 89

ACHANTA, R.; SUSSTRUNK, S. Superpixels and polygons using simple non-iterative clustering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. 30, 35, 36, 49, 64, 89

ALEXANDRE, E. B. et al. Ift-slic: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. In: *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*. [S.l.: s.n.], 2015. p. 337–344. 60

AN, J. et al. Extract and merge: Superpixel segmentation with regional attributes. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2020. p. 155–170. 30, 44, 45, 89

ARBELÁEZ, P. et al. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 5, p. 898–916, 2011. 51, 52

AURENHAMMER, F. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, v. 16, n. 1, p. 78–96, 1987. Disponível em: ⟨https://doi.org/10.1137/0216006⟩. 46

BAN, Z.; LIU, J.; CAO, L. Superpixel segmentation using gaussian mixture model. *IEEE Transactions on Image Processing*, v. 27, n. 8, p. 4105–4117, 2018. 30, 46, 64, 89

BAN, Z.; LIU, J.; FOURIAUX, J. Gmmsp on gpu. *Journal of Real-Time Image Processing*, Springer, v. 17, n. 2, p. 245–257, 2020. 30, 46, 89

BEJAR, H. H.; GUIMARÃES, S. J. F.; MIRANDA, P. A. Efficient hierarchical graph partitioning for image segmentation by optimum oriented cuts. *Pattern Recognition Letters*, v. 131, p. 185–192, 2020. ISSN 0167-8655. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0167865520300088⟩. 30, 40, 89

BEJAR, H. H.; MANSILLA, L. A.; MIRANDA, P. A. Efficient unsupervised image segmentation by optimum cuts in graphs. In: SPRINGER. *Iberoamerican Congress on Pattern Recognition*. [S.l.], 2018. p. 359–367. 40

BELÉM, F.; GUIMARÃES, S. J. F.; FALCÃO, A. X. Superpixel segmentation by object-based iterative spanning forest. In: SPRINGER. *Iberoamerican Congress on Pattern Recognition*. [S.l.], 2018. p. 334–341. 30, 39, 89

BELÉM, F. C.; GUIMARÃES, S. J. F.; FALCÃO, A. X. Superpixel segmentation using dynamic and iterative spanning forest. *IEEE Signal Processing Letters*, v. 27, p. 1440–1444, 2020. 30, 38, 39, 60, 64, 89

BELÉM, F. C. et al. Towards a simple and efficient object-based superpixel delineation framework. In: IEEE. *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2021. p. 346–353. 27, 30, 39, 64, 89

BENESOVA, W.; KOTTMAN, M. Fast superpixel segmentation using morphological processing. In: *Conference on Machine Vision and Machine Learning*. [S.l.: s.n.], 2014. p. 67–1. 14, 21, 22, 25

BERGH, M. Van den et al. Seeds: Superpixels extracted via energy-driven sampling. In: FITZGIBBON, A. et al. (Ed.). *Computer Vision – ECCV 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 13–26. ISBN 978-3-642-33786-4. 21, 33, 34, 64, 89

BEUCHER, S. The watershed transformation applied to image segmentation. *Scanning Microscopy*, v. 1992, n. 6, p. 28, 1992. 37

BHUGRA, S. et al. Unsupervised learning of affinity for image segmentation: An inpainting based approach. In: IEEE. *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)*. [S.l.], 2021. p. 1–6. 30, 50, 51, 89

BOBBIA, S. et al. Iterative boundaries implicit identification for superpixels segmentation: a real-time approach. *IEEE Access*, IEEE, v. 9, p. 77250–77263, 2021. 30, 34, 60, 64, 89

BOYD, S. et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, v. 3, n. 1, p. 1–122, 2011. ISSN 1935-8237. Disponível em: ⟨http://dx.doi.org/10.1561/2200000016⟩. 44

BRAGANTINI, J. et al. Graph-based image segmentation using dynamic trees. In: SPRINGER. *Iberoamerican Congress on Pattern Recognition*. [S.l.], 2018. p. 470–478. 39

BUYSSENS, P.; GARDIN, I.; RUAN, S. Eikonal based region growing for superpixels generation: Application to semi-supervised real time organ segmentation in ct images. *IRBM*, Elsevier, v. 35, n. 1, p. 20–26, 2014. ISSN 1959-0318. Biomedical image segmentation using variational and statistical approaches. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S1959031813001437⟩. 21, 64, 89

CHAI, D. Rooted spanning superpixels. *International Journal of Computer Vision*, Springer, v. 128, n. 12, p. 2962–2978, 2020. 30, 38, 64, 89

CHANG, J.-R.; CHEN, Y.-S. Pyramid stereo matching network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 5410–5418. 52

CHEN, J.; LI, Z.; HUANG, B. Linear spectral clustering superpixel. *IEEE Transactions on Image Processing*, v. 26, n. 7, p. 3317–3330, 2017. 30, 34, 37, 89

CHEN, J. et al. Wld: A robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 9, p. 1705–1720, 2010. 35

CHEN, L.-C. et al. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 50

CHEN, L.-C. et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. 33, 34

COMANICIU, D.; MEER, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 5, p. 603–619, 2002. 51, 54

CONDORI, M. A. et al. An extension of the differential image foresting transform and its application to superpixel generation. *Journal of Visual Communication and Image Representation*, v. 71, p. 102748, 2020. ISSN 1047-3203. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S1047320319303694⟩. 39

CONRAD, C.; MERTZ, M.; MESTER, R. Contour-relaxed superpixels. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 280–293. ISBN 978-3-642-40395-8. 21, 64, 89

CUEVAS-VELASQUEZ, H.; GALLEGO, A.-J.; FISHER, R. B. Segmentation and 3d reconstruction of rose plants from stereoscopic images. *Computers and Electronics in Agriculture*, v. 171, p. 105296, 2020. ISSN 0168-1699. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0168169919323919⟩. 13

DELONG, A. et al. Fast approximate energy minimization with label costs. *International journal of computer vision*, Springer, v. 96, n. 1, p. 1–27, 2012. 32

DI, S. et al. Image superpixel segmentation based on hierarchical multi-level li-slic. *Optics & Laser Technology*, Elsevier, v. 135, p. 106703, 2021. 30, 40, 89

DIJKSTRA, E. W. et al. A note on two problems in connexion with graphs. *Numerische mathematik*, v. 1, n. 1, p. 269–271, 1959. 45

FALCÃO, A. X.; STOLFI, J.; LOTUFO, R. de A. The image foresting transform: Theory, algorithms, and applications. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 26, n. 1, p. 19–29, 2004. 30, 38, 39, 89

FANG, L. et al. Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 53, n. 8, p. 4186–4201, 2015. 13

FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient graph-based image segmentation. *International journal of computer vision*, Springer, v. 59, n. 2, p. 167–181, 2004. 51, 52

FIELD, D. J. Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, Optical Society of America, v. 4, n. 12, p. 2379–2394, 1987. 54

FRANCIS, J.; BABURAJ, M.; GEORGE, S. N. An $l_{1/2}$ and graph regularized subspace clustering method for robust image segmentation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM New York, NY, v. 18, n. 2, p. 1–24, 2022. 30, 44, 89

GALVÃO, F. L.; FALCÃO, A. X.; CHOWDHURY, A. S. Risf: recursive iterative spanning forest for superpixel segmentation. In: IEEE. *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2018. p. 408–415. 30, 40, 89

GALVÃO, F. L.; GUIMARÃES, S. J. F.; FALCÃO, A. X. Image segmentation using dense and sparse hierarchies of superpixels. *Pattern Recognition*, v. 108, p. 107532, 2020. ISSN 0031-3203. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0031320320303356⟩. 40

GAUR, U.; MANJUNATH, B. S. Superpixel embedding network. *IEEE Transactions on Image Processing*, v. 29, p. 3199–3212, 2020. 30, 48, 49, 89

GIRAUD, R.; TA, V.-T.; PAPADAKIS, N. Evaluation framework of superpixel methods with a global regularity measure. *Journal of Electronic Imaging*, SPIE, v. 26, n. 6, p. 061603, 2017. 14, 23, 24, 25

GIRAUD, R.; TA, V.-T.; PAPADAKIS, N. Robust superpixels using color and contour features along linear path. *Computer Vision and Image Understanding*, Elsevier, v. 170, p. 1–13, 2018. 22, 30, 36, 64, 89

GODARD, C.; AODHA, O. M.; BROSTOW, G. J. Unsupervised monocular depth estimation with left-right consistency. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 270–279. 47, 48

GONG, J. et al. Superpixel segmentation via contour optimized non-iterative clustering. In: SPRINGER. *International Conference on Neural Computing for Advanced Applications*. [S.l.], 2021. p. 645–658. 30, 36, 89

GONG, Y.; SINNOTT, R. O.; RIMBA, P. Rt-dbscan: Real-time parallel clustering of spatio-temporal data using spark-streaming. In: SHI, Y. et al. (Ed.). *Computational Science – ICCS 2018*. Cham: Springer International Publishing, 2018. p. 524–539. ISBN 978-3-319-93698-7. 30, 37, 89

GRADY, L. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 28, n. 11, p. 1768–1783, 2006. 30, 43, 64, 65, 89

GUAN, J. et al. Peak-graph-based fast density peak clustering for image segmentation. *IEEE Signal Processing Letters*, IEEE, v. 28, p. 897–901, 2021. 30, 42, 64, 89

GUEVARA, A.; CONRAD, C.; MESTER, R. Boosting segmentation results by contour relaxation. In: IEEE. *2011 18th IEEE International Conference on Image Processing*. [S.l.], 2011. p. 1405–1408. 89

GUIGUES, L.; COCQUEREZ, J. P.; MEN, H. L. Scale-sets image analysis. *International Journal of Computer Vision*, Springer, v. 68, n. 3, p. 289–317, 2006. 29

GUPTA, A. K. et al. Awks: adaptive, weighted k-means-based superpixels for improved saliency detection. *Pattern Analysis and Applications*, Springer, v. 24, n. 2, p. 625–639, 2021. 30, 33, 89

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. 41, 54

HOU, X.; ZHANG, L. Saliency detection: A spectral residual approach. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2007. p. 1–8. 32

HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018. 53

HUANG, J.-Y.; DING, J.-J. Generic image segmentation in fully convolutional networks by superpixel merging map. In: *Proceedings of the Asian Conference on Computer Vision*. [S.l.: s.n.], 2020. 30, 50, 89

HUANG, J.-Y.; DING, J.-J.; HUANG, P.-C. Learning based superpixel merging model for image segmentation. In: IEEE. *2021 29th European Signal Processing Conference (EUSIPCO)*. [S.l.], 2021. p. 661–665. 30, 54, 89

HUANG, J. Z. et al. Automated variable weighting in k-means type clustering. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 27, n. 5, p. 657–668, 2005. 30, 33, 89

JAMPANI, V. et al. Superpixel sampling networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 352–368. 30, 47, 50, 52, 53, 54, 55, 89

JI, Z. et al. A spatially constrained generative asymmetric gaussian mixture model for image segmentation. *Journal of Visual Communication and Image Representation*, Elsevier, v. 40, p. 611–626, 2016. 89

KANG, X.; ZHU, L.; MING, A. Dynamic random walk for superpixel segmentation. *IEEE Transactions on Image Processing*, v. 29, p. 3871–3884, 2020. 30, 38, 64, 89

KE-CHEN, S. et al. Research and perspective on local binary pattern. *Acta Automatica Sinica*, Elsevier, v. 39, n. 6, p. 730–744, 2013. 37

KELM, A. P.; RAO, V. S.; ZÖLZER, U. Object contour and edge detection with refinecontournet. In: SPRINGER. *International Conference on Computer Analysis of Images and Patterns*. [S.l.], 2019. p. 246–258. 50, 54

KONG, S.; FOWLKES, C. C. Recurrent pixel embedding for instance grouping. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 9018–9028. 30, 49, 89

KRAUSE, A.; PERONA, P.; GOMES, R. Discriminative clustering by regularized information maximization. *Advances in neural information processing systems*, v. 23, 2010. 30, 47, 89

KWAK, S.; HONG, S.; HAN, B. Weakly supervised semantic segmentation using superpixel pooling network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2017. v. 31, n. 1. 47

LEE, H. et al. Structure-texture decomposition of images with interval gradient. *Computer Graphics Forum*, v. 36, n. 6, p. 262–274, 2017. Disponível em: ⟨https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12875⟩. 31

LEMPITSKY, V.; VEDALDI, A.; ULYANOV, D. Deep image prior. In: IEEE. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.], 2018. p. 9446–9454. 30, 47, 89

LEVINSHTEIN, A. et al. Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 31, n. 12, p. 2290–2297, 2009. 20, 21, 23

LI, H. et al. Superpixel segmentation based on spatially constrained subspace clustering. *IEEE Transactions on Industrial Informatics*, IEEE, v. 17, n. 11, p. 7501–7512, 2020. 30, 43, 44, 89

LI, X. et al. Cluster-based fine-to-coarse superpixel segmentation. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 102, p. 104281, 2021. 30, 35, 89

LI, Z.; CHEN, J. Superpixel segmentation using linear spectral clustering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. 64, 89

LI, Z.; WU, X.-M.; CHANG, S.-F. Segmentation using superpixels: A bipartite graph partitioning approach. In: IEEE. *2012 IEEE conference on computer vision and pattern recognition*. [S.l.], 2012. p. 789–796. 51

LIANG, Y. et al. Robust video object segmentation via propagating seams and matching superpixels. *IEEE Access*, IEEE, v. 8, p. 53766–53776, 2020. 13

LIN, Q.; ZHONG, W.; LU, J. Deep superpixel cut for unsupervised image segmentation. In: IEEE. *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.], 2021. p. 8870–8876. 30, 51, 89

LIU, G.; DUAN, J. Rgb-d image segmentation using superpixel and multi-feature fusion graph theory. *Signal, Image and Video Processing*, Springer, v. 14, n. 6, p. 1171–1179, 2020. 30, 32, 89

LIU, M.-Y. et al. Entropy rate superpixel segmentation. In: *CVPR 2011*. [S.l.: s.n.], 2011. p. 2097–2104. 21, 23, 30, 64, 89

LIU, Y. et al. Del: Deep embedding learning for efficient image segmentation. In: *IJCAI*. [s.n.], 2018. p. 864–870. Disponível em: ⟨https://doi.org/10.24963/ijcai.2018/120⟩. 30, 89

LOKE, S. C. et al. Accelerated superpixel image segmentation with a parallelized dbscan algorithm. *Journal of Real-Time Image Processing*, Springer, v. 18, n. 6, p. 2361–2376, 2021. 30, 37, 89

LU, C.-Y. et al. Robust and efficient subspace segmentation via least squares regression. In: SPRINGER. *European conference on computer vision*. [S.l.], 2012. p. 347–360. 44

MA, D. et al. Convex and compact superpixels by edge-constrained centroidal power diagram. *IEEE Transactions on Image Processing*, IEEE, v. 30, p. 1825–1839, 2020. 30, 45, 89

MACHAIRAS, V. et al. Waterpixels. *IEEE Transactions on Image Processing*, IEEE, v. 24, n. 11, p. 3707–3716, 2015. 20, 23

MAIERHOFER, G. et al. Peekaboo-where are the objects? structure adjusting superpixels. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2018. p. 3693–3697. 30, 32, 89

MANSILLA, L. A.; MIRANDA, P. A. Image segmentation by oriented image foresting transform: Handling ties and colored images. In: IEEE. *2013 18th International Conference on Digital Signal Processing (DSP)*. [S.l.], 2013. p. 1–6. 30, 40, 41, 89

MANSILLA, L. A. C.; MIRANDA, P. A. V. Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. In: *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2016. p. 289–296. 60, 65

MARTIN, D. R.; FOWLKES, C. C.; MALIK, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 26, n. 5, p. 530–549, 2004. 21, 22, 38, 65

MATHIEU, B.; CROUZIL, A.; PUEL, J. B. Oversegmentation methods: a new evaluation. In: SPRINGER. *Iberian Conference on Pattern Recognition and Image Analysis*. [S.l.], 2017. p. 185–193. 20

MESTER, R.; CONRAD, C.; GUEVARA, A. Multichannel segmentation using contour relaxation: fast super-pixels and temporal propagation. In: SPRINGER. *Scandinavian conference on image analysis*. [S.l.], 2011. p. 250–261. 89

MIRANDA, P. A.; MANSILLA, L. A. Oriented image foresting transform segmentation by seed competition. *IEEE Transactions on Image Processing*, IEEE, v. 23, n. 1, p. 389–398, 2013. 40, 41

MOORE, A. P. et al. Superpixel lattices. In: IEEE. *2008 IEEE conference on computer vision and pattern recognition*. [S.l.], 2008. p. 1–8. 14, 25, 65

NEUBERT, P.; PROTZEL, P. Superpixel benchmark and comparison. In: *Proc. Forum Bildverarbeitung*. [S.l.: s.n.], 2012. v. 6, p. 1–12. 18, 19, 21, 23, 27, 65

PARIHAR, V. R.; BOVEIRI, H. R. A survey and comparative analysis on image segmentation techniques. *Image Segmentation*, p. 1, 2018. 18

PATHAK, D. et al. Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2536–2544. 51

PENG, B.; ZHANG, L.; ZHANG, D. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, v. 46, n. 3, p. 1020–1038, 2013. ISSN 0031-3203. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0031320312004219⟩. 18

PENG, H.; AVILES-RIVERO, A. I.; SCHÖNLIEB, C.-B. Hers superpixels: Deep affinity learning for hierarchical entropy rate segmentation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. [S.l.: s.n.], 2022. p. 217–226. 27, 30, 41, 64, 69, 89

POLYA, G. *Mathematics and plausible reasoning, Volume 1: Induction and analogy in mathematics*. [S.l.]: Princeton University Press, 2020. 19

QIAO, N.; DI, L. An improved method of linear spectral clustering. *Multimedia Tools and Applications*, Springer, v. 81, n. 1, p. 1287–1311, 2022. 30, 34, 89

QIN, X. et al. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, Elsevier, v. 106, p. 107404, 2020. 39

RAMADAN, H.; LACHQAR, C.; TAIRI, H. A survey of recent interactive image segmentation methods. *Computational Visual Media*, v. 6, n. 4, p. 355–384, 2020. Disponível em: ⟨http://cvm.tsinghuajournals.com/EN/abstract/article_155638.shtml⟩. 18

REN, L.; ZHAO, L.; WANG, Y. A superpixel-based dual window rx for hyperspectral anomaly detection. *IEEE Geoscience and Remote Sensing Letters*, IEEE, v. 17, n. 7, p. 1233–1237, 2019. 13

REN, X.; MALIK, J. Learning a classification model for segmentation. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2003. v. 2, p. 10. 15, 18, 89

RODRIGUEZ, A.; LAIO, A. Clustering by fast search and find of density peaks. *Science*, v. 344, n. 6191, p. 1492–1496, 2014. Disponível em: ⟨https://www.science.org/doi/abs/10.1126/science.1242072⟩. 30, 42, 43, 89

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241. 49

SCHICK, A.; FISCHER, M.; STIEFELHAGEN, R. Measuring and evaluating the compactness of superpixels. In: IEEE. *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. [S.l.], 2012. p. 930–934. 13, 19, 22, 23, 65

SCHICK, A.; FISCHER, M.; STIEFELHAGEN, R. An evaluation of the compactness of superpixels. *Pattern Recognition Letters*, Elsevier, v. 43, p. 71–80, 2014. 13, 19, 23, 27

SELLARS, P.; AVILES-RIVERO, A. I.; SCHÖNLIEB, C.-B. Superpixel contracted graph-based learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 58, n. 6, p. 4180–4193, 2020. 13

SETHIAN, J. A. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. [S.l.]: Cambridge university press, 1999. v. 3. 64

SHAH, S. A. H. et al. Superpixel segmentation via density peaks. In: IEEE. *2021 4th International Conference on Information and Computer Technologies (ICICT)*. [S.l.], 2021. p. 93–98. 30, 42, 89

SHEN, J. et al. Real-time superpixel segmentation by dbscan clustering algorithm. *IEEE Transactions on Image Processing*, v. 25, n. 12, p. 5933–5942, 2016. 30, 34, 36, 37, 89

SHENG, B. et al. Retinal vessel segmentation using minimum spanning superpixel tree detector. *IEEE transactions on cybernetics*, IEEE, v. 49, n. 7, p. 2707–2719, 2018. 13

SHI, J.; MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 8, p. 888–905, 2000. 44

SHI, J. et al. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 38, n. 4, p. 717–729, 2015. 60

SHU, G.; DEHGHAN, A.; SHAH, M. Improving an object detector and extracting regions using superpixels. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 3721–3727. 13

SILVA, C.; BOUWMANS, T.; FRÉLICOT, C. An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos. In: *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2015*. Berlin, Germany: [s.n.], 2015. Disponível em: ⟨https://hal.archives-ouvertes.fr/hal-01227955⟩. 44

STRASSBURG, J. et al. On the influence of superpixel methods for image parsing. In: *VISAPP*. [S.l.: s.n.], 2015. 21

STUTZ, D. Superpixel segmentation: An evaluation. In: SPRINGER. *German conference on pattern recognition*. [S.l.], 2015. p. 555–562. 20, 23, 27

STUTZ, D.; HERMANS, A.; LEIBE, B. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, Elsevier, v. 166, p. 1–27, 2018. 13, 14, 15, 18, 20, 21, 22, 23, 24, 25, 27, 31, 64, 65

SUZUKI, T. Superpixel segmentation via convolutional neural networks with regularized information maximization. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2020. p. 2573–2577. 30, 47, 48, 64, 89

SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Thirty-first AAAI conference on artificial intelligence*. [S.l.: s.n.], 2017. 50

TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: IEEE. *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. [S.l.], 1998. p. 839–846. 45

TU, W.-C. et al. Learning superpixels with segmentation-aware affinity loss. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 568–576. 30, 50, 54, 89

ULLAH, S.; BHATTI, N.; ZIA, M. Adaptive tuning of slic parameter k. *Multimedia Tools and Applications*, Springer, v. 80, n. 17, p. 25649–25672, 2021. 30, 31, 89

VARGAS-MUÑOZ, J. E. et al. An iterative spanning forest framework for superpixel segmentation. *IEEE Transactions on Image Processing*, IEEE, v. 28, n. 7, p. 3477–3489, 2019. 30, 38, 64, 89

VEDALDI, A.; FULKERSON, B. Vlfeat: An open and portable library of computer vision algorithms. In: *Proceedings of the 18th ACM International Conference on Multimedia*. New York, NY, USA: [s.n.], 2010. (MM '10), p. 1469–1472. ISBN 9781605589336. Disponível em: ⟨https://doi.org/10.1145/1873951.1874249⟩. 43

WANG, G.; WEI, Y.; TSE, P. Clustering by defining and merging candidates of cluster centers via independence and affinity. *Neurocomputing*, v. 315, p. 486–495, 2018. ISSN 0925-2312. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0925231218308786⟩. 30, 42, 89

WANG, H. et al. Adaptive nonlocal random walks for image superpixel segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 30, n. 3, p. 822–834, 2020. 30, 43, 89

WANG, J. et al. Superpixel segmentation with squeeze-and-excitation networks. *Signal, Image and Video Processing*, Springer, p. 1–8, 2022. 30, 52, 53, 89

WANG, K.; LI, L.; ZHANG, J. End-to-end trainable network for superpixel and image segmentation. *Pattern Recognition Letters*, v. 140, p. 135–142, 2020. ISSN 0167-8655. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0167865520303421⟩. 30, 46, 89

WANG, L. et al. Learning parallax attention for stereo image super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. 53

WANG, M. et al. Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, Elsevier, v. 56, p. 28–39, 2017. 13, 14, 20, 23, 27

WANG, N.; ZHANG, Y. Adaptive and fast image superpixel segmentation approach. *Image and Vision Computing*, Elsevier, v. 116, p. 104315, 2021. 30, 37, 89

WANG, P. et al. An maize leaf segmentation algorithm based on image repairing technology. *Computers and Electronics in Agriculture*, v. 172, p. 105349, 2020. ISSN 0168-1699. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0168169919323737⟩. 13

WANG, W.; WU, C. Image segmentation by correlation adaptive weighted regression. *Neurocomputing*, v. 267, p. 426–435, 2017. ISSN 0925-2312. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0925231217311748⟩. 30, 44, 89

WANG, X. et al. A global/local affinity graph for image segmentation. *IEEE Transactions on Image Processing*, v. 24, n. 4, p. 1399–1411, 2015. 30, 51, 89

WANG, X. et al. Semasuperpixel: A multi-channel probability-driven superpixel segmentation method. In: IEEE. *2021 IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2021. p. 1859–1863. 30, 33, 89

WEI, X. et al. Superpixel hierarchy. *IEEE Transactions on Image Processing*, IEEE, v. 27, n. 10, p. 4838–4849, 2018. 41, 60, 64, 89

WEST, D. B. et al. *Introduction to graph theory*. [S.l.]: Prentice hall Upper Saddle River, 2001. v. 2. 65

WU, J.; LIU, C.; LI, B. Texture-aware and structure-preserving superpixel segmentation. *Computers & Graphics*, Elsevier, v. 94, p. 152–163, 2021. 27, 30, 31, 32, 89

WU, R. et al. Stereo superpixel segmentation via dual-attention fusion networks. In: IEEE. *2021 IEEE International Conference on Multimedia and Expo (ICME)*. [S.l.], 2021. p. 1–6. 30, 53, 89

WU, X. et al. Superpixel via coarse-to-fine boundary shift. *Applied Intelligence*, Springer, v. 50, n. 7, p. 2079–2092, 2020. 30, 34, 89

XIE, J. et al. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors. *Information Sciences*, v. 354, p. 19–40, 2016. ISSN 0020-0255. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S002002551630158X⟩. 42

XIE, S.; TU, Z. Holistically-nested edge detection. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. 41

YANG, F. et al. Superpixel segmentation with fully convolutional networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020. 27, 30, 52, 53, 64, 89

YAO, J. et al. Real-time coarse-to-fine topologically preserving segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 2947–2955. 21, 33, 64, 89

YIN, H.; GONG, Y.; QIU, G. Side window filtering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. 48

YU, J. et al. Generative image inpainting with contextual attention. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018. 51

YU, Y.; YANG, Y.; LIU, K. Edge-aware superpixel segmentation with unsupervised convolutional neural networks. In: IEEE. *2021 IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2021. p. 1504–1508. 30, 47, 64, 89

YUAN, H. et al. Nonlocal random walks algorithm for semi-automatic 2d-to-3d image conversion. *IEEE Signal Processing Letters*, v. 22, n. 3, p. 371–374, 2015. 30, 43, 89

YUAN, Q. et al. Sin: Superpixel interpolation network. In: SPRINGER. *Pacific Rim International Conference on Artificial Intelligence*. [S.l.], 2021. p. 293–307. 30, 54, 55, 64, 89

YUAN, Y. et al. Superpixels with content-adaptive criteria. *IEEE Transactions on Image Processing*, IEEE, v. 30, p. 7702–7716, 2021. 30, 35, 89

YUAN, Y. et al. Watershed-based superpixels with global and local boundary marching. *IEEE Transactions on Image Processing*, v. 29, p. 7375–7388, 2020. 30, 89

ZHANG, B.; KANG, X.; MING, A. Bp-net: deep learning-based superpixel segmentation for rgb-d image. In: IEEE. *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.], 2021. p. 7433–7438. 30, 55, 89

ZHANG, J. et al. Dynamic spectral residual superpixels. *Pattern Recognition*, Elsevier, v. 112, p. 107705, 2021. 30, 32, 89

ZHANG, J. et al. Spatiotemporal saliency detection based on maximum consistency superpixels merging for video analysis. *IEEE Transactions on Industrial Informatics*, IEEE, v. 16, n. 1, p. 606–614, 2019. 13

ZHANG, Y.; GUO, Q.; ZHANG, C. Simple and fast image superpixels generation with color and boundary probability. *The Visual Computer*, v. 37, p. 1061–1074, 2021. 30, 36, 89

ZHAO, W. et al. An improved image semantic segmentation method based on superpixels and conditional random fields. *Applied Sciences*, MDPI, v. 8, n. 5, p. 837, 2018. 13

ZHOU, X. et al. Ssg: superpixel segmentation and grabcut-based salient object segmentation. *The Visual Computer*, Springer, v. 35, n. 3, p. 385–398, 2019. 13

ZHU, H. et al. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, v. 34, p. 12–27, 2016. ISSN 1047-3203. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S1047320315002035⟩. 18

ZHU, L. et al. Learning the superpixel in a non-iterative and lifelong manner. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 1225–1234. 27, 30, 48, 49, 64, 89

ZOHRIZADEH, F.; KHEIRANDISHFARD, M.; KAMANGAR, F. Image segmentation using sparse subset selection. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2018. p. 1470–1479. 44

APPENDIX A - SUPERPIXEL SEGMENTATION METHODS

Method	Year	Iterative	#Iter.	#Superp.	Connec.	Compact.	Superv.	Color	Time complexity	Initial processing	Main processing	Final processing	Features: Pix.	Features: Mid.	Features: High.	Inspired	Code
SLIC	2012	✓	✓	✓	✓†	✓		CIELAB		Seed sampling	Neighborhood-based clustering	Merging step	✓				⟨https://www.ep.ch/labs/ivrl/research/slic-superpixels/⟩
K-SLIC	2021	✓	✓	✓		✓		RGB		Compute optimum K	Clustering with SLIC		✓			SLIC (2012)	
LSC	2015	✓	✓	✓	✓†	✓		CIELAB	$O(kn+nz)$ ‡‡	Seed sampling	Neighborhood-based clustering	Merging step	✓			SLIC (2012) NCut (2003)	⟨https://jschenthu.weebly.com/projects.html⟩
SCALP	2018	✓	✓	✓	✓	✓		CIELAB		Seed sampling	Neighborhood-based clustering			✓		SLIC (2012)	⟨https://github.com/rgiraud/scalp⟩
TASP	2021	✓	✓	✓				CIELAB		Seed sampling	Neighborhood-based clustering			✓		SLIC (2012)	
MFGS	2020			✓*	✓	✓		CIELAB		Seed sampling	Neighborhood-based clustering	Merging step		✓		SLICO (2012)	
DSR	2021	✓		✓		✓		CIELAB		Seed sampling	Neighborhood-based clustering	Merging step			✓	dSLIC (2018)	
Semasuperpixel	2021	✓	✓	✓	✓†		✓	CIELAB		*arch:* Encoder-decoder *train:* Semantic map	Neighborhood-based clustering	Merging step			✓	SLIC(2012)	
AWkS	2021	✓	✓	✓				CIELAB		Seed sampling	Neighborhood-based clustering	Merging step	✓			W-k-means (2005)	
IBIS, IBIScuda	2021	✓		✓	✓†	✓		CIELAB	$O(n)$	Grid segmentation	Boundary evolution clustering	Merging step	✓			SLIC (2012)	⟨https://github.com/xapha/IBIS⟩, ⟨https://github.com/xapha/IBIS_cuda⟩
SEEDS	2012	✓	✓	✓		✓		CIELAB		Grid segmentation	Boundary evolution clustering		✓				
CRS	2013	✓	✓	✓	✓	✓		YCrCb		Grid segmentation	Boundary evolution clustering			✓		CR (2011, 2011)	
ETPS	2015	✓	✓	✓	✓	✓		RGB		Grid segmentation	Boundary evolution clustering		✓			SEEDS (2012)	⟨https://bitbucket.org/mboben/spixel/src/master/⟩
CFBS	2020	✓		✓	✓	✓		CIELAB		Grid segmentation	Boundary evolution clustering		✓			SLIC (2012)	
SCAC	2021b			✓*	✓	✓		CIELAB		Grid segmentation	Boundary evolution clustering	Boundary evolution clustering		✓		WSBM (2020)	⟨https://github.com/YuanYeNeu/SCAC⟩
LSC-Manhattan	2022	✓		✓	✓	✓				Classification	Boundary evolution clustering				✓	LSC (2017)	
SNIC	2017			✓	✓	✓		CIELAB	$O(n)$	Seed sampling	Dynamic-center-update clustering		✓			SLIC (2012)	⟨https://github.com/achanta/SNIC⟩
CONIC	2021			✓	✓	✓		CIELAB	$O(n)$	Seed sampling	Dynamic-center-update clustering		✓			SNIC (2017), SCALP (2018)	
DRW	2020			✓	✓				$O(n)$	Seed sampling	Dynamic-center-update clustering	Label propagation		✓		RW (2006)	⟨https://github.com/zh460045050/DRW⟩
FCSS	2021	✓	✓*	✓	✓†	✓		CIELAB	$O(n+nt)$§		Dynamic-center-update clustering		✓			SNIC (2017)	
F-DBSCAN	2021			✓	✓			CIELAB	$O(n)$		Dynamic-center-update clustering		✓			RT-DBSCAN (2018)	⟨https://github.com/scloke/DBScanTest⟩
SCBP	2021			✓	✓	✓		RGB	$O(n)$		Dynamic-center-update clustering	Merging step		✓		DBSCAN (2016)	
A-DBSCAN	2021			✓	✓	✓		RGB	$O(n)$	Compute features	Dynamic-center-update clustering	Merging step			✓	DBSCAN (2016)	
ERGC	2014				✓	✓		CIELAB		Seed sampling	Path-based clustering			✓			
ISF	2019	✓	✓	✓	✓	✓		CIELAB	$O(n \log n)$	Seed sampling	Path-based clustering			✓		IFT (2004)	⟨https://www.ic.unicamp.br/afalcao/downloads.html⟩
RSS	2020			✓	✓	✓			$O(n)$	Seed sampling	Path-based clustering			✓		IFT (2004)	⟨https://github.com/dfchai/Rooted-Spanning-Superpixels⟩
DISF	2020	✓		✓	✓			CIELAB	$O(n \log n)$	Seed oversampling	Path-based clustering			✓		ISF 2019	⟨https://github.com/LIDS-UNICAMP/DISF⟩
ODISF	2021	✓		✓	✓			CIELAB	$O(n \log n)$ ‖	Seed oversampling	Path-based clustering				✓	DISF (2020), OISF (2018)	⟨https://github.com/LIDS-UNICAMP/ODISF⟩
SH	2018			✓	✓			RGB	$O(n)$		Hierarchical clustering			✓			⟨https://github.com/semiquark1/boruvka-superpixel⟩
UOIFT	2020			✓	✓			CIELAB		Clustering method	Hierarchical clustering			✓		IFT (2004), OIFT (2013)	
HMLI-SLIC	2021	✓	✓	✓*	✓	✓		CIELAB	$O(nd)$ ¶	Clustering method	Hierarchical clustering	Merging step	✓			SLIC (2012)	
RISF	2018	✓	✓	✓	✓	✓		CIELAB			Hierarchical clustering	Hierarchical region merging		✓		ISF (2019)	
DAL-HERS	2022			✓	✓		✓	RGB	$O(n)$‡	*arch:* Multi-scale Residual CNN *train:* Affinity map	Hierarchical clustering				✓	SEAL (2018), ERS (2011)	⟨https://github.com/hankuipeng/DAL-HERS⟩
PGDPC, SLIC-PGDPC	2021			✓	✓			CIELAB	$O(n \log n)$	Seed sampling	Density-based clustering			✓		DPC (2018)	⟨https://github.com/Guanjunyi/PGDPCforImageSegementation⟩
DPS	2021			✓*				CIELAB		Compute features	Density-based clustering	Clustering method		✓		DP (2014)	
ANRW	2020			✓	✓			CIELAB	$O(n^2)$	Seed sampling	Sparse linear system clustering				✓	NRW (2015)	⟨http://github.com/shenjianbing/ANRW⟩
GL$l_{1/2}$RSC	2022	✓		✓						Clustering method	Sparse linear system clustering	Encoding procedure		✓		CAWR (2017)	
SCSC	2020	✓	✓	✓				RGB		Clustering method	Sparse linear system clustering	Clustering method			✓		
EAM	2020			✓*	✓			RGB	$O(\log^2 n)$	Noise remotion	Regional attributes extraction	Merging step		✓			
ECCPD	2020	✓	✓	✓	✓			RGB		Seed sampling	Polygonal decomposition clustering	Boundary evolution clustering		✓			⟨https://github.com/madongyang-stack/ECCPD⟩
GMMSP	2018	✓	✓	✓*	✓†	✓		CIELAB	$O(n)$		Data distribution-based clustering	Merging step		✓		SCGAGMM (2016)	⟨https://github.com/ahban/GMMSP-superpixel⟩
gGMMSP	2020	✓	✓	✓*	✓†	✓		CIELAB	$O(n)$ **		Data distribution-based clustering	Merging step		✓		GMMSP (2018)	⟨http://github.com/ahban/gGMMSP⟩
ERS	2011				✓	✓		RGB			Graph-based clustering		✓				⟨https://github.com/akanazawa/collective-classification/tree/master/segmentation⟩
E2E-SIS	2020			✓	✓†		✓	CIELAB			*arch:* FCN *train:* Superpixels	Superpixel pooling layer and merging step			✓	DEL (2018), SSN (2018)	
ss-RIM	2020			✓*				RGB			*arch:* FCN *train:* Image reconstruction and Superpixels				✓	DIP (2018), RIM (2010)	⟨https://github.com/DensoITLab/ss-with-RIM⟩
EW-RIM	2021			✓	✓	✓		RBG			*arch:* FCN *train:* Image reconstruction and Superpixels				✓	ss-RIM (2020), DIP (2018)	⟨https://github.com/yueyu-stu/EdgeAwareSpixel⟩
SEN	2020			✓				RGB		*arch:* Encoder-Decoder *train:* Deep embeddings	Clustering method				✓	RPEIG (2018)	
DMMSS-FCN	2020				✓		✓	RGB			*arch:* Encoder-decoder *train:* Edge map decision				✓		⟨https://drive.google.com/drive/folders/1NcEsdGh7OkuyTJk9Kx_U4N33f-BIglRP⟩
UDAG	2021				✓			CIELAB		Clustering method	*arch:* Encoder-Decoder *train:* Inpainting	Clustering method			✓	GL Graph (2015)	
SuperAE-DSC	2021	✓	✓	✓	✓			RGB		*arch:* Autoencoder CNN *train:* Image reconstruction and Superpixels	Clustering method	Differentiable clustering			✓		
SSFCN	2020			✓*			✓	CIELAB			*arch:* Encoder-Decoder *train:* Superpixels	Merging step			✓	SSN (2018)	⟨https://github.com/fuy34/superpixel/superpixel_fcn⟩
SENSS	2022			✓*	✓	✓	✓	CIELAB			*arch:* Encoder-Decoder *train:* Superpixels				✓	SSFCN (2020)	
DAFnet	2021			✓	✓		✓	CIELAB			*arch:* Weight-shared CNN *train:* Superpixels				✓	SSFCN (2020)	
LNS-net	2021			✓		✓		LAB/RGB			*arch:* FCN *train:* Image reconstruction and Superpixels	Merging step			✓		⟨https://github.com/zh460045050/LNSNet⟩
DMMSS	2021	✓			✓					Clustering method	*arch:* FCN *train:* Binary classification	*arch:* FCN *train:* Binary classification			✓		⟨https://drive.google.com/drive/folders/16Q4zXldkUgnTQJI7lvouw6QcJ3YjNa82⟩
SIN	2021a			✓*	✓		✓				*arch:* Interpolation Network *train:* Superpixels				✓		⟨https://github.com/yuanqqq/SIN⟩
BP-net	2021			✓			✓	RGB-D		Seed sampling	*arch:* Multi-scale CNN *train:* Boundary map and superpixels	Merging step			✓		

* Partially † With post-processing ‡ Time complexity in HERS module § t is the number of relocations ¶ d is the number of hierarchy levels ‖ without the saliency map computation ** without paralelization †† i is the number of iterations ‡‡ k is the number of iterations and z represents the number of small isolated superpixels to be merged.