Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.4242

Learning on hierarchical trees with Random Forest

Raquel Almeida [**PUC Minas and IRISA** | ***raquel1908@gmail.com***]
Laurent Amsaleg [**PUC Minas** | ***Laurent.Amsaleg@irisa.fr***]
Zenilton Kleber G. do Patrocínio Júnior [**PUC Minas** | ***zenilton@pucminas.br***]
Simon Malinowski [**IRISA** | ***simon.malinowski@irisa.fr***]
Ewa Kijak [**IRISA** | ***ewa.kijak@irisa.fr***]
Silvio Jamil Ferzoli Guimarães [**PUC Minas** | ***sjamil@pucminas.br***]

Laboratory of Image and Multimedia Data Science (IMScience), Graduate Program in Informatics, Pontifical Catholic University of Minas Gerais, Av. Dom José Gaspar, 500 - CoraçãoEucarístico, Belo Horizonte, MG, 30535-901, Brazil

Received: 12 March 2024 • **Accepted:** 27 August 2025 • **Published:** 26 January 2026

Abstract Hierarchies, as described in mathematical morphology, represent nested regions of interest and provide mechanisms to create coherent data organization. They facilitate high-level analysis and management of large amounts of data. Represented as hierarchical trees, they have formalisms intersecting with graph theory and generalizable applications. Due to the deterministic algorithms, the multiform representations, and the absence of a direct quality evaluation, it is hard to insert hierarchical information into a learning framework and benefit from the recent advances. Researchers usually tackle this problem by refining the hierarchies for a specific media and assessing their quality for a particular task. The downside of this approach is that it depends on the application, and the formulations limit the generalization to similar data. This work aims to create a learning framework that can operate with hierarchical data and is agnostic to the input and application. The idea is to transform the data into a regular representation required by most learning models while preserving the rich information in the hierarchical structure. The proposed methods use edge-weighted image graphs and hierarchical trees as input, and they evaluate different proposals on the edge detection and segmentation tasks. The learning model is the Random Forest, a fast and scalable method for working with high-dimensional data. Results demonstrate that it is possible to create a learning framework dependent only on the hierarchical data that presents a state-of-the-art performance in multiple tasks.

Keywords: Morphological hierarchies, Random Forest, Machine learning, Graphs, Image processing

1 Introduction

Hierarchies are an inherent property composing several elements in real life, relating to how we perceive patterns, scenes, and movement [Marr, 1982]. According to Kurzweil [2013], a pattern identifier exists in the core of our visual perception, operating hierarchically to recognize parts, objects, and abstract concepts.

Hierarchies are broadly defined in the literature and could represent different concepts. For instance, literature presents hierarchies as a method's abstraction [Ilin *et al*., 2017], a description of model architectures [Liu *et al*., 2019], and a form to organize features [Lin *et al*., 2017] or concepts [Fan *et al*., 2017]. This broad definition reinforces that hierarchies are the natural organization form of data. The perceptual hierarchy is difficult to translate to computer models, but in visual media processing, mathematical morphology has an edge in defining, creating, and manipulating hierarchies.

Morphological hierarchies are arranged structures of nested regions that are easy to navigate and interpret, remaining very popular since their creation [Beucher, 1994; Najman and Schmitt, 1996; Krishnammal *et al*., 2022; Makrogiannis *et al*., 2021]. The nested regions of interest provide navigation and merging operations to build more semantically significant objects from lower-level instances. In multimedia processing, the region delineation considers the media's building blocks, such as pixels, voxels, frequency transformations, or sound waves [Bosilj *et al*., 2018]. In the hierarchical theory, image processing is undoubtedly the principal definition space and applications, most notably image segmentation [Soille and Najman, 2012] and remote sensing [Maia *et al*., 2021]. Nevertheless, similarly structured visual media, such as hyperspectral [Tochon *et al*., 2018] and multi-modal images [Kiran and Serra, 2015], videos [Xu *et al*., 2012], and structured time measurements [Nguyen *et al*., 2019], are also processed with hierarchical algorithms.

In practical applications, morphological hierarchies require thorough preprocessing of the data [Nguyen *et al*., 2019] and strategies to deal with issues like over/under-partitioning of the space [Zwettler and Backfrieder, 2015] or selecting an ideal number of regions [Meyer, 2001]. At the same time, hierarchies produce multiform representations, their algorithms are primarily deterministic, and there is no direct way to evaluate their quality. Therefore, it is difficult to generalize a successful approach to other media and tasks.

For a generalization of the media type, most challenges regard the characterization of the information, mainly: the media data presents different characteristics, and the media's building blocks composing the regions have different connotations. These differences in form and connotation eventually become limiting factors. The models created to solve a problem could only deal with that particular data type, despite their eventual similarities. In terms of task, the generalization is challenging due to the lack of a measure assessing

the quality of a hierarchy, which requires an empirical refinement through a series of trial-and-error fittings for a particular application.

Furthermore, creating a framework to operate on hierarchies presents some considerable additional challenges besides the problem of generalization, namely: (i) the product of the hierarchies is multiform, meaning they have different sizes, components, and interpretations; and (ii) the same data could create multiple hierarchical structures depending on the hierarchical operators and constraints. Therefore, applying the morphological hierarchies in an agnostic learning framework requires a strategy to overcome the determinism, the quality assessment, and the heterogeneous aspects.

This work aims to create a learning framework that can operate on hierarchical data and is agnostic to the media type and task. In doing so, it must deal with the generalization challenges and place a strategy to conform the hierarchical information to a learning framework. It requires: (i) defining an appropriate representation shared among most media types; (ii) providing a way to retain the information presented in the original media; and (iii) avoiding assumptions on the data source in the task definition. Each of these requirements guides the design of the proposed framework, shaping the choices in representation, modeling, and evaluation.

In this context, graphs serve as a unifying formalism: they are structures used to represent objects, with graph theory focusing on how these objects relate to each other [Bondy *et al*., 1976]. In relation to the outlined requirements: (i) graphs can be constructed to represent a wide variety of media types, fulfilling the need for a shared representation [Ortega *et al*., 2018]; (ii) graph structures and their attributes enable the retention and analysis of essential information from the original data; and (iii) because graphs are agnostic to domain or modality, they enable modeling and learning without imposing assumptions specific to a given data source or task. Importantly, hierarchies can be represented as graphs, specifically as hierarchical trees [Cousty *et al*., 2013]. Thus, both media and their hierarchical organization can be described and processed within the same theoretical and computational framework, enabling generalization across applications.

The literature in pattern recognition [Wu *et al*., 2020; Zhang *et al*., 2024], network science [Newman, 2018], and bioinformatics [Zitnik *et al*., 2019] demonstrates the wide applicability of graph structures. Our proposal aligns with these broader perspectives: by abstracting the core operations to work on general graphs and their hierarchies, the method can be directly adapted to non-image domains, such as temporal event graphs in time-series data [Isella *et al*., 2011], video data [Sakarya and Telatar, 2010; dos Santos Belo *et al*., 2016], and document structures [Mihalcea and Radev, 2012]. The shared requirement is that the data can be modeled and navigated as a graph [Chen *et al*., 2024; Tong *et al*., 2005], allowing the learning framework to operate agnostically with respect to the specific application domain.

While the proposed framework is, in principle, applicable to diverse data types and domains, in this work we focus our experimental validation on two image analysis tasks: edge detection and segmentation. Future work will explore applications in other domains to further demonstrate the method's generality.

A critical aspect in hierarchical studies is understanding how the media's building blocks relate at the low level to form homogeneous regions. Visual data are organized structures, and information such as color, spatial distance, or variance defines homogeneity. And although defining homogeneous regions and their connotations are particular for each media type, the grouping strategy and their storage in the hierarchical structure follow the same rules.

The main challenge in this proposal concerns the regular representation required by most machine learning algorithms. The regular representation is inherently opposed to the unconstrained nature of graphs. Hence, the proposed strategy is to represent the graph's components as vectors of selected attributes and assess its capability to retain the information modeled in the hierarchical trees while remaining discriminant for a task.

Using a selection of graph attributes as input to the learning framework allows it to be agnostic to the media type. Modeling at the graph component level enables each entry to be assigned a task label without imposing assumptions on the data source. Previous studies [Almeida *et al*., 2021, 2022] have demonstrated this strategy on non-hierarchical image graphs, introducing a graph-based image gradient operator (GIG) that produces gradients delineating strong object contours as well as minor components, textures, and uniform regions. Extensive analysis of these gradients in the segmentation task, using them as input for the watershed method [Beucher, 1994], demonstrated that GIG achieved good segmentation performance comparable to leading edge-map methods, thereby validating the graph attribute selection strategy.

Nonetheless, depending on the modeling choices of the graphs, it can create a particular structured space known as grid graphs close to the spatial domain of the media. Presuming generalization on a grid graph can be deceptive, and more than the structural information may be necessary for a discriminative representation. However, modeling the graphs from the hierarchical structure provides a non-regular characterization of regions with notions of order and navigation.

Considering the semantical arrangement within the hierarchies, any proposal must retain the structures and ordering relations consistent with the hierarchical principles. Also, because there is no direct way to evaluate the quality of a hierarchy, the learning model should support easy navigation between tasks to assess various aspects through experimentation.Furthermore, the framework should rely on something other than strategies to adequately prepare the data for a specific task or refine the structures in a particular application.

Although recent advances in deep learning, such as Graph Neural Networks (GNNs) [Wu *et al*., 2020], enable end-to-end learning from graph-structured data, these methods typically require large annotated datasets and often yield models that are less transparent or interpretable. In contrast, our framework extracts and uses well-defined hierarchical attributes, allowing the model to directly leverage the semantic meaning encoded in the structure. This promotes both data efficiency and model transparency, which are critical in many scientific and practical domains.

In summary, the main contributions of this work may be described as follows: (i) proposal of a learning framework

that can operate with hierarchical data and is agnostic to the input, considering that all multimedia data can be modeled as graphs, and application; (ii) discussion about the topology of the hierarchical structures alone could be used, and it is possible to directly insert the hierarchical structures in a learning framework and benefit from the embedded information to create a model for visual tasks that is agnostic to the media type and task; (iii) a fast and scalable method for working with high-dimensional data thanks to the learning model; (iv) proposal of a machine learning method on a non-regular graph for image processing provided by hierarchical structures; and (v) development of experiments with the trivial, topological, and regional approaches in two image tasks: edge detection and image segmentation.

We organized this work as follows. Section 3 provides the theoretical background on graphs and hierarchies. Section 4 discusses common generalization issues with hierarchies. Section 5 presents the proposed methodology, followed by the experiments and results in Section 6. Finally, Section 7 discusses the main findings of this work, while Section 8 draws some conclusions and future work proposals.

2 Related work

The study of hierarchical representations spans mathematical morphology, graph-based modeling, and modern machine learning. This section reviews foundational and recent work in each area, emphasizing advances in media-agnostic and interpretable learning on hierarchies. Our aim is to situate the proposed framework within this context and clarify its distinctions and advances over prior work.

Mathematical morphology provides the theoretical and algorithmic foundations for hierarchical representations. Classical techniques such as the watershed transform [Beucher, 1994; Najman and Schmitt, 1996; Krishnammal *et al*., 2022; Makrogiannis *et al*., 2021] and its hierarchical variants [Soille and Najman, 2012; Meyer, 2001] enable multiscale decompositions of images, supporting efficient region merging, filtering, and navigation. Developments including ultrametric watersheds, saliency trees, and related structures [Krishnammal *et al*., 2022; Makrogiannis *et al*., 2021; Cousty *et al*., 2013] formalize inclusion and adjacency relationships among regions. These methodologies have evolved to address increasingly complex data—multi-channel, volumetric, and multi-modal sources—with applications in remote sensing [Bosilj *et al*., 2018], biomedical imaging [Kiran and Serra, 2015], and other fields [Maia *et al*., 2021]. Critically, morphological hierarchies prioritize structural and relational properties over purely pixel-based approaches, underpinning meaningful and interpretable representations.

Machine learning approaches that leverage hierarchical representations primarily use them to define or select spatial regions for feature extraction, especially in medical [Grossiord *et al*., 2017; Padilla *et al*., 2021] and remote sensing domains [Hu *et al*., 2021]. Max-tree and tree-of-shapes representations segment volumetric medical data [Grossiord *et al*., 2017; Padilla *et al*., 2021], while superpixels and binary partition trees structure high-resolution aerial images [Hu *et al*., 2021]. Hierarchies also serve as spatial masks or filters to isolate relevant regions before extracting features, as in 3D point cloud classification using elevation-based and quasi-flat zone hierarchies [Serna and Marcotegui, 2014], or text detection with multi-channel max-trees [Sun *et al*., 2015]. Across these strategies, features describing region shape, intensity, texture, or geometry are extracted from pixels grouped by each hierarchical node and used to train classifiers such as Random Forests [Grossiord *et al*., 2017; Hu *et al*., 2021], Random Walks [Padilla *et al*., 2021], SVMs [Serna and Marcotegui, 2014; Sun *et al*., 2015; Díaz *et al*., 2009], or clustering models. To manage the complexity of hierarchical representations, practitioners often filter for stable nodes [Padilla *et al*., 2021], sample multiple scales [Hu *et al*., 2021], or aggregate features in a bag-of-features model [Clément *et al*., 2018]. However, the effectiveness of these approaches relies heavily on media-derived features, demanding careful, often domain-specific, feature design and limiting both interpretability and generalization.

The preceding approaches illustrate that hierarchical representations have predominantly served as a means to define regions for feature extraction in classical pipelines. To transcend these modality- and domain-specific constraints, recent work has shifted towards graph-based models, offering a unified formalism for both local and global relationships [Bondy *et al*., 1976; Ortega *et al*., 2018; Cousty *et al*., 2013]. In this framework, hierarchical trees and region adjacency graphs are particular cases of edge-weighted graphs, broadening the scope to domains such as bioinformatics [Zitnik *et al*., 2019], event data [Isella *et al*., 2011], and document structures [Mihalcea and Radev, 2012]. Machine learning on graphs enables algorithms to operate in a source-agnostic manner, with vertices and edges representing entities and their relationships [Wu *et al*., 2020; Zhang *et al*., 2024].

However, adapting graph representations to learning frameworks introduces new challenges. Graphs tend to be large and densely connected, and their arbitrary, non-Euclidean structure complicates the use of standard algorithms that expect fixed, systematic inputs [Makarov *et al*., 2021]. To address these challenges, a range of strategies have emerged, including graph embeddings [Perozzi *et al*., 2014; Grover and Leskovec, 2016; Wang *et al*., 2016], deep graph learning [Scarselli *et al*., 2009; Micheli, 2009; Wu *et al*., 2020], and feature vectorization approaches. Each of these seeks to preserve topological and semantic properties while enabling efficient learning. Our framework builds on this lineage by focusing on interpretable, structural features derived directly from the hierarchy itself.

Recent years have seen rapid growth in deep learning methods on graphs, with architectures such as graph convolutional and attention networks [Wu *et al*., 2020; Zhang *et al*., 2024]. These have been adapted to image, video, and multi-modal data by encoding pixels, superpixels, or spatial regions as graph nodes, and modeling relationships via edges [Chen *et al*., 2020; Ji *et al*., 2020; Huang *et al*., 2020; Selvan *et al*., 2020]. Applications now range from classical tasks like image segmentation and classification [Ji *et al*., 2020; Selvan *et al*., 2020] to semantic scene understanding and visual reasoning [Luo *et al*., 2019; Yang *et al*., 2020; Jing *et al*., 2020]. In each case, graph structure is key to capturing complex interactions not accessible in regular grids. Still, these models

face obstacles: constructing meaningful graphs from media data requires non-trivial choices regarding node grouping, edge definition, and encoding of geometric or temporal relationships [Qi *et al*., 2017; Chuang *et al*., 2018]. Further, most models act as black-boxes, making it difficult to interpret which properties of the structure drive their predictions.

In summary, while hierarchical and graph-based models have provided a solid foundation for representing complex data, and deep learning on graphs has expanded the reach of these methods, a persistent gap remains in learning directly from interpretable, structural features without relying on extensive media-derived attributes or opaque models. The present work seeks to address this gap by proposing a framework for transparent, structural, and media-agnostic learning on hierarchies, extending the landscape of machine learning with a focus on interpretability and broad applicability. The sections that follow describe our approach and demonstrate its practical benefits through rigorous experimental evaluation.

3 Hierarchies and graphs

The hierarchical functions on mathematical morphology are rooted in the algebraic theory of complete lattices, modeling non-linear transformations with set operators to correlate whole sets of values [Najman and Talbot, 2013]. The scale-set theory, a sub-area of mathematical morphology, formalizes the hierarchical principles guiding the morphological operators [Guigues *et al*., 2006]. In the scale-set theory formalization, a structure could be defined as a hierarchy if it follows two **hierarchical principles**: (i) the principle of causality: a particular element at one hierarchical level should be present at any consecutive level; and (ii) the principle of locality: regions must be stable when creating or removing partitions.

In [Cousty *et al*., 2013], the authors provided formal links between the morphological partitions and edge-weighted graphs. This section formalizes graph concepts describing their components and terminologies (Section 3.1), connects graphs and hierarchies (Section 3.2), and describes the different hierarchical model types contemplated in this work (Section 3.3).

3.1 Graph's formalism and notions

A graph $G = (V, E)$ consists of a finite set of vertices, denoted by V, and a finite set of edges denoted by E, in which $E \subseteq V \times V$. If $(u, v) \in E$ for two vertices $u, v \in V$, then u and v are adjacent vertices. The notion of vertices relates to the data's elemental components while edges to the connections and dynamics between the parts. A graph is non-empty if $V \neq \emptyset$, nontrivial if $E \neq \emptyset$, complete if $E = V \times V$, and direct if $(u, v) \neq (v, u), \forall u, v \in V$.

The set E induces a unique adjacency relation Γ on V, which associates $u \in V$ with $\Gamma(u) = \{u\} \cup \{v \in E | (u, v) \in E\}$. Γ is reflexive ($u \in \Gamma(u)$) and symmetric ($v \in \Gamma(u) \iff u \in \Gamma(v)$). In multimedia processing, the adjacency relation is usually in a regularly structured form as a grid invariant to translation. Standard grid adjacency in 2D spaces is the squared orthogonal shape named $4-$adjacency, the octilinear form in the $8-$adjacency, or the hexagonal structure in the $6-$ adjacency relation. Alternatives to the grid adjacency involve distance parameters determining the reach of each vertex or a selection criterion based on a pattern or media property.

An edge-weighted graph is denoted by $(G, \mathcal{F})$, in which $\mathcal{F} : V \times V \to \mathbb{R}$ is a function that weights the edges of $G = (V, E)$ and $\mathcal{F}(E)$ is the weighted map for the function $\mathcal{F}$ on the set E. The nature of $\mathcal{F}$ determines which characteristics the graph preserves, and selecting a function could be considered a similarity measure problem between two finite sets of points, where $\{w = \mathcal{F}(u, v) | (u, v) \in E\}$ is the weight w of an edge $(u, v) \in E$ that could describe the dissimilarity of u and v.

A path $\pi = (v_0, \ldots, v_\ell)$ is an ordered sequence of vertices with size ℓ connecting v_0 to v_ℓ if $(v_{i-1}, v_i) \in E$ for any $i \in \{1, \ldots, \ell\}$. In an edge-weighted graph, a path is descending if for any $i \in \{1, \ldots, \ell - 1\}$, $\mathcal{F}(v_{i-1}, v_i) \geq \mathcal{F}(v_i, v_{i+1})$. A connected graph has a path from v to u for all $u, v \in V$.

Another way to define and interpret a graph is through subsets of all possible vertices and edges. A graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$, then G and G' are ordered by the inclusion relation $G' \sqsubseteq G$, where G' is smaller than G. A lattice is a set of all subgraphs of G preserving the inclusion order.

A tree is a particular case of a direct graph. In a tree, we denote vertices as nodes and distinguish them based on their positions in the structure. The root is the single node at the top of the tree that connects all the other nodes. From the root, every subsequent node is a child. They can be either an internal node, from which other nodes branch, or a leaf with no children at the bottom of the tree. The root and internal nodes are the parents of their children. From the root, each node in the path to a leaf characterizes one level, and the maximum number of levels defines the depth of a tree. The altitude of a node starts from the leaves, ascending until reaching the node, and it is inversely proportional to the depth of the node.

3.2 Hierarchies from graphs to graphs

In classical mathematical morphology, structuring elements are the parameters for the algebraic operators on lattices. On graphs, the modeling choices for the edges, weights, and adjacency relation define the parameters. A hierarchy operating on the edge-weighted graph defines non-gridded regions as subsets of the vertices. For $G = (V, E)$ and the subgraph $G' = (V', E')$, the graph induced by V' is $G = (V', \epsilon)$ where $V' \subseteq V$ and $\epsilon = \{(u, v) \in E \,|\, u, v \in V'\}$. V' is a connected component of G if V' is connected for G and maximal.

A set $\mathcal{H} \subseteq \mathbb{V}$, where $\mathbb{V}$ denotes the set of all subsets on V, is a hierarchy on V if $H_1 \cap H_2 \in \{\emptyset, H_1, H_2\}$ for any two elements $H_1, H_2 \in \mathcal{H}$ and complete if $\{V\} \in \mathcal{H}$ and $\{\{v\} \in \mathcal{H} \,|\, \forall\, v \in V\} \in \mathcal{H}$. Without loss of generalization, for $\mathbb{G}$ denoting the set of all subgraphs of G, $\mathcal{H} \subseteq \mathbb{G}$ is a hierarchy on G if $H_1, H_2 \in \{(\emptyset, \emptyset), H_1, H_2\}$ for any for any $H_1, H_2 \in \mathcal{H}$, and it is complete if $G \in \mathcal{H}$ and $\{(\{v\}, \emptyset)\} \in \mathcal{H}$.

These notations characterize a direct forest and tree, respectively, which portray the hierarchy as a Hasse diagram, also known as a dendrogram representation [Sokal and Rohlf, 1962]. Therefore, a hierarchy is a graph in the form of a

hierarchical tree. In a hierarchical tree, for $H_1, H_2 \in \mathcal{H}$, H_2 is a child of H_1 if H_2 is the largest proper subset of H_1 and if $H_2 \subseteq H \subseteq H_1$, $H = H_2$ or $H = H_1$ for any $H \in \mathcal{H}$. An element of $\mathcal{H}$ without a child is a minimum of $\mathcal{H}$.

A partition $\mathbb{P}$ is a set of non-empty disjoint subsets of V, meaning that $\forall\, X, Y \in \mathbb{P}$, X and Y are regions, $X \cap Y = \emptyset$ if $X \neq Y$ and $\cup\{X \in \mathbb{P} = V\}$. Any element $v \in V$ belongs to a unique region, a singleton partition of $\mathbb{P}$, denoted $[\mathbb{P}]_v$. The partition set is ordered from finer in $\mathbb{P}'$ to coarser in $\mathbb{P}''$ if any region in $\mathbb{P}'$ is present in $\mathbb{P}''$ for any $\mathbb{P}', \mathbb{P}'' \in \mathbb{P}$. The ordered relation conveys the idea of refinement. Also, navigating the partition from finer to coarser, commonly codded as bottom-up, impart the concept of region aggregation. In contrast, the opposite, top-down, is the concept of region splitting.

A hierarchy of partitions $\mathcal{H} = (\mathbb{P}_0, \ldots, \mathbb{P}_k)$ is a sequence of partitions on V, such that $[\mathbb{P}]_{i-1}$ is a refinement of $[\mathbb{P}]_i$ $\forall i \in \{1, \ldots, k\}$ where k is the number of levels in the hierarchy characterizing its altitude and depth. The hierarchy preserves the non-empty disjoint sets notion and the ordered relation. The union of all partitions of $\mathcal{H}$ creates the set of regions of $\mathcal{R}_\mathcal{H}$, and the inclusion relation induces a tree structure.

The hierarchical partition tree $\mathcal{T}_\mathcal{H}$ is the tree representing the hierarchy $\mathcal{H} = (\mathbb{P}_0, \ldots, \mathbb{P}_k)$ where:

- the root node represents the single partition $\mathbb{P}_k = \{V\}$,
- the set of leaves $\mathcal{L}$ represents the partition $\mathbb{P}_0$, where $\mathbb{P}_0 = \{[\mathbb{P}]_v \mid \forall\, v \in V\}$,
- the parent of a node n in the set of nodes $\mathcal{N}$ representing the region $\mathcal{R}_n$ of $\mathcal{R}_\mathcal{H}$ is the smallest region of $\mathcal{R}_\mathcal{H}$ that is strictly larger than $\mathcal{R}_n$, and
- the depth d_n of a node $n \in \mathcal{N}$ is its number of parents.

There are multiple ways to represent a hierarchy of partitions, straightforward as a hierarchical partition tree with all the partitions in a single structure. Another way is by a cut presenting one partition of the hierarchy at a time. The cut can be a horizontal cut Perret *et al*. [2018] if all regions are extracted at the same hierarchical level or a non-horizontal cut [Guigues *et al*., 2006] if searching for regions at different levels for one representation.

3.3 Types of hierarchical models

Thus far, the discussions about hierarchies considered only the structural components of the graphs: the vertice and edge sets. The hierarchical construction algorithms use the weights to regulate how regions are formed, the criterion to merge and create new ones, and the order to pursue. This work contemplates two particular hierarchical model types grouped by their ordering method on the hierarchical tree. Namely: (i) altitudes ordering based on increasing values of edge-weights criterion: quasi-flat ones [Cousty *et al*., 2018]; and (ii) altitudes ordering based on a geometric criterion: hierarchical watersheds [Beucher, 1994].

The **quasi-flat zones (QFZ)** hierarchy is induced directly from the edge-weight graph. Its construction algorithm takes the set of ordered weights on the edges and defines each level as the set of connected component partitions whose weights are smaller than a threshold value λ. Formally, consider an edge-weighted graph $(G, \mathcal{F})$, the set of connected components of G denoted by $\mathbb{C}$, a subgraph G' of $(G, \mathcal{F})$, an weight value $\{w = \mathcal{F}(u, v) | (u, v) \in E\}$ and the range of values $\mathbb{E}$ for all weight values of E. The QFZ hierarchy induced by the edge-weighted graph is defined as $\text{QFZ}(G', w) = (\mathbb{C}(w^V_\lambda(G')) | \lambda \in \mathbb{E})$, where: (i) $w_\lambda(G')$ is the λ-*level set* of all edges of G' whose weight values are less than λ; (ii) $w^V_\lambda(G')$ is the λ-*level graph* whose edges are $w_\lambda(G')$ and vertices V; and (ii) $\mathbb{C}(w^V_\lambda(G'))$ is the λ-*level partition* of connected components partition induced by the λ-*level graph* of G'.

The **hierarchical watershed** extends the classical morphological watershed [Beucher, 1979], and it is an intuitive approach to map weights into partitions. One of the intuitions behind the classical watershed is the principle of the drop of water flowing on a topological surface. The watersheds are the lines separating the multiple downward regional minima. In media processing, the topological surface is usually created by magnitude values, in which mountains are the regions with comparatively higher magnitudes, and basins and valleys are the ones from lower magnitudes.

This principle is used in the hierarchies of watersheds to create a sequence of segmentations as connected elements formalized as a minimum spanning forest (MSF) representing the flooded regions in all possible levels. For edge-weighted graphs, the drop of water principle is interpreted as a graph cut, known as a watershed cut, that is not uniquely defined for a weight map. However, the watershed hierarchies as a relative MSF are optimal and unique for a watershed cut [Cousty *et al*., 2009]. To obtain a partition in the hierarchy, it takes the weighted graph, and a subset of graph vertices called markers representing regional minima on the weight map. If the markers are ranked and ordered, it creates a sequence of nested partitions where each hierarchy level represents a marker's extinction value [Vachier and Meyer, 1995] (the minimum value that makes a region be merged into another region). The extinction values are usually grouped and ranked based on a given geometric criterion that reflects its region's topological properties.

Each construction algorithm has its particular properties and interpretation of the data. However, the rules on the hierarchical principles and the ordered representation of regions create a shared space convenient for commuting from one type to another if one representation is inadequate for an application. Furthermore, efficient implementations [Najman and Couprie, 2006] make the hierarchies an appealing alternative to introducing a semantic interpretation into media processing.

4 Typical hierarchical pipeline

This section introduces a typical pipeline, illustrated in Fig. 1, that applies the hierarchies defined for an edge-weighted graph to an image processing task.

Usually, image applications are tasks defined for three-channel colored images, and despite the availability of existing hierarchical methods applied directly on the color channels [Soille, 2008], operating on colored images requires strategies to either map dissimilarities between pixels on mul-

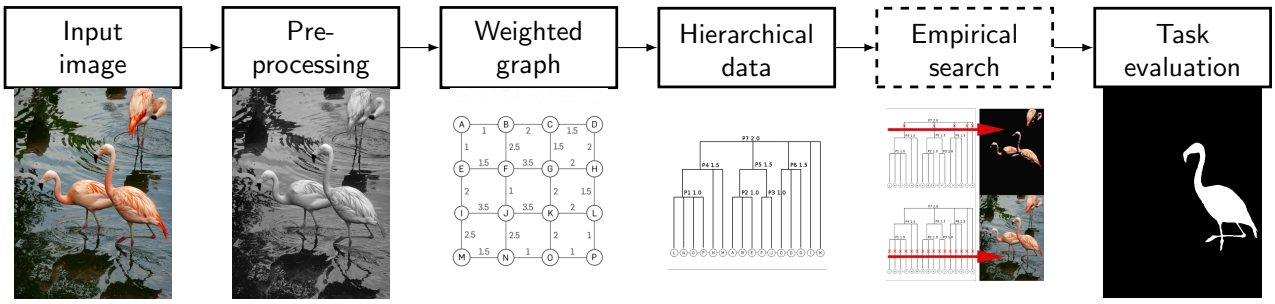


Figure 1. Illustration of a typical pipeline using hierarchies for image processing. First, it transforms each image to the gray-scale magnitudes used to create the edge-weighted graphs. Then, the hierarchical method computes the desired hierarchy based on its criterion. Because the hierarchical structure is multi-layered, selecting a certain level, a combination, or a specific number of regions is necessary to create a single output evaluated on the task.

tiple dimensions [Aptoula *et al*., 2013] or combine the hierarchies independently defined on each channel [Kurtz *et al*., 2014]. Therefore, the general approach is to model the graph from the monochromatic images, such as the grayscale representation of pixel intensities or image gradients.

During the development of our framework, we evaluated several gradient operators for generating edge weights in the image graph, namely GIG [Almeida *et al*., 2022], SED Dollar and Zitnick, 2015, and the kernel methods Sobel and Laplacian. Any operator that produces meaningful edge information can serve as input, since the hierarchical representation encodes structure beyond the specific gradient details. However, Laplacian gradients tend to lose important structural information, and SED, while effective, introduces significant computational overhead. Sobel provided reasonable results, but GIG consistently produced better outcomes in terms of detail, contour sharpness, and region separation, without requiring additional parameterization. In addition, GIG is efficient, producing the image gradient in a fraction of a second. For these reasons, we chose GIG as our default, as it offers an optimal balance between informative contours and computational efficiency.

After adequately preparing the image, the following steps on the pipeline are the graph generation and hierarchy construction. Defining the graph representation is a modeling question with various connotations, and each hierarchical model type has its particular characteristics, discussed in Section 3.

Once constructed, it is necessary to decide how to represent the hierarchies to be applied to a task since most ground-truth references need a flat (*i.e.*, non-hierarchical) form for comparison. In this step resides the central problem of this work. The **trivial** approach is a series of horizontal cuts selecting multiple independent partitions representing the hierarchy. The selection could indicate the desired number of regions portrayed on the partition or a threshold of the hierarchical levels. This process can be strenuous if searching for an ideal number of regions. One could search from a single region to the total number of regions in the hierarchy, which is variable among the many representations. Or, if thresholding the levels, one crucial detail present at one hierarchical level could be merged on the subsequent levels. Even further, as pointed out in Perret *et al*. [2018], the metric used to evaluate the selection can be misleading. Also, a good horizontal cut for one specific hierarchy does not guarantee that it will be ideal for another on the same dataset.

Other representation strategies include post-processing the hierarchies by flattening [Xu *et al*., 2013], realigning [Adão *et al*., 2020], or filtering the structure [Perret *et al*., 2019b]. These strategies rely on identifying less relevant regions and re-weight or merging these regions, creating more concise representations. The problem with these approaches is that defining the region's importance is subjective and strongly related to a media type or task.

Alternatively, one could search for the ideal representation with a non-horizontal cut [Arbelaez *et al*., 2014], which is, by all means, a combinatorial problem. One possible solution is to create a model that learns this ideal representation directly from the structure and uses the model to adapt unseen sets of hierarchies [Chierchia and Perret, 2020]. However, inserting the hierarchies in a learning framework is difficult since they have heterogeneous representations, for instance, in their altitudes and number of regions. Furthermore, construction algorithms are primarily deterministic, and there is no direct way to evaluate their quality other than applying them to a task.

To point out these notions is to highlight that beyond the data modeling, and, despite the abundance of information embedded in the hierarchies, without careful considerations in choosing the hierarchical type, the parsing strategy, the representation for the task, and the metrics, media processing strategies could overlook the potential in these structures. For instance, datasets without large quantities of labeled data or applications that require dependable outputs could rely upon regional analysis methods that provide a consistent data organization, such as those offered by hierarchical analysis.

5 Learning on hierarchical attributes

This section presents a learning framework, illustrated in Fig. 2, formulated on the structural components of the hierarchies and a regular representation of the structure attributes. We present two strategies for selecting attributes from the hierarchical structures: (i) a regular representation selecting topological properties from the hierarchical trees (Section 5.1); and (ii) regional features deduced from the hierarchies and their conjoined graph (Section 5.2).

The hierarchical construction contemplates the hierarchies in Section 3.3. Without loss of generality, the conjoined edge-weighted graphs in this pipeline are defined on the image

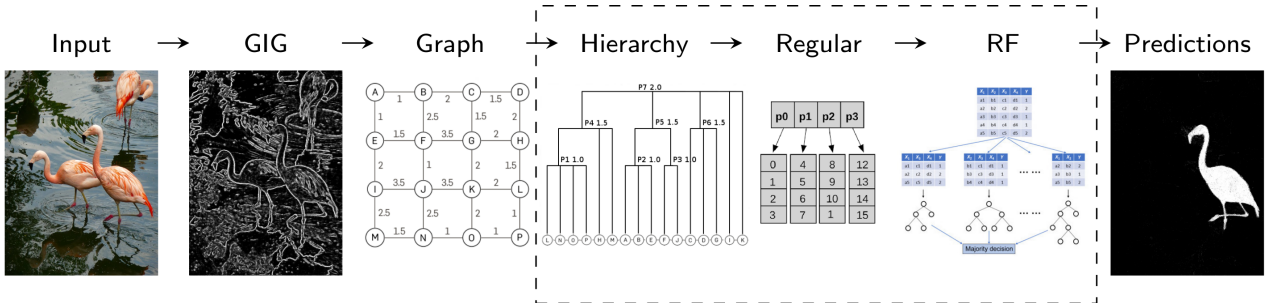


Figure 2. Illustration of the framework from the input image to the Random forest predictions performing the task. First, it computes the GIG gradient for each image in the dataset. Then, it calculates the edge-weighted graphs, here illustrated with the $4-$adjacency relation. The next step constructs the hierarchies from the graphs and creates a regular representation with topological attributes of the hierarchical trees to serve as input for the Random Forest model. The regular input for the training set includes the associated label: the unique discrete label on the task for each tree leaf. During the test, the Random Forest subject each leaf of the test hierarchies to prediction, where the estimated values are mapped back to the image coordinates for evaluation.

domain. A structured grid obtains the adjacency relation Γ, and the set of vertices $V = \{v_1, v_2, \ldots, v_N\}$ represents the N pixels of the image. Each vertex is associated with a function $f : V \subset \mathrm{Z}^2 \rightarrow \mathbb{R}$ mapping gray-scale magnitudes of the GIG [Almeida *et al*., 2022] gradient computed from the original image. For the edges, a weighting function $\mathcal{F}$ characterizes similarities.

The representations are aggregated using Random Forests (RF) [Breiman, 2001], a fast, simple, and scalable model capable of dealing with high dimensional data and with satisfactory results in multiple tasks. It allows for extensive experimentation without the need for additional model-specific engineering or scalability adjustments.

The RF described in Breiman [2001] is a non-parametric machine learning method for classification and regression. At the core of the RF is the randomization of sampled data distributed to supervise the training of independent decision trees and the aggregation of the results for the final prediction. The randomness performs as an implicit regularization process promoting consistency and noise suppression [Wyner *et al*., 2017].

For a graph created from the image pixels, there is a direct correspondence between the pixels, the vertices, and the leaves in the tree. Therefore, the task label attribution is performed at the leaf level at the bottom of the tree, and each leaf has a unique discrete label. It creates a model agnostic to the task since no assumptions are made on the media type for the label attribution, and the single label represents multiple regions that share a path on the tree. At inference time, the RF predictions are mapped to the image space to be evaluated on the task.

A notable property of our framework is that learning is driven entirely by the hierarchical and structural features of the data, independent of low-level appearance cues. During training, the model receives labels paired only with specific leaves of the hierarchy. The Random Forest thus learns to distinguish structural roles based on the hierarchy, rather than relying on raw visual similarity. This makes the approach naturally robust in settings with ambiguous boundaries, incomplete annotation, or the presence of multiple objects that may be visually similar but structurally distinct. As a result, our method is well suited to real-world situations where annotation may be sparse or ambiguous, and where classical, appearance-based methods may be misled.

5.1 Topological attributes

The first strategy creates a regular representation by selecting topological properties from the hierarchical trees. Formally, consider a hierarchical tree $\mathcal{T}_\mathcal{H}$ representing the hierarchy of partitions $\mathcal{H} = (\mathbb{P}_0, \ldots, \mathbb{P}_k)$ created from the edge-weighted graph $(G, \mathcal{F})$ has a set of nodes $\mathcal{N}$. The depth d_n of a node $n \in \mathcal{N}$ is its number of parents. At the bottom of this tree, there is a collection of leaves $\mathcal{L}$ representing the partition $\mathbb{P}_0$, where $\mathbb{P}_0 = \{[\mathbb{P}]_v \,|\, \forall\, v \in V\}$ and each $l \in \mathcal{L}$ corresponds to a $v \in V$. The proposed representation depicts each leaf $l \in \mathcal{L}$ as a vector $\mathbf{T}_l$ of selected attributes. The selection corresponds to one of the following attributes:

- **Altitude**: the value inversely proportional to the depth of the node n: $\texttt{alt}_n = 1/d_n$.
- **Area**: sum of the number of leaves on the subtree τ_n rooted on the node n: $\texttt{area}_n = |\{\mathcal{L}_n\}|$, for $\mathcal{L}_n = \{l \,|\, \forall\, l \in \tau_n\}, \mathcal{L}_n \subseteq \mathcal{L}$.

The selected attribute is computed for all parents of l. Each leaf has a variable number of parents; therefore, the dimension p_t of the vector $\mathbf{T}_l$ is standardized by the maximum depth in all $\mathcal{T}_\mathcal{H}$ computed for a dataset. Also, the leaves with a set of parents smaller than the maximum depth receive a padding value of -1 because the attributes considered for the selection have all positive values ($\texttt{alt} \in [\,0, 1\,]$ and $\texttt{area} \in [\,0, |\mathcal{L}|\,]$).

The semantical meaning is kept by representing the parents of a leaf node in the *order* they appear transversing the hierarchical tree. The order could be *ascending* (from leaf to root) or *descending* (from root to leaf). Early experiments showed that essential attributes occur at the initial positions of the feature vector and are favored by the RF model during training. Therefore, we use the *ascending order* in this work.

The **regular representation on topological attributes** is formalized as $\mathbf{T}_\mathcal{H} = ((\mathbf{T}_1, \mathbf{Y}_1), \ldots, (\mathbf{T}_{|\mathcal{L}|}, \mathbf{Y}_{|\mathcal{L}|}))$, where each leaf $l \in \mathcal{L}$ is represented as a vector $\mathbf{T}_l$ with a single label $\mathbf{Y}_l$. $\mathbf{T}_l = [[\texttt{topo}(\texttt{par}_1), \ldots, \texttt{topo}(\texttt{par}_{par})]]$ for all par parent nodes in the set $\mathbf{P}_l$ of parents of l, and $\texttt{topo} \in \{\texttt{alt}, \texttt{area}\}$ for the attribute candidates. The size of $\mathbf{T}_l$ is p_t and $p_t = \mathbf{max}(d_n)\,, \forall n \in \mathcal{N}$ in all $\mathcal{T}_\mathcal{H}$ in the set $\mathbb{T}$ of all hierarchies in a dataset. The training input $\mathcal{D}_t$ on topological attributes for the RF concatenates all the $\mathbf{T}_\mathcal{H}$ of the hierarchies $\mathcal{T}_\mathcal{H} \in \mathbb{T}$ that corresponds to a training instance on the dataset, where $\mathcal{D}_t = ((\mathbf{T}_1, \mathbf{Y}_1), \ldots, (\mathbf{T}_{T_l}, \mathbf{Y}_{T_l}))$ and T_l is the total number of leaves in the training set.

5.2 Regional attributes

The second strategy uses a set of regional attributes created on the conjoined graph by the hierarchical structure. Formally, each node $n \in \mathcal{N}$ represents a region $\mathcal{R}_n$ that is the union of all regions on the subtree τ_n rooted on the node n. A cut is a partition $\mathbb{P}$ of V made of regions of $\mathcal{H}$, where a horizontal cut is a partition $\mathbb{P} = \mathbb{P}_i$ for $i \in \{0, \ldots, k\}$ for all k altitude levels on the tree. A horizontal cut by altitude levels defines the partition by a threshold σ on its altitude values. Two regions $\mathcal{R}$ and $\mathcal{R}'$ are in the same region $\mathcal{R}_n$ if n is their lowest common ancestor that have $\texttt{alt}_n > \sigma$.

Consider β as a series of altitude levels to cut the hierarchy. The proposed representation depicts each leaf $l \in \mathcal{L}$ as a vector $\mathbf{R}_l$ of size $|\beta|$. At each position of this vector, there is a cut $\mathbb{P}_\sigma$ for $\sigma \in \beta$. Thus, the leaf l is represented by a selected regional attribute for the region $\mathcal{R}_n$ where n is the lowest parent of l whose $\texttt{alt}_n > \sigma$. The selection corresponds to one of the following:

- **Contour strength**: The contour ζ of a node is the number of edges on the conjoined weighted graph shared among the regions merged by a node. The contour strength is the average of edge weights on the contour: $\texttt{contour}_{\mathcal{R}_n} = \sum_{(u,v)\in\zeta} \mathcal{F}(u,v)$, in which $\zeta = \{(u,v) \in E \,|\, \forall\, u \in \mathcal{R} \wedge v \in \mathcal{R}' \text{ and } \forall\, \mathcal{R}, \mathcal{R}' \subseteq \mathcal{R}_n\}$.
- **Gaussian**: Estimates the Gaussian distribution of leaf weights in the region $\mathcal{R}_n$ defined by the node n. The function returns two values: the mean and the variance. The leaf weights could be defined for any attribute or set of attributes (on which one could calculate the covariance). Here, they are the sum of the weights of the edges that comprise the vertice equivalent of the leaf. Hence, $\texttt{gaussian}_{\mathcal{R}_n} = [\texttt{mean}_{\mathcal{R}_n}, \texttt{var}_{\mathcal{R}_n}]$, in which:

$$\texttt{area}_{\mathcal{R}_n} = |\{\mathcal{L}_{\mathcal{R}_n}\}|,\ \texttt{mean}_{\mathcal{R}_n} = \frac{W_{\mathcal{R}_n}}{\texttt{area}_{\mathcal{R}_n}} \text{ and}$$

$$\texttt{var}_{\mathcal{R}_n} = \frac{(\texttt{mean}_{\mathcal{R}_n})^2}{\texttt{area}_{\mathcal{R}_n}} - (\texttt{mean}_{\mathcal{R}_n})^2$$

for $\mathcal{L}_{\mathcal{R}_n} = \{l \,|\, \forall\, l \subseteq \mathcal{R}_n\}$, with $\mathcal{L}_{\mathcal{R}_n} \subseteq \mathcal{L}$ and $W_{\mathcal{R}_n} = \sum_{u\in\Gamma(u)}$ and $_{v=l\in L_{\mathcal{R}_n}} \mathcal{F}(u,v)$.

The selected attribute is computed for all regions created by the cut $\sigma \in \beta$, and the ordered representation is preserved on the cut despite not representing every possible region in the hierarchy. It is proposed to select only a few steps in the normalized altitudes creating a reduced set of features guaranteed to be present in all hierarchical types.

The **regular representation on regional attributes** is formalized as $\mathbf{R}_\mathcal{H} = ((\mathbf{R}_1, \mathbf{Y}_1), \ldots, (\mathbf{R}_{|\mathcal{L}|}, \mathbf{Y}_{|\mathcal{L}|}))$, where each leaf $l \in \mathcal{L}$ is represented as a vector $\mathbf{R}_l$ with a single label $\mathbf{Y}_l$. $\mathbf{R}_l = [\texttt{reg}(\sigma_1), \ldots, \texttt{reg}(\sigma_{|\beta|})]$ for all σ cuts in β and $\texttt{reg} \in \{\texttt{contour}, \texttt{gaussian}\}$. The size of $\mathbf{R}_l$ is $|\,\beta\,|$, defined in the range $]0, 1[$ with a 0.1 step adding 0.01 and 0.99 for the extremal regions in the structure. The training input $\mathcal{D}_r$ on regional attributes for the RF concatenates all the $\mathbf{R}_\mathcal{H}$ of the hierarchies $\mathcal{T}_\mathcal{H} \in \mathbb{T}$ that corresponds to a training instance, where $\mathcal{D}_r = ((\mathbf{R}_1, \mathbf{Y}_1), \ldots, (\mathbf{R}_{T_l}, \mathbf{Y}_{T_l}))$ and T_l is the total number of leaves in the training set.

The procedure for test instances in both proposed representations takes the regular representation of each hierarchy in the test set and individually subjects them to the RF estimations without the labels.

6 Experiments and results

This section presents experiments with the trivial, topological, and regional approaches in two image tasks: edge detection and segmentation. It also includes results using selected attributes extracted directly from graphs before hierarchy construction, as described in [Almeida *et al*., 2022]. In these experiments, the objective is to assess the learning framework based solely on hierarchical and structural information derived from the input graph, independent of how the graph was generated. While preprocessing could improve task-specific results, our focus is on validating the generality and effectiveness of the framework itself.

6.1 Datasets

The edge detection dataset is the Berkeley Segmentation Dataset and Benchmark (BSDS500) [Martin *et al*., 2001], illustrated in Fig. 3. It contains 500 (200 train, 100 validation, and 200 test) natural images, presenting complicated/high-contrast patterns, occluded objects, and objects indistinguishable from the background by color. Each image has multiple labels performed by different annotators; thus, we performed a majority vote to obtain a single label.

For segmentation, Birds [Mansilla and Miranda, 2016], a binary segmentation public dataset. It contains 50 images of birds with manual annotations and no official train/test sets division. Therefore, a random selection split the dataset into 35/15 train/test. Fig. 4 illustrates the Birds dataset and its challenges. Namely, the images usually portray the birds close to a body of water, with areas of high-intensity lights and annotations covering only one leading object, despite the presence of multiple similar objects in the surroundings. This design introduces a challenging scenario for segmentation, as both annotated and unannotated objects may be visually indistinguishable. We intentionally selected this dataset to assess the capacity of our method to go beyond low-level visual cues and instead leverage structural and hierarchical information.

6.2 Experimental setup

The pipeline takes the colored images and computes the GIG gradient without any additional preprocessing. Next, it constructs the graph with a $4-$adjacency relation and the Euclidean distance on the gradient magnitudes for the weighting function. The hierarchy construction explores the aforementioned quasi-flat zones (QFZ) and the hierarchical watershed using the number of parents (WATER-PAR) as topological criterion [Perret *et al*., 2018], which counts the number of parents a node has on the MST representing the graph to determine its extinction values. It does not perform additional post-processing, such as filtering, realigning, or balancing the hierarchical levels.

For the BSDS500, the pipeline uses an RF regressor as a model, where the average predictions are mapped back to

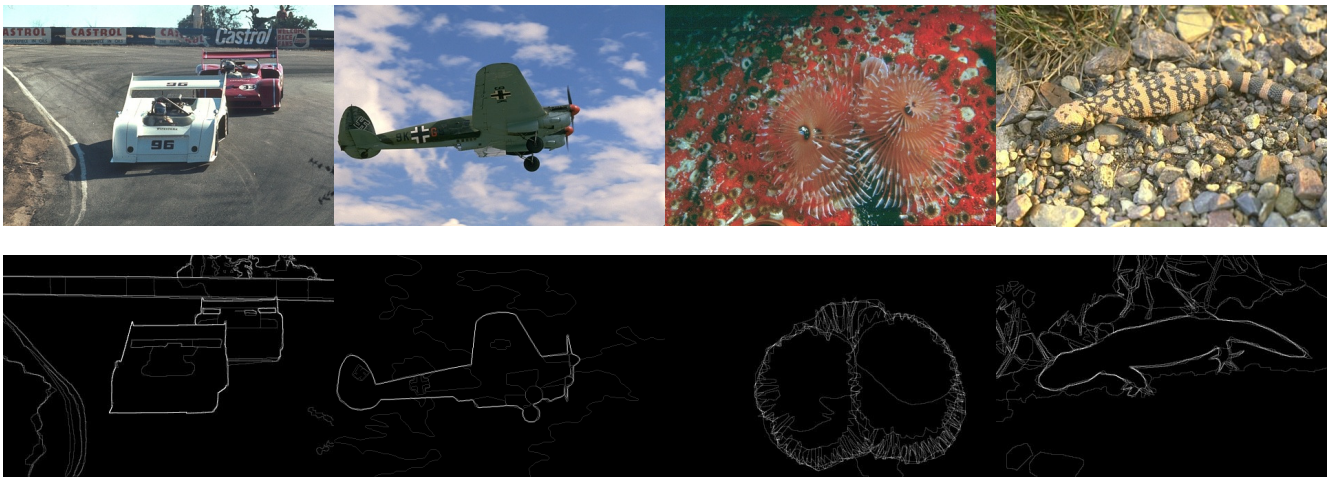


Figure 3. BSDS500 dataset sampled images with their respective boundary ground truths. It contains colored natural images presenting complicated patterns, occluded objects, main objects indistinguishable from the background by color, and objects with patterns of high contrast. Each image contains multiple labels where line intensities indicate the annotators' agreement.

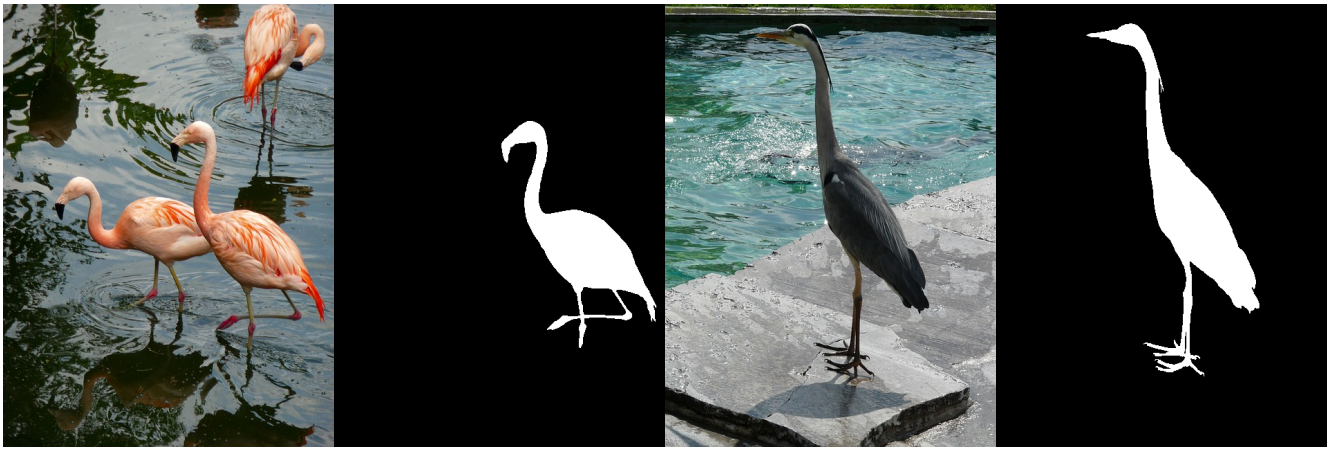


Figure 4. Birds dataset sampled images with their respective segmentation ground truths. The images usually portray the birds close to a body of water, with areas of high-intensity lights and the annotations for only one main object, despite the presence of multiple similar objects in the surroundings.

the image domain for evaluation. This dataset proposes an evaluation system for methods using it. The evaluation takes an edge map and threshold the values in the range $[0, 1[$ with a 0.01 step computing the precision-recall $F1-$score at all threshold values. The results were then assessed in terms of the optimal dataset scale (obtained in the threshold that best represents most of the images), the optimal image scale (obtained for each image at its best scale), the average precision through all scales. For non-hierarchical methods, this evaluation works to process soft edge maps. For hierarchical methods, this evaluation allows the assessment of different levels of details in the hierarchical partitions. However, for clarity, the results presented in this section for the BSDS500 dataset are only for the optimal dataset scale, which gives the average score obtained in the threshold that best represents most images, which is the most challenging and the best to evaluate the overall performance.

For the segmentation dataset, the pipeline considers an RF classifier whose predictions for each leaf on the binary segmentation labels are directly mapped back to the image space. The evaluation metric use the Jaccard similarity coefficient score as the metric, which measures in the interval $[0, 1]$ the intersection size divided by the union size of two sets. It is equivalent to the precision-recall $F1-$score on binary sets.

The parameters for the RF models were obtained using a grid search on the validation set of the BSDS500 dataset and set as 500 trees in the forest, 5 minimum number of samples to split an internal node, 20 the minimum number of samples to be a leaf node, 10% percent for the bootstrap sample size, squared function on the whole set of features to amount to the sampled features for the split, and 10 as the maximum depth of the trees. The trivial approach does not involve a learning step. The experiments explored a range of parameters defining the number of desired regions for a cut by the number of regions and multiple horizontal cuts by threshold. The results presented are for the best parameters, namely: (i) 1000 regions for QFZ and 60 for WATER-PAR using the cut by the number of regions; and (ii) threshold at 0.22 for QFZ and 0.53 for WATER-PAR using the horizontal cuts by threshold. The graph comparison uses an attribute selection belonging to two categories: (i) vertex attributes, representing low-level color descriptors proposed in Dollar *et al*. [2010] (named onlyColor); and (ii) edge weights, representing the weight values in every edge on the adjacency of a vertex (named

Table 1. Quantitative comparison of the results obtained in all datasets for the compared approaches. $F1-$score for the optimal dataset scale for the BSDS500 and average Jaccard score for Birds. The best score for approach variation is shown in bold and underline emphasis the best score per dataset. Perfect scores=1.

		BSDS		Birds	
Graph	GIG	**0.65**		**0.29**	
	GIG-Edge	0.64		0.28	
	onlyColor	0.61		0.27	
Trivial	Hierarchy	Threshold	Regions	Threshold	Regions
	QFZ	0.26	**0.28**	**0.14**	0.05
	WATER-PAR	0.24	**0.53**	**0.28**	0.24
Topological	Hierarchy	Altitude	Area	Altitude	Area
	QFZ	**0.60**	0.52	0.30	**0.37**
	WATER-PAR	**0.63**	0.54	0.32	**0.41**
Regional	Hierarchy	Contour	Gaussian	Contour	Gaussian
	QFZ	0.63	**0.67**	**0.53**	0.51
	WATER-PAR	0.63	**0.65**	**0.71**	0.64

GIG-Edge). The graph representation with both categories of attributes is named GIG.

6.3 Quantitative analysis

Table 1 shows the results for the proposed strategies on the two datasets compared with the typical trivial approach (cut by threshold on altitude levels and by the number of regions) and the representations from graph attributes.

While the results with the trivial approach are considerably worst compared with the other strategies, they are presented to establish a baseline, not to say that hierarchical structures are ineffectual for the edge detection task. On the contrary, many hierarchical proposals in this dataset present competitive results. However, each successful method also gives one strategy to improve or filter the hierarchical contours. For instance, Arbelaez *et al*. [2009] proposed a technique that constructs hierarchical boundary maps from an edge map where the boundaries between consistent regions are reinforced and small areas removed (reaching 0.71 on the optimal dataset scale). In Maninis *et al*. [2018], they take pre-computed contours using the side outputs of a convolutional network for constructing the hierarchies (with a 0.73 score). In Taylor [2013], they use normalized cuts to reduce internal regions and sharpen the contours between contrasting areas (with a 0.67 score). And Arbelaez *et al*. [2014] creates the hierarchies at multiple image scales independently and combines them into a single contour map weighing the strength of each contour using machine learning (with a 0.73 score). Furthermore, Perret *et al*. [2018] shows the gain quantitatively in score by filtering small areas on another dataset with the same task.

As for the segmentation task with Birds, the illumination conditions on the images create a scenario that is very challenging for many of the best image processing methods, as it creates peaks in the magnitude values that make it difficult to distinguish the main objects and the body of water in the background. With the hierarchical methods, the algorithms will create similar partitions for the many objects portrayed in the images, while only one is considered a valid answer.

Compared with the typical approach, the topological strategy improves the results for almost all hierarchical types for all datasets (except for WATER-PAR with altitudes in Birds). The additional benefit is that it does not require an empirical search on the hierarchical levels and regions for evaluation. Furthermore, the topological approach presents best results than the trivial and the graph in the Birds dataset. In edge detection, the graph and the topological perform better than using only the color features, with the GIG approach performing better than the best on the topological strategy.

The regular representation with topological attributes captures enough information for the learning model to better discriminate between classes. And the padding values did not disturb the model performance in any hierarchical type. Regarding the topological attributes, the altitudes perform better on the edge detection and the area on the segmentation, which matches the task goals with the attributes' properties.

The regional attributes present the best results in all datasets. Even for the challenging Birds, there is at least one attribute for all hierarchical types that give a satisfactory result. The Gaussian presents, in general, superior results on the different tasks. Because the Gaussian attribute quantifies the region distribution on the hierarchical trees, it assimilates the representation with the task. Future applications of this strategy may consider the hierarchical type that most agrees with the objectives and use the Gaussian attribute for the representation.

6.4 Computational efficiency and scalability

All experiments were run on a HPE ProLiant DL385 Gen 10+ v2 (AMD EPYC 7431, 8 cores, 32GB RAM, Linux). Hierarchical computations used the *Higra* Python library [Perret *et al*., 2019a], and learning models were implemented with *Scikit-learn* [Pedregosa *et al*., 2011], using parallel training over 50 CPU cores.

Despite the additional cost of graph and hierarchy construction, the overall runtime remains competitive with standard image-based pipelines. For the full BSDS500 dataset, creating the regular sets takes 500 seconds for the topological approach and 110 seconds for the regional approach; for the Birds dataset, these steps require just 40 seconds (topological) and 10 seconds (regional). Training the Random Forest models is also efficient, requiring at most 2 hours (topological) or 10 minutes (regional), which is considerably faster than typical deep learning models for similar tasks.

The framework is well suited to large datasets and scalable to even larger graphs or non-image domains, especially when annotation or visual data is limited. Efficient library implementations and hardware parallelism contribute to its practicality in real-world scenarios. For non-image data that are already graph-structured, the initial graph construction step is unnecessary, further reducing the runtime.

Overall, our method is competitive in computational cost with both traditional and learning-based approaches, while remaining straightforward to scale and deploy. The reliance on open-source libraries and commodity hardware makes the

framework accessible for both research and practical applications.

6.5 Qualitative analysis

One significant advantage of working with image data is the ability to directly and intuitively inspect outputs, making it possible to qualitatively assess the practical strengths and weaknesses of each method. In this section, we leverage this property to present a qualitative analysis that complements the global quantitative results and illustrates how image-based tasks enable a deeper understanding of model behavior and representational properties. This visual perspective is particularly valuable given the general, media-agnostic nature of the proposed framework: although the input data here are images, the results illustrate properties of the structural representations themselves, independent of domain.

Figure 5 presents segmentation results for a representative sample from the Birds dataset: the ”flamingos” image from Figure 4. This image poses several challenges—multiple visually similar birds near water, strong background reflections, and only the foreground bird annotated as ground truth. This example typifies the main experimental difficulties and illustrates the distinctive strengths and limitations of each method. From left to right: (a) GIG representation; (b) trivial approach (WATER-PAR, filtered by number of regions); (c) topological approach (WATER-PAR, area); and (d) regional approach (WATER-PAR, Contour).

The GIG representation in Figure 5(a) performs poorly (Jaccard 0.40), producing a mask that fails to distinguish the annotated bird from other similar objects. As GIG relies on edge weights and local color, it is sensitive to background clutter and reflections, which results in over-segmentation and confusion. The trivial approach in Figure 5(b) fares even worse (Jaccard 0.04). By selecting a fixed number of prominent regions (60), it isolates only small, irrelevant fragments and covers almost none of the target object. The lack of learning or adaptation means region selection depends purely on graph topology, not semantic content.

The topological approach in Figure 5(c) achieves a substantial improvement (Jaccard 0.63). By capturing parent relationships through area features in the hierarchical tree, it groups more relevant regions and suppresses some background noise. Nonetheless, faint outlines of other birds and background artifacts persist. This reflects the fact that while the topological representation captures the merging and relative importance of regions within the hierarchy, it does not encode more detailed region-level statistics that might help to resolve ambiguous or closely packed objects. Thus, regions that are structurally similar in the hierarchy may remain grouped or partially segmented together, particularly in visually complex images.

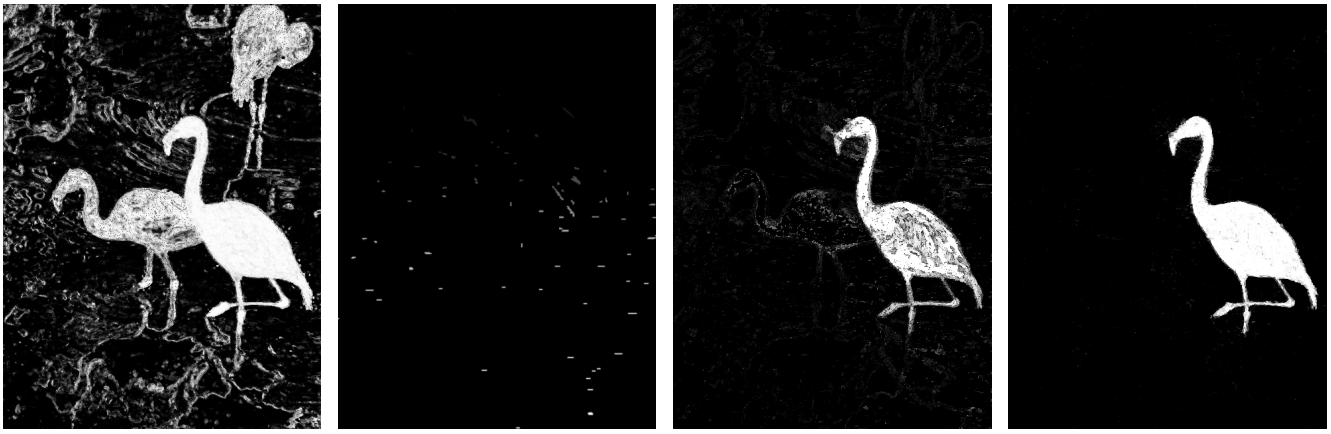
The regional approach Figure 5(d) yields the best result (Jaccard 0.96), accurately segmenting the annotated bird with minimal background interference or confusion from unannotated objects. Here, contour-related features quantify the strength and structure of region boundaries within the hierarchical graph. By systematically encoding how strongly each region is delineated, these features provide richer information about spatial and structural coherence. As a result, the model more reliably isolates the correct region even with visual ambiguity or cluttered backgrounds. The segmentation closely matches the ground truth, with only minor mislabeling inside the bird or along diffuse boundaries.

Figure 6 presents edge detection results for a representative sample from the BSDS500 dataset: the “airplane” image in Figure 3, which is frequently used in the literature for benchmarking. This image contains an object with many small, intricate components (airplane details), set against a complex background of clouds. The combination of detailed annotations and a visually busy scene makes it a strong test case for evaluating method behavior and limitations. From left to right: (a) GIG representation; (b) trivial approach (WATER-PAR, filtered by number of regions); (c) topological approach (WATER-PAR, altitude); and (d) regional approach (QFZ, Gaussian). Individual $F1$-score for this image: GIG: 0.59, trivial: 0.22, topological: 0.56, regional: 0.72.

The GIG method, Figure 6(a) with 0.59 $F1$-score, produces edge maps where the main object contours are strongly delineated, and fine details are preserved—including intricate airplane components such as wheels, markings, and propellers. The output typically highlights both object and background contours, resulting in an appearance similar to an image gradient. This characteristic arises from the construction of the GIG features, which aggregate local edge weights and relational differences across the graph structure. As discussed in our earlier work, the GIG method is designed to emphasize strong transitions and relational structure in the pixel grid, rather than focusing solely on semantic object boundaries. Consequently, the GIG edge maps retain not just the principal boundaries but also reinforce responses at textured regions and spurious gradients, leading to broader edge markings and the inclusion of background elements. This provides rich structural information but can result in thicker boundaries and some background clutter, particularly when fine details are densely annotated in the ground truth.

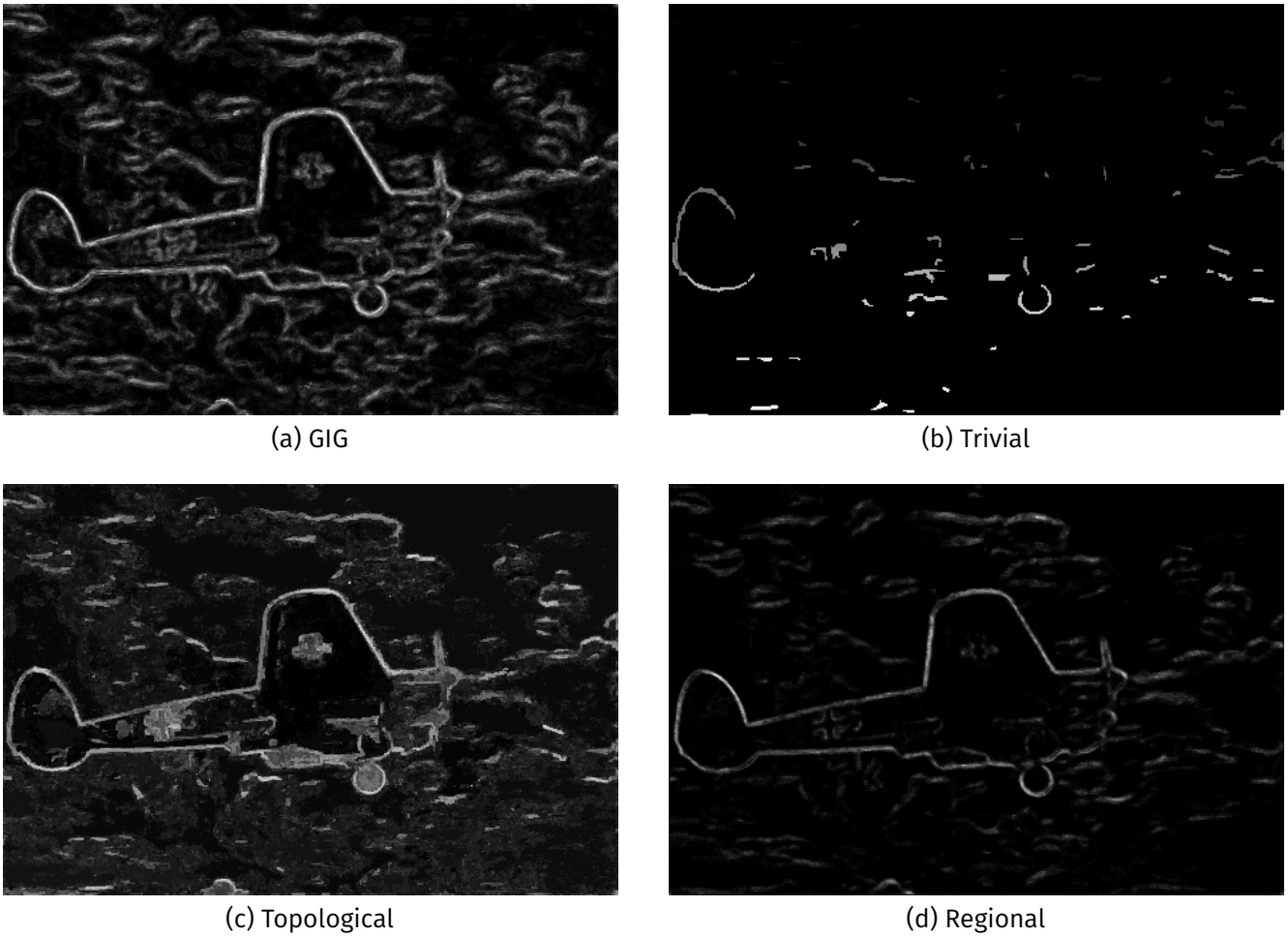
The topological approach in Figure 6(c) with $F1$-score of 0.56, produces results that are visually similar to the GIG representation, with the main object boundaries and small structural details clearly marked. However, this method sometimes predicts solid regions, filling in areas within the main object instead of only outlining them. This behavior arises because topological features such as altitude capture the persistence and merging of regions in the hierarchy. While these features effectively represent the stability and importance of different regions, they do not focus solely on the transitions between regions that define object boundaries. As a result, the learned model occasionally produces thick or filled boundaries, especially for regions that are prominent or merge late in the hierarchical tree.

The trivial approach Figure 6(b) with 0.22 $F1$-score, which uses WATER-PAR filtered by the number of regions, performs better in edge detection than in segmentation. Most selected regions align with object boundaries, but the number remains insufficient to reconstruct the complete contour, so the result is fragmented and incomplete. This reflects a well-known aspect of watershed-based methods: while they can produce precise boundaries, their practical success often depends on careful parameter selection and additional filtering of the partition results. In this task, the trivial method’s over-



(a) GIG (b) Trivial (c) Topological (d) Regional

Figure 5. Qualitative segmentation results for a challenging Birds dataset image with multiple similar birds by water, where only the foreground bird is annotated as ground truth. GIG and trivial both fail to distinguish the annotated bird: GIG includes excessive background detail, while trivial misses most of the object. The topological approach partially segments the target but retains some noise. The regional method clearly delineates the annotated bird with minimal confusion.



(a) GIG (b) Trivial

(c) Topological (d) Regional

Figure 6. Qualitative edge detection results for a standard BSDS500 image containing an airplane with detailed structure and a complex sky background. GIG and topological methods both detect the main object boundaries and fine details but include more background noise and thicker contours. The trivial approach yields fragmented and incomplete edges. The regional method produces thinner, more precise boundaries with less background clutter, providing the closest visual match to the ground truth.

all performance is closer to the learning-based approaches, but its outputs are still more fragmented and strongly affected by the chosen parameters. This outcome highlights the value of adaptive, data-driven strategies when working with hierarchical structures.

Finally, the regional approach Figure 6(d) with 0.72 $F1$-score (QFZ, Gaussian features) achieves the best performance for this image and for the dataset overall. Its prediction cap-

tures most of the object boundaries and key structural details, producing thinner and more precisely localized edges compared to the other methods. Regional features systematically summarize the statistical distribution of values within regions at multiple hierarchy levels, allowing the model to better suppress spurious edges from background texture and noise. While some background contours and small inaccuracies remain, the result is visually cleaner and more focused along the annotated contours. Although the global and individual scores for the GIG, topological, and regional methods are similar, the regional approach produces edges that are less cluttered and more concentrated along relevant structures.

These qualitative results show how structural representations influence performance in real image tasks. Visual inspection clarifies how each method approaches object localization, background suppression, and resistance to noise, aspects that are not always reflected in quantitative metrics alone. Regional and topological features derived from hierarchical structures improve robustness and interpretability, particularly when boundaries are ambiguous or annotation is sparse.

7 Discussion

The great incentive to center the considerations towards graph processing is that they are critical for hierarchical analyses, and machine learning operating on graphs provides a form to create an agnostic model regarding the media type. Machine learning on graphs is a topic of great interest due to: (i) its autonomy—once you have your learning system operating in the vertices and edges, the data's source becomes virtually irrelevant; (ii) the multiple possibilities of applications; and (iii) the capacity to represent multivariate information.

The hierarchical structure provides a non-regular characterization of regions with notions of order and navigation without needing many parametrizations other than those offered by the already modeled edge-weighted graph. They introduce a semantic interpretation into media processing through meaningful partitioning of the perceptual space. Hierarchical operators are idempotent and provide a consistent data organization.

By keeping the formulation on the structures, the proposed framework evades decisions at the media level. It avoids any feature extracted from the media and only uses the information on the hierarchical tree and their conjoined graph. Also, it does not select any particular region that better suits an application. Instead, the entire structure is represented in a vectorial form that preserves its semantical arrangement. Furthermore, the task label attribution is performed at the leaf level at the bottom of the tree; therefore, each leaf has a unique discrete label and does not demand any considerations specific to a task. This is illustrated in our experiments with the Birds dataset, where segmentation relies entirely on hierarchical and structural features. Our framework achieves robust results even when there are visually similar, unlabeled objects and ambiguous boundaries. Such scenarios are known to be difficult for appearance-based models.

By contrast, similar methods in the literature use attributes and regions defined in the hierarchies to gather features from the media for the learning model. For instance, in Grossiord *et al*. [2017], they use the hierarchies aggregated with RF, but the features used as input for the RF are taken from the media guided by the regions defined in the hierarchies. In Hu *et al*. [2021], they also use the RF as the learning method but only for a few sampled regions in the hierarchy described by media features and information about the regions' geometry. In Padilla *et al*. [2021], besides using the hierarchies to model the correspondence between different media, they also define the features to be applied in a Random walk method. However, they do not use all regions in the hierarchy. Instead, to reduce the number of nodes, they filter the structure by searching for stable areas regarding each attribute and perform a majority vote to determine the most critical regions.

On the topological approach proposed, the hierarchical structures are represented by taking the entire set of parents of a leaf that retains the semantical information embedded on the hierarchical trees without the need to filter or select a particular level for evaluation. Experiments with the topological approach showed that it not only contains crucial information about the hierarchies but also improves the typical approach's performance in both tasks. The topological strategy constructs a regular representation that could be used in most available learning models, but the representation's dimensions could be challenging regarding computational resources. The efficient implementations for hierarchical structures and the flexibility of the RF model allow working with these sizable structures.

The second strategy with the regional attributes performs best in both tasks. Procedurally, this approach is equivalent to performing horizontal cuts by altitude levels. However, rather than creating a representation for each cut and evaluating them individually, the method gathers all of them systematically as a regular representation. Equally to the topological approach, the regional strategy avoids any feature extracted from the media, the representation is leaf-centered, and only uses the information on the hierarchical tree and the conjoined graph.

Furthermore, the regional strategy is considerably easier to standardize than the topological approach. While the topological approach took the maximum possible depth in all datasets, resulting in high-dimensional data that often had multiple padding positions due to the multiform structures, the regional approach has only a fixed number of steps in the normalized altitudes available in all hierarchical model types.

One advantage of working with explicit hierarchical features is the ability to directly observe and interpret how different structural attributes influence outcomes for specific tasks. As shown in our qualitative analysis, we can identify which aspects of the hierarchical structure improve results in challenging scenarios. This interpretability enables targeted selection or adjustment of attributes for each application. In contrast, black-box methods provide little visibility into which attributes drive predictions or how representations can be adapted to domain-specific needs. This limitation is particularly relevant in fields where explainability or bias mitigation are critical, such as in medical or human-centered applications.

8 Conclusions

This work introduced a general learning framework for operating directly on hierarchical data, independent of media type or specific task. By focusing on structural and relational properties derived from hierarchical trees and their associated graphs, the proposed approach enables robust and interpretable learning while avoiding reliance on features extracted from the original media.

Unlike most previous methods, which define regions or masks to extract media-dependent features, our framework encodes the entire hierarchical structure in a vector representation. This preserves semantic relationships and enables label assignment without making application-specific assumptions. All components of the hierarchy are utilized, with label attribution performed at the leaf level, ensuring that the model remains generalizable and task-agnostic.

Experiments on edge detection and segmentation tasks demonstrate that learning with explicit hierarchical features can achieve performance comparable to state-of-the-art approaches that depend on pixel or appearance-based attributes. Furthermore, the qualitative analyses highlight the interpretability of the method, showing how structural attributes influence model predictions and making it possible to select or refine features for domain-specific requirements.

While Random Forests were chosen for their scalability and effectiveness in high-dimensional spaces, the framework is not limited to any single classifier and can be adapted to alternative models, including those optimized for large-scale or structured data. This flexibility, along with the ability to represent a variety of hierarchical models, supports a wide range of potential applications beyond the image domain.

Future work will explore the use of more sophisticated learning models within this framework, the incorporation of data reduction techniques that preserve hierarchical information, and the extension to new modalities and application areas. The approach may also be enhanced by selectively integrating certain media-specific attributes where appropriate, provided the hierarchical structure remains central to the representation and learning process.

Overall, this study demonstrates that structural, interpretable, and media-agnostic learning on hierarchies is both feasible and effective, offering a foundation for further developments in generalized, explainable, and domain-agnostic machine learning.

Declarations

Acknowledgements

This work was partially financially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – (Universal 407242/2021-0, PQ 306573/2022-9, 42950/2023-3), the Fundação de Amparo a Pesquisa do Estado de Minas Gerais – FAPEMIG – (PPM-00006-18 and APQ-01079-23), and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES (Finance code 001 and COFECUB 88887.191730/2018-00). The authors also thank PUC Minas and Inria under the project *Learning on graph-based hierarchical methods for image and multimedia data* for the support during this work.

Authors' Contributions

All authors have equally contributed in this work.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Our method can be found online at http://imscience.icei.pucminas.br/codes

References

Adão, M. M., Guimarães, S. J. F., and Patrocínio Jr, Z. K. G. (2020). Learning to realign hierarchy for image segmentation. *Pattern Recognition Letters*, 133:287–294. DOI: 10.1016/j.patrec.2020.03.010.

Almeida, R., Kijak, E., Malinowski, S., Patrocínio Jr, Z. K., Araújo, A. A., and Guimarães, S. J. (2022). Graph-based image gradients aggregated with random forests. *Pattern Recognition Letters*. DOI: 10.1016/j.patrec.2022.08.015.

Almeida, R., Patrocínio Jr., Z. K. G., Araújo, A. d. A., Kijak, E., Malinowski, S., and Guimarães, S. J. F. (2021). Descriptive image gradient from edge-weighted image graph and random forests. In *34th Conference on Graphics, Patterns and Images*, pages 338–345. DOI: 10.1109/SIBGRAPI54419.2021.00053.

Aptoula, E., Weber, J., and Lefèvre, S. (2013). Vectorial quasi-flat zones for color image simplification. In *Mathematical Morphology and Its Applications to Signal and Image Processing.*, pages 231–242. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-38294-9_20.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From contours to regions: An empirical evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301. IEEE. DOI: 10.1109/CVPR.2009.5206707.

Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., and Malik, J. (2014). Multiscale combinatorial grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335. IEEE. DOI: 10.1109/CVPR.2014.49.

Beucher, S. (1979). Use of watersheds in contour detection. In *International Workshop on Image Processing*, pages 2.1–2.12. CCETT/IRISA. Available at:`https://www.researchgate.net/publication/230837989_Use_of_Watersheds_in_Contour_Detection`.

Beucher, S. (1994). Watershed, hierarchical segmentation and waterfall algorithm. In *Computational Imaging and Vision*, volume 2, pages 69–76. Springer. DOI: 10.1007/978-94-011-1040-2_10.

Bondy, J. A., Murty, U. S. R., *et al*. (1976). *Graph theory with applications*, volume 290. Macmillan London. DOI: 10.1007/978-1-349-03521-2.

Bosilj, P., Kijak, E., and Lefèvre, S. (2018). Partition and inclusion hierarchies of images: a comprehensive survey. *Journal of Imaging*, 4:33. DOI: 10.3390/jimaging4020033.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32. DOI: 10.1023/A:1010933404324.

Chen, C., Wu, Y., Dai, Q., Zhou, H.-Y., Xu, M., Yang, S., Han, X., and Yu, Y. (2024). A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10297–10318. DOI: 10.1109/TPAMI.2024.3445463.

Chen, D., Wu, X., Dong, J., He, Y., Xue, H., and Mao, F. (2020). Hierarchical sequence representation with graph network. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2288–2292. IEEE. DOI: 10.1109/ICASSP40776.2020.9054195.

Chierchia, G. and Perret, B. (2020). Ultrametric fitting by gradient descent. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, page 124004, Red Hook, NY, USA. Curran Associates Inc.. DOI: 10.1088/1742-5468/abc62d.

Chuang, C.-Y., Li, J., Torralba, A., and Fidler, S. (2018). Learning to act properly: predicting and explaining affordances from images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 975–983. IEEE. DOI: 10.1109/CVPR.2018.00108.

Clément, M., Kurtz, C., and Wendling, L. (2018). Learning spatial relations and shapes for structural object description and scene recognition. *Pattern Recognition*, 84:197–210. DOI: 10.1016/j.patcog.2018.06.017.

Cousty, J., Bertrand, G., Najman, L., and Couprie, M. (2009). Watershed cuts: minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1362–1374. DOI: 10.1109/TPAMI.2008.173.

Cousty, J., Najman, L., Kenmochi, Y., and Guimarães, S. (2018). Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps. *Journal of Mathematical Imaging and Vision*, 60:479–502. DOI: 10.1007/s10851-017-0768-7.

Cousty, J., Najman, L., and Perret, B. (2013). Constructive links between some morphological hierarchies on edge-weighted graphs. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 86–97. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-38294-9_8.

Dollar, P., Belongie, S., and Perona, P. (2010). The fastest pedestrian detector in the west. In *Procedings of the British Machine Vision Conference*, pages 68.1–68.11. British Machine Vision Association. DOI: 10.5244/C.24.68.

Dollar, P. and Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1558–1570. DOI: 10.1109/TPAMI.2014.2377715.

dos Santos Belo, L., Caetano Jr, C. A., do Patrocínio Jr, Z. K. G., and Guimaraes, S. J. F. (2016). Summarizing video sequence using a graph-based hierarchical approach. *Neurocomputing*, 173:1001–1016. DOI: 10.1016/j.neucom.2015.08.057.

Díaz, G., González, F. A., and Romero, E. (2009). A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. *Journal of Biomedical Informatics*, 42:296–307. DOI: 10.1016/j.jbi.2008.11.005.

Fan, J., Zhao, T., Kuang, Z., Zheng, Y., Zhang, J., Yu, J., and Peng, J. (2017). Hd-mtl: hierarchical deep multi-task learning for large-scale visual recognition. *IEEE Transactions on Image Processing*, 26:1923–1938. DOI: 10.1109/TIP.2017.2667405.

Grossiord, E., Talbot, H., Passat, N., Meignan, M., and Najman, L. (2017). Automated 3d lymphoma lesion segmentation from pet/ct characteristics. In *International Symposium on Biomedical Imaging*, pages 174–178. IEEE. DOI: 10.1109/ISBI.2017.7950495.

Grover, A. and Leskovec, J. (2016). Node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM. DOI: 10.1145/2939672.2939754.

Guigues, L., Cocquerez, J. P., and Men, H. L. (2006). Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317. DOI: 10.1007/s11263-005-6299-0.

Hu, Z., Shi, T., Wang, C., Li, Q., and Wu, G. (2021). Scale-sets image classification with hierarchical sample enriching and automatic scale selection. *International Journal of Applied Earth Observation and Geoinformation*, 105:102605. DOI: 10.1016/j.jag.2021.102605.

Huang, J., Wang, M., Xu, X., Jie, B., and Zhang, D. (2020). A novel node-level structure embedding and alignment representation of structural networks for brain disease analysis. *Medical Image Analysis*, 65:101755–110767. DOI: 10.1016/j.media.2020.101755.

Ilin, R., Watson, T., and Kozma, R. (2017). Abstraction hierarchy in deep learning neural networks. In *International Joint Conference on Neural Networks*, pages 768–774, Anchorage, AK, USA. IEEE. DOI: 10.1109/IJCNN.2017.7965929.

Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F., and Van den Broeck, W. (2011). What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180. DOI: 10.1016/j.jtbi.2010.11.033.

Ji, W., Li, X., Wei, L., Wu, F., and Zhuang, Y. (2020). Context-aware graph label propagation network for saliency detection. *IEEE Transactions on Image Processing*, 29:8177–8186. DOI: 10.1109/TIP.2020.3002083.

Jing, Y., Wang, J., Wang, W., Wang, L., and Tan, T. (2020). Relational graph neural network for situation recognition. *Pattern Recognition*, 108:107544. DOI: 10.1016/j.patcog.2020.107544.

Kiran, B. R. and Serra, J. (2015). Braids of partitions. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 217–228. Springer International Publishing. DOI: 10.1007/978-3-319-18720-4_19.

Krishnammal, P. M., Therase, L. M., Devi, E. A., and Joany, R. M. (2022). Wavelets and convolutional neural networks-based automatic segmentation and prediction of mri brain images. In *IOT with Smart Systems*, pages 229–241, Singapore. Springer. DOI: 10.1007/978-981-16-3945-6_23.

Kurtz, C., Naegel, B., and Passat, N. (2014). Connected filtering based on multivalued component-trees. *IEEE Transactions on Image Processing*, 23:5152–5164. DOI: 10.1109/TIP.2014.2362053.

Kurzweil, R. (2013). *How to create a mind: the secret of human thought revealed*. Penguin Books. DOI: 10.5860/choice.50-6167.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 936–944, Honolulu, HI, USA. IEEE. DOI: 10.1109/CVPR.2017.106.

Liu, Y., Cheng, M.-M., Hu, X., Bian, J.-W., Zhang, L., Bai, X., and Tang, J. (2019). Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1939–1946. DOI: 10.1109/TPAMI.2018.2878849.

Luo, W., Zhang, C., Zhang, X., and Wu, H. (2019). Improving action recognition with the graph-neural-network-based interaction reasoning. In *IEEE Visual Communications and Image Processing*, pages 1–4. IEEE. DOI: 10.1109/VCIP47243.2019.8965768.

Maia, D. S., Pham, M.-T., Aptoula, E., Guiotte, F., and Lefevre, S. (2021). Classification of remote sensing data with morphological attribute profiles: a decade of advances. *IEEE Geoscience and Remote Sensing Magazine*, 9:43–71. DOI: 10.1109/MGRS.2021.3051859.

Makarov, I., Kiselev, D., Nikitinsky, N., and Subelj, L. (2021). Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ Computer Science*, 7. DOI: 10.7717/peerj-cs.357.

Makrogiannis, S., Annasamudram, N., Wang, Y., Miranda, H., and Zheng, K. (2021). A system for spatio-temporal cell detection and segmentation in time-lapse microscopy. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 2266–2273, Houston, TX, USA. IEEE. DOI: 10.1109/BIBM52615.2021.9669421.

Maninis, K.-K., Pont-Tuset, J., Arbelaez, P., and Gool, L. V. (2018). Convolutional oriented boundaries: from image segmentation to high-level tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:819–833. DOI: 10.1109/TPAMI.2017.2700300.

Mansilla, L. A. C. and Miranda, P. A. V. (2016). Oriented image foresting transform segmentation: connectivity constraints with adjustable width. In *Conference on Graphics, Patterns and Images*, pages 289–296. IEEE. DOI: 10.1109/SIBGRAPI.2016.047.

Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. MIT Press. DOI: 10.7551/mitpress/9780262514620.001.0001.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision*, pages 416–423. IEEE Comput. Soc. DOI: 10.1109/ICCV.2001.937655.

Meyer, F. (2001). Hierarchies of partitions and morphological segmentation. In *Scale-Space and Morphology in Computer Vision*, volume 2106, pages 161–182. Springer Berlin Heidelberg. DOI: 10.1007/3-540-47778-0_14.

Micheli, A. (2009). Neural network for graphs: a contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511. DOI: 10.1109/tnn.2008.2010350.

Mihalcea, R. and Radev, D. (2012). Graph-based natural language processing and information retrieval. DOI: 10.1017/cbo9780511976247.

Najman, L. and Couprie, M. (2006). Building the component tree in quasi-linear time. *IEEE Transactions on Image Processing*, 15:3531–3539. DOI: 10.1109/TIP.2006.877518.

Najman, L. and Schmitt, M. (1996). Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1163–1173. DOI: 10.1109/34.546254.

Najman, L. and Talbot, H. (2013). *Mathematical morphology: from theory to applications*. John Wiley & Sons, Inc., 1 edition. DOI: 10.1002/9781118600788.

Newman, M. (2018). *Networks*. Oxford university press. Book.

Nguyen, T. T., Krishnakumari, P., Calvert, S. C., Vu, H. L., and van Lint, H. (2019). Feature extraction and clustering analysis of highway congestion. *Transportation Research Part C: Emerging Technologies*, 100:238–258. DOI: 10.1016/j.trc.2019.01.017.

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828. DOI: 10.1109/jproc.2018.2820126.

Padilla, F. J. A., Romaniuk, B., Naegel, B., Servagi-Vernat, S., Morland, D., Papathanassiou, D., and Passat, N. (2021). Random walkers on morphological trees: a segmentation paradigm. *Pattern Recognition Letters*, 141:16–22. DOI: 10.1016/j.patrec.2020.11.001.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830. DOI: 10.48550/arxiv.1201.0490.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM. DOI: 10.1145/2623330.2623732.

Perret, B., Chierchia, G., Cousty, J., Guimarães, S. F., Kenmochi, Y., and Najman, L. (2019a). Higra: hierarchical graph analysis. *SoftwareX*, 10:100335. DOI: 10.1016/j.softx.2019.100335.

Perret, B., Cousty, J., Guimaraes, S. J. F., and Maia, D. S. (2018). Evaluation of hierarchical watersheds. *IEEE Transactions on Image Processing*, 27:1676–1688. DOI: 10.1109/TIP.2017.2779604.

Perret, B., Cousty, J., Guimarães, S. J. F., Kenmochi, Y., and Najman, L. (2019b). Removing non-significant regions in hierarchical clustering and segmentation. *Pattern Recognition Letters*, 128:433–439. DOI: 10.1016/j.patrec.2019.10.008.

Qi, X., Liao, R., Jia, J., Fidler, S., and Urtasun, R. (2017). 3d graph neural networks for rgbd semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 5209–5218. IEEE. DOI: 10.1109/ICCV.2017.556.

Sakarya, U. and Telatar, Z. (2010). Video scene detection using graph-based representations. *Signal Processing: Image Communication*, 25(10):774–783. DOI: 10.1016/j.image.2010.10.001.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80. DOI: 10.1109/TNN.2008.2005605.

Selvan, R., Kipf, T., Welling, M., Juarez, A. G.-U., Pedersen, J. H., Petersen, J., and de Bruijne, M. (2020). Graph refinement based airway extraction using mean-field networks and graph neural networks. *Medical Image Analysis*, 64:101751. DOI: 10.1016/j.media.2020.101751.

Serna, A. and Marcotegui, B. (2014). Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:243–255. DOI: 10.1016/j.isprsjprs.2014.03.015.

Soille, P. (2008). Constrained connectivity for hierarchical image partitioning and simplification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1132–1145. DOI: 10.1109/TPAMI.2007.70817.

Soille, P. and Najman, L. (2012). On morphological hierarchical representations for image processing and spatial data clustering. In *Applications of Discrete Geometry and Mathematical Morphology*, pages 43–67. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-32313-3_4.

Sokal, R. R. and Rohlf, J. (1962). The comparison of dendrograms by objective methods. *TAXON*, 11:33–40. DOI: 10.2307/1217208.

Sun, L., Huo, Q., Jia, W., and Chen, K. (2015). A robust approach for text detection from natural scene images. *Pattern Recognition*, 48:2906–2920. DOI: 10.1016/j.patcog.2015.04.002.

Taylor, C. J. (2013). Towards fast and accurate segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1916–1922. IEEE. DOI: 10.1109/CVPR.2013.250.

Tochon, G., Mura, M. D., Veganzones, M. A., Valero, S., Salembierr, P., and Chanussot, J. (2018). Advances in utilization of hierarchical representations in remote sensing data analysis. In *Reference Module in Earth Systems and Environmental Sciences*, pages 77–107. Elsevier. DOI: 10.1016/B978-0-12-409548-9.10340-9.

Tong, H., He, J., Li, M., Zhang, C., and Ma, W.-Y. (2005). Graph based multi-modality learning. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 862–871. DOI: 10.1145/1101149.1101337.

Vachier, C. and Meyer, F. (1995). Extinction value: a new measurement of persistence. In *IEEE Workshop on nonlinear signal and image processing*, volume 1, pages 254–257. Neos Marmaras Greece. Available at:`https://www.researchgate.net/publication/228957197_Extinction_value_A_new_measurement_of_persistence`.

Wang, D., Cui, P., and Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1225–1234. ACM. DOI: 10.1145/2939672.2939753.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24. DOI: 10.1109/tnnls.2020.2978386.

Wyner, A., Olson, M., Bleich, J., and Mease, D. (2017). Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18:1558–1590. DOI: 10.5555/3122009.3153004.

Xu, C., Whitt, S., and Corso, J. J. (2013). Flattening supervoxel hierarchies by the uniform entropy slice. In *IEEE International Conference on Computer Vision*, pages 2240–2247. IEEE. DOI: 10.1109/ICCV.2013.279.

Xu, C., Xiong, C., and Corso, J. J. (2012). Streaming hierarchical video segmentation. In *European Conference on Computer Vision*, pages 626–639. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-33783-3_45.

Yang, G., Cao, J., Chen, Z., Guo, J., and Li, J. (2020). Graph-based neural networks for explainable image privacy inference. *Pattern Recognition*, 105:107360. DOI: 10.1016/j.patcog.2020.107360.

Zhang, J., Tsai, P.-H., and Tsai, M.-H. (2024). Semantic2graph: graph-based multi-modal feature fusion for action segmentation in videos. *Applied Intelligence*, 54(2):2084–2099. DOI: 10.1007/s10489-023-05259-z.

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., and Hoffman, M. M. (2019). Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91. DOI: 10.1016/j.inffus.2018.09.012.

Zwettler, G. and Backfrieder, W. (2015). Evolution strategy classification utilizing meta features and domain-specific statistical a priori models for fully-automated and entire segmentation of medical datasets in 3d radiology. In *International Conference on Computing and Communications Technologies*, pages 12–18, Chennai, India. IEEE. DOI: 10.1109/ICCCT2.2015.7292712.