Learning on graphs and hierarchies

Raquel Pereira de Almeida

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par

Raquel PEREIRA DE ALMEIDA

Learning on graphs and hierarchies

Thèse présentée et soutenue à Rennes, le 27 mars 2023
Unité de recherche : Institut de Recherche en Informatique et Systèmes Aléatoires

Rapporteurs avant soutenance :

Davide BACCIU — Professeur associé, Université de Pise, Italie
Alexandre FALCÃO — Professeur, Unicamp, Brésil

Composition du Jury :

Président :	Laurent NAJMAN	Professeur, Université Gustave Eiffel, France
Examinateurs :	Yukiko KENMOCHI	Chargé de recherche DR CNRS, Greyc, France
	Zenilton PATROCÍNIO	Professeur, PUC-Minas, Brésil
	Davide BACCIU	Professeur associé, Université de Pise, Italie
	Alexandre FALCÃO	Professeur, Unicamp, Brésil
Dir. de thèse :	Laurent AMSALEG	Directeur de recherche CNRS, Université de Rennes 1, France
Co-dir. de thèse :	Silvio GUIMARÃES	Professeur, PUC-Minas, Brésil

Invité(s) :

Ewa KIJAK — Maître de Conférence, Université de Rennes 1, France
Simon MALINOWSKI — Maître de Conférence, Université de Rennes 1, France

Acknowledgement

First and foremost, I would like to thank my esteemed supervisor, Dr. Silvio Jamil Ferzoli Guimarães, for the encouragement that guided me through my academic life and whose invaluable advice completely changed my personal life. I am also extremely grateful to my supervisors, Dr. Ewa Kijak, Dr. Simon Malinowsky, and Dr. Laurent Amsaleg, for their knowledge, continuous patience, and kind help that made my study possible in a foreign country. I would also like to thank Dr. Zenilton K. G. do Patrocínio Jr and Dr. Arnaldo de A. Araújo for their tremendous support in my academic life. Finally, I would like to express my gratitude to my beloved husband, Tristan, for his unconditional love, immense understanding, and encouragement that made it possible for me to finish this work.

Abstract

Hierarchies, as described in mathematical morphology, represent nested regions of interest and provide mechanisms to create concepts and 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 applications that can be conveniently generalized. Due to the deterministic algorithms, the multiform and distinct representations, and the absence of a direct way to evaluate the hierarchical representation quality, it is hard to insert hierarchical information into a learning framework and benefit from the recent advances in the field. 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 the application. The idea is to study ways to transform the data to a regular representation required by most learning models while preserving the rich information in the hierarchical structure. It proposes to study and formalize the concepts as graphs, a common point for hierarchies and multimedia, and a topic of great interest for machine learning. The methods in this study use edge-weighted image graphs and hierarchical trees as input, evaluating different proposals on the edge detection and segmentation tasks. The primary model is the Random Forest, a fast, inspectable, and scalable method suited to work with high-dimensional data. Despite the media, tasks, and model choices, it focuses the formulations on graphs and hierarchical trees and only uses the tasks to evaluate the response produced by different characteristics. It gives the results in quantitative and qualitative terms and offers statistical analyses of the data distribution and dimensionality, assessing their impact on learning. Furthermore, it provides a critical systematic review of proposals in the literature that integrates machine learning and hierarchies. It demonstrates that it is possible to create a learning framework dependent only on the hierarchical data that performs well in multiple tasks.

Table of Contents

List of Tables

List of Figures

List of Definitions

List of Algorithms

List of abbreviations

Abbreviation	Meaning
2D/3D/4D	two/three/four dimensions
ACM	association for computing machinery
ALPHA	$\alpha-$tree hierarchy
AP	average precision
ARG	attribute relational graph
BOW	bag-of-words scheme
BPH	binary partition hierarchy
BPT	binary partition tree
BSDS500	Berkeley segmentation dataset and benchmark
CNN	convolutional neural network
CPU	central processing unit
CT	computed tomography
GB	gigabyte
GIG	graph-based image gradient
gPb-owt-ucm	Arbeláez, et al. (2011) hierarchical method
GPU	graphics processing unit
HED	holistically-nested edge detection
hGB	Guimarães, et al. (2017) hierarchical graph
HOG	histogram of gradient descriptors
HOPE	high-order proximity
IEEE	institute of electrical and electronics engineers
kNN	$k-$nearest neighborhood model
LiDAR	laser imaging, detection, and ranging
LLE	locally linear embedding
MRF	Markov random field
MRI	magnetic resonance imaging
MSF	minimum spanning forest
MST	minimum spanning tree
NN	neural network
ODS	optimal dataset scale
OIS	optimal image scale
PCA	principal component analysis
PET	positron emission tomography
PRI	probabilistic random index
QFZ	quasi-flat zones
RAG	region adjacency graph
RCF	richer convolutional features
RF	random forest
RGB	red, green, blue color channels in an image
RGBD	red, green, blue, and depth channels in an image
SAR	synthetic aperture radar
SDNE	structural deep network embedding
SED	structured edge detection
SLIC	simple linear iterative clustering
SpixelFCN	superpixel segmentation with fully convolutional networks
SVM	support vector machine
UCM	ultrametric contour map
WATER-AREA	watershed hierarchy by area
WATER-DYN	watershed hierarchy by dynamics
WATER-PAR	watershed hierarchy by number of parents
WATER-VOL	watershed hierarchy by volume

LIST OF SYMBOLS AND NOTATIONS

Symbols		Symbols		Notations	
	Graphs		**Hierarchies**		
G	graph	$\mathbb{C}$	connected component	$(\,,\,)$	ordered pair
$G = (V, E)$	graph	$\mathcal{H}$	hierarchy	$\subseteq$	is subset of
V	set of vertices	$H\ H_1\ H_2$	elements of $\mathcal{H}$	$\times$	cartesian product
E	set of edges	$\mathbb{P}\ \mathbb{P}'\ \mathbb{P}''$	partitions	$\in$	is a member of
$u\ v\ y$	vertices	X, Y	regions	$\neq$	not equal
Γ	adjacency relation	k	number of levels in $\mathcal{H}$	$\emptyset$	empty set
$(G, \mathcal{F})$	edge-weighted graph	$[\mathbb{P}]_v$	singleton partition	$=$	equal
$\mathcal{F}$	edge weight function	$\mathbb{P}_k$	single partition	$\Longleftrightarrow$	if and only if
$\mathcal{F}(E)$	weight map for the edges	$\mathcal{R}_\mathcal{H}$	set of regions in $\mathcal{H}$	$\forall$	for all
w	edge weight value	$\mathcal{T}_\mathcal{H}$	hierarchical partition tree	$(\cdot)$	function of
$\pi\ \pi_1\ \pi_2$	graph paths	$\mathcal{L}$	set of leaves	$\{\}$	set
$\ell\ l$	size of a sequence	n	node in a tree	$\vert$	such that
$G'\ G''$	subgraphs	$\mathcal{N}$	set of nodes in a tree	$\rightarrow$	maps to
V'	subset of vertices	$\mathcal{R}_n$	node region in $\mathcal{H}$	$\mathbb{R}$	set of real numbers
E'	subset of edges	d_n	depth of a node	$(\cdots)$	sequence
$G = (V', \epsilon)$	graph induced by V'	$\mathbb{E}$	range of weight values	$i\ j$	sequence auxiliary
ϵ	edges existing in V'	$\lambda\ \alpha$	threshold values	$\vert\cdot\vert$	the cardinality of
$\mathbb{V}$	set of all subsets on V	$\mathcal{M}$	graph of subsets minima	$[[\,]]$	array
$\mathbb{G}$	set of all subgraphs on G	$\mathbb{M}_\mathcal{F}$	set of weight minima	$\leftarrow$	array attribution
	Image graphs	$\mathcal{S}$	sequence of minima	$\sqsubseteq$	inclusion order
N	number of pixels	$\mathcal{G}$	sequence of subgraphs	$\cap$	set intersection
f	function on the vertices		**Regular topological**	$\cup$	set union
f_{gray}	grayscale function	l	leaf in $\mathcal{L}$	$\mathbb{R}^+$	positive real set
$\mathcal{F}_\mathbf{euc}$	edges Euclidean distance	$\mathbf{T}_l$	leaf topological vector	$[\,,]$	interval endpoints in
$\nabla_\mathcal{F}$	weight function gradient	$\mathtt{alt}_n$	altitude of a node	$]\,,[$	interval no endpoints
$\partial_v f(u)$	edge derivative at vertex v	$\mathtt{area}_n$	area of a node	$\mathbb{A}, \mathbb{B}$	generic sets
$\mathcal{F}_\mathbf{max}$	edges max function	τ_n	subtree rooted in n	$\leq$	is less or equal to
$\mathcal{F}_\mathbf{min}$	edges min function	$\mathcal{L}_n$	subset of leaves in τ_n	$\geq$	is greater or equal to
$\mathcal{F}_\mathbf{mean}$	average function on edges	$\mathtt{vol}_n$	volume of a node	$\mathbb{Z}$	set of intergers
$\mathcal{F}_{\ell_0}$	edges cardinality function	$\mathtt{par}_n$	parent of a node	$\subset$	proper subset of
	Regular graph	$\mathtt{dyn}_n$	dinamics of a node	∇	gradient operator
$\mathbf{X}_v$	vertex attribute	$\mathcal{E}_n$	extinction value of a node	∂	derivative operator
$\mathbf{X}_\mathcal{F}$	edge weight attribute	p_t	the dimension of $\mathbb{T}_l$	$\vert_u$	evaluation point
$\mathbf{X}_v$	vertex vector	$\mathbb{T}_\mathcal{H}$	$\mathcal{H}$ topological regular	$\wedge$	logical and
p	vertex vector dimension	$\mathbb{T}$	set of hierarchies in dataset	$\sim$	approximately
$\mathbf{G}_\mathbf{att}$	graph selected attributes	$\mathbf{Y}_l$	label of a leaf	$\sum$	sum expression
$\mathcal{X}_G$	regular graph	$\mathtt{topo}$	topological candidates set		
$\mathbf{Y}_v$	vertex label	par	number of parents		
$\mathcal{D}$	graph regular train	$\mathbf{P}_l$	set of parents of a leaf		
T	total number of vertices	$\mathcal{D}_t$	topological regular train		
$\mathbf{Y}$	set of labels in train set	T_l	total number of leaves		

Symbols		Symbols	
Random Forest		**Regular regional**	
M	number of trees in RF	σ	threshold value
m	internal node RF tree	β	series of altitude levels
h	split function	$\mathbf{R}_l$	leaf regional vector
$\mathbf{x}$	split query	$\mathbb{P}_\sigma$	partition at σ level
θ_m	split parameter on m	$\texttt{area}_{\mathcal{R}_n}$	area of a region
Region adjacency graph		$\mathcal{L}_{\mathcal{R}_n}$	subset of leaves in $\mathcal{R}_n$
I	image	$\texttt{contour}_{\mathcal{R}_n}$	region contour strength
$\mathbb{S}_I$	superpixel region set	ζ	contour length
r	superpixel region	$\texttt{inertia}_{\mathcal{R}_n}$	region moment of inertia
R	number of regions in $\mathbb{S}$	(x, y)	media's coordinates
G_{rag}	region adjacency graph	$\texttt{coord}$	set of coordinates
V_{rag}	vertices set of G_{rag}	$(\bar{x}, \bar{y})$	centroid's coordinates
E_{rag}	edge set of G_{rag}	$\mu_{02}\,\mu_{20}$	central moment
$v_p\,v_q$	vertices in V_{rag}	$\texttt{gaussian}_{\mathcal{R}_n}$	region gaussian distr.
$\mathbf{X}_{\mathcal{F}_{rag}}$	G_{rag} edge attribute	$\texttt{mean}_{\mathcal{R}_n}$	mean weights of a region
		$\texttt{var}_{\mathcal{R}_n}$	region weights variance
		$W_{\mathcal{R}_n}$	region leaf weights sum
		$\mathbb{R}_{\mathcal{H}}$	$\mathcal{H}$ regional regular
		$\texttt{reg}$	regional candidates set
		$\mathcal{D}_r$	regional regular train

INTRODUCTION

Hierarchies are an inherent property composing several elements in real life, relating to how we naturally perceive patterns, scenes, and movement [1]. According to KURZWEIL (2013) [2], there is a pattern identifier in the core of our visual perception, operating hierarchically to recognize parts, objects, and abstract concepts simultaneously. The perceptual hierarchy is difficult to translate to computer models mimicking our ability to perceive reality's intrinsic nature. But, in visual media processing, mathematical morphology has an edge in defining, creating, and manipulating hierarchies.

Hierarchical methods, formulated in mathematical morphology [3], provide semantically arranged structures of nested regions that are easy to navigate and interpret, remaining very popular since their creation [4] [5]. However, they are hard to evaluate and insert into learning frameworks to benefit from the recent advances in the field [6].

This thesis centers its study on hierarchies, aiming to create a learning framework that could operate on the hierarchical structures from the media processing perspective. This introduction presents the context defining hierarchies, details the problem of inserting them into a learning framework, and states the goals of this study, establishing some hypotheses and questions to answer. In the end, it presents the organization of the thesis to facilitate the navigation through the document.

1. MARR David (1982). *Vision: a computational investigation into the human representation and processing of visual information.*
2. KURZWEIL Ray (2013). *How to create a mind: the secret of human thought revealed.*
3. NAJMAN Laurent and TALBOT Hugues (2013a). *Mathematical morphology: from theory to applications.*
4. MEYER Fernand and BEUCHER Serge (1990). *Morphological segmentation.*
5. MASSARO Alessandro (2021). *Image vision advances.*
6. PERRET Benjamin, COUSTY Jean, GUIMARAES Silvio Jamil F., et al. (2018). *Evaluation of hierarchical watersheds.*

Contextualizing hierarchies

Hierarchies are broadly defined in the literature and could represent different notions. For instance, literature presents hierarchies as a method's abstraction [7], a description of model architectures [8], and a form to organize features [9] or related concepts [10]. This broad definition reinforces the notion that hierarchies are the natural organization form of data, particularly the visual data in multimedia, where multiple models try to mimic this organization [11] [12].

Morphological hierarchies use non-linear transformations to gather information based on the reaction they produce [13]. In this sense, it defines hierarchical principles as transformations obtained by applying the proper operators. The operators and concepts present a solid mathematical formalism using the non-linear geometric space to represent the formulations and generalize the set theory of complete lattices [14].

Their methods represent nested regions of interest that provide easy 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, and frequency, applied to tasks such as image segmentation, video action recognition, and time series processing [15]. At the same time, hierarchies produce multiform representations, their algorithms are primarily deterministic, and there is no direct way to evaluate their quality.

Recently, deep learning architectures drastically changed the computational paradigm for visual tasks [16]. The main advantage of deep learning methodology is that it does not require an engineered model to operate, meaning it can learn the features to represent the

7. Ilin Roman, Watson Thomas, and Kozma Robert (2017). *Abstraction hierarchy in deep learning neural networks.*
8. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*
9. Lin Tsung-Yi et al. (2017). *Feature pyramid networks for object detection.*
10. Fan Jianping et al. (2017). *HD-MTL: hierarchical deep multi-task learning for large-scale visual recognition.*
11. Lindsay Grace W. (2021). *Convolutional neural networks as a model of the visual system: Past, present, and future.* 10.
12. Dotsenko Viktor S. (1986). *Hierarchical model of memory.* 1-2.
13. Najman Laurent and Talbot Hugues (2013a). *Mathematical morphology: from theory to applications.*
14. Serra Jean (2006). *A lattice approach to image segmentation.*
15. Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey.*
16. O'Mahony Niall et al. (2019). *Deep learning vs. traditional computer vision.*

data and the models to describe it [17]. The success of these approaches relies on a hierarchy of concepts learned through the network [18]. For instance, in the object recognition task, the raw pixel on the input layer is understood as segments and parts until composing the object concept at the last layers.

The typical deep learning approach is far from ideal, as it imposes a rigid structure for the input, which limits their generalization capabilities for multiform data [19]. Furthermore, even with the recent advances in explaining and inspecting the networks, the reasoning behind the inferences remains obscure [20] [21] and needs to be more empirical than formal.

In both subject areas—hierarchies of partitions and deep learning—hierarchy is the reaction created by the applied operations. Hierarchies of partitions have the hierarchies as an integral part of the structures, but deterministic methods producing heterogeneous data are challenging to improve using machine learning. In contrast, deep learning presents implied hierarchical concepts, but the generalization and reasoning are limited.

17. Liu Weibo et al. (2017). *A survey of deep neural network architectures and their applications.*
18. Zeiler Matthew D. and Fergus Rob (2014). *Visualizing and understanding convolutional networks.*
19. Bacciu Davide et al. (2020). *A gentle introduction to deep learning for graphs.*
20. Kuo Jay (2016). *Understanding convolutional neural networks with a mathematical model.*
21. Montavon Grégoire, Samek Wojciech, and Müller Klaus-Robert (2018). *Methods for interpreting and understanding deep neural networks.*

Problem formulation

In practical applications, morphological hierarchies help perform semantic tasks in visual data processing, such as object proposal, semantic contour, and semantic segmentation [22]. However, they require thorough preprocessing of the data [23] [24] and strategies to deal with issues like over/under-partitioning of the space [25] [26] or selecting an ideal number of regions [27]. Therefore, it is difficult to generalize a successful approach to other media and tasks.

For a generalization in terms of the media, 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 deterministic, the quality assessment, and the heterogeneous aspects.

22. Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey.*
23. Clément Michaël, Kurtz Camille, and Wendling Laurent (2018). *Learning spatial relations and shapes for structural object description and scene recognition.*
24. Nguyen Tin T. et al. (2019). *Feature extraction and clustering analysis of highway congestion.*
25. Nandy Kaustav et al. (2011). *Supervised learning framework for screening nuclei in tissue sections.*
26. Zwettler Gerald and Backfrieder Werner (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.*
27. Meyer Fernand (2001). *Hierarchies of partitions and morphological segmentation.*

Thesis statement

This thesis argues that 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 and task.

Goals and questions

The main goal of this thesis is to design a learning framework that can operate on hierarchical data and is agnostic to the media 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. Therefore, the investigative study in this thesis aims to answer three main questions:

Question 1: How do hierarchical methods model various media information, and what are the practical challenges faced when applying them to a learning framework?

In the hierarchical study, a critical understanding is how the media's building blocks relate at the low level to group them in homogeneous regions. Visual data, such as images and videos, are organized data 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, the grouping strategy and their storage in the hierarchical structure follow the same rules.

Given these considerations, this thesis studies the hierarchical structures, inspecting their strengths and limitations. It also offers a systematic review of the literature on "Learning on hierarchies", which inquires how hierarchies are inserted in a learning framework. It assesses the advantages of hierarchical information in the learning process and the improvement machine learning can bring to the hierarchical representation.

Hypothesis 1

Hierarchical representations contain valuable information embedded in their structures for a generic learning framework, and the learning framework could assist in processing the structure.

Question 2: How to create a learning framework agnostic to media and tasks?

Answering this question requires defining an appropriate representation, ideally shared among most media types and provided with the capacity to retain the information presented in the original media. And also, the task definition should not impose assumptions on the data source.

Graphs are structures used to represent objects, and the primary concern in graph theory is how these objects are interconnected. They can depict many data and carry information about the objects in their components, including from different domains, such as numerical, textual, and logical. In this sense, despite their differences, multimedia data share the same rules once modeled as graphs. Also, one way to represent hierarchical data is as hierarchical trees. Therefore, both graphs and hierarchies have formalisms intersecting with graph theory and applications that can be conveniently generalized.

Given these considerations, this thesis proposes taking graph representations to model the learning framework. **To be explicit, it does not present a multimedia application.** Instead, the formulations and considerations focus on the graph structures which is a common point between hierarchies and multimedia modeling.

Furthermore, it proposes using the Random Forests [28], a fast, simple, and scalable model capable of dealing with high dimensional data and with satisfactory results in multiple tasks. The main challenge in this proposal concerns the regular representation required by most machine learning algorithms, including Random Forests. 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 graphs while remaining discriminant for a task.

28. Breiman Leo (2001). *Random forests.*

Hypothesis 2

Using a selection of graph attributes as input to the learning framework allows the formulation of a model agnostic to the media, and casting the information at the graphs' components level allows assigning each entry with a task label without imposing assumptions on the data source.

Question 3: Could the hierarchical structure provide useful information in an agnostic learning framework?

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.

Answering this question requires considering the semantical arrangement within the hierarchies, and 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 asses 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 for an application.

Given these considerations, this thesis proposes to use the topological and regional features of the hierarchical structure, transposed to an ordered representation that respects their original arrangement.

Hypothesis 3

The topology of the hierarchical structures alone could be used in a learning framework to solve multiple tasks if it preserves their semantical arrangement.

Thesis organization

Given these goals and questions, this thesis is organized into three main parts, each addressing one question and corroborating one hypothesis. Specifically:

Part 1: comprises Chapters 1 and 2, assessing the first question. Chapter 1 contextualize morphological hierarchies, presents the hierarchical methods' strengths and limitations, and describes the different hierarchical types used in the thesis. It also explains and formalizes graphs and hierarchies on the shared notation describing their components and terminologies, followed by a discussion in a typical framework delimiting the target problem for this thesis. Chapter 2 features a systematic review of the literature on "Learning on hierarchies", which is the first on the theme to the best of our knowledge. The search aims to gather the learning strategies applied to the hierarchical structures and portray the most promising approaches relevant to this work.

Part 2: comprises Chapters 3 and 4, assessing the second question. Chapter 3 provides some graph considerations contemplated as essential to this work's development and presents a literature review of machine learning on graphs, exploring the motivations, strategy, and main issues. It reviews deep learning on graphs to formulate the pertinence and identify the limitations, concentrating on the multimedia processing perspective. Chapter 4 presents the case study for a learning framework operating on a selection of graph attributes, establishing the framework for the hierarchical structures. It assesses the regular representation problem and contains investigative experiments, results, and analysis.

Part 3: comprises Chapter 5 , assessing the third and final question. Chapter 5 presents the culmination of the proposals, expanding the concepts and strategies to the hierarchical data. It creates and delivers a learning framework operating directly on the hierarchical data, focusing the formulations solely on the structural components of the hierarchies.

The final chapter **Conclusions**, at the end of this document, presents a discussion considering the different aspects of the experimental investigation, summarizes the observed properties, and draws some findings to guide future work in the hierarchical study.

Fig. 1 presents a graphical overview of the document organization, indicating the association between the main sections and their subject matter.

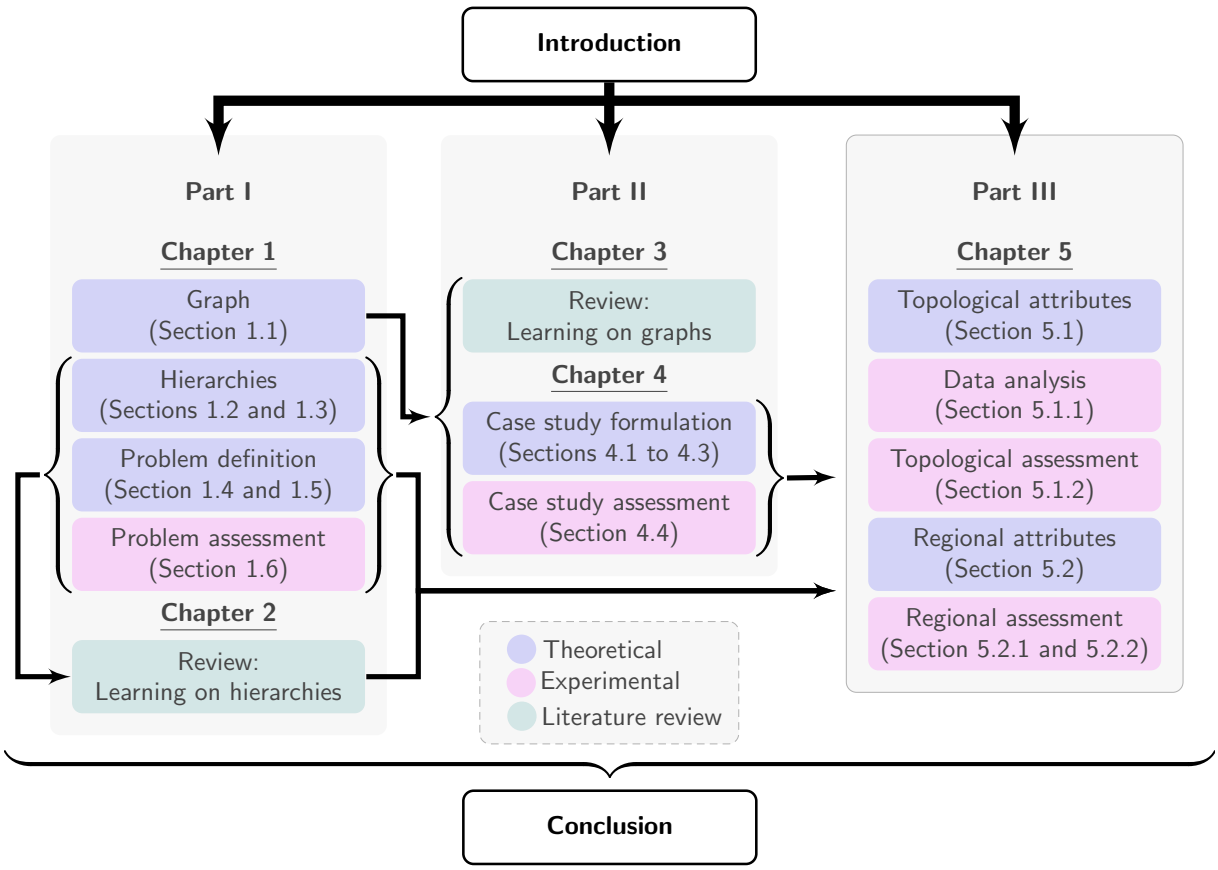


Figure 1: Figure presenting a graphical outline of the document grouped by parts and displaying the main sections. The colors indicate the theme (theoretical, experimental, and literature review), and the arrows show the conceptual dependency between the chapters and sections. All sections within a part are interdependent.

Part I

Hierarchical data

Chapter 1

Hierarchies and graphs

The theory of hierarchies is well defined in mathematical morphology, presenting formulations with a solid theoretical foundation working as a base for a broad range of applications and efficient implementations. 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 [1]. The scale-set theory, a sub-area of mathematical morphology, formalizes the hierarchical principles guiding the morphological operators [2].

Definition 1: Hierarchical principles

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.

Guigues, Cocquerez, and Men (2006) formalisms for the hierarchical principles are in the image domain, as do most literature [3] and mathematical morphology studies [4]. Primarily because of the natural correlation between image coordinates and the lattice structure but also because of the clear region connotation and the visually inspectable results.

However, hierarchies represent partitions defined for regions, often called hierarchies of partitions [5]. Partitions segment the space into disjoint regions with a perceptual meaning [6]. The regions could characterize desired characteristics from various sources, such as

1. Najman Laurent and Talbot Hugues (2013a). *Mathematical morphology: from theory to applications.*
2. Guigues Laurent, Cocquerez Jean Pierre, and Men Hervé Le (2006). *Scale-sets image analysis.*
3. Fehri Amin (2018). *Image characterization by morphological hierarchical representations.*
4. Serra Jean (2006). *A lattice approach to image segmentation.*
5. Ronse Christian (2014). *Ordering partial partitions for image segmentation and filtering: merging, creating and inflating blocks.*
6. Serra Jean (2006). *A lattice approach to image segmentation.*

pixels, voxels, frequency transformations, or sound waves [7]. Independently of the media, hierarchies provide ordered representations of regions at different scales given a criterion.

In Cousty, Najman, and Benjamin Perret (2013) [8], the authors provided formal links between the morphological partitions and edge-weighted graphs. This connection offers additional media generalization tools since graphs are often considered a generic data structure [9]. In practical terms, it models any sets of objects given their connections and relative position. Despite the difference between the multimedia data, once modeled as graphs, they share the same rules in graph theory.

This chapter introduces the theory of hierarchies as graphs. Section 1.1 formalizes graph concepts describing their components and terminologies and providing some essential considerations. Section 1.2 connects graphs and hierarchies, and Section 1.3 describes the different hierarchical types contemplated in the thesis.

The remainder of the thesis manuscript tackles the problem delineated in Sections 1.4 and 1.5. These sections illustrate a typical pipeline for hierarchies, presenting some important considerations and delimiting the problem. Ensuing, some experiments in Section 1.6 establish a baseline. Finally, a brief discussion in Section 1.7 summarizes the main points of this chapter.

1.1 Graph's formalism and notions

This section offers the main concepts of graph theory, defining components, terminologies, and notions relevant to this work.

Definition 2: Graph

A **graph** $G(V, E)$ consists of a finite set of vertices, denoted by V, and a finite set of edges denoted by E, where $E \subseteq V \times V$. 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 representing the data's elemental components, and a graph is **non-empty** if $V \neq \emptyset$. The notion of edges relates to the connections and dynamics between the parts, and a graph is **nontrivial** whenever $E \neq \emptyset$. Also a graph is **complete** if $E = V \times V$, undirected if $(u, v) \Longleftrightarrow (v, u)$ and **direct** if $(u, v) \neq (v, u), \forall u, v \in V$.

7. Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey.*
8. Cousty Jean, Najman Laurent, and Perret Benjamin (2013). *Constructive links between some morphological hierarchies on edge-weighted graphs.*
9. Najman Laurent and Cousty Jean (2014). *A graph-based mathematical morphology reader.*

Definition 3: Adjacency relation

The set E induces a unique **adjacency relation** Γ on V, which associates $u \in V$ with $\Gamma(u) = \{v \in V | (u, v) \in E\}$. Γ is reflexive ($u \in \Gamma(u)$) and symmetric ($v \in \Gamma(u) \iff u \in \Gamma(v)$).

The adjacency relation is the graph's architecture, guiding the edges' disposition. In multimedia processing, the adjacency relation is usually in a regularly structured form as a grid. The regular grid is invariant, meaning that translating the media elements create the same graph as in the original position. 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. The 4− and 8−adjacency are spatially close to the coordinate systems of most media and could be intuitively extended to higher dimensions. Contrary to the 6−adjacency that is hard to expand into higher dimensions [10], but is prominent in morphological processing for its isotropic properties. 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.

Multiple functions could be associated with each vertex and edge, enhancing the relational aspects, the data interpretation, and inserting metric properties.

Definition 4: Edge-weighted graph

An **edge-weighted graph** is denoted by $G(V, \mathcal{F})$, in which $\mathcal{F} : V \times V \to \mathbb{R}$ is a function that weights the edges of $G(V, E)$. Also, $\mathcal{F}(E)$ represents 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. Typically, the weights computed by the weighting function ponder the navigation on the graph.

Definition 5: Path and descending path

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 $i \in \{1, \ldots, \ell - 1\}$, $\mathcal{F}(v_{i-1}, v_i) \geq \mathcal{F}(v_i, v_{i+1})$.

10. Najman Laurent and Cousty Jean (2014). *A graph-based mathematical morphology reader.*

The paths on a graph define the navigation between the elements, and the edges determine the possible routes. Multiple contexts could be attributed using paths. Most notable are the discrete distance between vertices based on the number of edges necessary to navigate from one to another, the weighted interpretation of distance between components, or the relations on shared paths. 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. For instance, the subsets could be created by a given path or a filtering criterion on the weights or components.

Definition 6: Subgraph

A graph $G' = (V', E')$ is **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 $\sqsubseteq$.

Regarding the computational declaration of a graph, it is usually represented either as an adjacency list or an adjacency matrix. An **adjacency list** uses an array of vertices containing lists of $\{u \in \Gamma(v), \forall (u, v) \in E\}$, including the associated weights in the case of an edge-weighted graph. This representation captures all graph's components but can be challenging to parse. The **adjacency matrix** is a $|V| \times |V|$ matrix and $[[u, v]] \leftarrow$ *value*, if $(u, v) \in E$, where *value* is either 1, representing the presence of an edge, or the weight in an edge-weighted graph. This representation is more suitable for many algorithms but can be challenging to process for large sets of vertices.

The concepts and notions presented here are for a better comprehension of the theory in this work. Graph structures are versatile and adaptable to the desired context; hence most definitions are preceded by terms such as "usually", "commonly", or "often". The following sections will expand these definitions, adapting them accordingly to the context of media and hierarchical analysis.

1.2 Hierarchies from graphs to graphs

The hierarchical operators are delineated in the mathematical morphology domain. This section introduces the essential concepts for the hierarchical operators in mathematical morphology but focuses on the definitions for graphs. It refers the reader to the com-

prehensive work in Najman and Talbot 2013b [11] and the excellent general formalization in Serra (2006) [12] and Ronse (2014) [13] for more details on mathematical morphology.

Classical mathematical morphology is based on algebraic operations on lattices (a partially ordered set), defining operators and filters that produce information on the reaction they cause. The main operators are dilatation and erosion, which respectively retrieve the least upper bound (supremum) and the greatest lower bound (infimum) in any family of elements. These operators are related by adjunction and deliver new operators and morphological filters when applied successively. There are many properties of these operators, but crucially they are increasing and idempotent [14], delivering reliable results.

Serra (2006) formalizes hierarchies as algebraic operators in the complete lattice that creates faces of a tessellation, characterized as segmented regions. The image space is undoubtedly the principal definition space for the hierarchical theory, and most applications are for image processing, notably image segmentation [15] [16] and remote sensing [17]. Nevertheless, similarly structured visual media, such as hyper-spectral [18] and multi-modal images [19], videos [20] [21] [22], or even structured time measurements like sensor data time series [23] [24], are also processed with hierarchical algorithms if given a proper characterization of interrelations.

Hierarchies defined on graphs facilitate this characterization. In classical morphol-

11. Najman Laurent and Talbot Hugues (2013b). *Mathematical morphology: from theory to applications*. Complete work.
12. Serra Jean (2006). *A lattice approach to image segmentation*.
13. Ronse Christian (2014). *Ordering partial partitions for image segmentation and filtering: merging, creating and inflating blocks*.
14. Najman Laurent and Cousty Jean (2014). *A graph-based mathematical morphology reader*.
15. Soille Pierre and Najman Laurent (2012). *On morphological hierarchical representations for image processing and spatial data clustering*.
16. Randrianasoa Jimmy Francky et al. (2018). *Binary partition tree construction from multiple features for image segmentation*.
17. Maia Deise Santana et al. (2021). *Classification of remote sensing data with morphological attribute profiles: a decade of advances*.
18. Tochon Guillaume et al. (2018). *Advances in utilization of hierarchical representations in remote sensing data analysis*.
19. Kiran Bangalore Ravi and Serra Jean (2015). *Braids of partitions*.
20. De Souza Kleber Jacques et al. (2013). *Hierarchical video segmentation using an observation scale*.
21. Xu Chenliang, Xiong Caiming, and Corso Jason J. (2012). *Streaming hierarchical video segmentation*.
22. Wang Dezhao et al. (2021). *Combining progressive rethinking and collaborative learning: a deep framework for in-loop filtering*.
23. Alonso-Gonzalez Alberto, Lopez-Martinez Carlos, and Salembier Philippe (2014). *Polsar time series processing with binary partition trees*.
24. Nguyen Tin T. et al. (2019). *Feature extraction and clustering analysis of highway congestion*.

ogy, structuring elements are the parameters for the operators. On graphs, the modeling choices for the edges, weights, and adjacency relation define the parameters. While the graphs represent the interrelations, the hierarchical structure provides a non-regular characterization of regions with notions of order and navigation.

This work focus on the hierarchies of partitions modeling the space on edge-weighted graphs. The remainder of this chapter follows the notations in Cousty, Najman, and Benjamin Perret (2013) [25] and the comprehensive work in Najman and Talbot 2013b [26]. For insights connecting morphological hierarchies and the hierarchies on edge-weighted graphs, the work in Najman and Cousty (2014) [27]. And for generalizations of hierarchies of partitions and hierarchical types, the work in Cousty, Najman, Kenmochi, et al. (2018) [28].

A hierarchy operating on the edge-weighted graph defines non-gridded regions as subsets of the set of vertices. For the graph $G\,(V, E)$ and subgraph $G' = (V', E')$, the **graph induced by V'** is a graph $G = (V', \epsilon)$ where $V' \subseteq V$ and $\epsilon = \{(u, v) \in E \,|\, u, v \in V'\}$. V' is connected for G if the graph induced by V' is connected. V' is a **connected component** of G if V' is connected for G and maximal. Maximal means that for any other subset of V, if there is a connected superset of V', it represents the same V' set.

Definition 7: Hierarchy on graph vertices

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 the **dendogram** [29] representation of the hierarchy.

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

25. Cousty Jean, Najman Laurent, and Perret Benjamin (2013). *Constructive links between some morphological hierarchies on edge-weighted graphs.*
26. Najman Laurent and Talbot Hugues (2013b). *Mathematical morphology: from theory to applications.* Complete work.
27. Najman Laurent and Cousty Jean (2014). *A graph-based mathematical morphology reader.*
28. Cousty Jean, Najman Laurent, Kenmochi Yukiko, et al. (2018). *Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps.*
29. Sokal Robert R. and Rohlf James (1962). *The comparison of dendrograms by objective methods.*

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 leaves are the *parent*s 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 until reaching the node, and it is inversely proportional to the depth of the node.

Hierarchies are 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 called a minimum of $\mathcal{H}$.

Definition 8: Partition

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 $[\mathbb{P}]_v$.

Partitions characterize regions that go from the singleton element on the vertices to the entire set of vertices representing a single region in a *complete partition*. It regulates that there is no intersection between regions (no element could be present in two different regions simultaneously) and that the union of all regions composes the data. In terms of disjoint sets, the definition leads to a complete lattice [30] where operators are a dilation and an erosion if they preserve the union and intersection, respectively.

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 $\mathbb{P}', \mathbb{P}'' \in \mathbb{P}$. This 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 [31].

Definition 9: Hierarchy of partitions

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.

30. Serra Jean (2006). *A lattice approach to image segmentation.*
31. Ronse Christian (2014). *Ordering partial partitions for image segmentation and filtering: merging, creating and inflating blocks.*

As a sequence of partitions, 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 on the partitions induces a tree structure.

Definition 10: Hierarchical partition tree

In this context, the **hierarchical partition tree** $\mathcal{T}_\mathcal{H}$ is the tree representing the hierarchy $\mathcal{H} = (\mathbb{P}_0, \ldots, \mathbb{P}_k)$ created from the edge-weighted graph 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** [32] if all regions are extracted at the same hierarchical level or a **non-horizontal cut** [33] if searching for regions at different levels for one representation.

1.3 Types of hierarchies

Thus far, the discussions about hierarchies and partitions modeling the space on edge-weighted graphs considered only the structural components of the graphs: the vertice and edge sets. This section introduces the weights, which regulate how regions are formed, the criterion to merge and create new ones, and the order to pursue.

The hierarchical construction algorithms use the weights and regions on the partitions to characterize the type of hierarchy created. Among the many types [34], this thesis contemplates three particular hierarchical types grouped by their ordering method on the hierarchical tree. Namely:

32. Perret Benjamin, Cousty Jean, Guimaraes Silvio Jamil F., et al. (2018). *Evaluation of hierarchical watersheds.*
33. Guigues Laurent, Cocquerez Jean Pierre, and Men Hervé Le (2006). *Scale-sets image analysis.*
34. Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey.*

1. Altitudes ordering based on a minimum distance criterion: binary partition hierarchies [35];
2. Altitudes ordering based on increasing values of edge-weights criterion: quasi-flat zones [36] and strongly constrained connectivity hierarchies [37]; and
3. Altitudes ordering based on a geometric criterion: hierarchical watersheds [38] [39].

All these constructions algorithms could be defined on the edge-weighted graphs discerned as minimum spanning trees (MST) or minimum spanning forests (MSF) [40] [41].

Definition 11: Minimum spanning tree and Minimum spanning forest

The edge-weighted subgraph $(G', \mathcal{F})$ is a **minimum spanning tree** of $(G, \mathcal{F})$ if G' is connected, $V' = V$, and the sum of the weights defined by $\mathcal{F}$ in G' is less or equal than any other subgraph of $(G, \mathcal{F})$ whose vertex is V. Furthermore, a **minimum spanning forest** is the minimum spanning tree of all connected components in G.

The **binary partition hierarchies** [42] (BPH) is the hierarchy of partitions of V that solves the MST for the edge-weighted graph $G(V, \mathcal{F})$ with the Kruskal's algorithm [43] for the binary partition tree (BPT). The construction algorithm is recursive, starting from the partition of singletons $[\mathbb{P}]_v$ where each unitary region is a tree, the set of edges E, and the set of weights $\{w = \mathcal{F}(u, v) \,|\, (u, v) \in E\}$. At each iteration, the algorithm selects the edge of minimum weight if it connects two distinct trees representing a region to form a new one. The procedure repeats until the partition of a single region $\mathbb{P}_k$ is created, for $k = |\, V \,| - 1$. The internal nodes represent the MST of $G(V, \mathcal{F})$.

The **quasi-flat zones** (QFZ) hierarchies comprise the hierarchies induced directly

35. Salembier Philippe and Garrido Luis (2000a). *Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval.*
36. Cousty Jean, Najman Laurent, Kenmochi Yukiko, et al. (2018). *Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps.*
37. Soille Pierre (2008). *Constrained connectivity for hierarchical image partitioning and simplification.*
38. Meyer Fernand (1996). *The dynamics of minima and contours.*
39. Beucher Serge (1994). *Watershed, hierarchical segmentation and waterfall algorithm.*
40. Cousty Jean and Najman Laurent (2011). *Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts.*
41. Najman Laurent, Cousty Jean, and Perret Benjamin (2013). *Playing with Kruskal: algorithms for morphological trees in edge-weighted graphs.*
42. Salembier Philippe and Garrido Luis (2000a). *Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval.*
43. Kruskal Joseph B. (1956). *On the shortest spanning subtree of a graph and the traveling salesman problem.*

from the edge-weight graph [44]. Its construction algorithm takes the set of ordered weights on the edges and defines each level of the hierarchy as the set of connected component partitions whose weights are smaller than a threshold value λ. The threshold values cover the range of weights in the graph.

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, 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. A hierarchy of quasi-flat-zones induced in the edge-weighted graph is defined as:

$$\mathrm{QFZ}(G', w) = (\mathbb{C}(w_\lambda^V(G')) | \lambda \in \mathbb{E})$$

for the elements:

$w_\lambda(G')$ is the λ-*level set* of all edges of G' whose weight values are less than λ;

$w_\lambda^V(G')$ is the λ-*level graph* whose edges are $w_\lambda(G')$ and vertices V;

$\mathbb{C}(w_\lambda^V(G'))$ is the λ-*level partition* of connected components partition induced by the λ-*level graph* of G';

The **strongly constrained connectivity** hierarchy [45], also known as $\alpha-$trees, is a case of the QFZ, where the induced partitions are maximal for the connected components at a specific thresholding value. Formally, for the set of vertices V of the edge-weighted graph $G(V, \mathcal{F})$, we say V is $\alpha-$strongly connected for $\alpha \in \mathbb{R}^+$ if there is a path $\pi = (v_0, \ldots, v_\ell)$ connecting v_0 to v_ℓ where all the edge weights are smaller than α. The strongly constrained hierarchy is composed of all the $\alpha-$strongly connected components of G defined as the maximal $\alpha'-$connected sets of V where $\alpha' \in \mathbb{R}^+$ and $\alpha' \leq \alpha$.

The **hierarchical watershed** [46] [47] extends the classical morphological watershed [48], 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

44. Cousty Jean, Najman Laurent, Kenmochi Yukiko, et al. (2018). *Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps.*
45. Soille Pierre (2008). *Constrained connectivity for hierarchical image partitioning and simplification.*
46. Meyer Fernand (1996). *The dynamics of minima and contours.*
47. Beucher Serge (1994). *Watershed, hierarchical segmentation and waterfall algorithm.*
48. Beucher Serge (1979). *Use of watersheds in contour detection.*

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 MSF representing the flooded regions in all possible levels. In the context of edge-weighted graphs, the principle of the drop of water 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 [49].

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 [50]. The notion behind an extinction value is the minimum value that makes a region disappear (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. Common criteria are: (i) area, ranking regions by their size; (ii) dynamics, ranking regions by their depth; and (iii) volume, ranking regions by balancing the size and the depth.

The ensuing formal definition of the hierarchical watershed in the edge-weighted graph follows Cousty, Giles Bertrand, et al. (2009) and Cousty and Najman (2011).

First, consider G' and G'' as two subgraphs of G. G' is an extension of (or rooted on) G'' in G if: (i) $G'' \subseteq G'$, and (ii) G' contains exactly one component of G''.

Now, consider $\mathcal{M}$ a graph of all subsets minima of the edge-weighted graph $(G, \mathcal{F})$. In $\mathcal{M}$, each minimum is a subgraph G' of G that: (i) is connected, (ii) the weight map $\mathcal{F}(E')$ of the edges E' in G' has a unique value, and (iii) any adjacent edge to G' is strictly greater than the value in $\mathcal{F}(E')$.

For E' as a subset of the edges of G, E' is a **watershed cut** (or simply watershed) of the weight map $\mathcal{F}(E)$ if: (i) the complementary set $\overline{E'}$ of E' is an extension of $\mathcal{M}$, and (ii) if for any $(u, v) \in E'$ there exists two descending paths $\pi_1 = (v_0, \ldots, v_\ell)$ and $\pi_2 = (u_0, \ldots, u_{\ell'})$ such that v_ℓ and $u_{\ell'}$ are vertices of two distinct minima in $\mathcal{M}$, and $\mathcal{F}(u, v) \geq \mathcal{F}(v_0, v_1)$ whenever π_1 is non-trivial, and respectively, $\mathcal{F}(u, v) \geq \mathcal{F}(u_0, u_1)$ for a non-trivial π_2.

49. Cousty Jean, Bertrand Giles, et al. (2009). *Watershed cuts: minimum spanning forests and the drop of water principle.*
50. Vachier Corinne and Meyer Fernand (1995). *Extinction value: a new measurement of persistence.*

The watershed from markers computed from subsets of minima of the edge-weighted graph have a dual in terms of regions that are formalized as graph forests. Constructing such forest from an ordered set of minima is equivalent to a MSF relative to subgraphs. Formally, for G' and G'' as two subgraphs of G, G' is a relative (or rooted) MSF of G in G'' if: (i) G' is rooted in G''; (ii) $V' = V$; (iii) and the sum of weights of G' is less than any other possible subgraphs that satisfies (i) and (ii).

Taking $\mathbb{M}_{\mathcal{F}}$ as the set of weight minima, $\mathcal{S} = (M_1, \ldots, M_\ell)$ as a sequence of pairwise minima distinct on the weight values and $\mathcal{G} = (G_1, \ldots, G_\ell)$ as a sequence of subgraphs of G, $\mathcal{G}$ is a **MSF hierarchy** of $\mathcal{S}$ if for any $i \in [0, \ell]$, $G_{i-1} \sqsubseteq G_i$ and the subgraph G_i is a MSF rooted in $\sqcup[\mathbb{M}_{\mathcal{F}} \setminus \{M_j \,|\, [1, i]\}]$, for $\sqcup$ denoting the supremum of a family of values. The MSF hierarchy induces a hierarchy of partitions on V that is optimal.

All of these hierarchical methods have linear or quasi-linear implementations [51] [52], and as shown in Najman, Cousty, and Benjamin Perret (2013) and Cousty, Najman, Kenmochi, et al. (2018), one type of hierarchy can be inferred from another. For instance, removing any consecutive nodes with equal values in a BPT induces a QFZ, and filtering the QFZ by the maxima creates a strongly constrained hierarchy. Efficient implementations of the hierarchical watershed involve computing a weight map whose QFZ reflects the desired watershed if each connected component contains exactly one marker and the markers are ranked by the weight values.

Each construction algorithm has its particular properties and interpretation of the data. However, the set of 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 not adequate for an application. The mathematical formalism in the morphological domain may be daunting at the outset. Still, most methods provide intuitive notions, and the construction methods require hardly any parameters other than the ones provided by the already modeled edge-weighted graph. Combining these aspects with efficient implementations makes the hierarchies an appealing alternative to introduce a semantic interpretation into media processing. In this context, semantics relates to partitioning the perceptual space into regions with some meaningful relation embedded in a single structure.

51. Najman Laurent and Couprie Michel (2006). *Building the component tree in quasi-linear time.*
52. Najman Laurent, Cousty Jean, and Perret Benjamin (2013). *Playing with Kruskal: algorithms for morphological trees in edge-weighted graphs.*

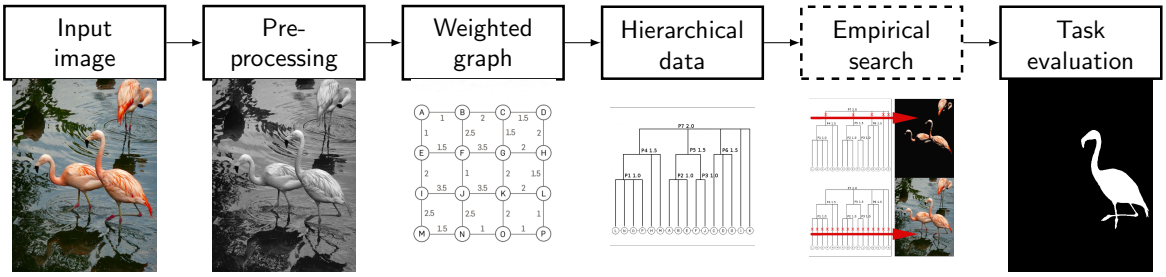


Figure 1.1: Figure illustrating a typical pipeline using hierarchies for image processing. First, it transforms each image to the gray-scale magnitudes used to create the edge-weighted graphs. Then, the hierarchical method computes the desired hierarchy based on its criterion. Because the hierarchical structure is multi-layered, selecting a certain level, a combination, or a specific number of regions is necessary to filter the structure and create a single output evaluated on the task.

1.4 Typical hierarchical pipeline

This section introduces a typical pipeline, illustrated in Fig. 1.1, that applies the hierarchies defined for an edge-weighted graph to an image processing task and, at each step, provides considerations relevant to this work's development.

Usually, image applications are tasks defined for three-channel colored images. Despite the availability of existing hierarchical methods applied directly on the color channels [53] [54] [55], operating on colored images requires strategies to either map dissimilarities between pixels on multiple dimensions [56] or combine the hierarchies independently defined on each channel [57]. Therefore, the general approach is to model the graph from the monocolored images, such as the grayscale representation of pixel intensities. The problem with simply working on the grayscale space is that it usually results in significant variation across regions with distinct absolute values, making it harder to map which value represents a region change [58].

An alternative is to use image gradients, which are transformation processes that

53. Salembier Philippe and Garrido Luis (2000a). *Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval.*
54. Soille Pierre (2008). *Constrained connectivity for hierarchical image partitioning and simplification.*
55. Merciol Franlois and Lefevre Sébastien (2012). *Fast image and video segmentation based on alpha-tree multiscale representation.*
56. Aptoula Erhan, Weber Jonathan, and Lefèvre Sébastien (2013). *Vectorial quasi-flat zones for color image simplification.*
57. Kurtz Camille, Naegel Benoit, and Passat Nicolas (2014). *Connected filtering based on multivalued component-trees.*
58. Meyer Fernand and Beucher Serge (1990). *Morphological segmentation.*

aim to enhance desirable properties of an image, particularly the visual perception of contours. Traditional gradient methods, such as Laplacian and Sobel, rely on kernel filters measuring local variation. In 2014, Dollar and Zitnick (2015) proposed the structured edge detection (SED), a fast method to create gradients. After the work in Benjamin Perret, Cousty, Silvio Jamil F. Guimaraes, et al. (2018), measuring the gain in using SED on the hierarchical pipeline, it is becoming a principal source for the edge-weighted graph [59] [60] on hierarchical processing.

After adequately preparing the image, the following steps on the pipeline are the graph and the hierarchical construction. Defining the graph representation is a modeling question with various connotations. Still, the graphs created from images present a unique transformed space close to the spatial domain in a grided form. As for the hierarchical construction, each type has its particular characteristics, and this work contemplates the hierarchies described in Section 1.3 to be used through the experimental investigations.

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 form for comparison. In this step resides the central problem this thesis tackles. The **trivial** approach is a series of horizontal cuts selecting multiple independent partitions representing the hierarchy [61] [62]. 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, as 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 Benjamin Perret, Cousty, Silvio Jamil F. Guimaraes, et al. (2018), the metric used to evaluate the selection can be misleading, and 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 [63],

59. Perret Benjamin, Cousty Jean, Guimarães Silvio Jamil Ferzoli, et al. (2019). *Removing non-significant regions in hierarchical clustering and segmentation.*
60. Otiniano-Rodríguez Karla et al. (2019). *Hierarchy-based salient regions: a region detector based on hierarchies of partitions.*
61. Perret Benjamin, Cousty Jean, Guimaraes Silvio Jamil F., et al. (2018). *Evaluation of hierarchical watersheds.*
62. Pont-Tuset Jordi and Marques Ferran (2012). *Supervised assessment of segmentation hierarchies.*
63. Xu Chenliang, Whitt Spencer, and Corso Jason J. (2013). *Flattening supervoxel hierarchies by the uniform entropy slice.*

realigning [64] [65] or filtering [66] [67] [68] [69] [70] the structure. These strategies rely on identifying less relevant regions and re-ponder or merge these regions, creating more concise representations. The problem with these approaches is that defining the importance is subjective and strongly related to a media or task.

Alternatively, one could search for the ideal representation with a non-horizontal cut on the hierarchy [71] [72] [73] [74], 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 [75] [76] [77]. However, inserting the hierarchies in a learning framework is difficult since they have heterogeneous representations, for instance, in their altitudes, number of regions, and component connections. Furthermore, the construction algorithms are primarily deterministic, and there is no direct way to evaluate their quality other than applying it to a task.

1.5 Illustrating the problem

Through a series of illustrations, this section shows the challenges of working and applying the hierarchies with the trivial approach. It aims to illustrate the problem and present some points for consideration.

64. Adão Milena M., Guimarães Silvio Jamil F., and Jr Zenilton K. G. Patrocínio (2020). *Learning to realign hierarchy for image segmentation.*
65. Chen Yuhua et al. (2016). *Scale-aware alignment of hierarchical image segmentation.*
66. Perret Benjamin, Cousty Jean, Guimarães Silvio Jamil Ferzoli, et al. (2019). *Removing non-significant regions in hierarchical clustering and segmentation.*
67. Barcelos Isabela Borlido et al. (2019). *Exploring hierarchy simplification for non-significant region removal.*
68. Xu Yongchao, Geraud Thierry, and Najman Laurent (2016). *Connected filtering on tree-based shape-spaces.*
69. Paris Sylvain and Durand Fredo (2007). *A topological approach to hierarchical segmentation using mean shift.*
70. Salembier Philippe, Liesegang Sergi, and Lopez-Martinez Carlos (2019). *Ship detection in SAR images based on maxtree representation and graph signal processing.*
71. Arbelaez Pablo, Pont-Tuset Jordi, et al. (2014). *Multiscale combinatorial grouping.*
72. Cousty Jean and Najman Laurent (2014). *Morphological floodings and optimal cuts in hierarchies.*
73. Guigues Laurent, Cocquerez Jean Pierre, and Men Hervé Le (2006). *Scale-sets image analysis.*
74. Bejar Hans H. C., Guimaraes Silvio Jamil Ferzoli, and Miranda Paulo A. V. (2020). *Efficient hierarchical graph partitioning for image segmentation by optimum oriented cuts.*
75. Chierchia Giovanni and Perret Benjamin (2020). *Ultrametric fitting by gradient descent.*
76. Kiran B. Ravi and Serra Jean (2014). *Global–local optimizations by hierarchical cuts and climbing energies.*
77. Kiran Bangalore Ravi and Serra Jean (2015). *Braids of partitions.*

For the hierarchical construction, it contemplates the discussed hierarchies and refers to them as:

— QFZ: quasi-flat zones;

— ALPHA: strongly constrained connectivity hierarchies; and

— WATER-*: hierarchical watersheds.

The complete nomenclature for the hierarchical watershed depends on the geometric criterion: (i) WATER-AREA; (ii) WATER-DYN; (iii) WATER-VOL; and (iv) WATER-PAR. The first three are the ones previously discussed in Section 1.3. The last one, WATER-PAR, refers to the topological criterion proposed in Benjamin Perret, Cousty, Silvio Jamil F. Guimaraes, et al. (2018) that counts the number of parents a node has on the MST representing the graph to determine its extinction values and use them as the criterion for hierarchical construction. The BPH is intuitive and extensively used in many contexts. Still, they are rarely applied directly into the vertex set representing each pixel of an image due to the considerable size the structure could assume. Therefore, here and in all experimental sections, it will be excluded.

Consider the typical pipeline presented in Fig. 1.2, which highlights the step in consideration and the two main cut strategies. The application illustrated is the segmentation task where either a threshold value selects an altitude level to create a partition or a certain number of desired regions determine the appropriate level to probe. In the hypothetical simplified case illustrated, both strategies return the same section to demonstrate that could be a correspondence between the two types of horizontal cuts.

While with the threshold approach, it is possible to search in a limited space, as in the case of normalized altitudes, one could infinitely divide this space, and the number of returned regions is uncertain. In contrast, performing the cut with a specific number of regions will give the closest approximation available in the structure to the parameter value. Depending on the task, selecting the number of regions could be more appropriate. However, the search space is limited only by the total number of regions in the hierarchy, and there is no way of knowing the number of regions that will return the complete object(s) of interest.

The goal of the cuts is to find the best partition for the task. And defining the best is subjective and often an accommodation in the degree of detail portrayed. As illustrated in Fig. 1.3, small regions could be either clutter or an essential part of the representation regarding the ground-truth. Furthermore, besides the subjectivity of this decision, defining one satisfactory cut for one instance in the dataset may not be adequate for another, as

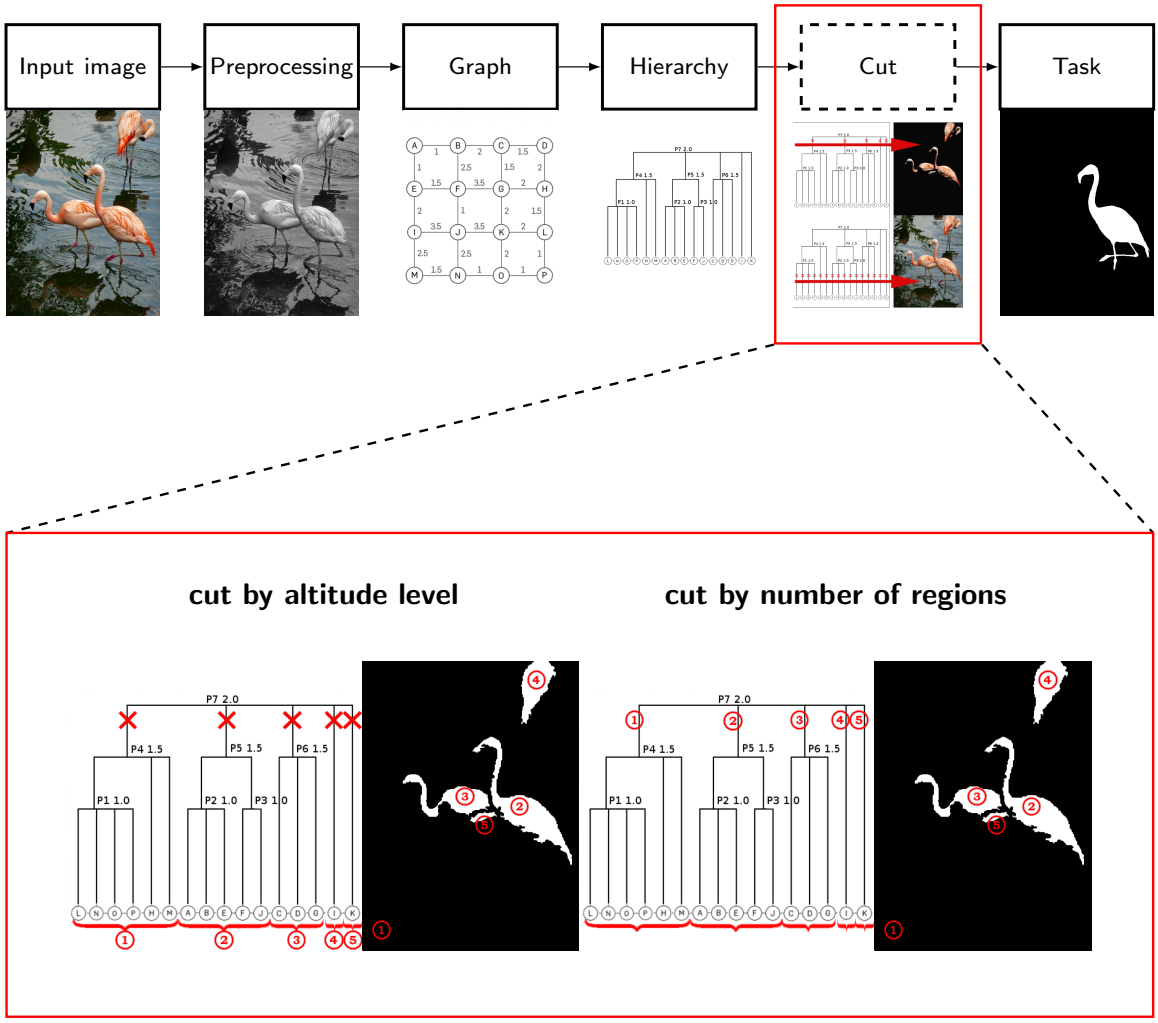


Figure 1.2: Figure illustrating with simplified images a typical pipeline using hierarchies for the segmentation task. It highlights the two most common strategies to create a more suitable representation for the task evaluation: horizontal cuts by thresholding the altitude levels or selecting the desired number o regions. In this hypothetical example, both cuts give the same partition. The parameter for the number of regions results in the closest approximation to the desired amount allowed by the structure. For instance, if the parameter were four instead of five, the selection must decide if it is *at most* 4, which would produce a single region, or *at least* four, which would give the five areas.

also illustrated in Fig. 1.3, even if using the same modeling strategy and hierarchical structure.

Another consideration regards the type of hierarchy defined by the construction algorithm. As illustrated in Fig. 1.4, different hierarchical construction strategy creates different representations of the same data. Among the types considered in this thesis,

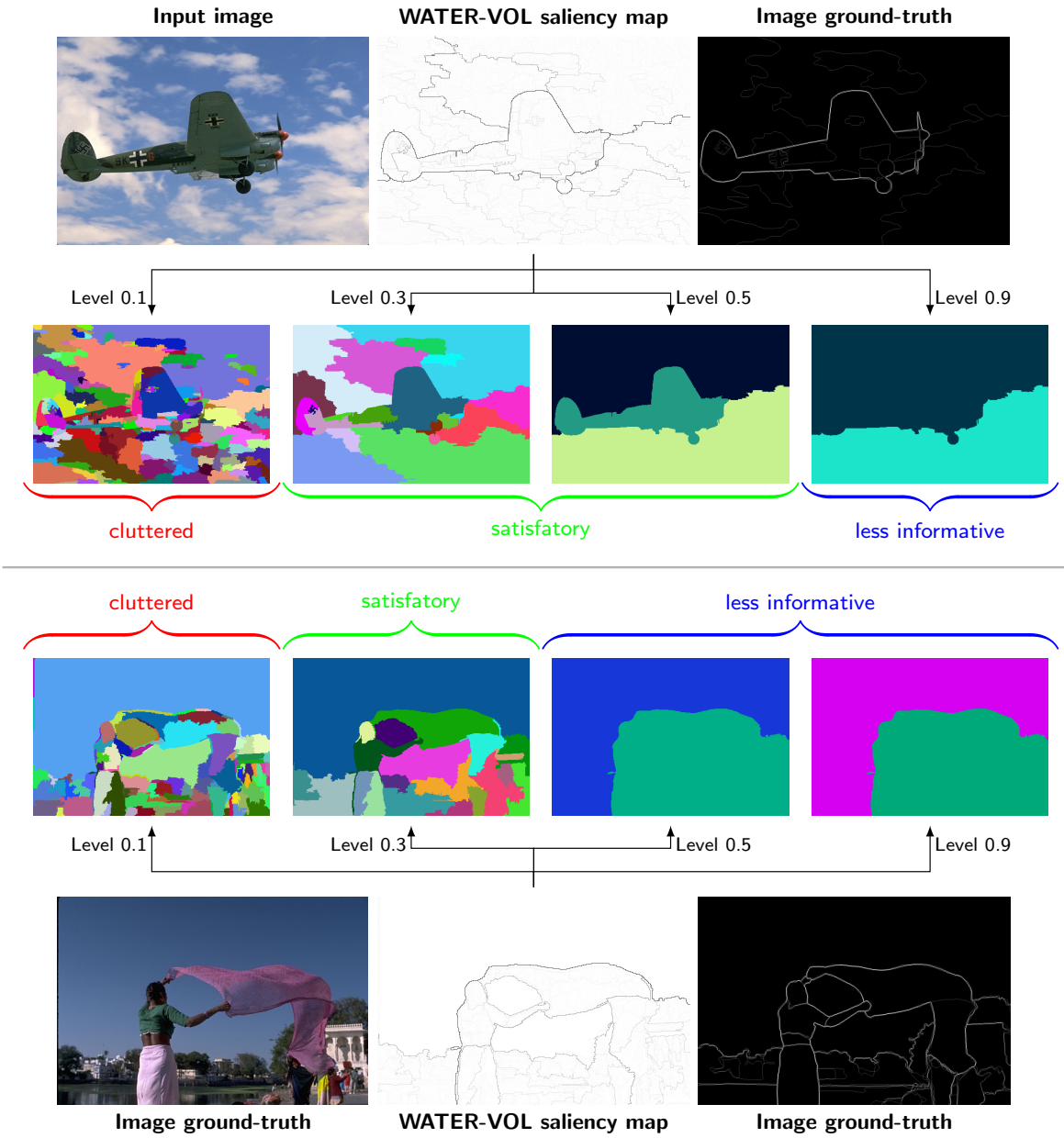


Figure 1.3: Figure illustrating the partitions created by thresholding the hierarchical altitudes. Selecting a low altitude level usually results in a clustered section. Conversely, choosing high levels results in a less detailed output, which may be uninformative depending on the task. Intermediary levels usually present a compromise between targeting the relevant objects and representing small components. For instance, level 0.3 on the plane image lose some details, such as the paddles and the logo at the wing. However, it keeps both wheels and the tail logo, but the main object remains over-segmented relative to the ground-truth. The main object is more concise at level 0.5 on the plane image, but most details are merged. Not all observations hold for the woman with a tissue image, where level 0.1 is closer to the ground-truth than level 0.5. Complete hierarchies are illustrated as saliency maps with magnitudes inverted for clearer visualization and balanced weights for better distribution.

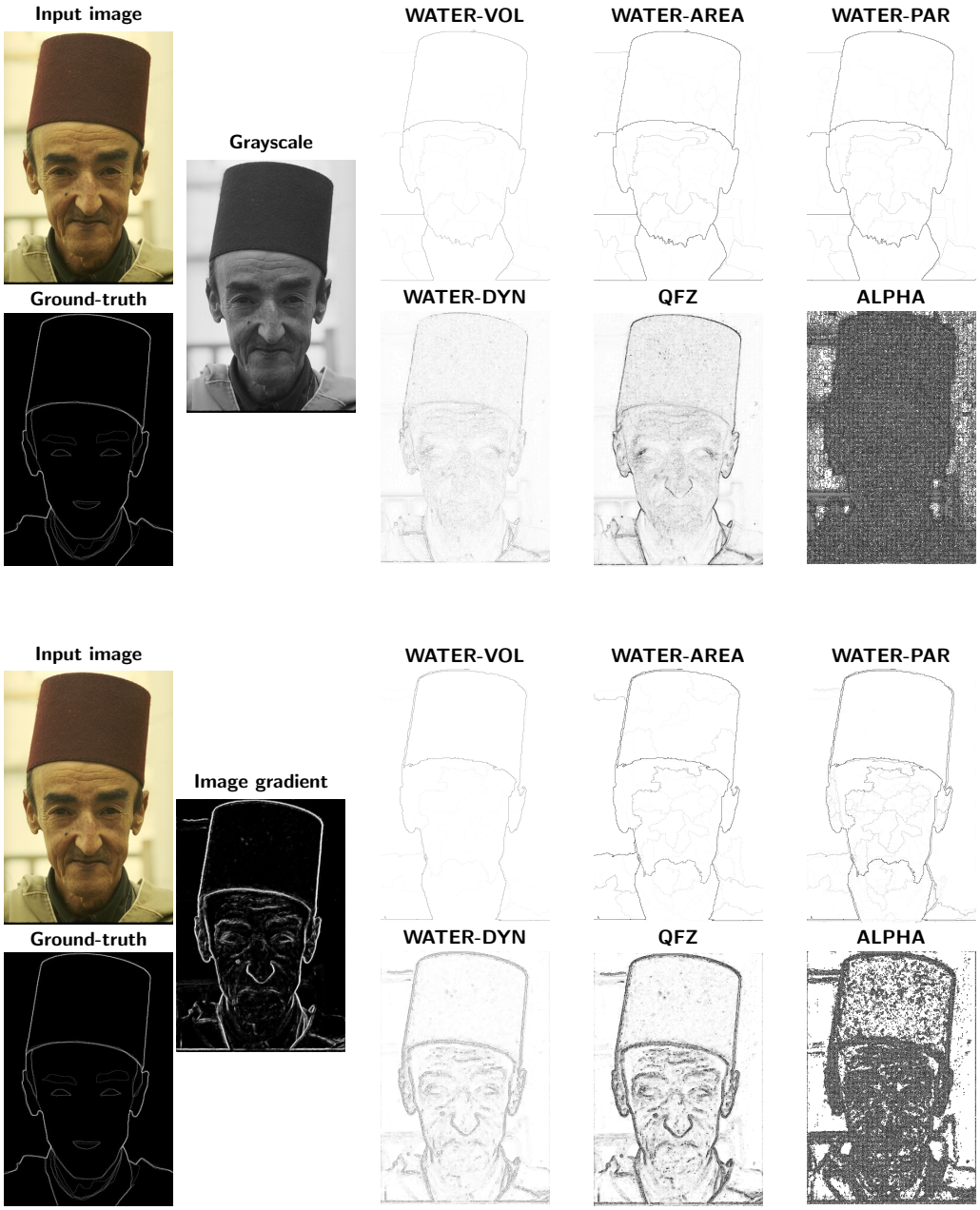


Figure 1.4: Figure illustrating the different outputs created by the hierarchical construction algorithms in the study. The hierarchical watersheds produce similar maps region-oriented, with very subtle differences in the details—except for the WATER-DYN, which ranks the regions by their depth, creating more contour-oriented maps. Similarly, the QFZ highlights more of the contours due to the organization of increasing values. The ALPHA hierarchies, more often than not, create a limited number of altitudes due to their strong constraints. The figure also demonstrates the impact the gradient input wields on the construction algorithm. From the stronger delimitation on the gradient, the saliency maps show more details in the region-oriented hierarchies, higher values for WATER-DYN and QFZ, and more distributed areas that meet the ALPHA constraints. Saliency maps illustrated with magnitudes inverted for more clear visualization.

the hierarchical watershed mechanism maps magnitudes into partitions and represents the topological surface of regions between the higher magnitudes at different levels. Consequently, the hierarchies of watersheds reflect these areas in their contour maps with very subtle visual differences among the distinct geometric criterion. For instance, the area criterion indicates the region's size distribution; the volume ponders both the size and the depth distribution, which creates more diminished contours for lower magnitudes between the regions. The criterion by the number of parents has the strongest values on lower topographical regions because in pondering the parents, it measures the number of climbings necessary to arrive at a new regional minimum. Therefore, the hierarchical map remains region-oriented, with contours expressing the contrast in the input.

The most distinguished contour map for the watersheds is the one taking dynamics as the geometric criterion. The notion of dynamics of a minimum relates to the depth notion or relative altitudes, reflecting the height to climb before reaching any point with a lower altitude than a said minimum. This criterion is known as uniform flooding, because it grows uniformly with the altitudes. Consequently, the hierarchical maps are more contour-oriented than the other watersheds.

The quasi-flat zones hierarchy also presents contour-oriented maps, but all contrasting contours in the input are present but flattened by increasing values. Finally, the α-tree imposes stringent constraints, which results in an over-partitioned hierarchy with, more often than not, only two hierarchical levels or a reduced number. This type of hierarchy has a practical application where the input needs simplification without splitting internal areas. It relies on highly contrasted input for a better distribution of regions.

These considerations lead to the second illustration set in Fig. 1.4, where the same hierarchical types receive a graph created from the image gradient instead of the grayscale magnitudes. The two sets of images highlight the importance of the media's modeling before applying it to hierarchical construction algorithms. And how independently of the input, the hierarchies will interpret and organize the data according to its rules.

Fig. 1.5 illustrates the effect of thresholding the hierarchical levels on different watershed hierarchies. Typically, in a balanced hierarchy, lower levels are cluttered, and intermediary levels have the best compromise between details and conciseness, as shown for the watershed by volume and area. However, this is different from a rule as seen for the watershed by dynamics and the number of parents. While for the number of parents, the lower levels are also representative, for dynamics, it is the only one. Furthermore, the range between proper representation and uninformative can vary largely between types.

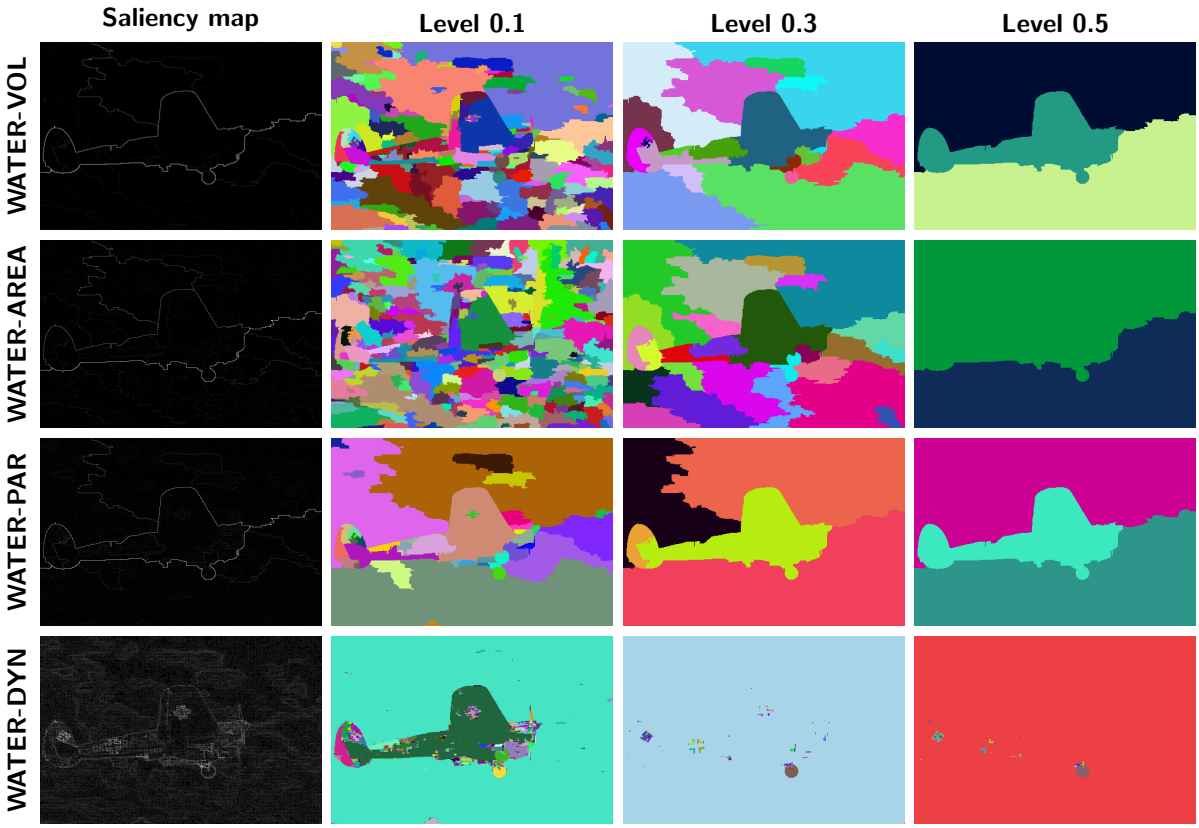


Figure 1.5: Figure comparing results produced by the same threshold value in different watershed hierarchies created for the graph modeling the grayscale image and one specific image. It aims to illustrate that no one cut solution is suitable for all, for instance. For some, lower levels (0.1) are usually cluttered representations (WATER-VOL and WATER-AREA). Still, it is suitable for others (WATER-PAR and WATER-DYN). Intermediary levels (0.3 and 0.5) are usually a good level for a cut, except for the contour-oriented representation (WATER-DYN). Also, a slightly fainted contour on the main object could cause it to be merged at much earlier stages (WATER-AREA at 0.5).

For instance, while for the number of parents, all levels illustrated could be considered a suitable partition, the area one merges the main object with the background much earlier in the tree.

The same considerations could be made when searching for the number of regions. For some representations, only a few regions would retrieve a concise representation (equivalent to good at intermediary levels); for others, a more significant number is required (equal to proper at lower levels). The quasi-flat zone represents most contours, and the $\alpha-$tree is overall over-segmented. In this type of visual analysis, they produce images perceived as noisy; therefore, they are not illustrated.

This final observation leads to the final consideration of this section: the evaluation

of the task. Assertions of good and bad representations and the best level or number of regions are all relative to the application. The analysis of the illustrations conforms with our perception of uniform regions and could be associated with the segmentation task. However, the detailed contours in some hierarchies could be more adapted to studies searching for high points in the data. For instance, the $\alpha-$tree in the context of these illustrations seems to produce only noise. Yet, for some applications, such as aerial analysis with high-resolution images, it presents a suitable arrangement between simplification and flexibility.

Furthermore, one should also be mindful of the metric chosen for the evaluation since some can be misleading. For instance, segmentation metrics that do not weigh true and false contributions tend to favor bad segmentations since the areas of interest are usually smaller than the background.

The goal of these considerations 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, sets of data without large quantities of labeled data or applications that require dependable outputs usually rely upon regional analysis methods that provide a consistent data organization, such as those provided by the hierarchical analysis.

1.6 Experimenting in the typical pipeline

This section shows some experiments with the typical pipeline and the trivial approach in two image tasks: edge detection and segmentation, that this work will use throughout.

The edge detection dataset is the Berkeley Segmentation Dataset and Benchmark (BSDS500) [78], which offers edge detection and segmentation labels. It contains 500 natural images (200 train, 100 validation, and 200 test) of the same size. Each image has multiple labels performed by different annotators; thus, this work performs a majority vote to create a single label. The challenging images in this dataset present: (i) complicated patterns, (ii) occluded objects, (iii) objects indistinguishable from the background by color, and (iv) high-contrast patterns.

78. Martin D. et al. (2001). *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.*

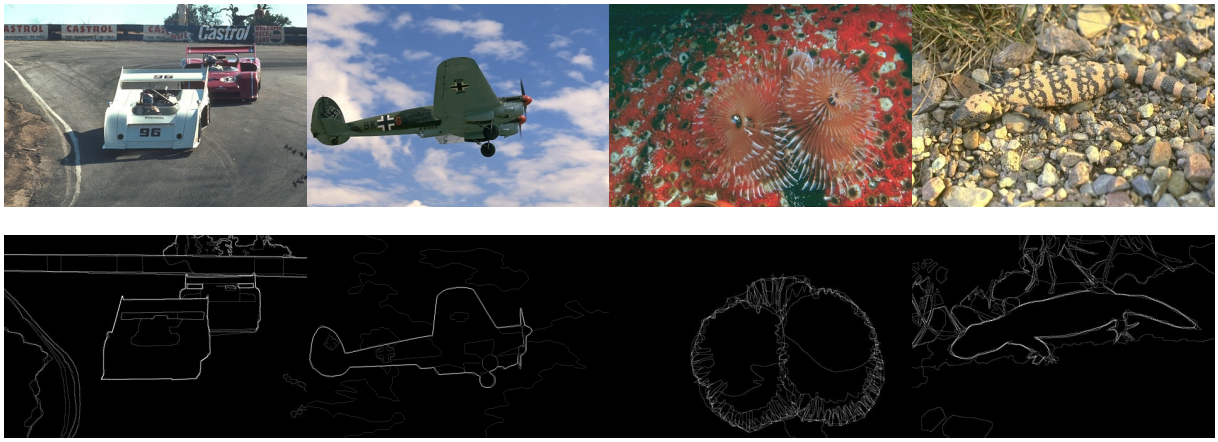


Figure 1.6: Sampled images from the BSDS500 dataset with their respective boundary ground truths. The dataset contains colored natural images and challenging images with complicated patterns, occluded objects, main objects indistinguishable from the background by color, and objects with patterns of high contrast. Also, each image contains multiple labels where line intensities indicate agreement among annotators.

For the segmentation task, it presents two binary segmentation public datasets: Birds [79] and Sky [80]. The Birds dataset contains 50 images of birds, and the Sky dataset includes 60 images of planes and the sky. Both datasets have manual annotations for the image segmentation task, and no official train/test sets division exists. Therefore, a random selection of the images split the datasets into 35/15 train/test for Birds and 40/20 for Sky. Fig. 1.7 illustrates the Birds dataset and the challenges it presents, namely, the images usually portraying 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. Fig. 1.8 illustrates the Sky dataset, which contains images annotated for the background instead of the main object. Usually, the region of interest covers large portions of the image but ignores the central part.

The pipeline takes the colored images in the datasets and computes the grayscale magnitudes without any additional preprocessing of the images. Next, it constructs the graph with a 4−adjacency (takes the four orthogonal values around a particular pixel) and the Euclidean distance on the magnitudes for the weighting function. The hierarchy construction explores the aforementioned hierarchies: QFZ, ALPHA, WATER-VOL,

79. Mansilla Lucy A. C. and Miranda Paulo A. V. (2016). *Oriented image foresting transform segmentation: connectivity constraints with adjustable width.*
80. Alexandre Eduardo Barreto (2017). *IFT-SLIC: geração de superpixels com base em agrupamento iterativo linear simples e transformada imagem-floresta.*

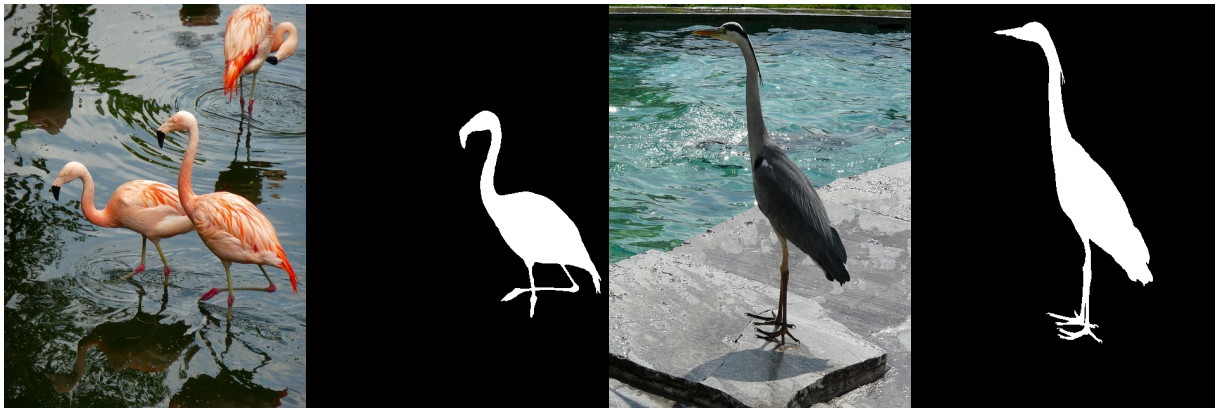


Figure 1.7: Sampled images from the Birds dataset with their respective segmentation ground truths. The dataset contains colored natural images, manually annotated. The images usually portray the birds close to a body of water, with areas of high-intensity lights and the annotations cover only one main object, despite the presence of multiple similar objects in the surroundings.

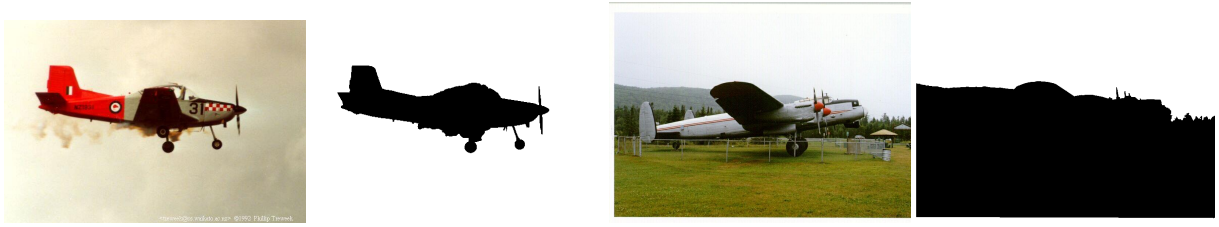


Figure 1.8: Sampled images from the Sky dataset with their respective segmentation ground truths. The dataset contains colored natural images, manually annotated. The ground truths in this dataset are for the image's background instead of the main object. Usually, the region of interest covers large portions of the image but ignores the central part.

WATER-AREA, WATER-DYN, and WATER-PAR. It does not perform additional post-processing, such as filtering the hierarchies, realigning, or balancing the levels.

The BSDS500 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 are then presented in terms of optimal dataset scale (obtained in the threshold that best represents most of the images), optimal image scale (obtained for each image at its best scale), and average precision through all scales. For non-hierarchical methods, this evaluation works to process soft edge maps, which is why methods producing fuzzy maps usually perform better, such as in Yun Liu et al. (2019) [81]. For hierarchical methods, this evaluation allows the assessment of different levels of details in the hierarchical partitions. For clarity, the

81. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*

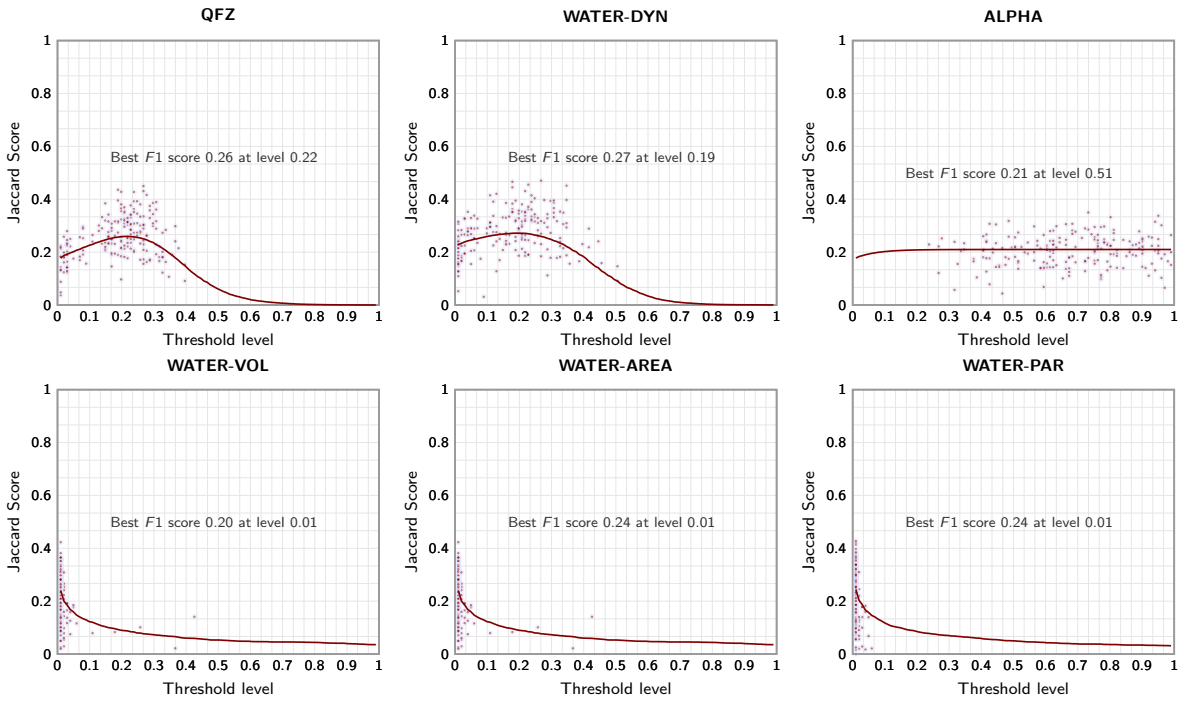


Figure 1.9: Results of the typical pipeline for the BSDS500 dataset. Shaded points represent the best score and threshold for individual images, and the solid line is the average score on the dataset scale. Perfect score=1.

results presented in this section for the BSDS500 dataset are only for the optimal dataset scale. It 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.

The segmentation datasets use the Jaccard similarity coefficient score as the metric, which measures in the interval $[0, 1]$ the intersection size divided by the union of two sets. Formally, given two sets, $\mathbb{A}$ and $\mathbb{B}$, the Jaccard score is computed as follows:

$$\text{Jaccard}(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|}$$

It is equivalent to the precision-recall $F1-$score on binary sets. Therefore, for evaluating the segmentation task, the horizontal cut strategy selects the partition and binarizes it for a direct assessment against the ground-truth.

1.6.1 Horizontal cut by threshold

Fig. 1.9 shows the results obtained in the BSDS500 dataset. Simply taking the contour maps does not yield good results, particularly for the region-oriented maps created by the watershed hierarchies by volume, area, and number of parents. Their application is usually through selecting the number of regions since, if not balanced, the significant areas are all in the lower end of the tree, and the evaluation method needs to take more slices in those regions. One could change the steps in the dataset evaluation threshold. However, there is an incentive to keep the evaluation parameters for comparison with other methods. The region-oriented watersheds are not the only ones with a bad performance since all types present unsatisfactory results.

The results presented here are 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, Maire, et al. (2009) [82] proposed a technique that constructs hierarchical boundary maps from an edge map where the boundaries between consistent regions are reinforced and small areas removed (scores 0.71 on the dataset scale). Maninis et al. (2018) [83] takes pre-computed contours using the side-outputs of a convolutional network for constructing the hierarchies (scores 0.73 on the dataset scale). Taylor (2013) [84] uses normalized cuts to reduce internal regions and sharpen the contours between contrasting areas (scores 0.67 on the dataset scale). Arbelaez, Pont-Tuset, et al. (2014) [85] 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 (scores 0.725 on the dataset scale). Furthermore, Benjamin Perret, Cousty, Silvio Jamil F. Guimaraes, et al. (2018) shows the gain quantitatively in score by filtering small areas on another dataset with the same task.

For the segmentation datasets, the horizontal cuts by threshold range are in $[0, 1]$ with a step of 0.001. The smaller steps count for the hierarchical watershed region-oriented methods that, without balancing the structure, produce most of its contours at the bot-

82. Arbelaez Pablo, Maire Michael, et al. (2009). *From contours to regions: An empirical evaluation.*
83. Maninis Kevis-Kokitsi et al. (2018). *Convolutional oriented boundaries: from image segmentation to high-level tasks.*
84. Taylor Camillo Jose (2013). *Towards fast and accurate segmentation.*
85. Arbelaez Pablo, Pont-Tuset Jordi, et al. (2014). *Multiscale combinatorial grouping.*

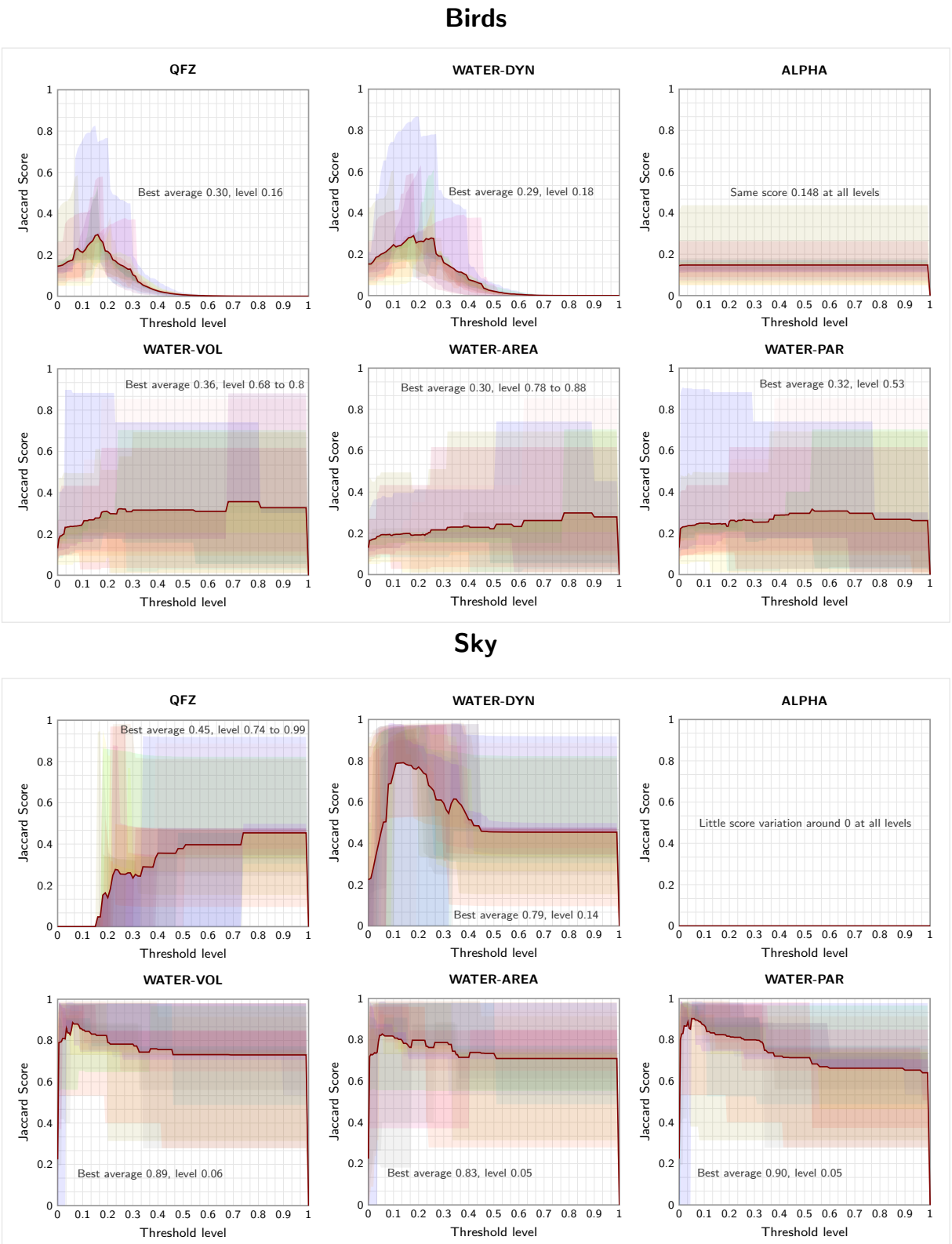


Figure 1.10: Results of the typical pipeline for the Birds and Sky datasets. Colored shaded areas represent individual image scores through the hierarchical levels, and the solid line is the average. Perfect score=1.

tom of the structure. Fig. 1.10 presents the results for the segmentation datasets. Both datasets give the average Jaccard score achieved by binarizing the images through different hierarchical levels. The first large region is taken as the background, and all the others are merged as the foreground. The challenge on the Sky dataset (large areas containing the object of interest where the background usually is) could be solved by inverting the region label, and all the watershed methods achieve, at some level, a good result (above 0.79 score). For QFZ, higher levels on the hierarchies perform better as multiple contour lines are merged to form a single region. However, it achieves a plateau close to the 0.7 threshold with an average score of 0.45. The only representation that does not present a satisfactory result is the ALPHA hierarchy, where the multiple small regions with a slight variation in the altitude levels created by the construction algorithm do not have a clear cut between the background and foreground for the binarization. The same occurs for ALPHA in the Birds dataset.

The other hierarchies also have an unsatisfactory performance in Birds. The image illumination conditions create peaks in the magnitude values that make it difficult to distinguish the main objects and the body of water in the background. Also, the algorithms will create similar partitions for the many objects portrayed in the images, while only one is considered a valid answer. As shown in the graphics in Fig. 1.10, for Birds, all the other methods perform well for some images, but the bad ones have close to zero scores, dragging the average.

1.6.2 Horizontal cut by the number of regions

For the cut by the number of regions, the experiments explore an extensive range of parameters defining the number of desired regions. Precisely: {2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}. The selection criterion follows the "at least" rule, meaning that the cut will take the first level available in the structure that contains at least the amount of regions passed as parameters.

In this approach, for the BSDS500 dataset, instead of passing all the contours in the hierarchy as saliency maps, each cut's contours are binarized and evaluated in a single scale. Fig. 1.11 illustrates the results obtained on the BSDS500. As shown, this strategy considerably improves the results on this dataset, particularly for the region-oriented maps. For them, the selection by the number of regions retrieves the most significant regions, independent of their position in the tree. The contour-oriented methods curves indicate that expanding beyond the search range may produce even better maps. However,

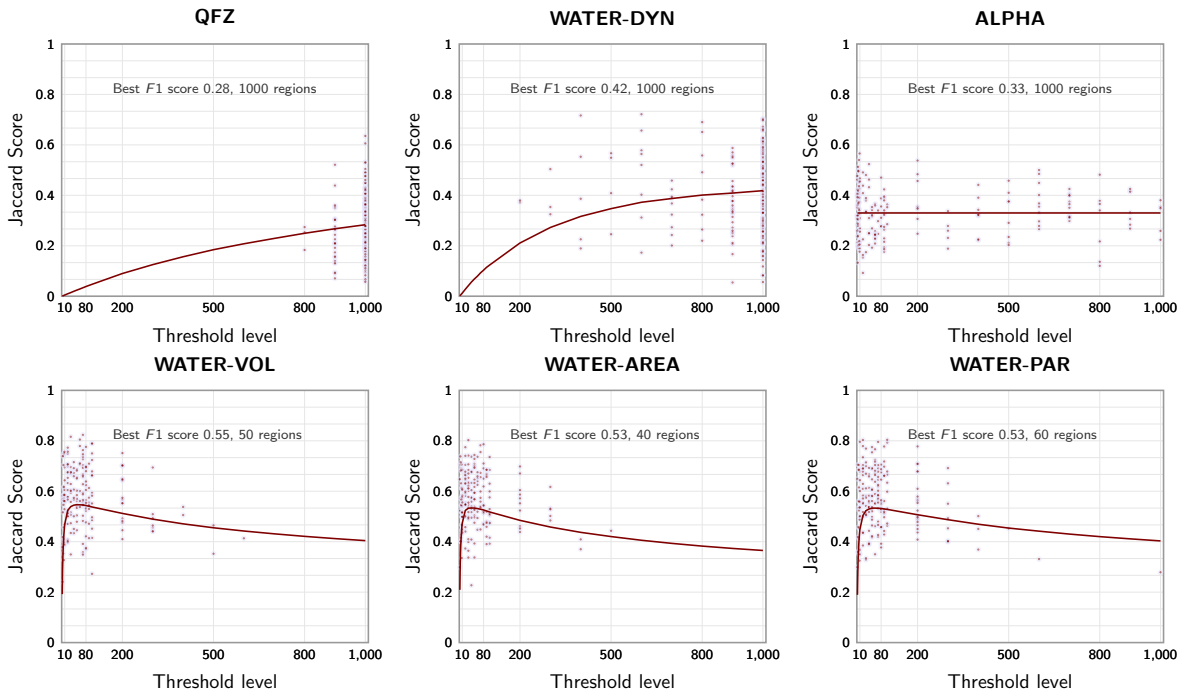


Figure 1.11: Results of the typical pipeline with a cut by the number of regions for the BSDS500 dataset. Shaded points represent the best score for individual images, and the solid line is the average score on the dataset scale. Perfect score=1.

the systematic search becomes arduous after a certain value. Furthermore, because of the "at least" policy, the ALPHA representation has almost the same edge map for all number of regions. A close inspection of this type in this particular dataset shows that the minimum amount of regions for most inputs is around 14000.

For the segmentation datasets, the binarization procedure is similar to the cut by a threshold but with a more explicit definition of background/foreground, where the regions returned by the cut parameter are taken as foreground. Fig 1.12 shows the results for both segmentation datasets. For Sky, the WATER-VOL, WATER-AREA, and WATER-DYN have very similar results as the cut by threshold with only a few regions. Emphasis on the WATER-PAR that kept an equal score through most of the range, slightly decreasing the average after arriving at the peak score with ten regions (a decrease of 3% with 1000 regions). It indicates no significant change in the number of parents during the construction for most of the region range evaluated. Contrasting these results with the cut by threshold, they indicate that the WATER-PAR has much smaller regions at the bottom of the tree than the other two region-oriented watersheds.

Because of the nature of the Sky dataset, there is much more leniency in adding

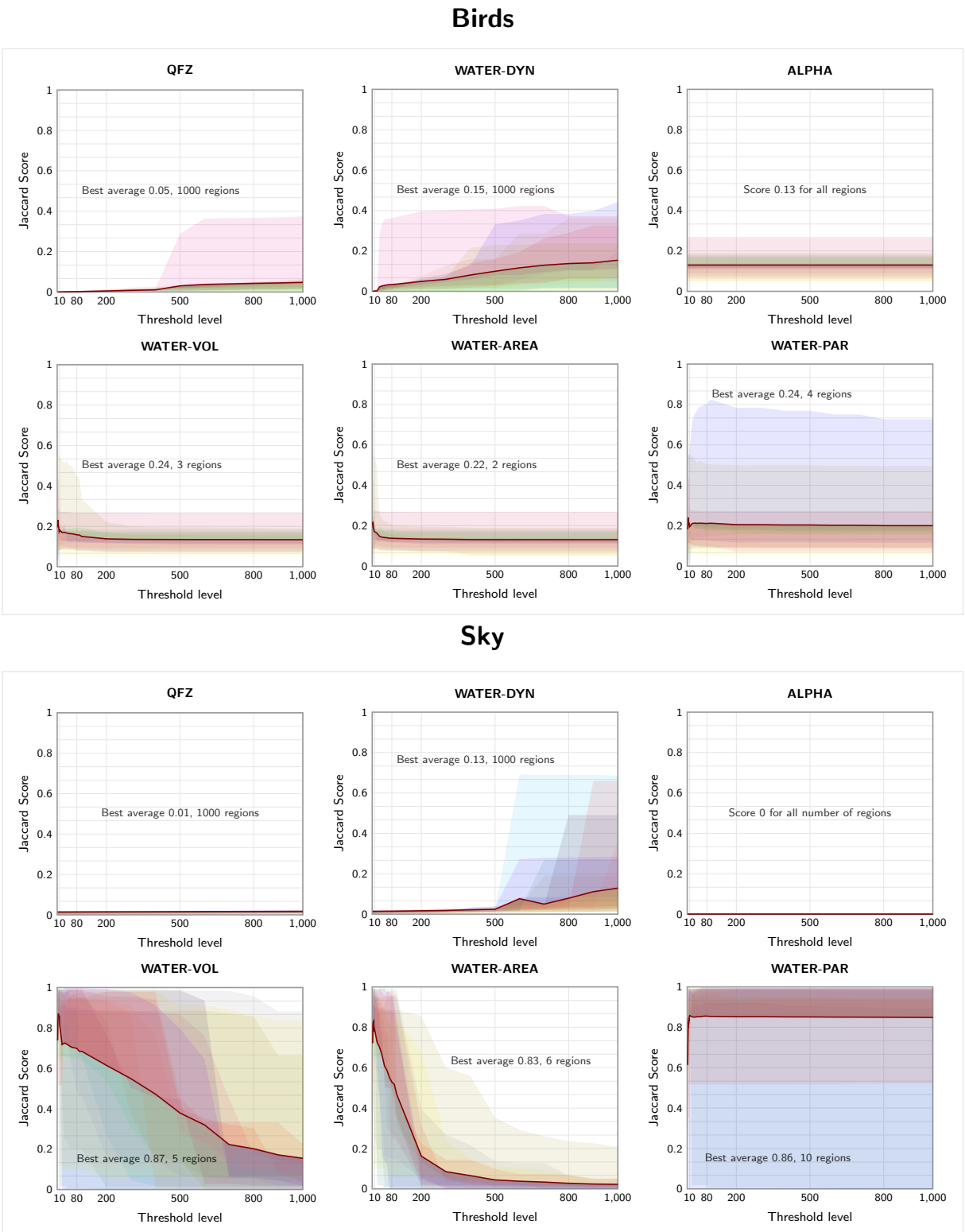


Figure 1.12: Results of the typical pipeline with cut by the number of regions for the Birds and Sky. Colored shaded areas represent individual images score for the number of regions, and the solid line is the average. Perfect score=1.

small regions to the large area composing the background or vice-versa. For instance, the same stable score is observed for all the region-oriented watersheds on the Birds dataset. However, for Birds, the score remains unchanged because most images present white outputs altogether. In the remaining, only a few areas are added to the output as the number of regions increases. Also, they achieve a lower score with the number of regions because, contrary to the Sky dataset, the highly contrasted areas on the input images spread the areas on the tree that are more easily covered by thresholding the structure than selecting an undefined number of ideal regions. This last observation is particularly true for the contour-oriented hierarchies, where the score slowly mounts as new regions are added to the partition without achieving similar scores to the threshold strategy. In fact, in both datasets, the region cut with a similar score to the threshold cut for WATER-DYN starts with 20000 regions and QFZ with 50000.

1.6.3 Final considerations on the typical pipeline experiments

The experiments were performed in two tasks (image segmentation and edge detection) using a single media (images). A single output on the image space is required to evaluate the structures on the tasks. To achieve that, the experimental pipeline uses the trivial approach by a horizontal cut using two common strategies: selecting the number of regions or selecting a threshold value on the altitude levels. The experiments extensively searched for the parameter value for each cut strategy, namely 27 different regions in the range $[2, 1000]$ and over a thousand altitude values. The experiment results with both cut strategies showed that a successful application is not always evident.

For instance, no set of parameters on either tasks or cut strategy gives good results with the ALPHA representation. On the edge detection task, the parameter search for a horizontal cut by a threshold on the altitude levels did not result in any satisfactory value for any hierarchical type. The cut by the number of regions improved the values, particularly for the region-oriented types (WATER-VOL, WATER-AREA, and WATER-PAR), but not enough to place the results among the state-of-the-art on the BSDS500 dataset.

The segmentation task on the Sky dataset with the region-oriented hierarchical types showed promising results with both cut strategies, although the WATER-VOL and WATER-AREA perform better with smaller values on the cut by the number of regions. The contour-oriented hierarchies are considerably more sensitive to the cut parameters on this dataset. QFZ and WATER-DYN only showed some satisfactory results with the cut by

altitude values in a small range. Finally, all types with both cut strategies performed very poorly on the segmentation task on the Birds dataset. From the results, one may conclude that the hierarchical structures are unsuitable for this task. Fairly, the illumination condition, the background, and the annotations on the images make in this dataset make it very challenging for most known image processing strategies in the literature.

Besides the search for the hierarchical type, the strategy cut, and the parameters, multiple other decisions must be taken to apply the hierarchies to the tasks, such as the binarization value, the determination of the regions characterizing the foreground or background, and the evaluation metric that properly reflects the quality of the images created. All of these factors have a significant impact on the final results. They may deter an interested researcher from further investigating this type of structure even if it provides a richer representation than the media alone. Furthermore, even if the ideal set of parameters is found and the results are satisfactory, there is no guarantee that it will perform similarly with a different related media and task. The experiments in this section showed precisely that and demonstrated that a successful application requires a deep understanding of media and tasks and multiple experimentations.

1.7 Discussion on hierarchies and the typical pipeline

The hierarchical operators are delineated in the mathematical morphology domain. Classical mathematical morphology is based on lattice algebraic operations, and the image scope is the principal definition space for the hierarchical theory. However, given a proper characterization of regions and interrelations, hierarchies could be used to process many media types.

Hierarchies defined on graphs facilitate this generic media characterization since graphs are versatile, adaptable to the desired context, and often considered a generic data structure. Intrinsically, hierarchies structured as hierarchical trees are a particular type of graph; therefore, the graph theory is a common point for both media and hierarchies.

The graph representation of media is a modeling question with various connotations (to be discussed in Chapter 3). Still, the primary concern in graph theory is how the components are related. The graph formalism for the elements, the relationships, and the operations are defined for sets of data. In hierarchical analysis, the region definition is in the structural components on the vertices and edges, and the edge weights and adjacency relation choices define the parameters for the hierarchical operators.

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. Each construction algorithm has particular properties, but they share common rules that facilitate commuting and inferring from one type to another if needed.

The output of a hierarchical algorithm is an elaborated scheme with an organic representation in the form of a tree, where each internal node represents a hierarchical level, the media components are portrayed on the leaves, and paths on the tree connect the modeled regions. However, many different aspects must be considered when applying a hierarchy to a task, from choosing the hierarchical type, the parsing strategy, the proper representation, and the evaluation metric. All of these aspects are crucial in a successful application.

Among the necessary considerations for the applications, selecting the hierarchical type may be the most straightforward because of their shared rules and the easy type change without pipeline alteration. However, it is not always clear if the inadequacy comes from the hierarchical construction or the evaluation method since some metrics may not correctly reflect subtle changes in the representation. Furthermore, some tasks may require additional parsing, such as binarization, which is in itself not evident and can undermine the representation.

Undoubtedly, the most crucial consideration is selecting how the hierarchy is represented for the application. The trivial approach with horizontal cuts creates a single partition that adequates the hierarchical structure to the usual ground-truths of a task for evaluation. The process could be strenuous if searching for an ideal number of regions or could disregard important details if thresholding by hierarchical levels. Also, a good horizontal cut for one specific hierarchy does not guarantee that it will be ideal for another on the same dataset. Non-horizontal cuts are potentially better, but the efforts thus far rely on combinatorial search. Finally, even considering all the different aspects and systematically searching for an ideal cut may not suffice for a good performance.

Preprocessing the data in the hierarchical pipeline are often necessary and requires a deep understanding of the media, the region connotation in the hierarchies, and the task. Post-processing may also be required by filtering, balancing, or realigning the structure, consequently facilitating the analysis, but it is media and task-dependent. Possible

solutions include creating a model that learns the ideal representation directly from the structure or uses the structure as a basis for the model applied to a task. Chapter 2 will provide an overview of the strategies in the literature dealing with these problems assessing how they are used for various media and tasks.

This thesis argues that the hierarchical information embedded in the hierarchies could be applied directly in a learning framework without making assumptions about the media or the task. It proposes (in Chapter 5) to represent the hierarchy by selecting attributes on the structure that preserve their original arrangement, creating a regular representation that a learning framework could use directly on a task.

Chapter 2

Hierarchies and machine learning

Hierarchies provide rich representations of nested regions that could help solve multiple computational problems, particularly multimedia processing. However, the applications usually require one single outline for evaluation. The arduous strategy of performing a series of horizontal cuts covering the entire hierarchy and evaluating each partition often leads to sub-par performances regarding the abundance of information embedded in the structure. Combinatorial non-horizontal search for cuts or flattening and region filtering are common strategies with no obvious solution. They all depend on the metric used to evaluate the selection, which can be misleading. Furthermore, finding an ideal representation for one specific structure does not guarantee that it will be suitable for any other hierarchy or task.

Machine learning techniques could facilitate this process by creating models leveraging the hierarchical data and the application. For instance, they could assemble the information in the structure and intermediate the task. Or they could parse the hierarchies to create more suitable representations and produce a replicable model. Furthermore, machine learning techniques usually consider entire datasets for modeling instead of analyzing each entry individually and are evaluated on unseen data for a more robust assessment. However, since there is no direct measurement of quality on the hierarchical structure itself, applying machine learning to hierarchies remains an open question.

This chapter features a literature review on the theme "Learning on hierarchies", gathering the strategies that combine machine learning and hierarchical data on the same framework. The search aims to collect information on methods that: (i) apply the hierarchies to a learning framework assessing how the structure assists on the task; and (ii) apply the learning strategies to the hierarchical structures assessing how the learning helps improve the representation.

The search by itself is complicated due to ambiguous and extensively used terms such as "hierarchies" and "hierarchical". Simply searching for "hierarchical"+"learning" or "hier-

archies"+"learning" retrieves a myriad of results that includes hierarchies of abstractions [1], architectures [2], features [3], or concepts [4]. Therefore, the search keys combined "learning" or "learn" with commonly used terms to describe morphological hierarchies such as "hierarchies of partitions", "partition tree", "inclusion tree", and "component tree". Also, it included specific hierarchical types nomenclatures of interest for this thesis, namely "quasi-flat zones", "binary partition trees", and "alpha-trees". For the "hierarchies of watersheds", there is no consistent naming in the literature, and simply "watershed" or "watershed merge trees" occasionally refers to the hierarchical form in question. The review includes all the watershed names and briefly outlines the strategies on the classical watershed outside the hierarchical structure for completeness.

These search keys retrieved 225 publications from 1990 to 2022 from the IEEE, ACM, and Science Direct databases. From the total retrieved, the review will omit 161 due to the following:

— **Not in the English language:** 2 publications.

— **Bad quality:** 8 publications with either an incomplete set of references or an unclear methodology for review.

— **Comparison:** 33 publications mention the target hierarchies but only to compare with their proposed methods outside the scope of the review.

— **No learning:** 20 publications mention a machine learning method but only to compare with a hand-engineered proposal in a hierarchical context, such as methods proposing a similarity measure for the tree nodes or a re-arrangement of the structures that do not involve a learning step.

— **No hierarchy:** 48 publications that, despite refining the search keys, still retrieve undesirable hierarchies or reference geological tasks that take the literal meaning of the word watershed.

— **Survey:** 17 surveys or book chapters without a method proposal, only a compilation of methods in a domain such as remote sensing or image processing. This review will take these publications only as guiding references.

1. Ilin Roman, Watson Thomas, and Kozma Robert (2017). *Abstraction hierarchy in deep learning neural networks.*
2. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*
3. Lin Tsung-Yi et al. (2017). *Feature pyramid networks for object detection.*
4. Fan Jianping et al. (2017). *HD-MTL: hierarchical deep multi-task learning for large-scale visual recognition.*

— **Post-processing:** 33 publications show methods that create the hierarchies after the learning step.

Methods expressed as post-processing use the learning step to create better low-level maps for the hierarchical construction [5] [6] [7] or use the hierarchies to improve the method's results in a task as a strategy to combine the learned features and data [8] [9] or add a semantic connotation [10]. There are some notable works in this category. For instance, the work in Arbeláez et al. (2011) presents a method (gPb-owt-ucm) that learns improved contour maps and uses them as local minima seeds for constructing the watershed hierarchies. The learning step takes clues from the brightness, color, and texture information cropped from image paths at different scales. And in conjunction with contour feature maps extracted from the eigenvectors of the spectral clustering algorithm [11], it trains a logistic regression model by gradient ascent on the edge detection task. After the learning step, it applies the hierarchical watershed and evaluates the hierarchies created from the improved contours on the segmentation task. The representation is as **ultrametric contour maps**, which transforms contour probability maps into a hierarchical boundary map.

Similarly, Maninis et al. (2018) uses the learning step to create contour maps at multiple intermediate layers of a convolutional network. Side-outputs of the network weights are mapped to the image domain as contour maps at different scales. The hierarchy is not strictly defined. Instead, it is implied from the different scales of the network. However, in the end, the learned contours are combined using ultrametric maps. But, since the learning step disregards the hierarchical information to perform the task or improve the hierarchical representation, the "post-processing" methods will be omitted from the review.

The remainder of this chapter will present the review results. Section 2.1 contains the methods employing the hierarchies to a learning framework, assessing how the structure assists in the task and how the authors format the hierarchical information for the applica-

5. Arbeláez Pablo et al. (2011). *Contour detection and hierarchical image segmentation.*
6. Kim Eunji et al. (2022). *Deep learning-based phenotypic assessment of red cell storage lesions for safe transfusions.*
7. Sinchuk Yuriy et al. (2021). *Geometrical and deep learning approaches for instance segmentation of CFRP fiber bundles in textile composites.*
8. Maninis Kevis-Kokitsi et al. (2018). *Convolutional oriented boundaries: from image segmentation to high-level tasks.*
9. Chen Yanlin et al. (2020). *Automatic segmentation of individual tooth in dental CBCT images from tooth surface map by a multi-task FCN.*
10. Zheng Wenfeng et al. (2021). *Improving visual reasoning through semantic representation.*
11. Shi Jianbo and Malik J. (2000). *Normalized cuts and image segmentation.*

tion. Section 2.2 presents the methods implementing learning strategies to the hierarchical structures assessing how the learning helps improve the representation and the evaluation choices made by the authors. Furthermore, Section 2.2 includes some hand-engineered techniques for transforming the hierarchies and some local-optimization strategies that are not framed as machine learning and, therefore, not retrieved by the search keys. For completeness, Section 2.3 presents the approaches for the non-hierarchical watershed, briefly describing the methods, and Section 2.4 the learning strategies inspired by the hierarchical construction algorithms.

Finally, Section 2.5 displays the review's conclusions, summarizing the method's goals, strategies, and motivations. It is particularly interested in the assessment of : (i) the types of media, how the authors model their representation both on the hierarchical structure and the task; (ii) the types of hierarchies and which role they play in the learning framework; and (iii) the machine learning methods and the reasons for choosing them.

2.1 Hierarchies applied to learning

This section presents the strategies that use the hierarchies to assist the machine learning algorithms in performing a task. It groups the methods into two categories regarding how they use the regions defined in the hierarchical structure. Namely: (i) the regions on the tree nodes define the space for feature extraction (Section 2.1.1); or (ii) the regions represent masks applied on the media for features extraction (Section 2.1.2).

2.1.1 Regions defined on the tree nodes

There are two medical applications in this category, E. Grossiord et al. (2017) [12], and Padilla et al. (2021) [13]. Both methods propose a strategy to combine Computed Tomography (CT) images, grayscale images often labeled, and 3D Positron Emission Tomography (PET) images that gives reliable information about changes in tumor tissues but are challenging to parse automatically.

12. Grossiord Eloise et al. (2017). *Automated 3D lymphoma lesion segmentation from PET/CT characteristics.*
13. Padilla Francisco Javier Alvarez et al. (2021). *Random walkers on morphological trees: a segmentation paradigm.*

In E. Grossiord et al. (2017), they propose to use max-tree [14] to model the PET images, which facilitates the detection of extremal intensity values in the structure for a fast match with the labeled boundary maps in the CT images. The description uses the region's shape, the pixels within the areas for intensity and histogram statistics, and the textural 3D spatial information. The target task is segmentation, where a supervised classification on the CT labels allows keeping only the desired regions for analysis. The biggest challenge in their framework is that the labels need to be precisely defined in the image space; instead, they are given within a bounding box. They choose to label positively any element in the region touching the box region without overlapping areas. The learning model is the Random Forest (RF) [15] because of the large space of features compared to the small number of data samples and the good generalization model.

Similarly, in Padilla et al. (2021), the authors propose to use the spatial correspondence between PET and CT images to compute complementary attributes for the task in a graph context. In their proposal, the PET images are re-scaled to the CT resolution. Both are represented as the hierarchical tree of shapes [16] (a dual hierarchical representation of min/max-trees [17] merged in another hierarchy to balance coarse and fine regions). Each region of the tree is described only by structural and regional features. Namely: relative distance between parent and node, number of voxels on the region, barycenter, and region compactness. To reduce the number of nodes, they propose to filter the structure by searching for stable areas (finite differences along all the branches induced by a node [18]) regarding each attribute and performing a majority vote to determine the most critical regions. The labels are hand-selected foreground/background seeds in the 3D voxel. The task is performed by label propagation using Random Walk [19] algorithm. Random Walk performs a statistical distribution analysis and is often considered a pattern finder in the graph context. Given subsets of non-overlapping vertices, it determines the probability of reaching one set or another by solving the combinatorial Dirichlet problem in the graph represented as a Laplacian matrix.

Hu, T. Shi, et al. (2021) [20] proposes an aerial analysis solution using high-resolution

14. Salembier Philippe, Oliveras Albert, and Garrido Luis (1998). *Antiextensive connected operators for image and sequence processing.*
15. Breiman Leo (2001). *Random forests.*
16. Monasse Pascal and Guichard Frédéric (2000). *Scale-space from a level lines tree.*
17. Salembier Philippe, Oliveras Albert, and Garrido Luis (1998). *Antiextensive connected operators for image and sequence processing.*
18. Nistér David and Stewénius Henrik (2008). *Linear time maximally stable extremal regions.*
19. Grady Leo (2006). *Random walks for image segmentation.*
20. Hu Zhongwen, Shi Tiezhu, et al. (2021). *Scale-sets image classification with hierarchical sample*

aerial images as input. Because of the large scale of the images, this method uses homogeneous regions pre-segmented as superpixels [21] instead of using the pixel as the unitary element on the graph. A spatially constrained hierarchy [22] (a Binary Partition Tree that takes color-texture information to compute the cost of merging two regions) is then created for the superpixels vertices and color-texture dissimilarity measure for the edges. Furthermore, small regions are merged to reduce the representation given a size parameter. The authors argue that selecting a single scale for an application depends on the task and is not robust. Therefore hand-picked samples from multiple scales are set for description together with different depths in the same path as candidates. The descriptors include information on spectral, textural, binary patterns, and region geometry. The labels for the classification task are propagated to all related nodes in a path, and regions with multiple labels are discarded. The model for the task is the Random Forest because of its good performance and high computational efficiency, trained with the features extracted from the structure and image associated with their respective labels. During the test, all nodes in the hierarchy are subjected to the predictions, taking the levels with the best estimations for the segmentation task, improving the robustness of selecting an optimal scale map.

In Clément, Kurtz, and Wendling (2018) [23], the authors first pre-group the image pixels using the Mean Shift clustering algorithm [24] then they construct a hierarchical variation of the Binary Partition Tree that probe the grouped pixels colors to select a union that is maximal for the similarity criterion [25]. The hierarchy organizes the image's objects and their subparts. Using these subdivisions, a structural description of the image can be computed through an attribute relational graph (ARG) and a Force Histogram decomposition [26] characterizing directional spatial relations. In the proposed ARG, the vertices represent the hierarchical nodes with attributes describing region shapes. The edges represent the hierarchical connections provided with attributes characterizing relative position.

enriching and automatic scale selection.

21. Felzenszwalb Pedro F. and Huttenlocher Daniel P. (2004). *Efficient graph-based image segmentation.*
22. Hu Zhongwen, Wu Zhaocong, et al. (2013). *A spatially-constrained color–texture model for hierarchical VHR image segmentation.*
23. Clément Michaël, Kurtz Camille, and Wendling Laurent (2018). *Learning spatial relations and shapes for structural object description and scene recognition.*
24. Comaniciu Dorin and Meer Peter (2002). *Mean shift: a robust approach toward feature space analysis.*
25. Ward Joe H. (1963). *Hierarchical grouping to optimize an objective function.*
26. Garnier Mickaël, Hurtut Thomas, and Wendling Laurent (2012). *Object description based on spatial relations between level-sets.*

Table 2.1: Summary of methods that use the hierarchies to assist the machine learning algorithm in performing a task, where the regions in the hierarchical structure define the space for feature extraction. The table includes information about the nature of the features.

Reference	Domain	Media	Task	Hierarchy	Model	Features
E. Grossiord et al. (2017)	medical	3D PET, CT images	segmentation	max-tree	RF	pixel, 3D texture, geometry
Clément, Kurtz, and Wendling (2018)	image analysis	RGB images	classification	BPT	BOW, SVM	HOG, structural decomposition
Padilla et al. (2021)	medical	3D PET, CT images	segmentation	tree of shapes	Random walker	structural
Hu, T. Shi, et al. (2021)	aerial analysis	high-resol. images	segmentation, classification	BPT	RF	spectral, textural, patterns, geometry

Both sets of attributes are extracted from the Force Histogram decompositions. Additionally, they compared the performance on the task using only the structural descriptor or combining it with descriptions of pixels within the objects and parts with the histogram of gradient descriptors (HOG [27]) of the pixels within the objects and parts. The representation with information about the image properties and structure gave considerably better results.

Furthermore, the authors proposed to aggregate this representation with a bag-of-features model [28]—variation of the bag-of-words (BOW) scheme that represents words (or features) by histograms of occurrence given a reference dictionary—named bags-of-shapes-and-relations, that not only conforms the representation to a learning-friendly format but also works as a consistent compression of the structure. The task is performed using a support vector machine (SVM [29]) with labels of object classes for classification.

All methods in this category place a strategy to reduce the size of the hierarchical representations, either by filtering, compression, or hand-picked samples. Applied for classification or segmentation tasks, they are defined for image analysis and with additional challenges to adapt the methods to 3D images in medical applications. The hierarchical types are all some variation of the binary partition trees. Regarding the learning model, Random Forest deals with large amounts of data and presents good generalization. At the

27. Dalal Navneet and Triggs Bill (2005). *Histograms of oriented gradients for human detection.*
28. Everingham Mark et al. (2010). *The pascal visual object classes (VOC) challenge.*
29. Cortes Corinna and Vapnik Vladimir (1995). *Support-vector networks.*

same time, the bag-of-features paradigm provides a good solution for the task and the size of the hierarchical structure. Furthermore, all methods rely on features extracted from both the media and the regions defined in the hierarchy, except for the random walker approach that takes only the structure of the hierarchical tree represented as graphs to propagate the task label. Table 2.1 presents a summary of the methods in this category.

2.1.2 Regions as media masks

The authors in Serna and Marcotegui (2014) [30] propose an image analysis solution for the 3D point clouds problem classification. Their strategy takes three steps: detection, segmentation, and classification. The image space defines the detection and segmentation steps. First, they project the point clouds as elevation images, where each pixel has the spatial dimension plus the altitude information. This projection creates the first set of candidates of interest using the spatial continuity of elements. The second set of objects includes candidates for removal created by a series of classical mathematical morphology operations, such as the *top-hat* to fill holes and the *opening* to remove unwanted noisy regions [31]. The problem is that these operations also remove some thin objects of interest. That is why they create a third set of candidates from the quasi-flat zone hierarchies [32]. They use the hierarchical structure thresholded by a high enough level to merge some elements on the ground but sufficiently low to keep the objects that the morphological operators will likely remove. The three sets of image object candidates are combined and filtered by a series of hand-engineered operations to create a segmented image with all the essential objects separated into coherent regions.

After the segmentation, they describe each segmented region with color and geometrical features plus some neighboring information, such as the number of adjacent areas and non-empty pixels in the vicinity. Finally, the learning step uses the set of features for a hierarchical classification [33]. Hierarchical classification is an iterative process, where at each iteration, the data is partitioned using generic labels to subsequent classifications with more specific ones with only a few samples. The few sampled classes are also why they have chosen the SVM as a learning model, plus the benefit of comparing their strategy

30. Serna Andrés and Marcotegui Beatriz (2014). *Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning.*
31. Hernandez Jorge and Marcotegui Beatriz (2009). *Filtering of artifacts and pavement segmentation from mobile LiDAR Data.*
32. Meyer Fernand (1998). *From connected operators to levelings.*
33. Avci Murat (2000). *A hierarchical classification of landsat TM imagery for land cover mapping.*

with similar methods on the task.

In Sun et al. (2015) [34], the authors propose a strategy to identify textual information in natural images. Their method relies on a thorough preprocessing of the images to identify and isolate possible text candidates and on the efficient implementation of the max-tree hierarchy [35]. The hierarchy delineates the regions with possible text and organizes the candidates based on their visual characteristics. Specifically, they convert the input image to grayscale and take the magnitudes, the hue, and the saturation [36] on the original and inverted intensities, resulting in six different images for the same input. The additional images help select less noise and extra pixels in the hierarchy, where one max-tree is built for each of the six images. Each region on the hierarchy is an extremal value, and the contrast of neighboring pixels helps identify text information on the images.

The method's core resides in the strategies to merge and prune the structure to create the best set of candidates representing individual text characters. The merge step simplifies the text line, grouping subtrees regions and reducing isolated components' ambiguity. Then the candidates' raw pixels information is taken as features, allowing pruning according to their shapes and textures. The training data is labeled based on the form of the regions, such as thin, elongated, and squared, and used in an iterative classification scheme. At each iteration of a shallow neural network, the quality of the predicted text proposals is verified by human annotators, selecting the unambiguous predictions that will guide a new iteration with more accurate pruning and labeling.

The method in Díaz, González, and Romero (2009) [37] has three learning steps for classifying parasitic infection in microscopic images. The first step takes thoroughly preprocessed images with a series of domain-specific low-level modifications, such as low-pass filters, luminance correction, and color channel selection. The preprocessing aims to increase the contrast between the background and the blood cells in the foreground for a $k-$nearest neighborhood (kNN) binary classification model using normalized RGB values as features.

Many cell centers are classified as backgrounds because of their non-uniform colors, justifying the construction of the dual min/max-tree Salembier, Oliveras, and Gar-

34. Sun Lei et al. (2015). *A robust approach for text detection from natural scene images.*
35. Najman Laurent and Couprie Michel (2006). *Building the component tree in quasi-linear time.*
36. Chong Hamilton Y., Gortler Steven J., and Zickler Todd (2008). *A perception-based color space for illumination-invariant image processing.*
37. Díaz Gloria, González Fabio A., and Romero Eduardo (2009). *A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images.*

rido (1998) hierarchies. Merging the dual regions fill holes while keeping the boundaries. Furthermore, filtering the merged hierarchy by small area size removes small parts that are usually a result of noise in the background. Conversely, large areas could indicate clumped cells that the second learning step could separate. From the image created from the large area selection on the tree, they use the expectation-maximization clustering strategy [38] to separate better candidates. The clustering method takes a template estimated ideal region, iteratively matching regions for a better correlation between the input shape and the template. Each iteration separates the found matches until the remaining area is smaller than a predetermined value. The final learning step takes all the suitable region features (color, shape, intensity, texture, and relational features) and trains a SVM to classify the cells. They also experimented with a multilayer perceptron model, but the SVM gave the best results.

In summary, the strategies in this category require a complete understanding of how the media's low-level components interact in the space and how they relate to the task. The authors model hand-engineered parameters to select subregions in the hierarchical structure. For instance, the hierarchical structure in Serna and Marcotegui (2014) is crucial because the known characteristics of the objects of interest could be easily filtered on the hierarchy and prevent their removal by the morphological operators. Also, in Sun et al. (2015), the hierarchy organizes and delineates the regions, and the learning algorithm gives information to prune and merge the hierarchy with the help of human annotators. Finally, in Díaz, González, and Romero (2009), the hierarchies are used as a filtering technique to fill holes, remove irrelevant objects, and indicate tangled regions based on the knowledge of the area size that the elements typically appear. Overall, once all these methods identified the objects of interest for the target task, they could be described with media features and applied to well-known learning models. Table 2.2 presents a summary of the methods in this category.

2.2 Learning applied to hierarchies

This section comprises the strategies that use the learning step to improve the hierarchical segmentation. The review here is less concerned with the application and more focused on the strategy applied to the structure. The methods in this section are grouped

38. Dempster Arthur P., Laird Nan M., and Rubin Donald B. (1977). *Maximum likelihood from incomplete data via the em algorithm.*

Table 2.2: Summary of methods that use the hierarchies to assist the machine learning algorithm in performing a task, where the regions define masks in the media for feature extraction. The table includes information about the nature of the features.

Reference	Domain	Media	Task	Hierarchy	Model	Features
Serna and Marcotegui (2014)	urban analysis	3D point clouds	detection, segmentation, classification	quasi-flat zones	SVM	color, geometry, neighborhood
Sun et al. (2015)	image analysis	RGB images	text detection	max-tree	Neural network	raw pixel data
Díaz, González, and Romero (2009)	medical	RGB images	classification	min/max-tree	kNN, SVM, density clustering	color, intensity, shape, texture, relational

based on the strategy used to transform the hierarchies with the learning model. Namely: (i) non-horizontal-cuts; (ii) node selection either by filtering important nodes or pruning less significant ones; (iii) realignment; and (iv) flattening.

These strategies aim to simplify the hierarchies or create a representation more suited to perform a task using a learning model to facilitate the process. Table 2.3 presents a summary of the methods retrieved by the search keys. However, each category section includes some reference methods without a learning step and strategies the authors did not formulate as machine learning and, therefore, were not retrieved by the search.

2.2.1 Non-horizontal cuts

Non-horizontal cut strategies select multiple regions at several levels in the hierarchy. The selection usually involves a combinatorial search of various partitions [39] or an optimization function on energy measures. The primary energy functional applied in hierarchies was formulated by Mumford and Shah (1985) [40], where the energy optimization is presented as a trade-off metric pondering data fidelity and a boundary regularization term.

B. Ravi Kiran and Serra (2014) [41] presents a theoretical study of optimization func-

39. Cardelino Juan et al. (2013). *A contrario selection of optimal partitions for image segmentation.*
40. Mumford David and Shah Jayant (1985). *Boundary detection by minimizing functionals.*
41. Kiran B. Ravi and Serra Jean (2014). *Global–local optimizations by hierarchical cuts and climbing energies.*

Table 2.3: Summary of methods that use machine learning models to to simplify or improve a hierarchical representation. The table includes only the methods retrieved by the search keys.

Reference	Strategy	Hierarchy	Model	Features
Benjamin Perret and Collet (2015)	filtering	max-tree	density clustering	Tree attributes
Wonder A. L. Alves, C. F. Gobber, et al. (2020)	filtering	tree of shapes	neural network	residual, color,region shape, tree attributes
Chenliang Xu, Whitt, and Corso (2013)	flattening	tree of shapes	local energy optimization	color, contour, or average magnitudes
Pinto et al. (2014)	flattening	hierarchical watershed	kNN	graph similarity matrix
Y. Xu, Géraud, and Najman (2016)	flattening	tree of shapes	local energy optimization	color, contour, or average magnitudes
Kurtz, Passat, et al. (2012)	non-horizontal cut	BPT	BOW	color, morphology, tree attributes
Kurtz, Stumpf, et al. (2014)	non-horizontal cut	BPT	BOW	region decomposition
Uzunbas, Chao Chen, and Metaxas (2016)	non-horizontal cut	hierarchical watershed	CRF	regional, boundary
Yuhua Chen et al. (2016)	realignment	multiple hierarchies	proposed regressor forest	color,graph metrics, region shape,textural
Adão, Silvio Jamil F. Guimarães, and Jr (2020)	realignment	hierarchical watershed, hGB	neural network, RF	color,graph metrics, region shape,textural

tions on energy measures for non-horizontal cuts in hierarchies. They argue that there are only two ways to obtain a unique minimal cut: limiting the number of partitions or imposing constraints on the energy. They investigate energy optimization methods and their unique optima assessing how to combine regions on energy, particularly energies computed in different feature spaces. Also, they determine how the methods simplify the combinatorial formulations and how they guarantee an optimal solution. Their work defines two families of climbing energies: by addition and by supremum in optimal cuts based on global and local constraints. Furthermore, they contrast hierarchical cuts as a less complex and more flexible solution over non-hierarchical optimal flow graph cuts [42] and conditional random field models.

42. Boykov Yuri, Veksler Olga, and Zabih Ramin (2001). *Fast approximate energy minimization via graph cuts.*

Guigues, Cocquerez, and Men (2006) [43] interpret the energy of a partition as an additive function of Mumford and Shah energies within its components, which simplifies the combinatorial search. Koepfler, Lopez, and Morel (1994) [44] and Salembier and Garrido (2000b) [45] also use global constraints on additive energies. Koepfler, Lopez, and Morel (1994) builds the hierarchy by iteratively increasing the weight on the boundary regularization term. Salembier and Garrido (2000b) presents a practical application of these principles that streamline the optimal solution as the most accurate image simplification for a given parameter. Soille (2008) presents an alternative to the additive function and creates a constrained connectivity defined in terms of the supremum of energies, and Akçay and Aksoy (2008) [46] defines a local constraint instead of global.

Another theoretical work on non-horizontal cuts and their properties is presented in Cousty and Najman (2014) [47], where the authors establish a direct correlation between non-horizontal cuts and morphological flood filtering in image analysis.

Kurtz, Passat, et al. (2012) [48] and the extended version in Kurtz, Stumpf, et al. (2014) [49] proposes a cut strategy for multiresolution images relying on user-defined example and a model that learns how to mimic the user. The core of the strategy is the correspondence of regions in multiresolution images modeled with BPT, where the manually defined cut at one resolution could be reproduced in another related tree. In Kurtz, Passat, et al. (2012), they model the user interaction with a BOW scheme where k-means extract centroids characterizing the user-selected regions by color and morphological features, and normalized histograms describe them. The subsequent interactions are automatic, where they minimize the comparative distance of histograms at different levels and create the next set of centroids for the following resolution. The extended version replaces the features characterizing the regions by the region decomposition on the subsequent resolution. Therefore, the histograms model the region composition in terms of radiometric

43. Guigues Laurent, Cocquerez Jean Pierre, and Men Hervé Le (2006). *Scale-sets image analysis.*
44. Koepfler Georges, Lopez Christian, and Morel Jean-Michel (1994). *A multiscale algorithm for image segmentation by variational method.*
45. Salembier Philippe and Garrido Luis (2000b). *Connected operators based on region-tree pruning strategies.*
46. Akçay Gokhan and Aksoy Selim (2008). *Automatic detection of geospatial objects using multiple hierarchical segmentations.*
47. Cousty Jean and Najman Laurent (2014). *Morphological floodings and optimal cuts in hierarchies.*
48. Kurtz Camille, Passat Nicolas, et al. (2012). *Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology.*
49. Kurtz Camille, Stumpf André, et al. (2014). *Hierarchical extraction of landslides from multiresolution remotely sensed optical images.*

clusters. Another change from the original proposal is the insertion of domain-specific a priori knowledge that reduces the user interactions in the cut-by-example stage.

Uzunbas, Chao Chen, and Metaxas (2016) [50] presents the segmentation task as optimal labeling in a graph where the hierarchical trees (hierarchical watershed) model the space of all possible segmentations. The features describing the regions are extracted from 3D data color, volume, and boundaries. The learning step selects a non-horizontal collection of nodes by training a conditional random field on labels indicating the segmentation quality (under, proper, over-segmented -1,0,1) derived from the max score with the ground-truth propagated with dynamic programming and restricted on hierarchical principles. The maximum a priori prediction corresponds to the segmentation and presents regions in multiple levels as significant. Their application is neuron segmentation, which is a sensitive output; therefore, their method allows user correction by offering only the regions of uncertainty (marginals of the graph model) to facilitate the interaction. Provided with the user input and the model predictions, the nodes on the hierarchy are adjusted, and the process repeats until the user is satisfied with the result.

2.2.2 Node selection

Strategies that select important nodes or remove others based on a particular importance criterion. Traditionally outside a learning framework, the selection is based on a specific attribute, energy values, iterative mergings, or using a hierarchy to cluster another [51] [52]. The methods presented in this section are the ones that entrust the selection to a learning model.

Wonder A. L. Alves, C. F. Gobber, et al. (2020) [53] presents a filtering strategy that removes undesirable regions based on the concepts of residual operators and ulti-

50. Uzunbas Mustafa Gokhan, Chen Chao, and Metaxas Dimitris (2016). *An efficient conditional random field approach for automatic and interactive neuron segmentation.*
51. Perret Benjamin and Collet Christophe (2015). *Connected image processing with multivariate attributes: an unsupervised Markovian classification approach.*
52. Grossiord Éloïse et al. (2020). *Shaping for PET image analysis.*
53. Alves Wonder A. L., Gobber Charles F., et al. (2020). *Image segmentation based on ultimate levelings: from attribute filters to machine learning strategies.*

mate levelings introduced in earlier proposals [54] [55] [56] [57]. First, the authors define residual values as the difference between an image and a transformed image using any morphological operator such as top-hat or skeleton. Then they define an ultimate leveling as the residual value of two consecutive morphological operations on a hierarchical scaled space, specifically the tree of shapes. The authors then establish an equivalence between the ultimate leveling and a pruning operation on the tree nodes, arguing that some residues may characterize undesired regions and guide the hierarchical filtering. Most of the earlier proposals explore recursive algorithms that propagate the residual values in the hierarchy in a top-down approach exploring multiple region criterion and constraints strategies.

In C. Gobber, Wonder A. L. Alves, and Ronaldo F. Hashimoto (2018), they include the Mumford-Shah energy functional optimization as a filtering criterion on the residual regions. In the more recent publication, Wonder A. L. Alves, C. F. Gobber, et al. (2020), they argue that some residuals are computed from undesirable areas and should be disregarded; hence, they introduce a learning step on the framework that assists in retrieving the desirable residues. More specifically, they propose to: (i) construct a new hierarchy using the ultimate leveling; (ii) learn the similarity between ground-truth and residual regions using a neural network; and (iii) filter the hierarchy combining the residual levels and the learned similarity.

In the pipeline, first, they compute the tree of shapes from the input images. Then the ultimate levelings prune the hierarchy and define the residual regions, where each is described using features from residual values, color, region shape, and tree attributes. In the learning step, the initial labels provided to the model take the maximum match between the ground-truth classes and a measure of similarity between a residual region and the ground-truth. In the testing phase, the regions are subjected to the network predictions, and a threshold on the estimated value decides if a residual region is undesirable. The final step in the pipeline defines the best residual regions as the ones with the highest prediction values. It reconstructs the hierarchical tree taking only disjoint residual regions that cover the entire space.

Benoît Naegel et al. (2007) presents an early attempt to learn the filtering of a com-

54. Alves Wonder Alexandre Luz, Morimitsu Alexandre, et al. (2013). *Extraction of numerical residues in families of levelings.*
55. Alves Wonder Alexandre Luz and Hashimoto Ronaldo Fumio (2014). *Ultimate grain filter.*
56. Alves Wonder A. L., Hashimoto Ronaldo F., and Marcotegui Beatriz (2017). *Ultimate levelings.*
57. Gobber Charles, Alves Wonder A. L., and Hashimoto Ronaldo F. (2018). *Ultimate leveling based on Mumford-Shah energy functional applied to plant detection.*

ponent tree. They construct multivariate attribute vectors with features from the image and the hierarchical structure on a selected set of nodes on the hierarchical tree. The model is a multivariate Gaussian, learning a Mahalanobis distance that expresses the probability that a node belongs to the desired class of nodes manually selected. Thresholding the learned distance filters undesired nodes on the tree. Similarly, Benjamin Perret and Collet (2015) performs a probabilistic analysis of attribute distribution on a max-tree hierarchy without pre-selecting the nodes or manually providing the labels. Instead, the max-tree defines a markovian probabilistic model [58] that aims to find each tree node's most probable hidden label, where the tree attributes at each node serve as observation values. This distribution is then classified, unsupervised, using the expectation-maximization algorithm for estimated labels. The classification results associate a value with each node that could be thresholded for a class-oriented node selection.

2.2.3 Realignment

An alternative hierarchical simplification is performed directly in the contour map. The strategies change the contour depths and provide a solution for the whole set of images. The realignment strategies combine two simplification procedures into the same framework: a scale learning step equivalent to a non-horizontal cut in the structure and a flattening step that provides an easy-to-cut non-trivial solution for the segmentation problem while keeping the other nested regions on different scales.

Yuhua Chen et al. (2016) [59] proposes a prediction scheme that learns the scale of the partitions and re-aligns the structure to provide a middle section in the hierarchy that ideally retrieves all salient objects with a single cut without internal subregions. The scale learning step uses a proposed regressor forest that applies a dynamic optimization function on energy levels and restricts the predictions to sustain the hierarchical principles. To create the learning data, each region in the hierarchy with an area bigger than 50 pixels is described using media features. The regions receive a label indicating if it is under, properly, or over-segmented. The labels are computed as an Intersection over Union measure with the ground-truth, where negative values indicate under, positive over, and 0 is the proper segmentation. After training the regressor, the predicted scales are used to re-align the hierarchies' ultrametric contour maps by a local linear transformation

58. Bouman Charles A. and Shapiro Michael (1994). *A multiscale random field model for Bayesian image segmentation.*
59. Chen Yuhua et al. (2016). *Scale-aware alignment of hierarchical image segmentation.*

placing the learned ideal scale at the threshold of 0.5. They test their strategy on multiple hierarchies.

Adão, Silvio Jamil F. Guimarães, and Jr (2020) [60] proposes a similar approach experimenting with a RF and a neural network for the regression step. They also use two different types of hierarchies hGB [61] (a hierarchical segmentation method based on graphs) and the gPb-owt-ucmArbeláez et al. (2011). Furthermore, the authors also propose an alternative scoring measure for label attribution that incorporates the region size in the measure.

2.2.4 Flattening

Methods in this category flatten the hierarchy into a non-trivial segmentation.

Chenliang Xu, Whitt, and Corso (2013) [62] introduced the flattening solution for the tree of shapes modeling supervoxels in a segmentation task. Their methods use a greedy algorithm that provides a local optimization for the Mumford-Shah energy functional. The optimization process gives an importance value for some attribute that favors removing levels with weak contrast or a complex pattern. However, the order that the levels are removed is not local and impacts other nodes on the path. They solve this problem by sorting the level lines by the attribute importance and progressively eliminating the levels that minimize the energy functional. The output of this process is a single segmentation solution that no longer preserves the hierarchical principles.

The extension of this work in Y. Xu, Géraud, and Najman (2016) [63] introduces an attribute function in the energy functional. The attribute function in this new formulation characterizes the persistence of each shape in the hierarchy under energy minimization. They also propose the saliency map representing the entire simplified hierarchy in a single image where the extinction values characterize the contour strengthens. Therefore, the new solution presents the flattened hierarchy with the semantical meaning preserved in the contours intensities.

Pinto et al. (2014) presents a strategy that learns the distance on the hierarchical

60. Adão Milena M., Guimarães Silvio Jamil F., and Jr Zenilton K. G. Patrocínio (2020). *Learning to realign hierarchy for image segmentation.*
61. Guimarães Silvio et al. (2017). *Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity.*
62. Xu Chenliang, Whitt Spencer, and Corso Jason J. (2013). *Flattening supervoxel hierarchies by the uniform entropy slice.*
63. Xu Yongchao, Géraud Thierry, and Najman Laurent (2016). *Hierarchical image simplification and segmentation based on Mumford–Shah-salient level line selection.*

watershed, re-weighting its edges for an easier cut in the segmentation task. Their method constructs the hierarchical trees from the input images and computes the similarity matrix of the regions defined in the structure. The learning step uses a kNN model to learn the distances separating regions across multiple instances. Iteratively it updates the values on the similarity matrix to better describe the inter and intra-relations between regions. In practical terms, the iterative distance update is a re-weight condition on the hierarchical regions. The final segmentation is obtained with a normalized cut [64] on the re-weighted similarity matrix.

2.3 Learning on classical Watershed

The review retrieved 28 publications with the non-hierarchical watershed [65] [66] [67] because the search keys included "watershed" to circumvent nomenclature inconsistencies. For completeness, the assessment comprises these publications assessing their applications, strategies, justification for the watershed, and the use of the machine learning models.

Medical applications

Most of the publications propose a medical application with images, with varying sources, including: (i) histological [68] [69] [70], (ii) microscopic [71] [72] [73], (iii) cytological [74], (iv) x-ray [75],

64. Shi Jianbo and Malik J. (2000). *Normalized cuts and image segmentation.*
65. Meyer Fernand and Beucher Serge (1990). *Morphological segmentation.*
66. Bleau Andrè and Joshua Leon (2000). *Watershed-based segmentation and region merging.*
67. Vincent Luc and Soille Pierre (1991). *Watersheds in digital spaces: an efficient algorithm based on immersion simulations.*
68. Yoon Ji-Seok et al. (2019). *Automated integrated system for stained neuron detection: An end-to-end framework with a high negative predictive rate.*
69. Xie Lipeng et al. (2020). *Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images.*
70. Whitney Jon et al. (2022). *Quantitative nuclear histomorphometry predicts molecular subtype and clinical outcome in medulloblastomas: preliminary findings.*
71. Li Kaiyue, Ding Guangtai, and Wang Haitao (2018). *L-FCN: A lightweight fully convolutional network for biomedical semantic segmentation.*
72. Chakravarthy Adithi D. et al. (2020). *A thrifty annotation generation approach for semantic segmentation of biofilms.*
73. Liu Ting et al. (2013). *Watershed merge forest classification for electron microscopy image stack segmentation.*
74. George Yasmeen Mourice et al. (2014). *Remote computer-aided breast cancer detection and diagnosis system based on cytological images.*
75. Demir Fatih (2021). *DeepCoronet: a deep LSTM approach for automated detection of Covid-19 cases from chest X-ray images.*

(v) biofilm [76] [77], and (vi) cervical [78] [79]. In those applications, the classical watershed segments the input image, and the boundary map is used to help isolate the regions of interest for the target task.

Some medical applications use the watershed in higher dimensional data, such as 3D microscopic [80], cellular culture [81], magnetic resonance imaging (MRI) [82], and CT [83] images, or neural activity maps in 3D (spatial dimensions + sensor) [84] or 4D (spatial+signal+time) [85]. For those, the channels are: (i) independently segmented in Roy, Mazumdar, and Chowdhury (2020); (ii) projected into the two-dimensional space in Nandy et al. (2011) Oztan et al. (2011), and Zwettler and Backfrieder (2015); or (iii) processed separately and later combined with graph-optimizing geometrical constraints in Matejek et al. (2019). In Whitney et al. (2022), the watershed segment the spatial dimension, and the signal over time data localize specific regions matching the signal activity.

In all medical applications, the authors often justify using the watershed over another segmentation method because it produces coherent regions consistent with the image gradients or because of the lack of large annotated sets of data. Indeed, in Chakravarthy et al. (2020) and Nandy et al. (2011), their target task is to create annotated data using the watershed contour map as a guide. In Chakravarthy et al. (2020), they preprocess and binarize the input images. The binarized gradient image works as the marker for the watershed. The watershed contours create binary edge/non-edge labels that train a convolutional network that produces segmentation, the U-net [86]. Experts' annotations

76. Chakravarthy Adithi D. et al. (2020). *A thrifty annotation generation approach for semantic segmentation of biofilms.*
77. Molina Angel et al. (2021). *Automatic identification of malaria and other red blood cell inclusions using convolutional neural networks.*
78. Alush Amir, Greenspan Hayit, and Goldberger Jacob (2009). *Lesion detection and segmentation in uterine cervix images using an ARC-level MRF.*
79. Alush Amir, Greenspan Hayit, and Goldberger Jacob (2010). *Automated and interactive lesion detection and segmentation in uterine cervix images.*
80. Nandy Kaustav et al. (2011). *Supervised learning framework for screening nuclei in tissue sections.*
81. Oztan Basak et al. (2011). *Classification of breast cancer grades through quantitative characterization of ductal structure morphology in three-dimensional cultures.*
82. Zwettler Gerald and Backfrieder Werner (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.*
83. Roy Rukhmini, Mazumdar Suparna, and Chowdhury Ananda S. (2020). *MDL-IWS: multi-view deep learning with iterative watershed for pulmonary fissure segmentation.*
84. Matejek Brian et al. (2019). *Biologically-constrained graphs for global connectomics reconstruction.*
85. Diego Ferran et al. (2013). *Automated identification of neuronal activity from calcium imaging by sparse dictionary learning.*
86. Ronneberger Olaf, Fischer Philipp, and Brox Thomas (2015). *U-net: convolutional networks*

later refine the segmentation output. After meticulous preprocessing, in Nandy et al. (2011), they compute the watershed for the images, use k-means to create clusters on the image intensity values, and reject the watershed regions intersecting with the lowest intensity clusters. The remaining areas are then iteratively merged using differences in the magnitudes to create new sets of markers. The final regions are then described with morphological and textural features and classified using a neural network with a single hidden layer. The manually annotated regions portray their quality as well or poorly segmented. The authors advocate for their strategy to accelerate manual annotations of unlabeled data.

Another common claim among all methods is the problem with over-segmentation. To deal with this problem, Alush, Greenspan, and Goldberger (2009) and Alush, Greenspan, and Goldberger (2010) only take the contours overlapping with the ground-truth as paths to extract features of the pixel's magnitudes and create two dictionaries of edge/non-edge using principal component analysis (PCA) and k-means in a BOW framework. They normalize the BOW histograms as probabilities distribution for a belief-propagation scheme taking the regions on the watershed as Markov Random Fields (MRF) for the final segmentation. Some strategies rely on specialist markers before the segmentation, such as in Whitney et al. (2022), or depend on domain specif preprocessing, such as the green channel filtering in Molina et al. (2021). Demir (2021) only computes the Sobel gradient for the input but uses a variation of the classical watershed that associates a minimizing function with the original markers to recalculate the boundaries.

More commonly, the over-segmentation issue is dealt with a series of preprocessing steps, such as the denoising, color normalization, and median signal filter in Yoon et al. (2019) to crop images patches on the watershed regions to be used as input for a convolutional neural network (CNN). Similarly, in George et al. (2014), they perform histogram and contrast equalization, gradient computation, high-signal filter, and pre-clustering to the input image to reduce over-segmentation and isolate the regions to be described by their shape and texture for a SVM classification of breast cancer cells. In Zwettler and Backfrieder (2015), there is no preprocessing. Instead, they use an evolutionary classification model on all watershed regions described as similarity histograms of local and regional features.

for biomedical image segmentation.

Aerial analysis

The second leading domain is aerial analysis, where usually the watershed mask regions on aerial images for description. The justification for using watershed regions is that it is a fast discrete operation with reliable results to reduce complexity. They argue that it is simpler to process regions instead of individual pixels, particularly for high-resolution and hyperspectral images as in Sébastien Derivaux et al. (2007) [87] and S. Derivaux et al. (2010) [88]. It also facilitates processing signal data such as laser imaging, detection, and ranging (LiDAR) [89] and synthetic aperture radar (SAR) [90].

Like in medical applications, aerial analysis methods also require preprocessing the images to reduce over-segmentations. However, the strategies are much more domain specific. For instance, in Cretu and Payeur (2013) [91], among other procedures, they filter the green channel in the aerial images because they are searching for buildings, and the green color is typically associated with trees and vegetation. They also remove shadow areas to avoid mixing their contour with the buildings. After creating the watershed, they further process the segmented images by applying morphological operators to filter small areas. The remaining regions are described and classified using the SVM model.

The proposal in Tao Liu, Im, and Quackenbush (2015) is specifically conceived to improve the watershed contours and diminish the over-segmentation problem. To improve the contours, they propose a recursive step to drag the initial borders closer to the delineated object by measuring and thresholding the distance from the region's center and the borders, pondering the height information in the sensor data and the neighboring pixels as reference parameters. To reduce over-segmentation, they perform a wavelet decomposition of the vertical distribution of LiDAR points and train the Random Forest on labels indicating if a region is inside a tree region or separates two tree regions. The areas classified as inside the tree should are merged. The labels are manually attributed by sampling multiple regions at multiple scales and used as references. The classification is recursive until there is no change in the regions.

87. Derivaux Sébastien et al. (2007). *On machine learning in watershed segmentation.*
88. Derivaux S. et al. (2010). *Supervised image segmentation using watershed transform, fuzzy classification and evolutionary computation.* 15.
89. Liu Tao, Im Jungho, and Quackenbush Lindi J. (2015). *A novel transferable individual tree crown delineation model based on Fishing Net Dragging and boundary classification.*
90. Mao Xueyue, Xiao Xiao, and Lu Yilong (2022). *PolSAR data-based land cover classification using dual-channel watershed region-merging segmentation and bagging-ELM.*
91. Cretu Ana-Maria and Payeur Pierre (2013). *Building detection in aerial images based on watershed and visual attention feature descriptors.*

In Sébastien Derivaux et al. (2007) and S. Derivaux et al. (2010), they also propose a strategy to reduce over-segmentation by inserting two learning steps in the watershed algorithm. The first one uses fuzzy classification to create probability maps learned from feature vectors extracted from the input images. The watershed is then computed in the probability map. The second learning step added is a supervised evolutionary segmentation on the watershed parameters: threshold, catchment basins, the euclidean distance between two regions, and a region membership flag. The evaluation criterion for the genetic algorithm optimizes criterion representing over and under-segmentation. In txsm08 [92], they also propose an evolutionary algorithm strategy, but they provide a co-occurrence matrix of magnitude levels and a wavelet decomposition as features. The evolutionary function learns the cluster label of the watershed regions defined for the images. Their proposal is a generic approach with one application in remote sensing and two others in artificial and natural images.

Lopez-Fandino et al. (2018) [93] proposes a distinctive approach among the reviewed, where the authors present a strategy to process multi-temporal hyperspectral images for change detection. They use autoencoders to extract features for all channels, reducing the dimension of the spectral channels in the process. In parallel, they create the watershed for individual spatial dimensions, the grayscale space averaging the magnitudes inside a region. Then they take consecutive images in the temporal axis, marking the pixels that changed in time and removing the non-changed pixels from the image. The final classification takes only the masked pixels represented by the autoencoders' learned features and applies them to a SVM model. Without watershed mapping and masking, this type of processing would be infeasible due to the dimensions of the images and spectral channels.

Other applications

Cheng Chen and G. Fan (2010) [94] combines localization and segmentation on the same framework. They argue that a correct location will encourage accurate shape-constrained segmentation. The watershed creates segmentations. Each region is taken individually to be matched with the contour priors for optimization. New regions are recursively proposed

92. Jiao Licheng et al. (2010). *Natural and remote sensing image segmentation using memetic computing.*
93. Lopez-Fandino Javier et al. (2018). *Stacked autoencoders for multiclass change detection in hyperspectral images.*
94. Chen Cheng and Fan Guoliang (2010). *Coupled region-edge shape priors for simultaneous localization and figure-ground segmentation.*

or merged based on the contours inside a sliding window.

Distinctly, Levner and H. Zhang (2007) [95] they propose to train a linear model on pixel features to create a probability map used as image input for the watershed and experimented with different threshold parameters on the maps to propose other markers. However, the markers often did not lie within the object boundary, and multiple markers were provided for the same region. Their final approach was to use the morphological erosion of the ground-truth to create the set of markers.

Other applications are mostly segmentation proposals applied for RGB images [96] [97] [98]. The input in Börold et al. (2020) [99] is depth images. They rely upon extensive signal and image preprocessing to level all the information in the same space (depth, space, and gaussian distribution). Once adequately prepared, they proceed with the usual: create the watershed segmentation, crop patches in the image, and classify the data, in their case using a pre-trained convolutional adapted to count components in an industrial context.

Table 2.4 presents a summary of all methods reviewed in this section.

Table 2.4: Summary of methods that use the non-hierarchical watershed to assist the machine learning algorithm in performing a task, or methods where the machine learning supports improving the watershed contours.

Reference	Domain	Media	Task	Model
Alush, Greenspan, and Goldberger (2009)	medical	cervical images	segmentation	BOW, MRF
Alush, Greenspan, and Goldberger (2010)	medical	cervical images	segmentation	BOW, MRF
Nandy et al. (2011)	medical	microscopy 3D	Annotation	Neural networks
Oztan et al. (2011)	medical	celular culture 3D	classification	SVM
Diego et al. (2013)	medical	neural activity 4D	segmentation	Sparse coding
Ting Liu et al. (2013)	medical	microscopy	segmentation	RF
George et al. (2014)	medical	cytological	classification	SVM
Zwettler and Backfrieder (2015)	medical	MRI 3D	segmentation	Evolutionary

95. Levner Ilya and Zhang Hong (2007). *Classification-driven watershed segmentation.*
96. Xin Hai et al. (2011). *Human head-shoulder segmentation.*
97. Nagoda Nadeesha and Ranathunga Lochandaka (2018). *Rice sample segmentation and classification using image processing and support vector machine.*
98. Ma Wenping et al. (2012). *Image segmentation based on a hybrid Immune Memetic Algorithm.*
99. Börold Axel et al. (2020). *Deep learning-based object recognition for counting car components to support handling and packing processes in automotive supply chains.*

Reference	**Domain**	**Media**	**Task**	**Model**
K. Li, Ding, and H. Wang (2018)	medical	microscopy	semantic segmentation	Proposed
Yoon et al. (2019)	medical	histological	detection	CNN
Matejek et al. (2019)	medical	neural activity 3D	segmentation	Greedy opt.
Chakravarthy et al. (2020)	medical	microscopy, biofilms	annotation	U-net
Roy, Mazumdar, and Chowdhury (2020)	medical	CT 3D	segmentation	CNN
L. Xie et al. (2020)	medical	histological	segmentation	CNN
Demir (2021)	medical	X-ray	classification	CNN
Molina et al. (2021)	medical	blood smear	classification	CNN
Whitney et al. (2022)	medical	histological	classification	RF
Sébastien Derivaux et al. (2007)	aerial	high-resolution, hyperspectral	semantic segmentation	Fuzzy, evolutionary
S. Derivaux et al. (2010)	aerial	high-resolution, hyperspectral	semantic segmentation	Fuzzy, evolutionary
Jiao et al. (2010)	aerial	aerial images	segmentation	evolutionary
Cretu and Payeur (2013)	aerial	aerial images	classification	SVM
Tao Liu, Im, and Quackenbush (2015)	aerial	LiDAR	segmentation	RF
Lopez-Fandino et al. (2018)	aerial	multitemporal hyperspectral	change detection, classification	Autoencoders, SVM
X. Mao, Xiao, and Yilong Lu (2022)	aerial	SAR	classification	Neural net. bagging
Cheng Chen and G. Fan (2010)	image	RGB	segmentation, localization	Proposed
Xin et al. (2011)	image	RGB	segmentation	Adaboost
Nagoda and Ranathunga (2018)	agriculture	RGB	segmentation, classification	SVM
Börold et al. (2020)	industrial	3D images	classification, counting	CNN
Levner and H. Zhang (2007)	learn markers	granulometry	segmentation	Linear model
W. Ma et al. (2012)	learning	RGB, SAR	segmentation	Proposed

2.4 Learning algorithms inspired by watershed

The final section in this review regards three retrieved publications that present a learning scheme inspired by the watershed algorithm.

Challa et al. (2022) propose two methods with a watershed-inspired classifier. The first one [100] extends the watershed algorithm for edge-weighted graphs to a semi-supervised classification scheme. In their proposal, the markers represent known labeled data points, and the algorithm partitions the remaining vertices by their sorted edge weights in the same fashion as in the watershed. Their study shows the properties of this classifier and establishes a correlation between a traditional classifier maximum margin principle (such as in the SVM) with the maximal margin partition in their method. Furthermore, they also propose an ensemble scheme to deal with redundant features by taking subsamples of data and markers to construct multiple classifiers and compute a weighted average taken as final labels.

They present the second method [101] as an alternative to the softmax layer in convolutional networks. In this approach, they propose a triplet loss function to train the neural network that induces a trainable parameter in the first watershed classifier and enforces order distances. More specifically, the neural network produces a set of features taken alongside some labeled markers for the watershed classifier label attribution. The triplet loss function collects each labeled entry and compares it with positive and negative input, minimizing the distance for the former and maximizing for the latter. The network uses the triplet loss as the cost, and a new classifier is computed at each epoch. The authors report superior performances with experiments in multiple hyperspectral image datasets, compared with the state-of-the-art neural network architectures and different classifiers for the triplet function.

In Bai and Urtasun (2017) [102], they incorporate the watershed strategy in a convolutional neural network architecture for semantic segmentation. They propose a two-stage network. In the first stage, they modified a VGG16 network [103] to avoid spatial reduction and keep the input's dimensions. It takes the input images filtered to keep only the relevant pixels regarding the segmentation ground-truth on the color channels and incorporates the ground-truth as an additional channel. The VGG16 learns the direction

100. Challa Aditya et al. (2019). *Watersheds for semi-supervised classification.*
101. Challa Aditya et al. (2022). *Triplet-watershed for hyperspectral image classification.*
102. Bai Min and Urtasun Raquel (2017). *Deep watershed transform for instance segmentation.*
103. Simonyan Karen and Zisserman Andrew (2015). *Very deep convolutional networks for large-scale image recognition.*

to the nearest border for each image pixel by estimating the direction of descent of the energy. Occluded objects will have opposing directions, which improves the distinction of different elements in the image.

The second stage is a generic convolutional network module inspired by the watershed algorithm that learns to map the directions to energy values. Instead of gray levels, the module uses the learned energy for topology. The first stage produces a two-channel vector with the image resolution size, while the second creates a bin bucket vector representing possible energy levels in pixel distance. In the bin bucket, bin 0 indicates the background and very close pixels (2-pixel distance), and higher-numbered bins correspond to regions in the object's interior. The second stage learning function attributes the bin by maximizing the energy level near zero to facilitate a cut that gives different classes at different bins. The object instances are represented as energy basins followed by cuts into a single energy level. In their formulation, bin 1 gives small objects such as people, bicycles, and motorcycles, and bins 2 to 4 give cars, buses, trains, and so forth. They evaluated their proposal in the Cityscapes dataset [104], and their results were more than doubled the state-of-the-art when published and remain relevant in multiple class categories.

2.5 Review discussion

The review found that hierarchies assisting the machine learning algorithms in performing a task define regions delimiting areas for feature extraction or represent masks applied on the media. Almost all methods rely on media features for the learning step and often require reducing the size of the hierarchical representations, either by filtering, compression, or hand-picked samples. The strategies in this category require a complete understanding of how the media's low-level components interact in the space and how they relate to the task.

Unsurprisingly the predominant media is images, and most applications are classification, segmentation, or detection. There are many domains, but the most dominant is the aerial and medical analysis and generic image processing. Regarding the models, Random Forest, SVM, and neural networks are often the models of choice for their robustness and generalization capabilities.

Among the methods using machine learning applied in the hierarchical structure, the typical approach is the energy optimization strategy to identify regions of interest inside

104. Cordts Marius et al. (2016). *The cityscapes dataset for semantic urban scene understanding.*

the hierarchical structure. Another common technique is to transfer the learning target to a parallel task that induces a response on the hierarchical nodes. Most of the methods present complex solutions or combinatorial analysis. Learning the hierarchical structure remains an active open research topic.

The majority of retrieved results in the review are for the non-hierarchical watershed. It is prevalent among medical applications that rely on coherent and consistent regions. A widespread problem among all methods using the classical watershed is over-segmentation. Many strategies rely on thorough preprocessing for successful applications, while others propose learning techniques to merge some regions or select areas of interest.

Finally, one unexpected result of the search is learning models inspired by the watershed algorithm that presents promising results for inference in multiple tasks.

Part II

Learning on graphs

Chapter 3

Graphs, media, and machine learning

Graphs are structures used to represent objects, and the primary concern in graph theory is how these objects are interconnected. They can depict many data and carry information about the objects in their components, including from different domains, such as numerical, textual, and logical. All deliberations in this work are centered on graph theory as they could provide generalization tools for: (i) the hierarchical structures depicted in a tree structure; (ii) the multimedia data; and (iii) a media-independent learning framework.

This chapter presents a literature review of machine learning on graphs, exploring the motivations, strategy, and fundamental issues (Section 3.1). To formulate the pertinence and identify the limitations, it provides a systematic review of deep learning on graphs, concentrating on the multimedia processing perspective (Section 3.1.2). At this chapter's end are a brief discussion and some final considerations that advocate for the framework choices (Section 3.2).

3.1 Review learning on graphs

Machine learning on graphs is a topic of great interest due to: (i) its autonomy—once you have your learning system operating in terms of vertices and edges, the data's source becomes virtually irrelevant; (ii) the multiple possibilities of applications; and (iii) the capacity to represent multivariate information.

However, employing graphs to known learning frameworks presents a few challenges, that is to say: (i) the **size**, particularly for graphs representing digital media, due to the original media dimension (often large) combined with dense adjacency relation and all the additional information stored on vertices and edges; and (ii) the graph's arbitrary structure—generally not well-defined beginning and end, two connected vertices are not

necessarily close, multiple possible paths—where machine learning algorithms usually expect systematic inputs.

The size issue demands large amounts of computational time and resources. Usually, researchers approach this problem by working on subgraphs, pairwise comparisons, and compact representations. Or even by choosing a machine learning method suited to operate in high-dimensional feature space. Yet, each of these solutions presents its own set of issues.

As for the arbitrary structure issue, depending on the graph's size, working on the adjacency matrix can be a desirable solution, but it is rarely an option when working with multimedia. There are strategies to transform the graph properties into vectorial representations, which may result in information loss from constraining the graph's structure and requires careful consideration. It requires selective parsing and must account for the graph type, the task it is trying to solve, and the proximity between the graph and the original data.

This section reviews the leading machine learning strategies that deal with graphs, including their strengths and limitations. The goal is to assess the recent approaches, their proposals to adapt the methods for media processing tasks, and the connections between what is learned and what is represented on the graphs. More precisely, it will discuss: (i) graph embedding, as methods dedicated to creating vectorial representations of graphs (Section 3.1.1); (ii) deep learning on graphs with a systematic review of applications on multimedia data (Section 3.1.2); and (iii) random forests on graphs, a fast learning method adapted to work with high-dimensional data (Section 3.1.3).

3.1.1 Graph embedding

In machine learning, graph embedding methods aim to codify graph components into a vectorial representation as a preprocessing step in a pipeline. The goal is to preserve the most relevant properties of the graph without losing too much information on the process. Many researchers resort to graph embeddings, considering that available machine learning methods on graphs are limited. At the same time, they are widely available in vectorial space, where operations are more straightforward and faster.

An ideal embedding method would keep all the relevant information, such as the topology, the modeled relationships, and essential features. But defining and finding this information is not always trivial. There is a general agreement that longer embeddings preserve relatively more information, but they can increase the embedding time and create

a high-dimensional feature space.
There are three main categories of strategies for graph embedding methods:

1. **Matrix factorization:** Algebraic models. Main methods: Locally Linear Embedding (LLE) [1], Laplacian EigenMaps strategy [2] and Generalization of High-Order Proximity (HOPE) [3].
2. **Random walk:** Probabilistic strategies. Main methods: Deep Walk [4] and Node2Vec [5].
3. **Deep approaches:** multi-layered neural networks. Main method: Structural Deep Network Embedding (SDNE) [6].

The matrix factorization category incorporates algebraic methods filtering with arrays of values or functions the graphs represented as adjacency matrices. They use a series of decomposition matrix operations to project the response into a linear representation. In LLE, for instance, matrix row vectors are projected to the k-near neighbors and then reconstructed back to the original vectors. The final representation includes the projections that best reconstruct the original matrix. The Laplacian EigenMaps strategy adapts the Laplace-Beltrami operator for manifold, in which linear projections aim to preserve local information and suppress outliers. The HOPE algorithm is proposed for large graphs, aiming to maintain high-order proximities using single-value decomposition as a generalized eigenvalue problem, with incremental perturbation of the values to capture the dynamics of the graph. These models have a solid theoretical base and produce meaningful representations, but they are very computationally expensive for large graphs [7].

In the random walk category, probabilistic methods aim to identify similarities in a path within the graph using random node sampling. The word2vec strategy [8] is the inspiration for this category, which transforms word sentences into embedded vectors by applying the skip-gram model—a one-hidden-layer neural network operating in a fake

1. Roweis Sam T. and Saul Lawrence K. (2000). *Nonlinear dimensionality reduction by locally linear embedding.*
2. Belkin Mikhail and Niyogi Partha (2001). *Laplacian eigenmaps and spectral techniques for embedding and clustering.*
3. Ou Mingdong et al. (2016). *Asymmetric transitivity preserving graph embedding.*
4. Perozzi Bryan, Al-Rfou Rami, and Skiena Steven (2014). *Deepwalk: online learning of social representations.*
5. Grover Aditya and Leskovec Jure (2016). *Node2vec: scalable feature learning for networks.*
6. Wang Daixin, Cui Peng, and Zhu Wenwu (2016). *Structural deep network embedding.*
7. Makarov Ilya et al. (2021). *Survey on graph embeddings and their applications to machine learning problems on graphs.*
8. Goldberg Yoav and Levy Omer (2014). *Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method.*

task: given a word, the skip-gram model would try to predict its neighboring words. Objectively, the word2vec framework is not interested in the predictions of the skip-gram; it just wants to learn the weights of the hidden layer, reasoning that similar words should have similar values. Deep Walk uses random walks to produce embeddings by sampling the graph with random walks coded as one-hot vector input for training on skip-gram and the embedded output is the hidden layer weights. Node2vec is a similar method, but with a biased random walk that could exploit the paths in-depth and broad and better infer the graph structure. Because of their procedures, the methods in the random walk category are less resource intensive than the matrix factorization, and they usually have an excellent capability to identify the graph structure but fail to incorporate the graph features in the final representation.

In the deep approach category, there is the structural deep network embedding (SDNE) which has two autoencoders with shared weights. The embedding is the distance of weights calculated for every single node on the graph and measured for all pairs of connected nodes. SDNE is highlighted because it is explicitly a graph embedding method. Still, many deep learning networks for graphs are a graph embedding method since the loss functions could be used to penalize dissimilarities and transmute the data to a set of features extracted from the network weights. The deep approaches usually produce meaningful sets of features representing the graph features. Still, their algorithms impose many restrictions on the input form and are memory intensive for large graphs.

3.1.2 Deep learning on graphs

Machine learning on graphs is in concert with the dominance of deep learning methods proposals in recent years [9]. Since the original proposal of the graph neural network in Scarselli et al. (2009) [10] and Micheli (2009) [11], there has been a fast-growing number of methods with end-to-end networks for graphs. Many now-common mechanisms such as convolution, attention, and recursion keep up the architectures for graphs with the trend advances of general convolutional networks. There are a multitude of application of deep networks operating on graphs, particularly for solving tasks in complex network analysis, natural language processing, and chemistry design.

Recently these methods have been generalized for media problems, specially for spatio-

9. Bacciu Davide et al. (2020). *A gentle introduction to deep learning for graphs.*
10. Scarselli F. et al. (2009). *The graph neural network model.* 1.
11. Micheli A. (2009). *Neural network for graphs: a contextual constructive approach.* 3.

temporal modeling. This section presents a systematic review on multimedia processing in deep learning to provide clues on how authors model multimedia data as graphs and how they apply them to deep graph networks.

A search on the terms *graph neural* or *graph networks* combined with the terms *image*, *video* and *multimedia* within the period of 2013 to 2021 from the databases IEEE, ACM and Science Direct retrieved 162 publications in total, of which 97 were excluded due to: (i) methods targeting undesired media types (26 publications); (ii) model not targeting graphs (29 publications); (iii) a general application strategy (34 publications); and (iv) presentation of survey compilation (8 publications).

The 65 publications reviewed, the methods are grouped based on the input media, namely: (i) videos, combining proposals for traditional videos, sequence of sensor and skeleton frames, multi-video data, and video recommendation systems; and (ii) images, combining proposals for point cloud information, aerial images, images with *red, green, blue* color channels (RGB) and their variant with an additional depth channel (RGBD). Then six identified categories for the reviewed publications based on how the authors incorporate the graphs into the task they are trying to solve:

1. Traditional tasks (16%): Media represented as graphs in well-known tasks.
2. Multi-modal association (12%): Graphs used to join data from different domains.
3. Higher dimensions (10%): Graphs used to incorporate an additional dimension.
4. Enhanced tasks (26%): Graphs to expand the task concept to a more complex goal.
5. Semantic tasks (28%): Graphs to perform tasks with semantic signification.
6. Data augmentation (8%): Graphs to expand datasets with few labels or images.

Regarding the network architectures, proposals are applications of known graph networks, namely the original Graph Neural Network in Scarselli et al. (2009) and its variations as Gated Graph Neural Network [12], and Graph Convolutional Networks [13] [14] [15] as variations from Micheli (2009). Some researchers propose variations of these mod-

12. Li Yujia et al. (2016). *Gated graph sequence neural networks.*
13. Bruna Joan et al. (2014). *Spectral networks and locally connected networks on graphs.*
14. Niepert Mathias, Ahmed Mohamed, and Kutzkov Konstantin (2016). *Learning convolutional neural networks for graphs.*
15. Atwood James and Towsley Don (2016). *Diffusion-convolutional neural networks.*

els [16] [17] [18], while others explore novel methods [19], other architectures such as autoencoders [20] and recursive messaging passing [21], and even embedding [22] [23] [24] applied to traditional deep methods.

The following sections briefly discuss each category that resulted from the review, presenting a summary of the methods by their media and task.

Traditional tasks

This category encapsulates proposals for well-known tasks in multimedia processing, such as classification, recognition, and segmentation. By media and task, they could be outlined them as follows:

Videos: methods aim to solve the problem of human action recognition and classification, in which: (i) skeleton data for action recognition have the joints represented as vertices and connecting bone structures as edges [25] [26] [27]; and (ii) feature extraction on the video frame sequence through the graph network for classification [28].

Images: methods use a cluster of pixels, superpixels, or regions of interest of the images represented as vertices. Then these graphs are fed into a graph network for vertice classification or are embedded for structural change comparison. More specifically: (i) saliency detection: superpixels are pooled together as nodes, and the edge information separates the background and the foreground [29]; (ii) disease

16. Qi Xiaojuan et al. (2017). *3D graph neural networks for RGBD semantic segmentation.*
17. Acuna David et al. (2018). *Efficient interactive annotation of segmentation datasets with polygon-RNN++.*
18. Li Zongmin et al. (2019). *Graph attention neural networks for point cloud recognition.*
19. Chen Siheng et al. (2019). *PCT: large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving.*
20. Gidaris Spyros and Komodakis Nikos (2019). *Generating classification weights with GNN denoising autoencoders for few-shot learning.*
21. Li Wanhua et al. (2020). *Graph-based kinship reasoning network.*
22. Huang Jiashuang et al. (2020). *A novel node-level structure embedding and alignment representation of structural networks for brain disease analysis.*
23. Herzig Roei et al. (2019). *Spatio-temporal action graph networks.*
24. Wu Le et al. (2020). *Learning to transfer graph embeddings for inductive graph based recommendation.*
25. Chen Yuxin et al. (2020). *Graph convolutional network with structure pooling and joint-wise channel attention for action recognition.*
26. Liu Jinde et al. (2020). *Kinematic skeleton graph augmented network for human parsing.*
27. Si Chenyang et al. (2020). *Skeleton-based action recognition with hierarchical spatial reasoning and temporal stack learning network.*
28. Chen Da et al. (2020). *Hierarchical sequence representation with graph network.*
29. Ji Wei et al. (2020). *Context-aware graph label propagation network for saliency detection.*

prediction: extracts regions of interest in images (brain, eyes), cluster information represented as vertices, embed the graph and compare structural change or classify vectors [30] [31]; (iii) object recognition: volume in images represented as trees to find optimal subgraphs for the task [32]; (iv) image captioning: implicitly model the relationship among regions of interest [33] [34]; and (v) image recognition: propose hypergraph labels on pixel-level for knowledge graph classification [35].

Most methods in this category use graphs to enrich the representation with connections between different media elements, guide networks' attention, and structure irregular information like in superpixels. Their goal is to improve the performance on the task, but it usually requires prior knowledge of the data to model the graph connections. This modeling is not always evident, as pointed out by Yuxin Chen et al. (2020) and Si et al. (2020). Also, inserting the graph information into the learning processes adds new challenges, such as regularization in Ji et al. (2020), and measures strong responses to minor structural changes mentioned in J. Huang et al. (2020).

Multi-modal association

Methods that use the graph structure to combine information from multiple domains besides the original media, such as text, sounds, or contextual attributes. By media and task, they could be outlined them as follows:

Videos: all studied proposals targeted video recommendation systems, which consider the interactions between users and items and the item contents from various

30. Huang Jiashuang et al. (2020). *A novel node-level structure embedding and alignment representation of structural networks for brain disease analysis.*
31. Sakaguchi Aiki, Wu Renjie, and Kamata Sei-ichiro (2019). *Fundus image classification for diabetic retinopathy using disease severity grading.*
32. Selvan Raghavendra et al. (2020). *Graph refinement based airway extraction using mean-field networks and graph neural networks.*
33. Wang Junbo et al. (2020). *Learning visual relationship and context-aware attention for image captioning.*
34. Fu Sichao, Yang Xinghao, and Liu Weifeng (2018). *The comparison of different graph convolutional neural networks for image recognition.*
35. Shi Lei et al. (2019). *Skeleton-based action recognition with directed graph neural networks.*

modalities (*e.g.,*visual, acoustic, and textual) [36] [37] [38] [39].

Images: authors propose to use graphs to represent different domains of information on vertices and use the edges to model the proximity between them by quantification or messaging passing between interactions. Specifically: (i) object matching: compatibility between two objects based on their visual features, as well as their contexts (social attitudes, time, and place) [40]; (ii) image captioning: uses graphs to formulate more complex, non-sequential dependencies among proposals of image regions and phrases [41]; (iii) disease prediction: vertices represent patients or healthy controls accompanied by a set of features, while the graph edges incorporate associations between subjects containing imaging and non-imaging information [42]; and (iv) object tagging: formulate item tagging as a link prediction problem between item vertices and tag vertices. [43].

Contents such as pixels of an image and user's preferences or video sequences and interactions are different in their form and semantic meaning. Traditional ways to codify high-level information (such as interests, preferences, and interactions) as sequences of words or values ignore semantics and the relationship to the original data, as pointed out by Jinguang Wang et al. (2020). Therefore, the graphs in this category are vital in connecting heterogeneous information that can be used and interpreted in a learning framework.

36. Wang Jinguang et al. (2020). *Multimodal graph convolutional networks for high quality content recognition.*
37. Wei Yinwei et al. (2019). *MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video.*
38. Gong Jibing et al. (2020). *Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view.*
39. Wu Le et al. (2020). *Learning to transfer graph embeddings for inductive graph based recommendation.*
40. Cucurull Guillem, Taslakian Perouz, and Vazquez David (2019). *Context-aware visual compatibility prediction.*
41. Bajaj Mohit, Wang Lanjun, and Sigal Leonid (2019). *G3raphGround: graph-based language grounding.*
42. Parisot Sarah et al. (2018). *Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer's disease.*
43. Mao Kelong et al. (2020). *Item tagging for information retrieval: a tripartite graph neural network based approach.*

Higher dimensions

In this category, the methods that use the graphs to incorporate an additional dimension to the conventional data to solve an enhanced task on the traditional formulation. they could be outlined them as follows:

Videos: graphs combine features extracted from multiple frames in different camera footage or different videos for the tasks of: (i) cross-view fusion [44], in which frame descriptions from multi-view cameras are aggregated and updated through a graph network; and (ii) multi-video summarization [45], in which a graph network measures the importance and relevance of each video shot in its video and the whole video collection.

Images: graphs combine image features on the pixel level for geometric positioning in the global scenario. More specifically, for: (i) RGBD semantic segmentation: each node in the graph corresponds to a set of points on the depth dimension and is associated with features extracted from 2D images [46]; (ii) point cloud recognition: 3D space coded into voxels and graph networks to represent 3D points' position for each voxel [47] [48] [49].

Overall, the graphs in this category gather multiple points in different spaces in an ordered manner. The networks can measure and refine affinities between the components through the connections modeled in the graphs. Connections that are otherwise ignored or subjected to discretization or merging errors, as pointed out by He, Q. Liu, and Y. Yang (2020) and S. Chen et al. (2019).

Enhanced tasks

The methods in this category use graphs to take the concepts associated with a traditional task and expand them into a more complex goal through modeling connections.

44. He Xin, Liu Qiong, and Yang You (2020). *MV-GNN: multi-view graph neural network for compression artifacts reduction.*
45. Wu Jiaxin, Zhong Sheng-hua, and Liu Yan (2020). *Dynamic graph convolutional network for multi-video summarization.*
46. Qi Xiaojuan et al. (2017). *3D graph neural networks for RGBD semantic segmentation.*
47. Bouritsas Giorgos et al. (2019). *Neural 3D morphable models: spiral convolutional networks for 3D shape representation learning and generation.*
48. Li Zongmin et al. (2019). *Graph attention neural networks for point cloud recognition.*
49. Chen Siheng et al. (2019). *PCT: large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving.*

For instance, instead of just recognizing a person in a scene, one could identify them every time they re-appear in another location. Alternatively, train a network for classification and expand the inference for classes never seen. By media and task, they could be outlined them as follows:

Videos: (i) multiple object tracking and action recognition: data association problems enhance their models with the graph relationship information between numerous objects [50] [51] [52]; (ii) zero/few-shot learning: classification task to unseen objects or object co-segmentation task uses the relationship between objects or regions represented as graphs to propagate information [53] [54] [55]; and (iii) person re-identification: person re-identification: local features and iterative feature affinity connections to construct graphs used to identify a person and locate its reappearance [56].

Images: (i) zero/few-shot learning and change detection: classification to unseen image classes, correlating images features or clusters to iteratively updates edges weights for final prediction [57] [58] [59]; (ii) feature characterization and matching: hand-engineered graph representation from evident visual structure to find patterns or to transform the position into local features [60] [61] [62] [63]; (iii) similarity reasoning: relational reasoning of extracted features in networks with a kinship graph of features [64];

50. Schulter Samuel et al. (2017). *Deep network flow for multi-object tracking.*
51. Ma Cong et al. (2019). *Deep Association: end-to-end graph-based learning for multiple object tracking with conv-graph neural network.*
52. Herzig Roei et al. (2019). *Spatio-temporal action graph networks.*
53. Gao Junyu and Xu Changsheng (2020). *CI-GNN: building a category-instance graph for zero-shot video classification.*
54. Wang Wenguan et al. (2019). *Zero-shot video object segmentation via attentive graph neural networks.*
55. Gao Junyu, Zhang Tianzhu, and Xu Changsheng (2021). *Learning to model relationships for zero-shot video classification.*
56. Wu Yiming et al. (2020). *Adaptive graph representation learning for video person re-identification.*
57. Kim Jongmin et al. (2019). *Edge-labeling graph neural network for few-shot learning.*
58. Liu Hongying et al. (2019). *A novel deep framework for change detection of multi-source heterogeneous images.*
59. Gidaris Spyros and Komodakis Nikos (2019). *Generating classification weights with GNN denoising autoencoders for few-shot learning.*
60. Zhang Zhen and Lee Wee Sun (2019). *Deep graphical feature learning for the feature matching problem.*
61. Sidorov Oleksii and Hardeberg Jon Yngve (2019). *Craquelure as a graph: application of image processing and graph neural networks to the description of fracture patterns.*
62. Shin Seung Yeon et al. (2019). *Deep vessel segmentation by learning graphical connectivity.*
63. Jimenez-Sanchez Daniel, Ariz Mikel, and Ortiz-de-Solorzano Carlos (2020). *Unsupervised learning of contextual information in multiplex immunofluorescence tissue cytometry.*
64. Li Wanhua et al. (2020). *Graph-based kinship reasoning network.*

and (iv) person re-identification and group identification: expand the identification task to multiple shots or groups of people by representing individual features as vertices and edges, modeling similarities between them [65] [66].

In general, the graphs in this category do not represent the media. Instead, the vertices represent a set of features, a cluster of data, or object concepts. Like in the category of the traditional task that requires prior knowledge to model the data, it is even more significant in this category. The modeling here often relates the information to a particular position or a link to other known classes. The considerable advantage gained with the graphs' use is that instead of performing multiple tasks independently and consecutively, they can all be encapsulated and improve the inferences.

Semantic tasks

So far, the categories covered used the graphs to integrate high-level concepts or enhance high-level data comprehension. Now it presents methods that formulate their problems in a complex semantic task. However, similarly, they use graphs to improve the traditional representation and link the data to higher concepts. By media and task, they could be outlined them as follows:

Videos: (i) scene parsing: explore objects interactions by modeling nodes as detected objects and discriminative paths with class activation maps for connections [67]; (ii) subtle visual communication: spatio-temporal graph neural network to explicitly represent interactions in social scenes to infer gaze communications [68]; and (iii) reasoning: a scene graph is built on top of segmented object instances within and across video frames to predict pedestrians' intent [69].

65. Li Yaoyu et al. (2019). *Adaptive feature fusion via graph neural network for person re-identification.*
66. Huang Ziling et al. (2019). *DoT-GNN: domain-transferred graph neural network for group re-identification.*
67. Luo Wu et al. (2019). *Improving action recognition with the graph-neural-network-based interaction reasoning.*
68. Fan Lifeng et al. (2019). *Understanding human gaze communication by spatio-temporal graph reasoning.*
69. Liu Bingbin et al. (2020). *Spatiotemporal relationship reasoning for pedestrian intent prediction.*

Images: (i) reasoning [70] [71], scene understanding [72] [73] [74] [75], emotion recognition [76] [77] and visual question answering [78] [79] [80]: relate identified objects with context, regions, and interactions with other objects or components. The graph representations are usually straightforward based on information and objects, and researchers applying these graphs to graph networks obtain good results; and (ii) semantic segmentation: image represented by a graph, which nodes contain different feature maps for richer representation and edges reflecting relationships of the nodes [81]

Luo et al. (2019) stated that ignoring the interactions between components fails the scene understanding task, making graphs crucial. This statement holds for many of the tasks in this category. Particularly for modeling complex relationships with subtle cues such as emotion and visual expression in Singh et al. (2019) and L. Fan et al. (2019) or relating agents and effects as in Chuang et al. (2018) or even intention and position as in B. Liu et al. (2020). The rigid grid on the media is usually not enough for the task, and the graphs can provide a more flexible structural layout.

Data augmentation

The methods in this category aim to solve a well-known problem with deep learning: its performance is conditional to large amounts of data, and there is not enough annotated data. Consequently, the proposals here create more annotated data by assigning labels of known data to others closely related in the graphs. Specifically: (i) connect and analyze the similarity between images for datasets with few labels or few images to expand

70. Chuang Ching-Yao et al. (2018). *Learning to act properly: predicting and explaining affordances from images.*
71. Yang Guang et al. (2020). *Graph-based neural networks for explainable image privacy inference.*
72. Suhail Mohammed and Sigal Leonid (2019). *Mixture-kernel graph attention network for situation recognition.*
73. Chen Gongwei et al. (2020). *Scene recognition with prototype-agnostic scene layout.*
74. Jing Ya et al. (2020). *Relational graph neural network for situation recognition.*
75. Li Ruiyu et al. (2017). *Situation recognition with graph neural networks.*
76. Shao Jingzhi et al. (2019). *Emotion recognition by edge-weighted hypergraph neural network.*
77. Guo Xin et al. (2020). *Graph neural networks for image understanding based on multiple cues: group emotion recognition and event recognition as use cases.*
78. Singh Ajeet Kumar et al. (2019). *From strings to things: knowledge-enabled VQA model that can read and reason.*
79. Yu Jing et al. (2020). *Cross-modal knowledge reasoning for knowledge-based visual question answering.*
80. Sawatzky Johann et al. (2019). *What object should I use?: task driven object detection.*
81. Lu Yi et al. (2021). *CNN-G: convolutional neural network combined with graph for image segmentation with theoretical analysis.*

data [82] [83]; (ii) group different datasets on the same domain [84]; and (iii) to map image labels to embeddings or features that would serve as labels rather than the traditional class labels [85] [86]. Overall, their strategies benefit from the graphs' capability to relate entities and, consequently, help improve the performance of deep networks in many tasks.

Final considerations

Deep learning and graphs help solve multiple tasks, especially those that involve high-level reasoning, complex iterations, and semantic meaning. The graphs provide ways to integrate dimensions, features, and heterogeneous data, model proximity between multiple components, and track relationships through media's space and time. Some task completion seems achievable only with graphs because their complex relationships are not evident in a rigid grid, while others are enhanced. Finally, graphs even help advance the deep learning study field itself by helping create annotated data. As a downside, modeling the graphs usually requires a profound prior knowledge of the data, and inserting the graph information into the learning processes adds new challenges regarding dimensions, form, and regularization.

3.1.3 Random Forest on graphs

First proposed by Breiman [87], the Random Forest (RF) is a fast, simple and scalable machine learning algorithm for classification and regression [88]. Random Forests has proved successful in many applications and is referred to as a general-purpose learning algorithm [89]. RF is a non-parametric ensemble method for supervised classification and regression, consisting of randomized independent trees. It relies on randomizing selected data and features and has extensive practical uses in many domains.

Although the RFs are empirically successful in suppressing noises, the statistical and

82. Acuna David et al. (2018). *Efficient interactive annotation of segmentation datasets with polygon-RNN++.*
83. Schroeder Brigit, Tripathi Subarna, and Tang Hanlin (2019). *Triplet-aware scene graph embeddings.*
84. Renton Guillaume et al. (2019). *Graph neural network for symbol detection on document images.*
85. Li Jinghui and Fang Peiyu (2019). *FVGNN: a novel GNN to finger vein recognition from limited training data.*
86. Shao Huikai and Zhong Dexing (2019). *Few-shot palmprint recognition via graph neural networks.*
87. Breiman Leo (2001). *Random forests.*
88. Cutler Adele, Cutler D. Richard, and Stevens John R. (2012). *Random forests.*
89. Biau Gérard and Scornet Erwan (2016). *A random forest guided tour.*

mathematical properties of the procedure are still obscure [90] [91]. Some authors [92] [93] believe that randomness performs as an implicit regularization process, promoting consistency and noise suppression. In the presence of complex signals permeated with noise, RFs behave as interpolating classifiers that encourage large consistent regions and reduce the effect of noise. Also, most authors agree that most unwanted behavior occurs when the input data is highly correlated. Scornet (2016) is one of the theoretical pieces aiming to explore the mathematical properties of the method, and it provides an interesting parallel between RFs and kernel methods.

RF is somewhat accepted on graphs as a viable method to perform tasks such as learning features, mining data, predicting labels and connections, and measuring similarity. Particularly in recent years, due to its good results and fast processing. It is possible to find methods integrating graphs and RF in many areas, such as image [94], text [95] [96], and natural language processing [97], as in social network analysis [98] [99] [100] and medical applications [101] [102] [103](*non-exhaustive citation*). Most research results on terms relating to RF and graphs are associated with medical applications and social network graphs. These methods profit from large graph embeddings and deep graph features combined with RF predictions. Only possible due to the graphs being not densely connected—although the high-dimensional feature does not impose a limitation for the RFs method, it can be a

90. Biau Gérard and Scornet Erwan (2016). *A random forest guided tour.*
91. Scornet Erwan (2016). *Random forests and kernel methods.*
92. Wyner Abraham et al. (2017). *Explaining the success of adaboost and random forests as interpolating classifiers.*
93. Gérard Biau, Devroye Luc, and Lugosi Gábor (2008). *Consistency of random forests and other averaging classifiers.*
94. Temir Askhat, Artykbayev Kamalkhan, and Demirci M. Fatih (2020). *Image classification by distortion-free graph embedding and KNN-random forest.*
95. Ayotte Blaine et al. (2020). *Fast free-text authentication via instance-based keystroke dynamics.*
96. Awajan Arafat (2015). *Keyword extraction from arabic documents using term equivalence classes.*
97. Gao Yali et al. (2018). *Graph mining-based trust evaluation mechanism with multidimensional features for large-scale heterogeneous threat intelligence.*
98. Pachaury Shruti et al. (2018). *Link prediction method using topological features and ensemble model.*
99. Ma Jiangtao et al. (2018). *De-anonymizing social networks with random forest classifier.*
100. Sharad Kumar and Danezis George (2014). *An automated social graph de-anonymization technique.*
101. Pouyan Maziyar Baran and Nourani Mehrdad (2017). *Clustering single-cell expression data using random forest graphs.*
102. Priya Michael Mary Adline and Jawhar Joseph (2020). *Advanced lung cancer classification approach adopting modified graph clustering and whale optimisation-based feature selection technique accompanied by a hybrid ensemble classifier.*
103. Zhou Shuang et al. (2019). *LncRNA-miRNA interaction prediction from the heterogeneous network through graph embedding ensemble learning.*

challenge for the embedding process.

As in many machine learning algorithms, RF requires a systematic input, and a strategy must be placed to deal with the arbitrary structure of the graphs. Besides the embeddings mentioned above [104] [105] [106] [107] [108] [109], authors also propose to use the adjacency matrix [110], network topology measures (*e.g.,*centrality, community, degrees) [111] [112], selection of graph attributes [113] [114], pairwise comparison of vertex or edges [115], and graph feature inference methods [116]. In Karunaratne and Boström (2009), the authors studied different forms to adapt the graph structure for RF without losing too much information and concluded that the choice has a significant impact on the performance and that keeping as much information about the structure is beneficial.

Some authors focus their interest on a modified RF to better operate on graphs. In Guillame-Bert and Dubrawski 2017, for instance, they use a constrain value during the split that is itself the determinant value of a relation between two vertices. Likewise, in Liang and D. Huang 2019, the importance of different features on the graph is weighted and later used to bias the sampling to give a bigger chance to sample more critical elements.

In summary, RFs use graphs from diverse data types, although media is not the most

104. Temir Askhat, Artykbayev Kamalkhan, and Demirci M. Fatih (2020). *Image classification by distortion-free graph embedding and KNN-random forest.*
105. Wang Xiaochan et al. (2019). *Predicting gene-disease associations from the heterogeneous network using graph embedding.*
106. Ou Mingdong et al. (2016). *Asymmetric transitivity preserving graph embedding.*
107. Demirci M. Fatih and Kacka Serdar (2016). *Object recognition by distortion-free graph embedding and random forest.*
108. Zhou Shuang et al. (2019). *LncRNA-miRNA interaction prediction from the heterogeneous network through graph embedding ensemble learning.*
109. Karunaratne Thashmee, Bostrom Henrik, and Norinder Ulf (2010). *Pre-processing structured data for standard machine learning algorithms by supervised graph propositionalization: a case study with medicinal chemistry datasets.*
110. Liang Jianheng and Huang Dong (2019). *Laplacian-weighted random forest for high-dimensional data classification.*
111. Ayotte Blaine et al. (2020). *Fast free-text authentication via instance-based keystroke dynamics.*
112. Sharad Kumar and Danezis George (2014). *An automated social graph de-anonymization technique.*
113. Pachaury Shruti et al. (2018). *Link prediction method using topological features and ensemble model.*
114. Kim Jaehwan and Lee Junsuk (2020). *Adaptive directional walks for pose estimation from single body depths.*
115. Guillame-Bert Mathieu and Dubrawski Artur (2017). *Classification of time sequences using graphs of temporal constraints.*
116. Gao Yali et al. (2018). *Graph mining-based trust evaluation mechanism with multidimensional features for large-scale heterogeneous threat intelligence.*

prominent source. However, since the source becomes irrelevant to the RF learning steps once modeled as graphs, the crucial decision is the strategy to transpose the graph to the regular representation required. Graph embeddings and network topology measures are the most frequent strategy. However, executing the former imposes limitations on the graph size, and the latter produces highly correlated values from graphs sourced from gridded data.

3.2 Discussion on graphs, media, and learning

The great incentive to center the considerations towards graph processing is that they are critical for hierarchical analyses, as a model operating on generic graphs can be later generalized to hierarchical structures. Also, machine learning operating on graphs provides a form to create an agnostic model regarding the media since the data's source becomes virtually irrelevant once the system works in terms of vertices and edges. As seen, machine learning on graphs is a topic of great interest.

In a graph representing digital media with arbitrary dimensions, the vertices may correspond to the media's units, such as pixels, voxels, or data points. This approach usually results in large sets of vertices but favors back-and-forth operations. Alternatively, the vertices could correspond to objects inferred from the data, such as superpixels, partitions, and surfaces, creating a more concise representation but requiring complex mappings dependent on the grouping strategy.

Graph embedding methods are suitable for creating a systematic representation of the graphs that allow their utilization in multiple learning frameworks. But embeddings are very expensive in terms of computational resources and are prohibitive for large graphs.

Deep learning methods on graphs are a contemporary solution to many tasks, primarily semantic and high-level analysis. Despite their improvements in inferring information, they impose limitations regarding the underlying graph and the modeling choices. A prevalent approach when handling deep learning on graphs in a multimedia context is to favor modeling the concepts and abstractions rather than the raw media data and placing careful designing decisions such as sampling and randomization.

Finally, Random Forest on graphs provides solutions for many computational problems, particularly medical applications and social network analysis. The most significant limitation of aggregating graphs and Random Forests is the systematic input required by the model. Careful graph parsing must take place, considering the type of graph, its

proximity to the original data, and the expected results.

Chapter 4

Case study: Learning on graphs

This chapter presents the case study for a learning framework operating on a selection of graph attributes aggregated with the Random Forest model. Motivation goes beyond a good performance in an application; it relies on a proposal of a machine learning framework working on graphs that could later be exploited for the hierarchical structures.

The main challenge in this framework concerns the regular representation required by most machine learning algorithms, which is inherently opposed to the unconstrained nature of graphs. Nevertheless, conforming to it allows processing graphs with general learning methods and avoids the long computations of graph networks operating on raw data and embedding strategies. Furthermore, the framework must also consider the high-dimensional space usually presented with graphs representing digital media and the label attribution strategy that should not impose assumptions on the data source to later generalize to other tasks.

The discussions about graph creation and manipulation can be made generic enough to model many media [1]. However, the proposals of the case study are for image graphs. Dealing with graphs created from images has a unique modeling space. The spatial connectivity gives a structured representation of a grid graph: close to the spatial domain and strengthened by the relational aspects of the graphs. Cognizant of this unique space, this chapter investigates the relational aspects while pondering the particular conditions for image processing. It evaluates the impact of the characteristics on the results obtained on two different tasks and discourses on elements that could not be generalized.

The study proposes to use edge-weighted graphs aggregated with the Random Forest (RF) [2]. The edge-weighted graph acts as a transformation filter based on local differences [3]. And, as in the case of many spatial filters based on local differences, it tends

1. Bertrand Gilles et al. (2013). *Mathematical morphology: from theory to applications.*
2. Breiman Leo (2001). *Random forests.*
3. Elmoataz A., Lezoray O., and Bougleux S. (2008). *Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing.*

to respond strongly to noise [4]. It is expected that the RF attribute selection and implicit regularization process [5] can mitigate this aspect while reinforcing desirable characteristics. Also, the RF mechanics allows it to work with high-dimensional data, making it a fast, simple, and scalable method [6] for the investigation. Finally, it proposes to describe the graphs at the vertice's level, which allows training the model on the discrete space by associating each entry with a single label.

This chapter has three parts. The first part describes the study's methodological components and comprises Sections 4.1, 4.2, and 4.3. More precisely, Section 4.1 formalizes the edge-weighted image graphs, Section 4.2 discusses the strategy to create the regular representation based on a selection of graph attributes, and Section 4.3 describes the RF mechanics and the application in the proposed framework. The second part contains the experimental investigation, where Section 4.4 describes the investigative steps and the methodology pipeline. It presents the inquiry in three progressive stages:

1. Section 4.4.1 assesses the selection of the attributes of the graph;
2. Section 4.4.2 applies the results as image gradients in the segmentation task; and
3. Section 4.4.3 addresses some identified limitations and extends the formalism.

Finally, Section 4.5 makes the final part, which presents a discussion considering the different aspects of the experimental investigation, summarizes the observed properties, and draws some conclusions to guide the hierarchical study.

4.1 Image graphs

For the graph G defined on the image domain, the adjacency relation Γ between the pixels is typically a structured adjacency relation, such as $4-$ or $8-$adjacency in a grid form. **Neighborhood** denotes a collection of adjacent vertices and usually is taken clockwise, starting by the north of a pixel and following the adjacency relation. The set of vertices $V = \{v_1, v_2, \ldots, v_N\}$ represents the N pixels of the image. The collection of functions associated with each vertex is denoted by $f : V \subset \mathrm{Z}^2 \to \mathbb{R}$. Common functions in f include low-level descriptors, variations in the color space, or the gray-scale magnitudes. The grayscale magnitude function, denoted by f_{gray}, plays an essential role in the image

4. Fogel I. and Sagi D. (1989). *Gabor filters as texture discriminator.*
5. Wyner Abraham et al. (2017). *Explaining the success of adaboost and random forests as interpolating classifiers.*
6. Cutler Adele, Cutler D. Richard, and Stevens John R. (2012). *Random forests.*

graph as the most common source to calculate the weighting function $\mathcal{F}$ in the edge-weighted graph $G(V, \mathcal{F})$.

Edge weighting functions ideally should characterize dissimilarities. Therefore, distance functions are more suitable, where the Euclidean distance is the most common, defined in E as:

$$\mathcal{F}_{\mathbf{euc}}(u, v) = \sqrt{(f_{gray}(u) - f_{gray}(v))^2}, \forall u \in \Gamma(v) \tag{4.1}$$

In representing dissimilarities, the edge weights may characterize the local variation around a vertex and serve as an image gradient operator bounded by the adjacency relation. Weighting edges as an image gradient operator acts as a transformation filter on the image, creating a transformed space by changing the contrast of the original image and spreading the intensity levels [7].

Definition 12: Graph gradient operator

The **graph gradient operator** for edge-weighted graph $G(V, \mathcal{F})$ at vertex u could be defined as:

$$\nabla_{\mathcal{F}} f(u) = (\partial_{v_1} f(u), \ldots, \partial_{v_i} f(u)), \forall v_i \in \Gamma(u) \tag{4.2}$$

where $\partial_v f(u)$ is the edge derivative of f at a vertex $u \in V$ along the edge $e = (u, v) \in E$:

$$\partial_v f(u) = \left.\frac{\partial f}{\partial e}\right|_u = \mathcal{F}(u, v) \tag{4.3}$$

The topology choices, such as the adjacency relation and the weighting function's properties, condition the interaction between the image data and the preserved characteristics on the edge-weighted graph.

4.2 Regular representation of graph's attributes

The case study proposes to use the information on the edges and vertices to represent the graph in a learning framework. The main challenge concerns the strategy to parse the data and create a regular input required by most learning algorithms without losing too much information.

The proposed strategy depicts each vertex of the edge-weighted image graph as a vector of selected attributes. The selection belongs to two categories of attributes: (i) **vertex**

7. Elmoataz A., Lezoray O., and Bougleux S. (2008). *Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing.*

attributes ($\mathbf{X}_V$), representing the vertices functions; and (ii) **edge weights** ($\mathbf{X}_\mathcal{F}$), representing the weight values in every edge on the adjacency of v. Thus, the vertex is a vector $\mathbf{X}_v$ with dimension $p = |\mathbf{G_{att}}|$, where $\mathbf{G_{att}}$ is the set $\mathbf{G_{att}} = \{\mathbf{X}_V \mathbf{X}_\mathcal{F}\}$.

Definition 13: Regular representation of the edge-weighted image graph

The proposed **regular representation** of a edge-weighted image graph $G = (V, \mathcal{F})$ is $\mathcal{X}_G = ((\mathbf{X}_1, \mathbf{Y}_1), \ldots, (\mathbf{X}_{|V|}, \mathbf{Y}_{|V|}))$, where $|V|$ is the number of vertices each represented as a vector $\mathbf{X}_v \in \mathbb{R}^p$ and a single label $\mathbf{Y}_v$.

Repeating the procedure for all graphs in the training set and concatenating the $\mathcal{X}_G$ outputs makes a regular training input $\mathcal{D}$ for the learning framework, where $\mathcal{D} = ((\mathbf{X}_1, \mathbf{Y}_1), \ldots, (\mathbf{X}_T, \mathbf{Y}_T))$ and T is the total number of vertices in the training set. For the test set, the procedure takes the regular representation of each graph in the validation/test set and individually subjects them to the estimations.

4.3 Random Forest as regularizers

The Random Forest predictor described in Breiman (2001) is a non-parametric machine learning method for classification and regression. And although there are many variations on the framework to create random trees and random forests [8], the original algorithm is known for its successful performance in multiple tasks [9].

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 [10] and noise suppression [11]. The RF acting as a regularizer on the spatial filters can diminish the noise, mitigate any eventual poor topology choice and accentuate strong connections.

Following the notations in Biau and Scornet (2016), the RF predictor consists of M randomized trees. In each internal node m of a tree in the forest, there is a split function $h(\mathbf{x}, \theta_m)$ for a query point $\mathbf{x}$ with parameters θ_m. During training, the parameters θ_m are learned, usually by maximizing the information gain (in classification) or minimizing the

8. Biau Gérard and Scornet Erwan (2016). *A random forest guided tour.*
9. Cutler Adele, Cutler D. Richard, and Stevens John R. (2012). *Random forests.*
10. Gérard Biau, Devroye Luc, and Lugosi Gábor (2008). *Consistency of random forests and other averaging classifiers.*
11. Wyner Abraham et al. (2017). *Explaining the success of adaboost and random forests as interpolating classifiers.*

(a) Input image (b) Graph gradient operator (c) RF output

Figure 4.1: Illustration of the graph gradient operator and the Random Forest(RF) regularization effect. It presents a challenging input image with transparent materials, a body of water, object occlusion, and objects with similar colors. The graph gradient operator is an image projection of the graph. The RF output is the predictions trained on the edge detection for the graph and mapped back to the image domain.

mean square error (in regression) to split the data samples covered by m into two subsets with the maximum proportion of instances belonging to the same label. In the test phase, it applies an unseen set of data to h at each split node, and the test result determines the path the data will perform until it reaches a terminal node with the label prediction.

Dollar and Zitnick (2015) [12] proposed a method for structured edge detection (SED) that was fast and precise in predicting object edges in an image. The SED method extends the RF formalism to the general structured output space using local segmentation masks on cropped patches in the image feature space and the ground truth. The core idea was to map similar structured labels in a given node to the same discrete label. But because the similarity in structured space is complex, the authors proposed a reduced intermediary space instead of calculating the continuous variance of entropy on the nodes.

In contrast, this study proposes using image graphs as RF inputs, where instead of using the complex structured output space, it creates a structured input on the standard discrete label. In that, it defines: (i) the neighborhoods inside the graph delineate the regions of analysis; (ii) the graph attributes define the feature space; and (iii) the discrete label attribution for the edge detection task is centered on the graph's vertices. Therefore a single label assigns the entirety of a neighborhood. At inference, it uses the RF as a regression estimator averaging all predicted values of the M trees. It then maps the RF predictions back to the image space in the form of image gradients, where they can be evaluated qualitatively and quantitatively.

Fig. 4.1 illustrates the regularization effect of the RF. It shows the input image, the

12. Dollar Piotr and Zitnick C. Lawrence (2015). *Fast edge detection using structured forests.*

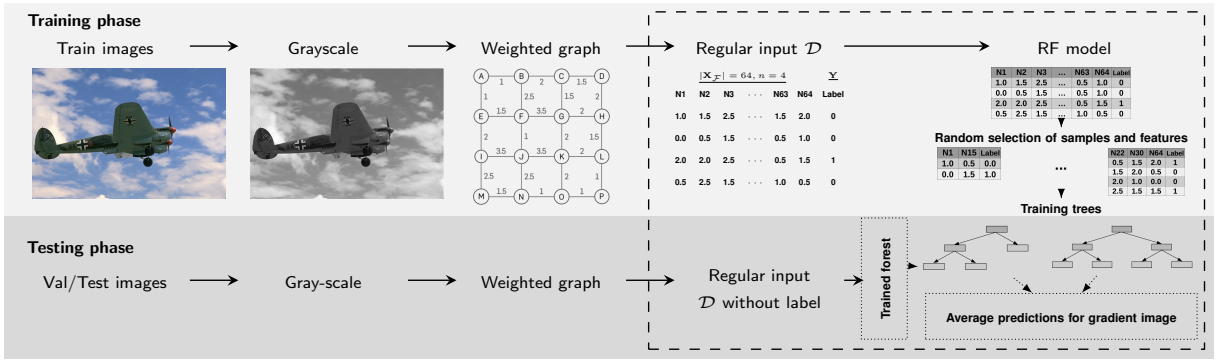


Figure 4.2: Figure illustrating the framework from the input image to the Random forest predictions computing the gradients. First, it transforms each image to the gray-scale magnitudes. Then, it calculates the weights for each image pixel as a grid graph, here illustrated with the $4-$adjacency relation. The next step transforms the graph structure to a regular representation with the selected attributes 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 edge detection ground truth. During the test, the Random Forest subject each vertex of the test graphs to prediction, where the estimated values are mapped back to the image coordinates as an intensity value for evaluation.

Khalimsky grid projection [13] of the weighted graph, and the RF predictions mapped back to the image domain. As shown, the RF predictions capture and reinforce the main characteristics modeled in the graph and remove isolated features counted as noise.

4.4 Investigative steps

This study analyses the viability of the regular representation of the graph's attributes in a learning pipeline through experimentation. Fig. 4.2 illustrates with simplified examples the pipeline for the proposed framework. It uses the Berkeley Segmentation Dataset and Benchmark (BSDS500) [14], which offers edge detection and segmentation labels.

For the task, it trains the RF on the edge detection task, where the test labels $\mathbf{Y} \in \{0, 1\}$, and all entries in $\mathcal{D}$ have a unique discrete label on the vertices. In the test step, it makes the regular representation $\mathcal{X}_G$ for each graph in the validation/test set, omitting the label, and individually subjects them to the estimations of the RF. The final estimated values are mapped back to the image coordinates as gradients.

13. Khalimsky Efim, Kopperman Ralph, and Meyer Paul R. (1990). *Computer graphics and connected topologies on finite ordered sets.*
14. Martin D. et al. (2001). *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.*

The experimental evaluation is a series of investigative steps. At each step there are four stages:

1. create the edge-weighted graph gradient operator from the input image (Section 4.1);
2. create the regular representation of the graph (Section 4.2);
3. train the RF on the edge detection task to obtain the gradients (Section 4.3); and
4. evaluate the quality of the gradients on image tasks.

The first investigative step assesses the choices of graph attributes, and Section 4.4.1 presents a qualitative analysis examining the resulting image gradient from each selection. The second step, in Section 4.4.2, evaluates the quality of the image gradients created from the best set of graph attributes on the segmentation task applying the gradients as input to a segmentation algorithm. The third and last step, in Section 4.4.3, extends the formalism to exploit better the relationships modeled by the graphs, mainly focusing on the RF mechanics and limitations. And besides the extended formalism, it adds evaluations on edge detection and comparisons with deep learning approaches on edge detection and region segmentation.

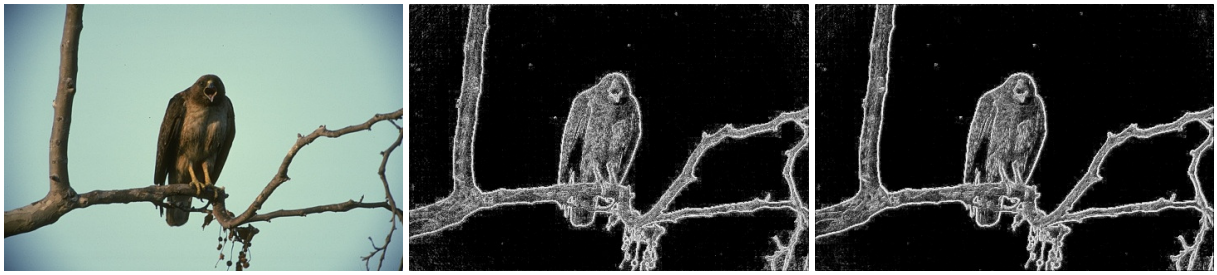
4.4.1 Inspecting the graph's attributes

This section contains the investigative step assessing the attributes choices through a qualitative analysis of the resulting gradients. For the qualitative analysis, the images presented are the result of the estimated values for the images on the validation set, predicted by the trained RF with $M = 150$.

This step investigates the following characteristics:

1. the adjacency relation;
2. the weighting function;
3. the neighboring size; and
4. the vertex attributes.

The neighboring size is entirely related to the graph's properties, therefore, detached from the media source. The adjacency relation and the weighting function refer to the modeling choices, which are often concerned with graphs (see Chapter 3). From the standpoint of the framework, it is indifferent to the modeling choices and bears upon only the values and connections. But from the task perspective, the choices condition the interaction between the data and the graph, impacting the results. The last aspect, the vertex



(a) Input image **(b)** 4-adjacency **(c)** 8-adjacency

Figure 4.3: Illustration of the adjacency relation effect on the final gradient.

attributes, is the most critical considering an agnostic framework. Likewise, from the framework perspective, it is just a set of values stored on the vertices to be or not considered during execution. However, conceptually, it is a direct reference to the media properties, particularly if the function represents image descriptors.

Adjacency relation

The first assessment is the adjacency relation in 4−adjacency or 8−adjacency. As illustrated in Fig. 4.3, the predicted images have subtle differences, but the ones created from the 8−adjacency are less noisy overall, and the objects' borders and regions are clearer and more defined. These aspects indicate that more connections make representations bounded on mutual characteristics, can assist the learning process, and enhance desirable features. **In the following, it will thus only consider** 8−**adjacency.**

Weighting function

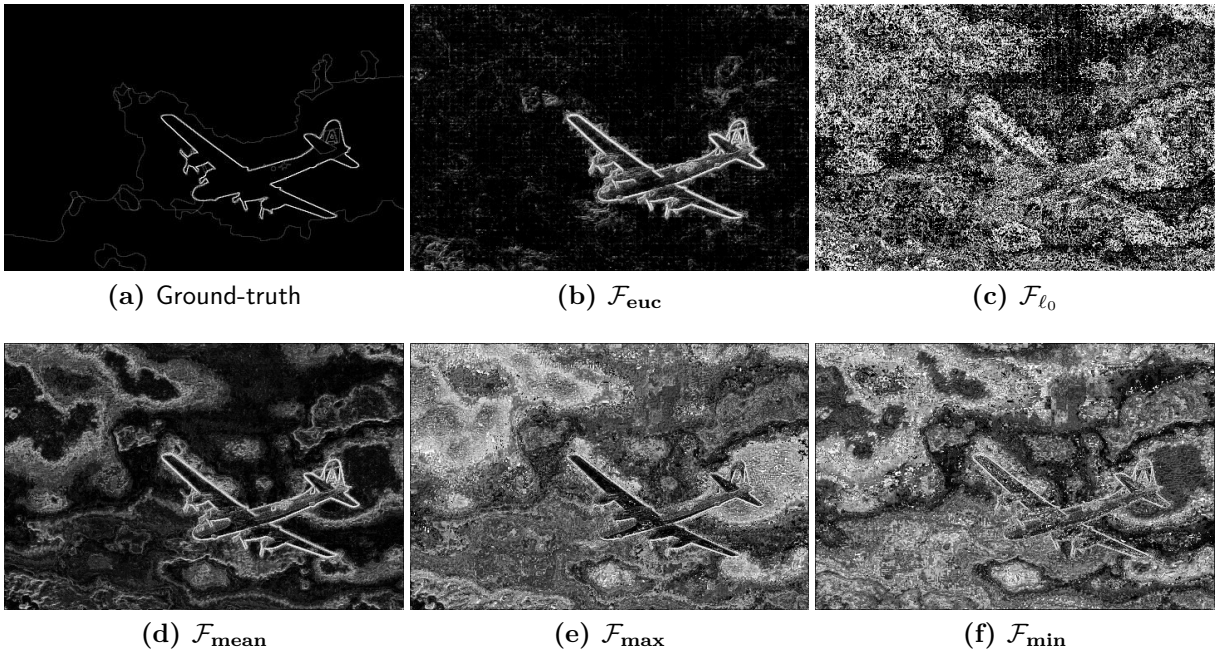
Another assessment is the choice of the weighting function on the edges of the graph. Besides the $\mathcal{F}_{\mathbf{euc}}$ in Equation 4.1, standard weighting functions provided by many popular tools to create and manipulate graphs [15] [16] includes:

$$\mathcal{F}_{\mathbf{max}}(u,v) = \mathbf{max}\{f_{gray}(u), f_{gray}(v)\}, \forall u \in \Gamma(v) \tag{4.4}$$

$$\mathcal{F}_{\mathbf{min}}(u,v) = \mathbf{min}\{f_{gray}(u), f_{gray}(v)\}, \forall u \in \Gamma(v) \tag{4.5}$$

$$\mathcal{F}_{\mathbf{mean}}(u,v) = \mathbf{mean} = (f_{gray}(u) + f_{gray}(v))/2, \forall u \in \Gamma(v) \tag{4.6}$$

15. Perret B. et al. (2019). *Higra: hierarchical graph analysis.*
16. *MATLAB version 7.10.0 (R2010a)* (2010).



(a) Ground-truth (b) $\mathcal{F}_{\mathbf{euc}}$ (c) $\mathcal{F}_{\ell_0}$

(d) $\mathcal{F}_{\mathbf{mean}}$ (e) $\mathcal{F}_{\mathbf{max}}$ (f) $\mathcal{F}_{\mathbf{min}}$

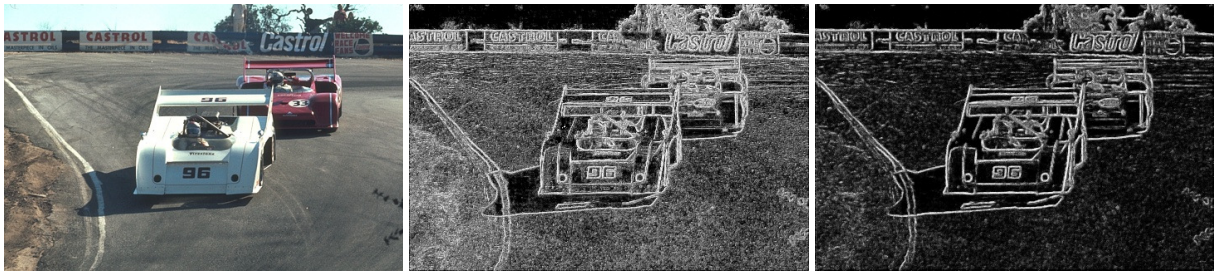
Figure 4.4: Illustration of the weighting function effect on the final gradient image.

Although any of those functions are valid functions for the edges, as they could represent the aspects in a vertex neighborhood, these functions do not characterize similarities between vertices. Another function commonly available is a cardinality function, denoted by $\mathcal{F}_{\ell_0}(u, v)$, corresponding to the total number of nonzero values among $f(u)$ and $f(v)$. Therefore $\mathcal{F}_{\ell_0} : \mathbb{R} \to \{0, 1, 2\}$ and it is not very descriptive.

As illustrated in Fig. 4.4, the $\mathcal{F}_{\mathbf{euc}}$ is, as expected, the best weighting function to model the neighborhood of a vertex among the evaluated functions. It presents a good result due mainly to the expected behavior induced by the function properties on the metric space. The other functions that do not represent the variation around a vertex, $\mathcal{F}_{\mathbf{max}}$, $\mathcal{F}_{\mathbf{min}}$, and $\mathcal{F}_{\mathbf{mean}}$, show spread noise to more significant regions in the resulting images. The particular case of $\mathcal{F}_{\ell_0}$ does not represent variation or is very descriptive, and the results demonstrate the RF's limitations for an extremely noisy and correlated input. **In the following, it will only consider $\mathcal{F}_{\mathbf{euc}}$ as the weighting function.**

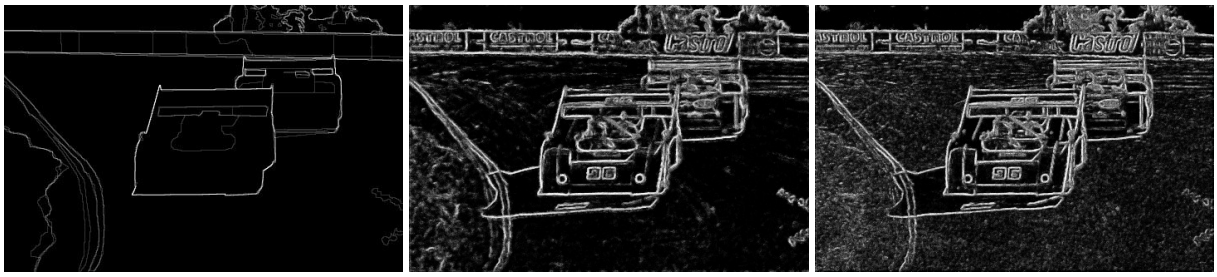
Neighboring size

This assessment pertains to the **edge weights ($\mathbf{X}_{\mathcal{F}}$)** attributes for a given vertex v. It represents v by the set of edge weights between the adjacent vertices. Therefore



(a) Input image **(b)** $|\mathbf{X}_{\mathcal{F}}| = 8$ **(c)** $|\mathbf{X}_{\mathcal{F}}| = 64$

Figure 4.5: Illustration of the neighboring size effect on the final gradient image



(a) Ground truth **(b)** $\mathbf{X}_V$ = low-level descriptors **(c)** $\mathbf{X}_V = \emptyset$

Figure 4.6: Illustration of the inclusion of vertex attributes impact's in the final gradient

$\mathbf{X}_{\mathcal{F}} = \{\mathcal{F}_{\mathbf{euc}}(u, v) \mid \forall u \in \Gamma(v)\}$ and $|\mathbf{X}_{\mathcal{F}}| = 8$. The investigation goes further and includes the adjacency of the immediate neighbors of v. Therefore, $\mathbf{X}_{\mathcal{F}} = \{\mathcal{F}_{\mathbf{euc}}(u, v), \mathcal{F}_{\mathbf{euc}}(y, u)\}$ for all $u \in \Gamma(v)$ and $\forall y \in \Gamma(u)$ and $|\mathbf{X}_{\mathcal{F}}| = 64$.

Fig. 4.5 illustrates the gradients from the two variations. As shown, as the representation size grows, the more information the RF has to decide if a particular vertex is, in fact, an edge. It translates into higher confidence values on the edges and less on the other elements of the original image. Nonetheless, different textures and uniform regions continue to present homogeneous values to distinguish them. **In the following,** $|\mathbf{X}_{\mathcal{F}}| = 64$ **for the neighboring size.**

Vertex attributes

The final assessment concerns the inclusion or not of the **vertex attributes** ($\mathbf{X}_V$) belonging to the set of vertices functions f. Without the vertex attributes, $\mathbf{X}_V = \emptyset$ in the regular representation. When included, it maps $v \in V$ into a set of low-level color

descriptors proposed in Dollar, Belongie, and Perona (2010) [17]. The descriptor takes the original RGB colors on the image pixel. It calculates three color channels in CIE-LUV color space [18], two normalized gradient magnitude channels, and eight gradient orientation channels, resulting in a 13−dimension vector of features for each vertex. Therefore $|\mathbf{X}_V| = 13$.

As shown in Fig. 4.6, including the vertex attributes results in less noise, stronger borders, and more details on the final image gradients. It could be due to the additional information inserted on the representation or other clues about local variation provided by the gradient magnitude and orientation in the descriptor. Nonetheless, including the low-level descriptors as vertex attributes on the representation establishes a solid link to the media source. Most of the following investigative steps will include the vertex attributes on the representation to maintain the most desirable characteristics on the gradient and facilitate the image analysis on the segmentation task in Sections 4.4.2 and 4.4.3.

RF parameters search

Completeness requires one final assessment unrelated to the attributes of the graph but crucial to the task: the RF parameters. The RF is simple and intuitive but requires setting many parameters for its execution. The framework used the Random Forest Regressor included in the *scikit-learn* Python package [19], which provides a parallelized implementation over the trees. To set the RF parameters, a grid search on the number of estimators (number of trees), the bootstrap sample size, and the number of sampled features for the split explored the best set of parameters for the application. It used the validation set and the regular representation with the best set of attributes. The score is the $F1-$measure for the precision-recall on the edges.

Fig. 4.7 illustrates that regardless of the parameter, among all executions, the score varies in $[\,0.5777, 0.6249\,]$, less than 5% gain. The number of estimators and the number of sampled features impacted the training time greatly. Furthermore, the search briefly explored limiting the depth of the trees, which resulted in reduced training times but much lower score values for all combinations (below 0.40 score).

In the end, a compromise on the parameters among the best score results and reason-

17. Dollar Piotr, Belongie Serge, and Perona Pietro (2010). *The fastest pedestrian detector in the west.*
18. Tkalcic Marko and Tasic Jurij (2003). *Colour spaces: perceptual, historical and applicational background.*
19. Pedregosa Fabian et al. (2011). *Scikit-learn: machine learning in Python.*

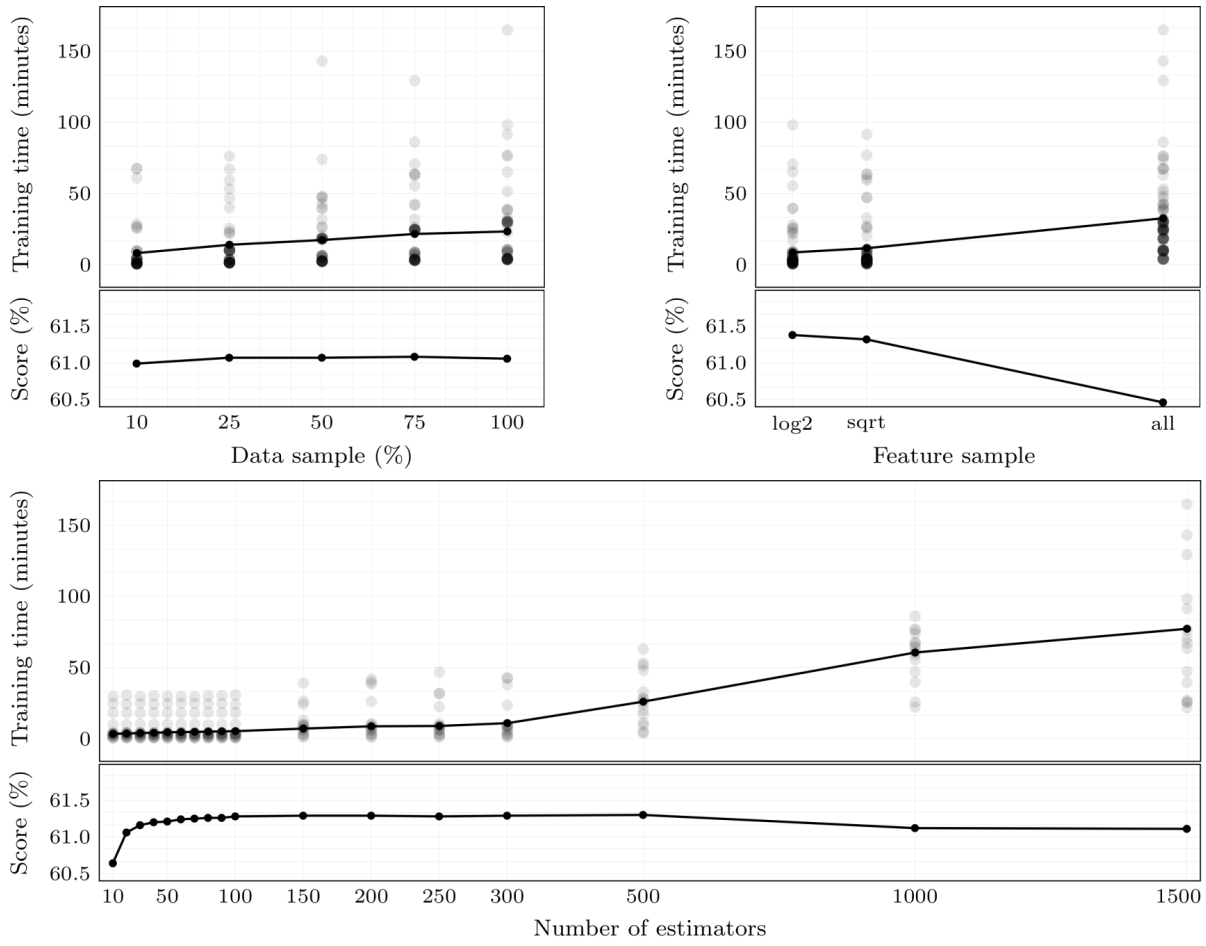


Figure 4.7: Grid search results for the RF parameters: data sample size, feature sample, and number of estimators. Evaluated in terms of training time (average on the dark line and occurrences as lighter dots) and $F1-$measure score for the precision-recall on the edges (average).

able training time resulted in $estimators = 500$, $samples = 25\%$ of T, $\#features = \log_2(p)$ and no control over the depth of the trees.

Final considerations on graph attributes

The case study proposes a straightforward approach to create a systematic graph input applied in a learning framework: use the available information on the graph edges and vertices to represent each vertex as a vector of selected attributes. The first investigative step assessed the selection and established: (i) $|\mathbf{X}_V| = 13$, the original image information stored in each vertex as the low-level descriptors proposed in Dollar, Belongie, and Perona (2010). (ii) $|\mathbf{X}_{\mathcal{F}}| = 64$, the weight values produced by the $\mathcal{F}_{\mathbf{euc}}$ function in $8-$adjacency relation, including also the weights of the adjacency of the immediate

neighbors. Therefore, $p = |\mathbf{G_{att}}| = 77$.

From now on, **graph-based image gradient (GIG)** refers to the proposed method with the selected attributes. [20] Algorithm 1 describes the steps to create the regular GIG representation for one graph $\mathcal{X}_G$ using an edge-weighted graph $G(V, \mathcal{F})$ and $p = 77$ for the selected attributes. For clarity, the operations not detailed in Algorithm 1 are:

— **ones**[[*size*]]: creates an array filled with the one value of size *size*. The operation **ones** plays a dual function: allocates the necessary memory and acts as a padding value (the maximal dissimilarity) for the vertices with an incomplete adjacent set (vertices created from the pixels close to the image border).

— **getLabel**(v): gets the ground-truth label for the vertex v.

— **getDescriptors**(v): gets the attributes stored in v. Here, the set of low-level color descriptors proposed in Dollar, Belongie, and Perona (2010), extracted during the graph creation, mapping the RGB colors of an image pixel into three color channels in CIE-LUV color space, two normalized gradient magnitude channels, and eight gradient orientation channels.

— *append*(array): appends row-wise the computed 1D array vector for each vertex to the 2D matrix representing the graph.

In general, the expected result of the proposed framework is a very descriptive image gradient in which: (i) object boundaries are highlighted (including very small components); (ii) image textures are firmly represented with different simplified patterns; and (iii) large regions are uniform with the distinction of shadow regions. By contrast, unsatisfactory results are usually created not descriptive weighting function and source images with: (i) a low signal-to-noise ratio; and (ii) very similar colors on different objects and patterns.

4.4.2 Evaluating image gradients on segmentation

Creating an image gradient is a transformation process that aims to enhance the desirable properties of an image while leaving aside the noise and non-descriptive elements [21]. Many algorithms in image processing rely on a good image gradient to adequately per-

20. Gradient computation code available at `https://github.com/RaquelAlmeida/GIG.git`
21. Gonzalez Rafael (2009). *Digital image processing.*

Algorithm 1: Regular representation GIG

Input : $G = (V, \mathcal{F})$: an edge-weighted graph, a flag *isTrainSet* indicating if G is a train instance, and the expected representation size p.

Output : $\mathcal{X}_G = \{(\mathbf{X}_1, \mathbf{Y}_1), \ldots, (\mathbf{X}_{|V|}, \mathbf{Y}_{|V|})\}$: set of regular representations of the graph vertices attributes $\mathbf{X}_v \in \mathbb{R}^p$ and its associated labels $\mathbf{Y}_v$ in case when G is a train instance.

Function *getAttributes(v)*:

if *isTrainSet* **then**
  $\mathbf{X}_v = \mathbf{ones}[[p+1]]$
  $\mathbf{Y}_v = \mathbf{getLabel}(v)$
  $\mathbf{X}_v[[p+1]] \leftarrow \mathbf{Y}$ `// at` $p+1$ `position`
else $\mathbf{X}_v = \mathbf{ones}[[p]]$
$colorFeatures \leftarrow \mathbf{getDescriptors}(v)$
$\mathbf{X}_v \leftarrow colorFeatures$
$firstNeighbors \leftarrow \{u \mid \forall u \in \Gamma(v)\}$
$secondNeighbors \leftarrow \{q \mid \forall q \in \Gamma(u)$ **and** $\forall u \in firstNeighbors\}$
$\mathbf{X}_v \leftarrow \{\mathcal{F}(u, v) \mid \forall u \in firstNeighbors\}$
$\mathbf{X}_v \leftarrow \{\mathcal{F}(q, u) \mid \forall q \in secondNeighbors$ **and** $\forall u \in firstNeighbors\}$
return $\mathbf{X_v}$

Main:

for *vertex* v ***in*** V **do**
  $\mathbf{X}_v = getAttributes(v)$
  $\mathcal{X}_G \leftarrow append(\mathbf{X}_v)$
end
return $\mathcal{X}_G$

form tasks [22] [23] such as classification [24] and segmentation [25]. For this reason, this section evaluates the quality of the image gradients created from the best set of graph attributes assessed in Section 4.4.1 on the image segmentation task. It argues that although the borders constitute an essential characteristic of the objects depicted in images, other properties that reflect uniformity, homogeneity, and continuity are also important for the interpretation of coherent regions, particularly in the task of image segmentation.

Image segmentation may be considered a semantic task, and it is an active topic of

22. Tzimiropoulos G et al. (2010). *Robust FFT-based scale-invariant image registration with image gradients.*
23. Neggers J. et al. (2016). *On image gradients in digital image correlation.*
24. Sharifi M., Fathy M., and Mahmoudi M.T. (2002). *A classified and comparative study of edge detection algorithms.*
25. Hirata Roberto et al. (2000). *Color image gradients for morphological segmentation.*

research [26]. This task consists in partitioning perceptually similar pixels into sets of regions representing areas of interest. Usually, this task is done in two stages: (i) the extraction of image characteristics that facilitates interpretation and further analysis; and (ii) the mapping of these characteristics into coherent regions.

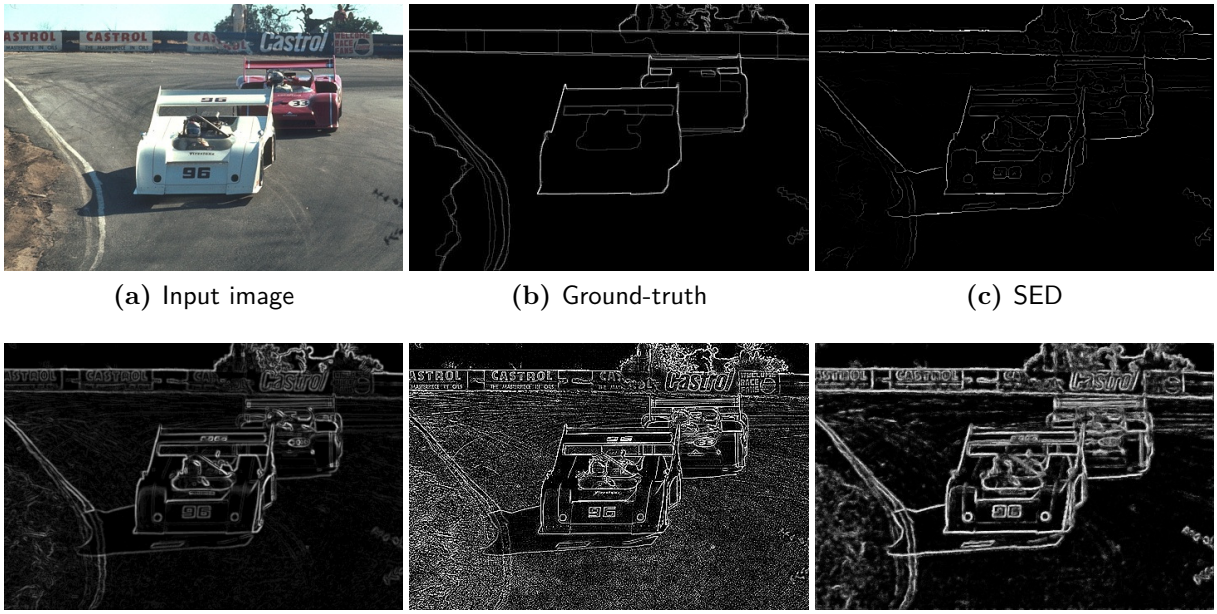
A coherent region is a subjective concept, but according to to Domínguez and Morales (2016) [27], it must present characteristics such as: (i) uniformity; (ii) continuity; (iii) contrast between adjacent regions; and (iv) well-defined boundaries.

Independently on how well-designed a mapping method is, most of them are limited by the characteristics extracted in the first stage. For instance, taking the grey-level contrast in the first stage produces a great variation between regions, and very distinct absolute values make it hard to determine which value actually represents a region change.

Fast and inexpensive to compute, image gradients are commonly used as a pre-processing step in multiple applications, such as medical analysis [28], text extraction [29], video processing [30] [31] and segmentation [32]. Even with the advent of deep networks, gradient use continues to be relevant due to its performance [33]. Also, the gradients are used as support for some networks, providing enhanced features or reducing complexity [34] [35] [36].

Traditional gradient methods, such as Laplacian and Sobel, are kernel filters for local variation, highlighting the borders of objects, and are usually very sensitive to abrupt changes in the original image. As detailed in Section 4.3 the method SED [37] is fast and

26. Mittal Himanshu et al. (2022). *A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets.*
27. Domínguez Didier and Morales Roberto Rodriguez (2016). *Image segmentation: advances.*
28. Soni Akanksha and Rai Avinash (2021). *Automatic cataract detection using Sobel and morphological dilation operation.*
29. Jeong Hyeonwoo, Choi Ye-Chan, and Choi Kang-Sun (2021). *Parallelization of levelset-based text baseline detection in document images.*
30. Honeycutt Wesley T. and Bridge Eli S. (2021). *UnCanny: exploiting reversed edge detection as a basis for object tracking in video.*
31. Eetha Sagar, Agrawal Sonali, and Neelam Srikanth (2018). *Zynq FPGA based system design for video surveillance with Sobel edge detection.*
32. Junejo Aisha Zahid et al. (2018). *Brain tumor segmentation using 3D magnetic resonance imaging scans.*
33. Lakshmi M. Muthu and Chitra P. (2020). *Tooth decay prediction and classification from X-ray images using deep CNN.*
34. Naveen P. and Sivakumar P. (2021). *Adaptive morphological and bilateral filtering with ensemble convolutional neural network for pose-invariant face recognition.*
35. Tu Zhuowen et al. (2008). *Brain anatomical structure segmentation by hybrid discriminative/generative models.*
36. Prabaharan L. and Raghunathan A. (2021). *An improved convolutional neural network for abnormality detection and segmentation from human sperm images.*
37. Dollar Piotr and Zitnick C. Lawrence (2015). *Fast edge detection using structured forests.*



(a) Input image **(b)** Ground-truth **(c)** SED

(d) Sobel **(e)** Laplacian **(f)** GIG

Figure 4.8: SED gradient presents reinforced fuzzy borders of the main objects and small details are in large ignored. Sobel presents very thin edges for both large and small objects, while large uniform regions, such as the asphalt and vegetation are discretely represented. For Laplacian, it is perceived a large amount of noise for objects, edges and patterns. GIG computed enhanced borders for both large and small objects and image textures are firmly represented with different simplified patterns.

precise in predicting object edges and became a common approach as image gradient creator for the segmentation task [38] [39] [40]. In turn, GIG firmly depicts the edges of large and small objects as well as uniform regions and patterns on the image, making it a very descriptive image gradient. Fig. 4.8 illustrates the gradient computed by GIG, SED, Sobel, and Laplacian.

This investigative step evaluates the proposed GIG both qualitatively and quantitatively on the segmentation task and compares it to the widely used gradients from SED, Sobel, and Laplacian. It does not propose a segmentation approach; instead, it evaluates the strategy to extract image characteristics in an application. It applies the compared

38. Perret Benjamin, Cousty Jean, Guimaraes Silvio Jamil F., et al. (2018). *Evaluation of hierarchical watersheds.*
39. Perret Benjamin, Cousty Jean, Guimarães Silvio Jamil Ferzoli, et al. (2019). *Removing non-significant regions in hierarchical clustering and segmentation.*
40. Otiniano-Rodríguez Karla et al. (2019). *Hierarchy-based salient regions: a region detector based on hierarchies of partitions.*

methods as input for the watershed hierarchies [41].

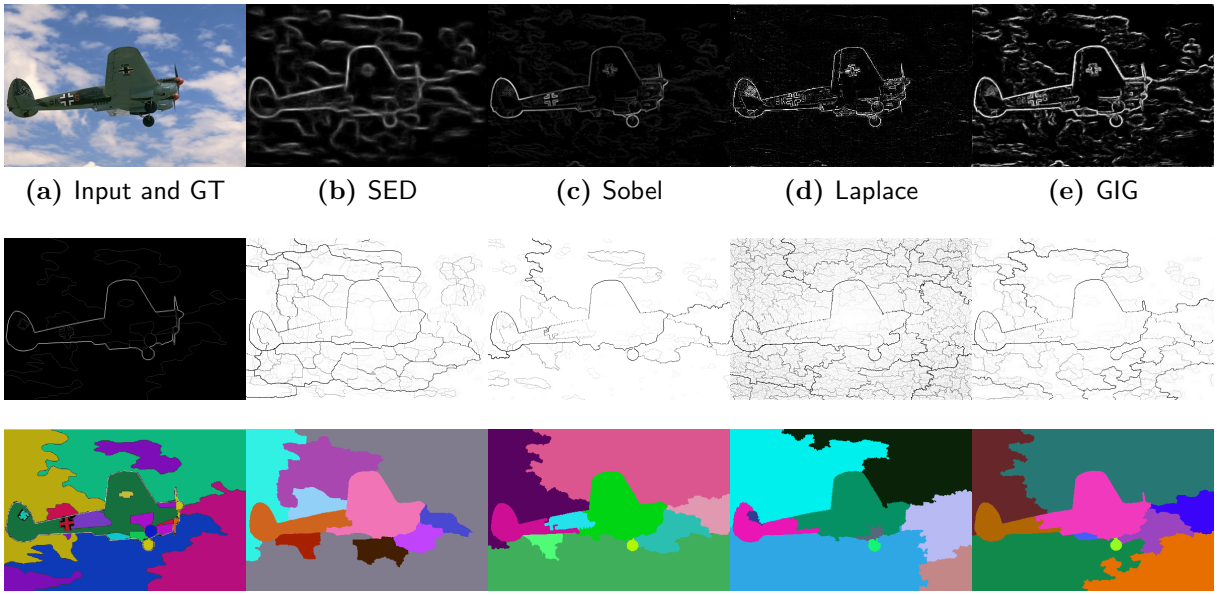
The watershed hierarchy maps image gradients to segmentation, and its performance depends on the gradient input, making it the ideal candidate for evaluating the methods. It is worth mentioning that, thanks to this hierarchical structure, it is straightforward to compute segmentation with an exact number of regions, for instance, from 2 to 5000 regions. This allows the analysis of a small number of regions closer to the ground truth, a medium number of regions for region consistency, and a very large number of regions (1000 and 5000), in which results are similar to a superpixel segmentation method.

Qualitative analysis on the segmentation task

Fig. 4.9 illustrates the gradient images obtained from the compared methods for the input images on the first row. As SED is a method for edge detection, it generally produces gradient images with soft edges close to the ground-truth boundaries, which guarantees its success on the edge detection task. Nonetheless, other aspects present in the input image, such as textures and small details, are wildly ignored. Sobel presents more details without significant distinctions (regarding the magnitude of values) for components other than the main object. Laplace, in turn, is permeated by noise on the object and background. For GIG, the gradients have a balance between highlighted firm edges, different textures, and uniform regions presented with homogeneous values distinguishing them. It is important to consider that Sobel and Laplacian depend on parameters definition, such as the kernel size. For the Sobel gradients, the parameters are the gradient magnitude with the ℓ_2-norm and a 3×3 kernel size calculated from the gray-scale image. For the Laplacian, the zero-crossing with a threshold at 0.04 of maximum value.

The 2nd row of Fig. 4.9 presents visual representations of the watershed hierarchies created from the gradients. The hierarchies presented as saliency maps allow the visualization and understanding of the hierarchies [42]. The saliency maps show the regions of importance mapped by the watershed method, indicating the strength and limitations of the final segmentation. As a hierarchical model, the watershed regions are stable and causal, meaning that no new region is created or removed, only merged and split, depending on the number of regions criteria. Therefore, the borders visualized on the saliency maps will not change their contours, and their strength indicates the region's proximity.

41. Cousty Jean and Najman Laurent (2011). *Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts.*
42. Cousty Jean, Najman Laurent, Kenmochi Yukiko, et al. (2018). *Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps.*



(a) Input and GT (b) SED (c) Sobel (d) Laplace (e) GIG

Figure 4.9: 1st row: Input image and the gradients created by the compared methods. 2nd row: Boundary ground-truth and watershed hierarchies represented as saliency maps that allow us to visualize the hierarchy of regions. 3rd row: Segmentation ground-truth and the segmented images with 10 regions as criterion.

In knowing that, the 3rd row of Fig. 4.9 shows the result of the segmentation created using ten regions as a criterion. The firm contours on SED and GIG give a good delineation of the main object while it invades part of the plane with Sobel. All details are lost using the SED gradient, while Sobel partially recovers part of the cross, GIG recovers the paddle, and both present the wheel. The background invades both the object and details using Laplace as a result of the noise on the gradient.

Fig. 4.10 presents more examples of segmented images from GIG, SED, and Sobel gradients, illustrating some variation in the number of regions, including the superpixel effect with a large number of regions (3rd column). In the 1st column, a successful instance of the proposed method in which the presence of strong borders and large uniform regions on the input image, captured by the GIG gradient, created a better segmentation. Using SED, the fuzzy edges limit the delineation of the main object, while the weak contour in Sobel prevents its detection. An observed limitation of the proposed method is presented in the 2nd column. When the input image shows objects with high-contrast patterns, such as zebras and tigers, the GIG gradient's detail works against the object's distinction. This is also partially observed on Sobel, but not with the SED gradient, in which the pattern details are softly represented inside the object. On the super-segmented images in the 3rd

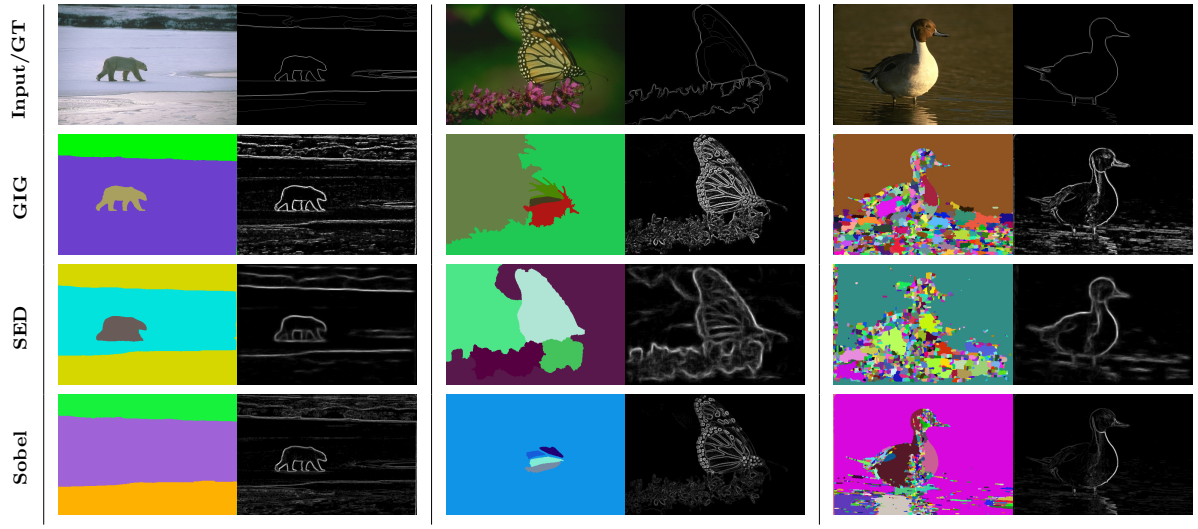


Figure 4.10: Segmentation results obtained from GIG, SED and Sobel gradients with varying number of regions (Left: 3, middle: 5, right: 1000).

column, the soft edges on SED do not produce a good segmentation. At the same time, large regions with Sobel are indistinct, creating many very small regions on the main object with little to no large parts of the duck, the water, and the shadow.

Quantitative analysis on the segmentation task

In terms of training time, in the standard discrete label and the graph attribute selection on GIG, all the 150 trees on the RF are trained in less than three minutes, while for SED, in the same CPU, each tree (of the eight trees for the presented results) takes approximately four hours. The inference takes a fraction of a second for each image in all compared methods.

For the quantitative metrics, there is two types of image partition interpretation measures, as categorized and defined in Pont-Tuset and Marques (2013) [43]: (i) Precision-recall for regions, using a pixel-wise comparison for overall performance in terms of $F1-$measure; and (ii) Probabilistic Rand Index (PRI), a pixel-wise measure that considers the multiple ground truths presented for each image on the BSDS500 dataset. Table 4.1 presents the results for both metrics. The $F1-$measure results for regions are presented in terms of the optimal dataset scale (ODS), optimal image scale (OIS), and average precision (AP) through all scales, and the PRI in terms of ODS. The superior results of GIG

43. Pont-Tuset Jordi and Marques Ferran (2013). *Measures and meta-measures for the supervised evaluation of image segmentation.*

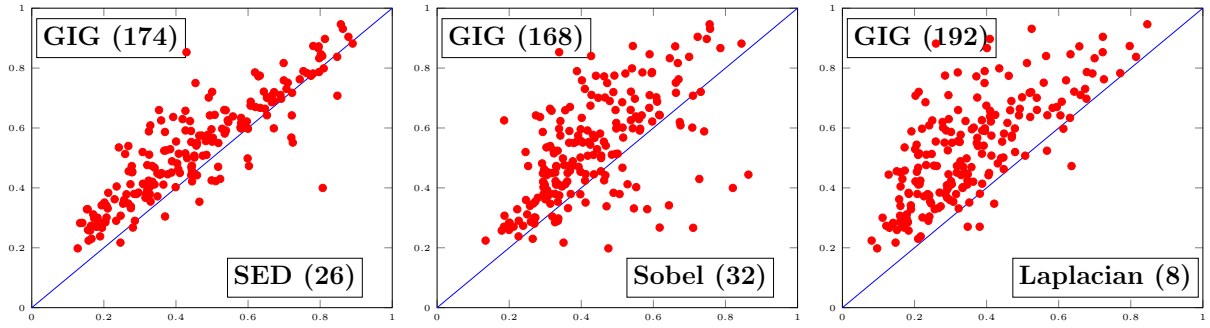


Figure 4.11: Pair-wise $F1-$measure results (red dots) on the best scale for each method, counting each image on the test set. The values in the boxes is the number of images that are better for a particular method. Proposed is better than all compared method, with statistical significance ($p-$value less than 10e-17).

on all metrics indicate that the strong borders combined with the uniform region information positively impact the hierarchies of watershed segmentation. Finally, in Fig. 4.11, we present a pair-wise comparison of individual images on the best scale of each compared method. As one can see, the proposed GIG produces considerably better-segmented images and are all statistically significant ($p-$value less than 10e-17).

Table 4.1: $F1-$measures for regions and PRI presented in terms of the optimal dataset scale (ODS), optimal image scale (OIS) and average precision (AP) through all scales. Perfect score=1.

	$F1-$**measure for regions**			**PRI**
Gradient	**ODS**	**OIS**	**AP**	**ODS**
GIG	**0.620**	**0.688**	**0.507**	**0.786**
SED	0.559	0.617	0.477	0.746
Sobel	0.579	0.655	0.481	0.742
Laplacian	0.511	0.583	0.476	0.741

Final considerations on the segmentation task

This investigative step verified the viability of the proposed representation as a gradient applied to the segmentation task. GIG and other popular gradient methods, namely SED, Sobel, and Laplace, were used as input for the watershed hierarchies segmentation method that relies on a good image gradient. GIG on the structured input proved to be not only viable but also considerably faster to train than the structured output in SED. Also, a quantitative analysis of the produced segmentations confirmed the visual results and demonstrated that GIG is a better candidate for creating image gradients for the

segmentation task.

4.4.3 Extended formalism

The GIG representation considers the type of graph, its proximity to the original data, and the expected results, allowing the processing of the graphs as regular data with a fast machine learning algorithm and avoiding the long computations of graph networks. The last investigative step extends the GIG formalism to exploit better the relationships modeled by the graphs, mainly focusing on the RF mechanics and limitations. Namely, it proposes:

1. **Region adjacency graphs (GIG-RAG):** Extends the formalism from the bijective correspondence of vertices and pixels in GIG to vertices and a set of regions produced by an initial segmentation into image superpixels. This approach could reduce the number of data points during training and impact the gradient and the computational cost;
2. **Positional features (GIG-Positional):** The vertices corresponding to the pixels on the image's border have an incomplete set of neighbors. In GIG, they received padding values that disregarded the missing value's position on the regular representation. In the positional feature approach, it only takes the vertices with a complete set of adjacent vertices to avoid changing the feature connotation during training;
3. **Unique paths (GIG-Unique):** The regular representation of the grid graph in GIG is redundant, meaning not all values are unique as the vertices on the grid path share some neighbors. The unique path approach considers only the first instance of a neighboring vertex in a region, reducing the representation's size and allowing the region's expansion.

Besides the extended formalism, it adds evaluations on edge detection and comparisons with deep learning approaches [44] [45] on edge detection and segmentation. It aims to advocate for the assertion that a good gradient for image segmentation should present more than precise object contours by comparing it with even more accurate edge maps. And for completeness, it also evaluates GIG-Edge, the GIG variant considering only edge attributes.

This step presents the quantitative analysis for edge detection and segmentation tasks, using the $F1-$measure for the precision-recall metrics for boundaries and regions. It also

44. Xie Saining and Tu Zhuowen (2017). *Holistically-nested edge detection.*
45. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*

assesses the Probabilistic Rand Index (PRI) [46] measure. Results are in terms of the optimal dataset scale (ODS), optimal image scale (OIS), and average precision (AP) through all scales. The scales for the boundaries are different thresholds applied to the edge maps to create a binary image and, for the regions, the desired number of segmented regions.

GIG defined on region adjacency graphs

The proposed region adjacency graph approach (GIG-RAG) reduces the number of vertices by presenting regions of grouped pixels in the set of vertices instead of a single pixel as in GIG. The GIG-RAG requires a strategy to group the pixels and considerations for edges, weights, and label attribution.

Given an edge-weighted grid graph $G = (V, \mathcal{F})$ and a set $\mathbb{S}_I = \{r^1, \ldots, r^R\}$ of grouped pixels into R regions for an image I, the *edge-weighted RAG graph* G_{rag} has one vertex for each labeled region in $\mathbb{S}_I$, thus $V_{rag} = \{v_l \,|\, l \in \{1, R\}\}$. There is an edge between two vertices in V_{rag} if an edge connects two vertices in the original grid graph G, hence: $E_{rag} = \{\{v_p, v_q\} \,|\, v_p, v_q \in V_{rag} \wedge p \neq q \wedge \exists\{u, v\} \in E \,|\, u \in r^p \wedge v \in r^q\}$. The set E_{rag} induces a unique adjacency relation on V_{rag}, which associates $v_q \in V_{rag}$ with $\Gamma(v_q) = \{v_p \in V_{rag} | (v_p, v_q) \in E_{rag}\}$. For the weighting function, it averages the edges' weights in G, $\mathcal{F}_{rag}(u_q, v_q) = \mathbf{mean}\{\mathcal{F}(u, v) \,|\, \forall\{u, v\} \in E \,|\, u \in r^p \wedge v \in r^q\}$.

Finally, label attribution in GIG-RAG takes a majority vote of the pixels within a region to determine the region label. Also, because it has multiple vertices within each region, it does not use the vertex attributes $\mathbf{X}_V$; thus, GIG-RAG attributes are only the edge weights $\mathbf{X}_{\mathcal{F}_{\mathbf{rag}}}$ of a regular number of closest adjacent regions. For GIG-RAG, $p = 64$ (no vertex attributes), and it uses the terminology GIG-RAG-R to indicate the number of desired regions of grouped pixels, where R is in $\{1\text{k}, 5\text{k}, 10\text{k}, 50\text{k}\}$.

Despite removing the vertex attributes, GIG-RAG continues to be connected to the media source because of the strategy required to group the pixels. However, this strategy is closer to a modeling choice like the weighting function and adjacency relation than a media-dependent feature.

The initial segmentation into image superpixels to group the pixels proposes a well-consolidated method called *simple linear iterative clustering* (SLIC) [47]. SLIC is an iterative method, which clusters the pixels with the closest center, initially distributed in a

46. Pont-Tuset Jordi and Marques Ferran (2013). *Measures and meta-measures for the supervised evaluation of image segmentation.*
47. Achanta Radhakrishna et al. (2010). *Slic superpixels.* 149300.

regular grid, evaluates the similarity within a cluster, and recalculates the centers until convergence. SLIC execution is fast and easy to set the parameters, and the number of produced regions is the closest approximation to the number of desired regions passed as a parameter.

To evaluate the impact of the superpixel quality in the GIG-RAG strategy, it adds some comparisons with a more modern method: the *superpixel segmentation with fully convolutional networks* (SpixelFCN) [48]. The SpixelFCN is a deep network trained to assign each pixel of an image, initially partitioned into a regular grid, to one of its neighboring grids. The network is an auto-encoder, in which the encoder learns the features, and the decoder aims to group pixels with similar features and enforce compactness. SpixelFCN is easy to set and overall produces better regions, but as in many deep network methods, it is limited in the number of created regions since the regular grid has a fixed size. To increase the number of superpixels, one should increase the scale of the input image.

Positional features

RF is an ensemble of multiple decision trees in which each independent tree takes local decisions to split the data considering a combination of features. The position of the features is an essential factor, as the model would assume that any subsequent data in that specific position would represent the same feature. GIG added padding values to the vertices that do not have all neighbors in the grid path, disregarding the position that the missing value would assume if present. GIG-Positional will take only the vertices with a complete set of adjacent vertices, creating a more regular representation for training. Like GIG, GIG-positional has $p = 77$.

Unique path

The unique path consideration tackles the redundancy created when transposing the edge-weighted graph to a regular representation. In Algorithm 1, when it takes the first and second adjacent neighbors, it inevitably casts repeated values to the regular representation as many vertices share some neighbors within the grid path. While this redundancy is not necessarily a problem, out of the 64 values obtained with two levels of neighbors with an 8−adjacency relation, only 24 of these values are unique. The GIG-Unique approach considers only the first instance of a neighboring vertex within a region. Removing the

48. Yang Fengting et al. (2020). *Superpixel segmentation with fully convolutional networks.*

redundant values allows the expansion of the region of analysis while maintaining a similar-sized representation.

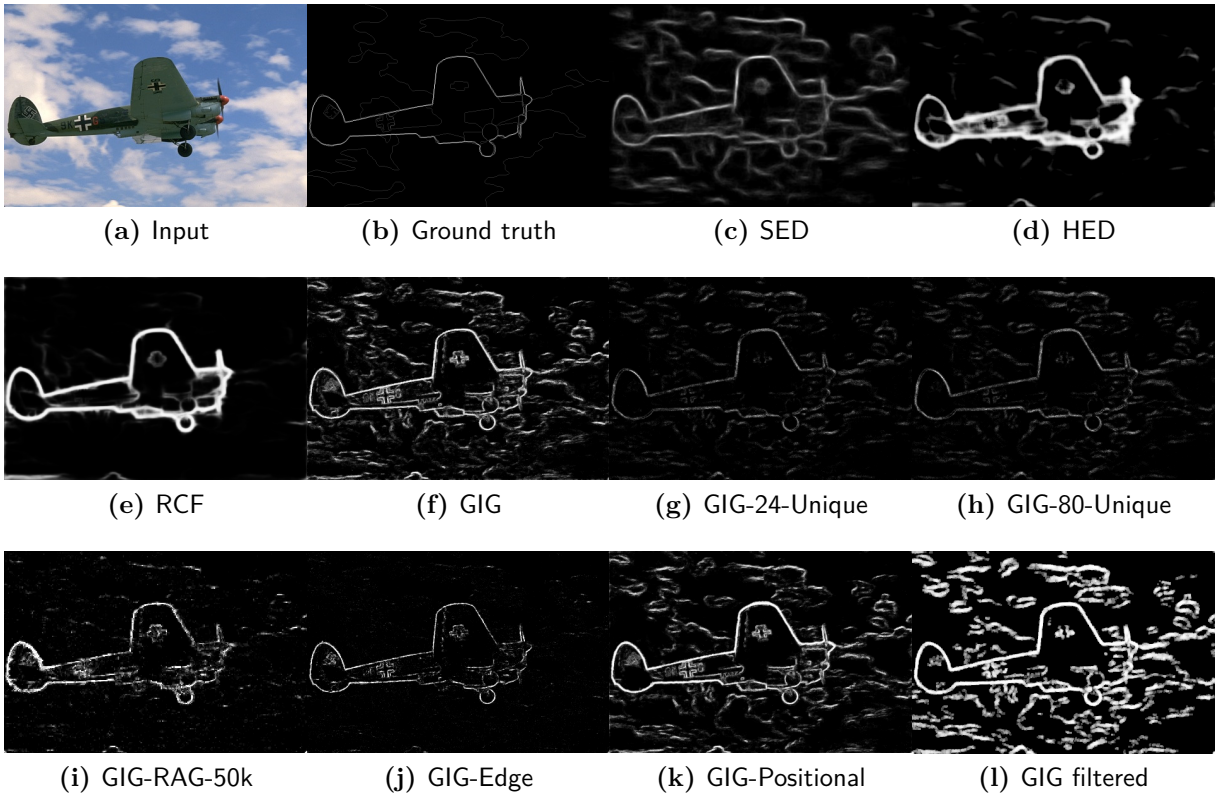
GIG-Unique proposes two variations with an 8−adjacency relation: (i) two levels of neighbors, a 5×5 grid on the original image, 24 values instead of 64; and (ii) four levels of neighbors, a 9×9 grid, 80 values instead of 4096. For all variations, the regular representation has the same attributes as GIG, $\mathbf{G_{att}} = \{\mathbf{X}_V \mathbf{X}_\mathcal{F}\}$, but different dimensions for the edge's attributes. The terminology GIG-U-Unique, with U in $\{24, 80\}$ indicates the number of unique values, thus $|\mathbf{X}_\mathcal{F}|$, leading to $p = 37$ and $p = 93$, respectively.

Results on edge detection - extended formalism

First, it evaluates each proposed strategy's impact on the representation performance regarding the quality of the edge maps (illustrated in Fig. 4.12(f)-(k)). Table 4.2 presents the relevant results in the validation set for the edge detection task. It omitted some similar values in Tables 4.2-4.4 to avoid repetition. All variations of the GIG-U-Unique presented similar gradients (thin contours and discreet textures) and results (less than 1.5% difference in all metrics). Moreover, all the GIG-Unique strategies had similar score metrics to the original GIG, indicating that there is not much gain in expanding the region of analysis and that the GIG redundancy does not compromise the RF generalization. The GIG-Positional results indicated that the feature position is, in fact, an essential factor during training, and the edge maps created have the best performance on the task among all the compared proposals. GIG-Edge shares the number of regions with GIG while not considering the vertex attributes as GIG-RAG, and the results reinforced the importance of vertex attributes in the gradients and the score.

The GIG-RAG strategy considerably reduced the training time proportionally to the number of regions. Nevertheless, it also reduced the performance of the task. As shown in Table 4.2, SpixelFCN has a slight advantage over SLIC superpixels, but it is limited on computational resources to create a larger number of regions. For instance, to make 50k regions with SpixelFCN, one must work with images scaled to ~ 9 times the original size. As illustrated in Fig. 4.13, the computed gradients vary with the number of regions, fewer regions create larger superpixels, and the predicted label is applied to a larger area creating gradients of regions instead of contours. Increasing the number of regions creates more contour-oriented gradients, which is crucial for the edge detection task.

Table 4.3 presents the best of the proposed methods, GIG and GIG-Positional. We compared with some leading deep methods on the task: the Holistically-Nested Edge



(a) Input (b) Ground truth (c) SED (d) HED

(e) RCF (f) GIG (g) GIG-24-Unique (h) GIG-80-Unique

(i) GIG-RAG-50k (j) GIG-Edge (k) GIG-Positional (l) GIG filtered

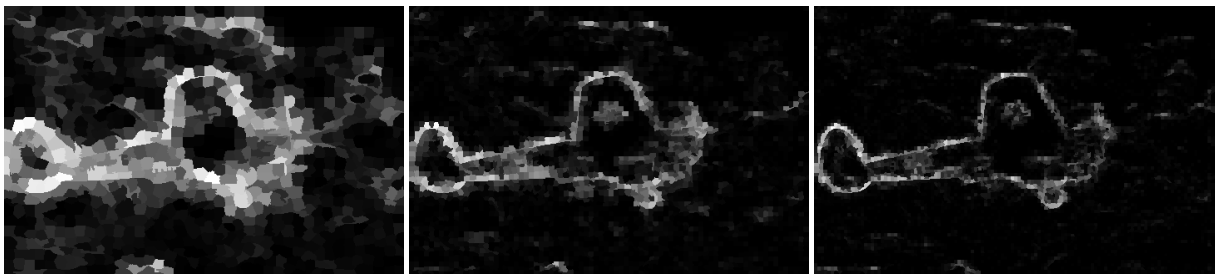
Figure 4.12: Gradient computations for the input image in (a). SED, HED, and RCF present reinforced contours of the main objects but almost no details; also, most contours are fuzzy, particularly for SED and HED. GIG computes enhanced borders for large and small objects and firm representations of textures. GIG-Unique gradients present thin contours close to the ground-truth and discreet textures (g, h). GIG-RAG gradient with 50k regions computed with SLIC presents a contour-oriented gradient with little texture information (i), the same is true for the GIG-Edge variation (j). GIG-Positional gradients (k) have slightly stronger borders for the main objects than GIG. In (l), the GIG gradient filtered by the morphological opening operation presenting thicker contours.

Detection [49] (HED, illustrated in Fig.4.12(d)) and the Richer Convolutional Features for Edge Detection [50] (RCF, illustrated in Fig.4.12(e)). Both HED and RCF produced better edge maps ($F1-$measures reported by the authors). Still, they required considerably longer training times (measured using a GPU during $10,000$ iterations on the required augmented dataset, following S. Xie and Tu (2017)). It also includes results from SED (illustrated in Fig. 4.12(c)), which formalism is parallel to GIG. SED, GIG, and GIG-Positional were all trained using the same CPU, with parallelized computation over the trees in 8 CPU cores, wherein the SED's RF is composed of 8 trees and ours of 500. The

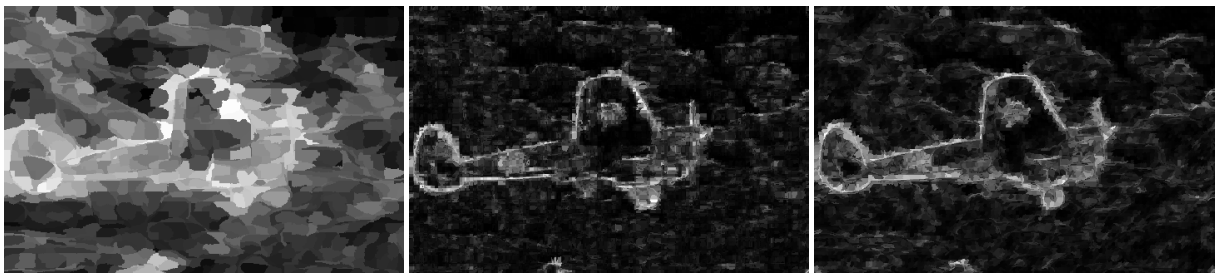
49. Xie Saining and Tu Zhuowen (2017). *Holistically-nested edge detection.*
50. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*

Table 4.2: Quantitative results on edge detection. $F1-$score for boundaries in terms of optimal dataset scale (ODS), optimal image scale (OIS) and average precision (AP) through all scales (perfect scores=1). Executed on the validation set.

Method	ODS	OIS	AP
GIG	0.623	0.651	0.619
GIG-RAG-600 (SLIC)	0.441	0.471	0.461
GIG-RAG-1k (SLIC)	0.472	0.502	0.463
GIG-RAG-5k (SLIC)	0.522	0.546	0.505
GIG-RAG-10k (SLIC)	0.542	0.566	0.541
GIG-RAG-50k (SLIC)	0.593	0.623	0.587
GIG-RAG-600 (SpixelFCN)	0.498	0.525	0.448
GIG-RAG-1k (SpixelFCN)	0.509	0.538	0.474
GIG-RAG-5k (SpixelFCN)	0.553	0.581	0.562
GIG-RAG-10k (SpixelFCN)	0.546	0.571	0.554
GIG-RAG-50k (SpixelFCN)	-	-	-
GIG-Unique-24	0.618	0.645	0.621
GIG-Unique-80	0.615	0.640	0.619
GIG-Edge	0.605	0.611	0.599
GIG-Positional	**0.632**	**0.667**	**0.669**



(a) SLIC **1k** regions **(b)** SLIC **5k** regions **(c)** SLIC **10k** regions



(d) SpixelFCN **1k** regions **(e)** SpixelFCN **5k** regions **(f)** SpixelFCN **10k** regions

Figure 4.13: Illustration of the gradient computations for the GIG-RAG strategy comparing the outputs of both tested superpixel algorithms. Gradients from SLIC present more apparent emphasis on the contours, while the ones from SpixelFCN preserve more texture information.

training time reflects the gain of the structure input on GIG instead of the structured output on SED. Furthermore, the GIG-Positional scores were comparable to SED. For the inference time, all methods took only a fraction of a second for each image, whereas RCF and ours were slightly faster.

Table 4.3: Quantitative results on edge detection.Comparison of the best proposed methods on the edge detection task with some of the well-acknowledged methods on the dataset. It presents the $F1-$scores for boundaries in terms of optimal dataset scale (ODS), optimal image scale (OIS) and average precision (AP) through all scales, the training and inference time (per image) for all compared methods. Perfect scores=1. Executed on the test set.

	$F1-$score boundaries			Train time	Inference time
Method	ODS	OIS	AP	(hh:mm:ss)	(s/image)
SED	0.712	0.724	0.750	03:53:18	0.452
HED$*$	0.782	0.804	0.833	11:03:42	0.215
RCF$*$	**0.811**	**0.830**	**0.947**	10:43:25	**0.141**
GIG	0.635	0.661	0.648	**00:09:18**	0.179
GIG-Positional	0.660	0.718	0.739	00:11:22	0.167

$*$ Trained using GPU and $F1-$score as reported by the authors.

Results on segmentation - extended formalism

It evaluates the segmentation task for all the proposed GIG variants, the deep methods, and SED. Table 4.4 shows that the worst metrics are for the GIG-RAG with a small number of regions (1k and 5k) computed from SLIC. The GIG-RAG gradients computed from SpixelFCN and SLIC with a larger number of regions (10k and 50k) perform like some of the best edge detection methods (SED and GIG-Positional). HED and GIG-Edge have similar performance on the task, whereas GIG-Edge has an advantage on the PRI metric and the AP. GIG performs better than both in all metrics. GIG-Unique underperforms compared to GIG, except for the $F1-$measure dataset scale, meaning that there is a certain number of segmented regions to choose and have more consistent results.

Finally, the RCF results outperformed GIG and the others proposed in all metrics. The gradients produced by GIG and RCF are very different, from the detail level to the contours' thickness. To investigate the thickness factor, it applies the operation *opening*—a well-known mathematical morphology filtering operation, consisting of one *erosion* to remove small regions followed by one *dilation* to increase object boundaries—on the GIG gradients to expand the contours. The erosion operation used a kernel 3×3 to avoid enlarging small points, followed by a 4×4 dilation kernel. Figure 4.12(l) illustrates the

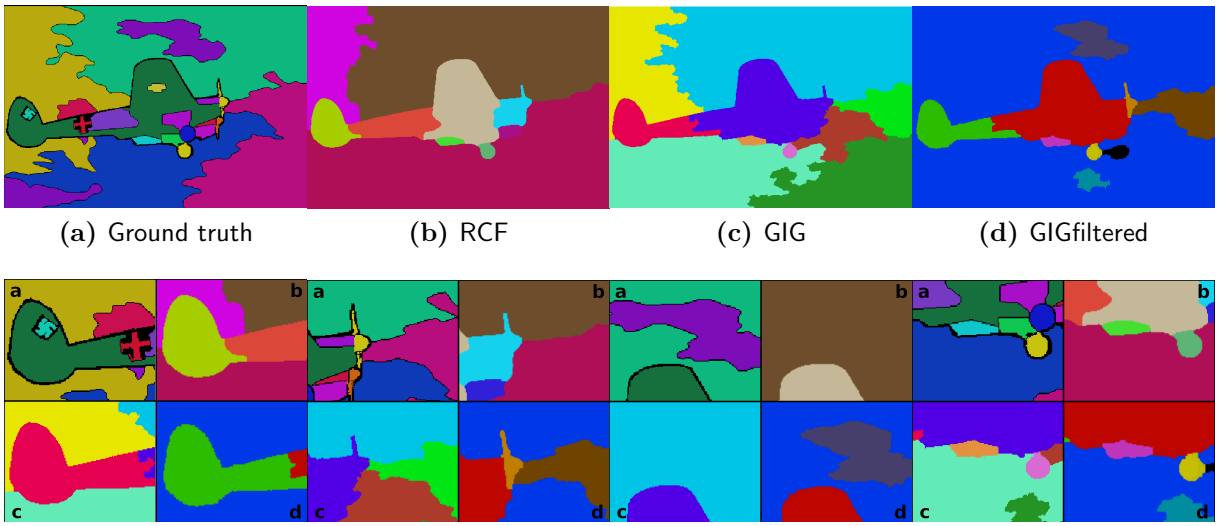
Table 4.4: Quantitative results on segmentation task. Segmentation results for all compared methods when applied as gradient input to the hierarchical watershed method. Results presented as $F1-$scores for boundaries in terms of optimal dataset scale (ODS), optimal image scale (OIS), and average precision (AP) through all scales, and for the Probabilistic Rand Index (PRI). Perfect $F1-$score and PRI=1. Executed on the test set.

	$F1-$ecore regions			PRI	
Gradient	ODS	OIS	AP	ODS	OIS
SED	0.559	0.617	0.477	0.746	0.742
HED	0.616	0.687	0.485	0.747	0.746
RCF	**0.721**	**0.787**	0.548	**0.835**	0.877
GIG-RAG-1k (SLIC)	0.537	0.571	0.420	0.718	0.727
GIG-RAG-5k (SLIC)	0.540	0.580	0.427	0.728	0.729
GIG-RAG-10k (SLIC)	0.549	0.598	0.438	0.737	0.739
GIG-RAG-50k (SLIC)	0.582	0.638	0.482	0.758	0.788
GIG-RAG-1k (SpixelFCN)	0.550	0.607	0.470	0.749	0.780
GIG-RAG-5k (SpixelFCN)	0.558	0.624	0.535	0.758	0.786
GIG-RAG-10k (SpixelFCN)	0.559	0.625	0.518	0.756	0.786
GIG-RAG-50k (SpixelFCN)	-	-	-	-	-
GIG-24-Unique	0.624	0.652	0.456	0.777	0.795
GIG-80-Unique	0.625	0.656	0.449	0.781	0.791
GIG	0.620	0.689	0.508	0.788	0.820
GIG-Edge	0.613	0.674	0.487	0.768	0.798
GIG-Positional	0.599	0.619	0.465	0.742	0.751
GIG (filtered)	0.645	0.715	**0.556**	0.832	**0.885**

result, where the GIG gradient presented thicker contours while retaining most of its details.

As shown in Table 4.4, this operation resulted in better segmentation, indicating that thick contours are crucial to the task. Also, despite the overall improved RCF results, the GIG filtered representation outperforms RCF in the AP. It is comparable or better in the PRI metrics, which ponder areas without consent among the annotators, such as the small details better captured in GIG. Fig. 4.14 illustrates the segmentations with highlights on some critical areas.

A statistical analysis for GIG to validate the segmentation results is presented in Fig. 4.15 scatter graphics to illustrate. GIG is better than the best edge maps methods (SED, HED, and GIG-Positional) with statistical significance ($p-$values $< 10e-17$) and comparable to the GIG-Unique representations ($p-$values $\sim$0.02). GIG is statistically better than GIG-Edge ($p-$value $< 10e-14$) despite both presenting similar segmentation metrics. The RCF comparison is with the GIG filtered version, which was still inferior to



(a) Ground truth (b) RCF (c) GIG (d) GIGfiltered

Figure 4.14: Samples of the segmentations with 10 regions produced by RCF, GIG and GIG filtered by the morphological operation. The black areas in the ground-truth (a) indicate regions without consent among the annotators. Overall, the RCF (b) boundaries between regions are cleaner. GIG (c) and its filtered version (d) have more details from the object along with some background information. Also, the filtered version is more concentrated in certain details and areas of the background. In the second row, region highlights to illustrate the remarks.

the RCF ($p-\text{value} < 10e-9$) despite the improvement from GIG.

Final considerations on the extended formalism

The extended formalism presented three strategies to explore larger image areas and make changes driven by the RF mechanics to achieve a well-considered learning framework operating on graphs. It demonstrated that reducing the number of data points by grouping the image pixels before the graph creation reduced the training time but compromised the performance on the edge detection and segmentation tasks. Also, expanding the analysis region by removing redundancy yielded similar results to the original proposal, indicating that the initial area of study already captured the necessary information, and the redundancy did not diminish the RF generalization in this particular application. Finally, the strategy that considered the position of the features regarding the RF mechanism resulted in better results in the edge detection task. Overall, it validated the original selection of attributes, where GIG was superior to the grouped pixels and the positional strategies and equivalent to the extended region.

Edge maps as gradients are commonly used as a preprocessing step in many applica-

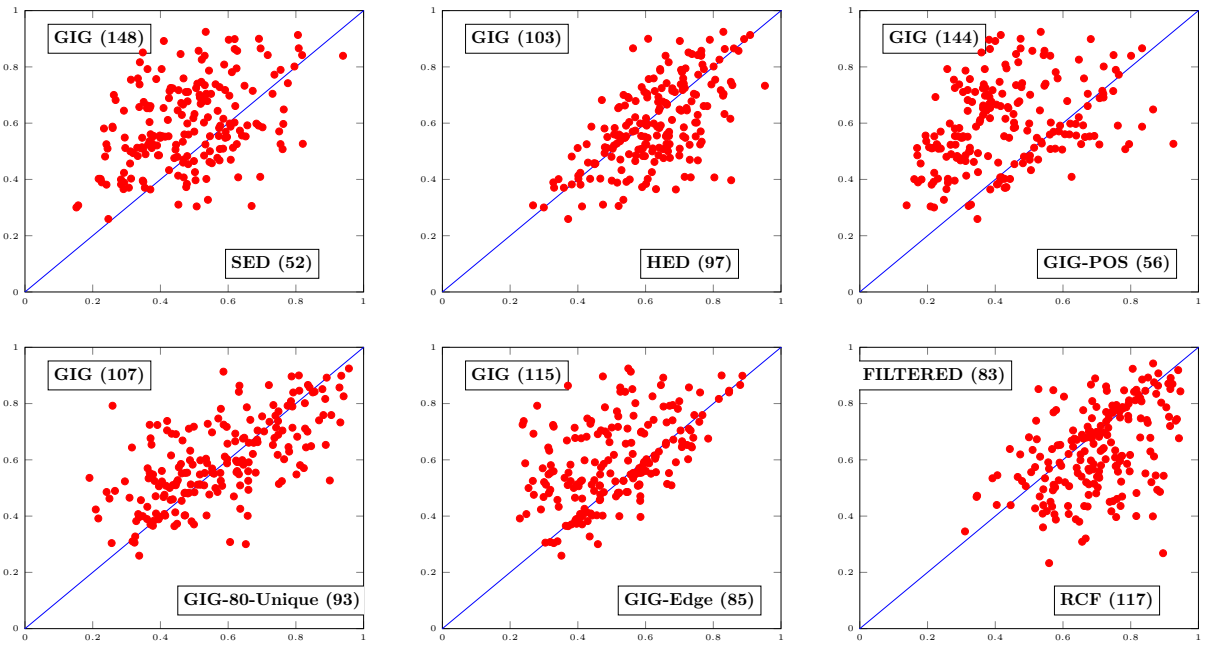


Figure 4.15: Pair-wise comparison of the $F1-$measure results (red dots) on the best scale for each method. The boxes' values are the number of images that are better for a particular method. GIG-Positional named as GIG-POS and GIG filtered as FILTERED.

tions because they are fast to compute and usually facilitate image analysis. Knowing the application and the type of analysis, one should always consider if the contour-oriented image simplification is enough for the task. The experiments proposed an application on the hierarchical watershed. It is important to notice that the core of the hierarchical watershed resides on the cost value and the markers representing the topology of a region, both extracted from the image magnitudes.

From this perspective, the success of this method relies on a good gradient image that reflects the distribution of the original image. The usual gradients with well-delineated contours provide clear extreme values for ranking, but these values are constantly contrasted with neighboring regions. Therefore, the depiction of uniformity and small details in conjunction with strong contours could provide additional context.

The results showed that better edge maps do not necessarily translate to better segmentation. For instance, gradients with fuzzy contours like those produced by SED and HED are not the best candidates, despite having good metrics on the edges. The small GIG-RAG representations are primarily gradients of regions instead of edges, yet, their segmentation results are better than one could imagine, except for the precision metric.

HED and GIG-Edge have similar segmentation metrics but distinct gradients: HED has fuzzy, thick contours with little details, while GIG-Edge has thin, detailed contours. GIG performed better than both in all metrics: GIG shares the thick contours but not the fuzziness with HED and shares the details with GIG-Edge, plus additional information about patterns.

To identify which characteristics command the improvement, one could examine the GIG-Unique methods, which also performed better than HED and GIG-Edge. GIG and GIG-Unique share sharp contours, details, and information patterns, indicating that these are the factors that differ and improve from HED and GIG-Edge. In contrast, GIG performs better than GIG-Unique and is varied by the thicker contours. The importance of the details and pattern information is evident in the results obtained with GIG-RAG from SLIC and SpixelFCN. While a larger number of superpixels with SLIC arrives at more contour-oriented gradients, the pattern information preserved with SpixelFCN gives superior segmentation metrics and values comparable with SED and HED, even with as little as 1k regions.

Overall, gradients with thick, sharp contours perform better. But as indicated in the qualitative analysis and the precision and PRI metrics for the filtered GIG compared to RCF, additional details and information about uniform regions positively contributed to the segmentation results regarding small objects and uniformity.

4.5 Case study discussion

Using the available information on the graph edges and vertices is a viable method to represent the image graph in a learning framework. It allows controlling the representation size and selecting the information depicted considering the type of graph, its proximity to the original data, and the expected results. Furthermore, representing the graphs at the vertex level allows maintaining the analyses on the discrete space by assigning a single label for an entry. This assignment is particularly advantageous with the graph strategy since it represents an entire region on a single vertex, and the task makes no assumptions about the media.

The RF as the learning paradigm is fast to train with the proposed representation in the discrete analysis space and, therefore, ideal for the experimental pipeline that requires the investigation of multiple aspects. Also, the RF paired with the edge-weights gradient operator acts as a regularizer diminishing noise, accentuating strong connections, and

mitigating any eventual poor topology choice. Furthermore, mapping the RF predictions back to the image space in the form of image gradients allows the evaluation of the results qualitatively and quantitatively.

A quality assessment of the topology choices addressed the considerations about the type of graph and its proximity to the original media. From the framework perspective, all attributes are just sets of values stored on the vertices and edges of the graph. But conceptually, the image graph creates a unique transformed space close to the spatial domain of the images, strengthened with relational aspects on the edges of the graph.

The experimental investigation on the attribute selection established that representing larger regions through the neighboring size and the number of connections with the adjacency relation translates into higher confidence values on the edges and less noise in the resulting images. Also, the weighting function must characterize similarities in the original data to be descriptive. The adjacency relation and the weighting function are modeling choices, conditioning the interaction between the data and the graph. The RF regularization mitigates most poor topology choices, except when the input is extremely noisy and correlated. The final assessment of the attribute selection regards the vertex attributes representing low-level descriptors of the image. Including the vertex attributes in the regular representation makes a direct reference to the media but results in less noise, stronger borders, and more details on the final image gradients, hence crucial to the practical applications assessed.

For the rest of this document, GIG refers to the image gradients obtained by mapping the predictions of the RF, trained on edge detection labels, and receiving the regular representation of the selected attributes of the edge-weighted graphs as input. GIG's gradients are generally very descriptive, with firm contours of the objects and other characteristics such as minor components, textures, and large uniform regions.

Gradients are commonly used as a preprocessing step in many applications because they are fast to compute and usually facilitate image analysis, particularly for the segmentation task. Compared with other popular gradient strategies, GIG's gradients, as input for the watershed hierarchies segmentation method, produced better-segmented images than traditional gradient methods like SED, Sobel, and Laplace. Comparing it with more elaborated edge maps, like the ones made by deep approaches HED and RCF, demonstrated that the segmentation task's performance depends on the characteristics portrayed on the gradient. Overall, better segmentations result from gradients with thick and sharp contours and additional details that contribute to identifying small objects,

and information about uniform regions provides consistency.

Regarding the assessment of the extended formalism, exploring larger regions did not yield better results or gradients, indicating that the initial attribute selection already captured the necessary information, and the redundancy did not diminish the RF generalization in this particular application. The strategy addressing the question of missing values in the regular representation, created by the vertices of the image's border, directly influences the feature connotation considering the RF mechanics. Therefore, addressing this aspect improved the results of the edge detection task.

Finally, regarding the performance of edge detection (the task the model is trained on), the results could be better compared with the deep methods or SED. However, observing the outputs showed that the procedures that perform better on the task are the ones with thicker contours. This is a result of the evaluation method proposed for the dataset, which compares each pixel with the multiple ground-truth pixels. Therefore, the representations with a more significant margin on the contours are more likely to match the ground-truth. Because GIG is centered on the analysis of the vertices, the more confident the RF is in distinguishing a vertice as a contour from its surrounding vertices, the more precise the predictions are, resulting in thinner contours. Nonetheless, the other aspects portrayed on the gradients, such as the large uniform regions and simplified patterns, could be considered a failure on the task, even if beneficial for other applications and descriptive of the properties relayed by the graphs.

Part III

Learning on hierarchies

Chapter 5

Learning on hierarchical attributes

Thus far, this thesis has shown that: (i) Hierarchies are rich structures that require careful considerations when applied to a task (Chapter 1); (ii) There is great interest in the literature to integrate hierarchies and machine learning in the same framework, but the application could also be challenging and dependent on the media and task (Chapter 2); (iii) Graphs are dynamic structures for modeling multimedia, but like hierarchies, require thoughtful considerations when applied in a machine learning framework (Chapter 3); and (iv) Aggregating regular representations of graph attributes with Random Forests create a viable pipeline that is generic for the task when the label attribution is at the graph components level but dependent on media features for a good performance (Chapter 4).

This chapter presents the culmination of the proposals, expanding the concepts and strategies to the hierarchical data. It delivers a learning framework operating directly on the hierarchies, focusing the formulations on the structural components. The pipeline, illustrated in Fig. 5.1, is similar to the case study, but instead of selecting graph attributes, it creates the regular representation from the hierarchical structure attributes.

The hierarchical construction in this chapter contemplates the same hierarchies described in Chapter 1, with similar nomenclatures: QFZ, ALPHA, WATER-AREA, WATER-DYN, WATER-VOL, and WATER-PAR. It presents two strategies for selecting attributes from the hierarchical structures for regular representation. The first one, shown in Section 5.1, uses the topological properties taking the hierarchical trees as inputs. The second one, in Section 5.2, computes regional features deduced from the hierarchies and their conjoined graph.

Each strategy section presents the representation, a discussion about its properties, and experiments on the edge detection and segmentation tasks. The topological approach provides additional information about the distribution of values in the hierarchies that could be valuable to understand the structures within the learning framework. Therefore,

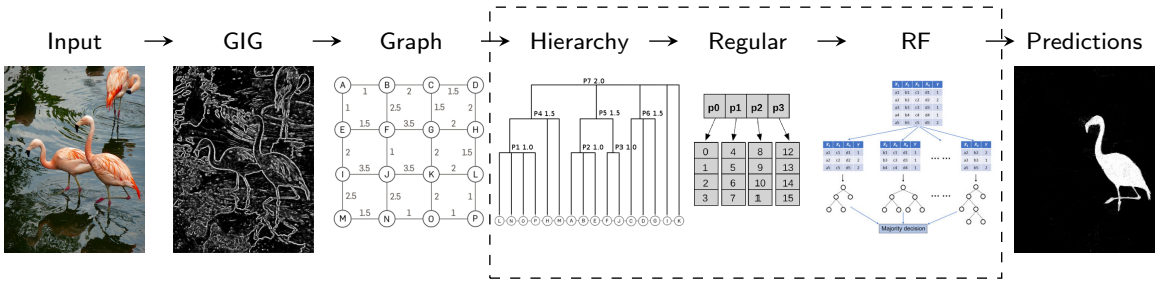


Figure 5.1: Figure illustrating the framework from the input image to the Random forest predictions performing the task. First, it computes the GIG gradient for each image in the dataset. Then, it calculates the edge-weighted graphs, here illustrated with the $4-$adjacency relation. The next steps constructs 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 leaf of the tree. 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.

Section 5.1.1 presents a data analysis of this representation. Section **??** offers experiments combining different attributes and evaluations on both tasks for completeness. Finally, Section 5.3 presents a brief discussion of the main findings in this chapter.

5.1 Topological attributes

This section presents a strategy to create a regular representation of the hierarchies to be applied in the RF by selecting topological properties from the hierarchical trees.

Some methods presented in Section 2.1.1 propose similar approaches. For instance, in E. Grossiord et al. (2017) [1], 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, T. Shi, et al. (2021) [2], 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) [3], besides using the hierarchies to model the correspondence between different media, they also define the features to be applied in a Random walker method. Precisely, each represented region

1. Grossiord Eloise et al. (2017). *Automated 3D lymphoma lesion segmentation from PET/CT characteristics.*
2. Hu Zhongwen, Shi Tiezhu, et al. (2021). *Scale-sets image classification with hierarchical sample enriching and automatic scale selection.*
3. Padilla Francisco Javier Alvarez et al. (2021). *Random walkers on morphological trees: a segmentation paradigm.*

in the hierarchy is characterized as a set of attributes describing the relative distance between a parent and a node, the number of components within the region, the barycenter value, and the region compactness metrics. 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. Their final representation is therefore suited for that task and that media (segmentation on PET/CT images).

The proposal presented here follows a different direction. First, it avoids any feature extracted from the media and only uses the information on the hierarchical tree. 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.

To be precise, 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 $G(V, \mathcal{F})$ has a set of nodes $\mathcal{N}$, and 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 topological characteristics representing a node $n \in \mathcal{N}$ by one of the following attributes:

— **Altitude**: the value inversely proportional to the depth of the node n.

$$\mathtt{alt}_n = 1/d_n$$

— **Area**: sum of the number of leaves on the subtree τ_n rooted on the node n.

$$\mathtt{area}_n = |\{\mathcal{L}_n\}|\,, \text{ for } \mathcal{L}_n = \{l \,|\, \forall\, l \in \tau_n\}, \mathcal{L}_n \subseteq \mathcal{L}$$

— **Volume**: in a tree is a value computed recursively, pondering the area, relative altitude regarding its parent $\mathtt{par}_n$, and the sum of volumes of all nodes in the subtree τ_n rooted in the node n. The volume of a leaf node is 0.

$$\mathtt{vol}_n = \mathtt{area}_n \times |\,\mathtt{alt}_n - \mathtt{alt}_{\mathtt{par}_n}\,| + \sum \mathtt{vol}_c\,, \text{ for } c \in \mathtt{children}_n$$
$$\mathtt{vol}_n = 0 \text{ if } n \text{ is } l \in \mathcal{L}_n.$$

— **Dynamics**: correspond to the extinction value $\mathcal{E}_n$ for the height of a node n. The extinction value is a measure of each local minimum for a regional attribute. The attribute height is the difference between the altitude of the node parent $\texttt{par}_n$ and the altitude of the deepest non-leaf node in the subtree τ_n rooted on the node n. In a tree with increasing altitudes, the node extinction value $\mathcal{E}_n$ for the height corresponds to a threshold value such that n remains minima when all nodes with heights smaller than the threshold are removed. Intuitively, the extinction value indicates when a region will be merged into another region.

$$\texttt{dyn}_n = \mathcal{E}_n \text{ for the attribute height}$$

In $\mathbf{T}_l$, 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 smaller set of parents than the maximum depth receive a padding value of -1 because the attributes considered for the selection have all positive values. To be precise, the range for the attributes `alt` and `dyn` is $[\,0, 1\,]$, and for `area` and `vol` is $[\,0, |\mathcal{L}|\,]$.

To keep the semantical meaning in the regular representation, the attributes representing the parents of a leaf node are put in the *order* they appear transversing the hierarchical tree. The order could be *ascending* (from leaf to root) or *descending* (from root to leaf).

From the case study in GIG representation, there is the question of label attribution. Considering that there is a direct correspondence between the set of vertices in the graph and the set of leaves in the tree, the same assumptions about a single label for an entry agnostic for the task are also valid for this part of the study. However, while the assignment on the graph allowed representing an entire region on a single vertex, on the hierarchy, the single label represents multiple regions that share a path on the tree.

Definition 14: Regular representation of hierarchical topological attributes

The proposed regular representation on **topological attributes** $\mathbf{T}_\mathcal{H}$ of a hierarchical tree $\mathcal{T}_\mathcal{H}$ in the set $\mathbb{T}$ of all hierarchies in a dataset is: $\mathbf{T}_\mathcal{H} = ((\mathbf{T}_1, \mathbf{Y}_1), \ldots, (\mathbf{T}_{|\mathcal{L}|}, \mathbf{Y}_{|\mathcal{L}|}))$. In $\mathbf{T}_\mathcal{H}$, 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 $\texttt{topo} \in \{\texttt{alt}, \texttt{area}, \texttt{vol}, \texttt{dyn}\}$ 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 $\mathcal{T}_\mathcal{H} \in \mathbb{T}$. If $|\mathbf{P}_l| < p_t$, a padding value -1 fills the remaining positions in the vector $\mathbf{T}_l$.

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. For the test instances, the procedure takes the regular representation of each hierarchy in the test set and individually subjects them to the RF estimations without the labels.

Algorithm 2 describes the steps to create and store the regular representation on topological attributes for both the training input $\mathcal{D}_t$ and the individual test instances. For clarity, the operations not detailed in Algorithm 2 are:

— *getDepth*(hierarchical tree): gets the maximum depth in the hierarchical tree.

— **max**(list): retrieves the maximum value in a list of values.

— **empty**[[nrows, ncols]]: allocates an array memory space of size number of rows by the number of columns.

— **getLabel**(leaf): gets the ground-truth label for a leaf node.

— **negativeOnes**[[*size*]]: creates an array filled with the negative one value of size *size*. The operation plays a dual function: it allocates the necessary memory and acts as a padding value outside the valid range of the candidate attributes for the leaf whose set of parents is smaller than the maximum depth in the entire dataset.

— *getParents*(leaf): gets the list of parents for a leaf node.

— *invert*(list): invert list order.

— *getAttribute*(attribute, node): computes the node specified attribute.

— **index**(element, list): retrieves the index of an element on a list.

— *append*(array): appends array vector to another array row-wise.

The following sections will investigate this representation where: (i) Section 5.1.1 inspects the question about the representation order regarding the way to transverse the hierarchical tree and presents a data analysis of the feature distribution in the topological representation; and (ii) Section 5.1.2 shows some experiments, evaluating the proposed strategy in the image tasks.

Algorithm 2: Regular representation topological attributes

Input : $\mathbb{T}$: a set of hierarchical trees computed for a dataset, a parameter `topo` $\in \{$`alt`, `area`, `vol`, `dyn`$\}$ indicating the topological attribute to be computed, and a flag *order* indicating if the representation is from root to leaf (*descending*) or leaf to root (*ascending*).

Output : $\mathcal{D}_t$: a regular training input for the learning framework, and a regular representation $\mathbf{T}_{\mathcal{H}}$ for all $\mathcal{T}_{\mathcal{H}}$ in the test set.

Function *getRegular*$(\mathcal{T}_{\mathcal{H}}, p_t)$**:**

$\mathbf{T}_{\mathcal{H}} =$ **empty**$[[\,|\,\mathcal{L}\,|\,, p_t + 1\,]]$
if $\mathcal{T}_{\mathcal{H}}$ *isTrainInstance* **then**
  $\mathbf{Y} =$ **getLabel**(l) **for all** $l \in \mathcal{L}$
  $\mathbf{T}_{\mathcal{H}}[[:, p_t + 1\,]] \leftarrow \mathbf{Y}$ `// leaves labels at added column at the end`
else $\mathbf{T}_{\mathcal{H}} = \mathbf{T}_{\mathcal{H}}[\,|\,\mathcal{L}\,|\,, p_t\,]$
for $l \in \mathcal{L}$ **do**
  $\mathbf{T}_l =$ **negativeOnes**$[[\,p_t\,]]$
  parentList $=$ *getParents*(l)
  if *order*==*descending* **then** parentList=*invert*(parentList)
  for `par` $\in$ *parentList* **do**
    $\texttt{att}_p =$*getAttribute*(`topo`, `par`)
    $\mathbf{T}_l[[$**index**(`par`, parentList)$]] \leftarrow \texttt{att}_p$
  end
  $\mathbf{T}_{\mathcal{H}}[[\,leaf, :]] \leftarrow$ *append*$(\mathbf{T}_l)$
end
return $\mathbf{T}_{\mathcal{H}}$

Main:

for $\mathcal{T}_{\mathcal{H}} \in \mathbb{T}$ **do**
  depthList=*getDepth*$(\mathcal{T}_{\mathcal{H}})$
end
$p_t =$ **max**(depthList)
$\mathcal{D}_t = [[\,]]$
for $\mathcal{T}_{\mathcal{H}} \in \mathbb{T}$ **do**
  $\mathbf{T}_{\mathcal{H}}$=*getRegular*$(\mathcal{T}_{\mathcal{H}}, p_t)$
  if $\mathcal{T}_{\mathcal{H}}$ *isTrainInstance* **then**
    $\mathcal{D}_t \leftarrow$ *append*$(\mathbf{T}_{\mathcal{H}})$
  else *save*$(\mathbf{T}_{\mathcal{H}})$
end
save$(\mathcal{D}_t)$

5.1.1 Data analysis: Topological representation

Understanding the distribution of the values using the topological representation is beneficial to guide the decisions regarding the learning step and better comprehend the hierarchical structure.

Representation order assessment

The first assessment regards the order of the regular representation on the leaves. The goal is to keep the semantical meaning in the hierarchical tree by preserving the order modeled in the structure. However, the set of parents for a leaf could either be taken from root to leaf (descending) or leaf to root (ascending).

It is crucial to notice that another consequence of the multiformity in the hierarchies is that different leaves have a variable amount of parents on the structure. Furthermore, it may not be an alignment between the feature position in the regular representation and the parent position in the hierarchical tree. For instance, in the ascending order, the first parent of a leaf node occupying the first position on the feature vector could be close to the root if, in its path, there are few hierarchical levels. While for another leaf node, there are many levels in the path, and the parent with a semantic equivalence with the first one will be many positions later in the feature vector. The GIG extended experiments in Section 4.4.3 indicated that the RF performs better when there is a meaningful correspondence between the features and the position they assume in the regular representation.

To investigate this aspect and select an order for the topological approach, the experiments in this section train a RF classifier with 150 estimators and the regular representation created with both orders, referred to as Ascending and Descending. The labels for the training set are for the binary segmentation task on the Birds dataset—the smallest and more critical collection of images in the experiments with the typical pipeline (Section 1.6). It trains one model for each hierarchy (named ALPHA, QFZ, WATER-VOL, WATER-PAR, WATER-AREA, and WATER-DYN) and each attribute in question: altitudes (named alt), area, dynamics (named dyn), and volume (named vol). The classifier predictions for each leaf on the binary segmentation labels are directly mapped back to the image space for evaluation. The evaluation metric is the Jaccard score.

Table 5.1 presents the results obtained with each representation for each topological attribute grouped by the representation order in question. The results demonstrate that

Table 5.1: Quantitative results (Jaccard score metric for the classifiers predictions) on the segmentation task. Comparison of the results obtained with the two representation orders on the different hierarchical types for the proposed attributes. Bold values are the best between the two and red the best among all. Perfect score=1.

		Ascending				**Descending**			
Hierarchy	Dimensions	Alt	Area	Dyn	Vol	Alt	Area	Dyn	Vol
QFZ	106	**0.139**	**0.308**	**0.154**	**0.263**	0.070	0.148	0.123	0.104
ALPHA	13	**0.167**	**0.177**	**0.165**	**0.175**	0.153	0.153	0.140	0.142
WATER-VOL	232	**0.196**	**0.256**	**0.190**	**0.240**	0.062	0.172	0.132	0.149
WATER-AREA	82	**0.203**	**0.208**	**0.165**	**0.210**	0.178	0.179	0.149	0.170
WATER-DYN	81	**0.178**	**0.303**	**0.173**	**0.278**	0.063	0.109	0.137	0.165
WATER-PAR	66	**0.222**	0.240	**0.217**	**0.209**	0.197	**0.270**	0.111	0.166

the ascending order produces a better model for this strategy.

Although a complete interpretation of the features and values in the RF model may be challenging due to the size of the model and the data, some valuable clues could be excerpted by probing the model nodes. Therefore, to provide some clarity on the relation between the model and the order of the features on the training data, it is proposed to inspect the feature position importance on the decision nodes of the model and probe the values used for the split to investigate how one distribution favors the learning step. Specifically, after training, it inspects the model Gini importance of a feature position, computed as the normalized total reduction of the criterion brought by that feature. Furthermore, it gathers per feature position the value used for the split in every split node and every tree in the forest.

Fig. 5.2 presents some selected representations for this dataset. Namely: (i) QFZ-area (the best result); (ii) WATER-VOL-altitudes (the worst result); and (iii) WATER-PAR-area (the only one with descending order performing better than ascending).

The charts in Fig. 5.2 show that essential features for the classifier in the ascending order occur at the initial positions of the feature vector for all representations. It indicates that most decisions are based on the values at the bottom of the hierarchical structure. For the descending order, the importance distribution has a normal shape, occurring at different stages for different hierarchies. Presumably, the importance also favors lower levels of the hierarchical tree, but the features start to represent them later in the vector.

Interestingly, when contrasted with the value distribution, it is possible to verify that the decision nodes in the ascending order favor valid values in the representation and, at large, ignore the padding values. The same is not true for the descending order, where the

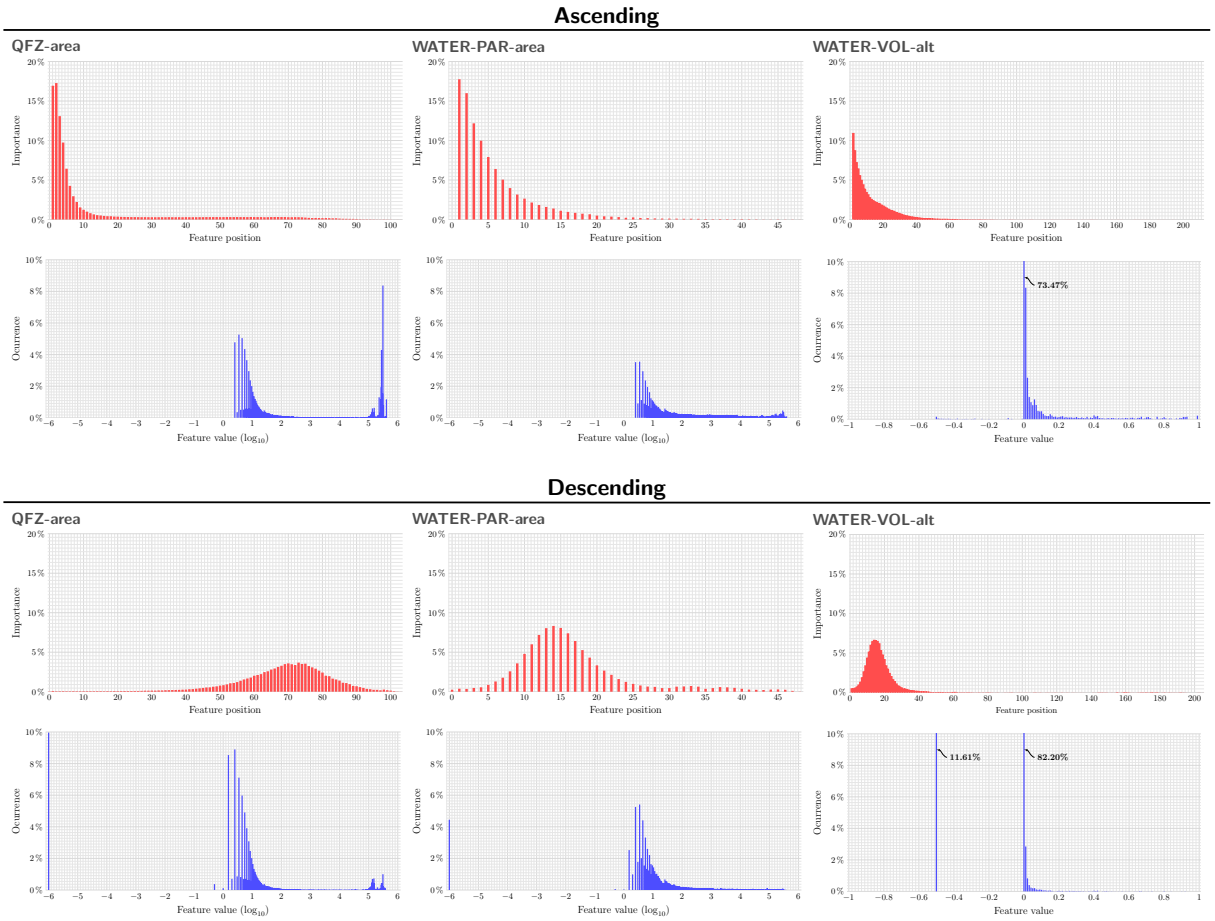


Figure 5.2: Illustration of the feature importance and data value distribution inside decision nodes of the RF. Red charts (top-rows) are for the feature importance in the model, showing in percent the importance a feature has on the RF decisions. Blue charts (bottom-rows) are the value frequency occurrence in a decision node for the split, where -1 indicates padding values for the altitude attribute and -6 for the area (area presented in log_{10} scale for better visualization).

padding value frequently occurs in all charts in the decision nodes. WATER-PAR-area is the only representation in descending order that performs better than ascending and the one where the padding value is less frequent in the model (4.23%) among all types.

Another critical observation regards the value distribution on the worst representation for this dataset (WATER-VOL-altitudes). Besides representing the largest dimension, most of the model's decisions are made in a very small range of values. The following section will assess the data distribution of each representation and attribute.

The other results that are not illustrated held for the observations regarding the representation order and distribution. Therefore, for the remainder of the experiments with the topological strategy, explore only the **ascending order**.

Table 5.2: The table shows the training data overview per dataset. It contains the total number of leaves (#—training sample), the feature vector dimension (Dim—maximum depth in dataset), and the percent of padding values in the regular representation (Padding %—number of positions in the feature vector where parent set is smaller than maximum depth).

	BSDS			Birds			Sky		
Hierarchy	#	Dim	Padding(%)	#	Dim	Padding(%)	#	Dim	Padding(%)
QFZ	30,880,200	149	25.46	10,331,520	107	17.42	10,390,186	101	18.53
WATER-DYN		122	37.16		82	20.61		79	21.25
ALPHA		18	79.99		14	74.47		15	75.40
WATER-VOL		239	78.80		233	73.75		241	64.51
WATER-AREA		80	69.85		83	65.98		76	57.41
WATER-PAR		69	65.51		67	58.05		58	51.77

Data distribution: topological representation

Individual inputs will construct a different hierarchy, but analyzing the values in the training set gives a notion about the global distribution.

Table 5.2 presents a summary of the training data per input dataset. Each leaf on the hierarchy corresponds to a vertice on the graph, which in turn corresponds to a pixel on the image. Creating the training data on the leaf level produces sizable data sets. The RF mechanics allows to work with such dimensions, but a different model would require a strategy to deal with the sample size.

Regarding the feature vector dimension, the length varies between the hierarchical types but is partially constant for the same type between the datasets. The most abrupt change happens for the BSDS500 dataset and the contour-oriented representations QFZ and WATER-DYN if compared to the other two datasets. This indicates that more hierarchical levels were constructed, which could result from the highly patterned images.

The two extremes in the feature vector size are ALPHA and WATER-VOL. At one end, ALPHA has a maximum of 18 dimensions, while WATER-VOL presents over 230 levels (which could pose difficulties for the learning process). Interestingly, both types represent the most significant amount of padding values, where over 70% of all values are padding. Other hierarchical types also have a considerable amount of padding, namely WATER-AREA and WATER-PAR (more than half of all values). The only two types with more valid values than padding are QFZ and WATER-DYN. The excessive padding indicates a large variety on the hierarchical levels, which is not surprising but can negatively impact the learning process. Inspecting the data by feature position shows that only the first 3 to 5 positions (number of parents) do not contain any padding.

Regarding the distribution of valid values, Fig. 5.3 illustrates, for all hierarchies and

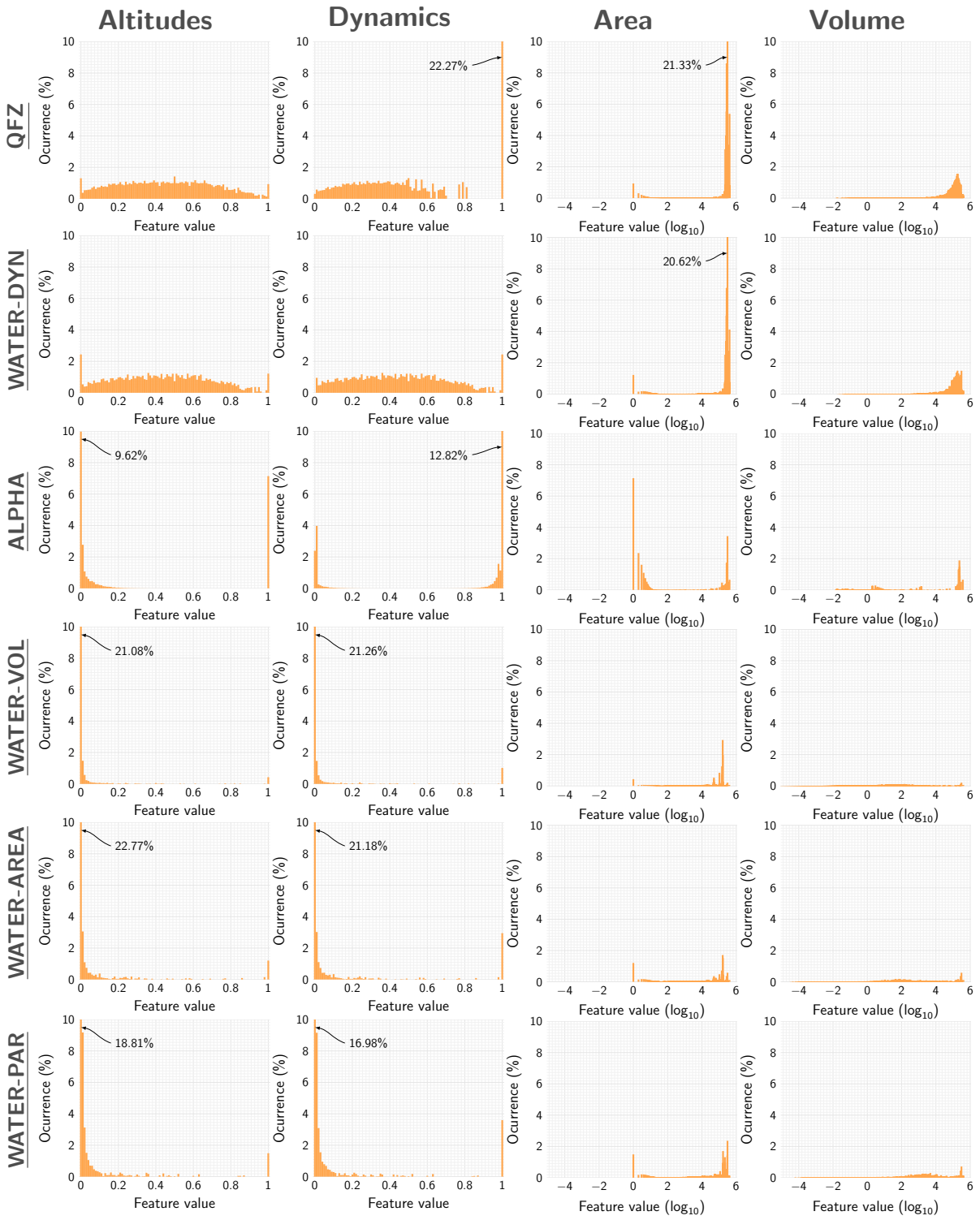


Figure 5.3: Visual distribution of the values distribution on the Birds dataset. Presents all considered hierarchical types and attributes.

attributes considered, the values' occurrence in the Birds' training set, regardless of their position on the feature vector. The distribution portrayed considers only the valid values without the padding.

Most values have minor occurrences in all representations (less than 2%), which is not a problem if the model could capture these unique values for the prediction. QFZ and WATER-DYN present similar contour maps; as expected, the distribution of their values is also similar for all attributes except dynamic. In WATER-DYN, higher values of the dynamics attribute (above 0.8) are better distributed than in QFZ, indicating that WATER-DYN construction still discerns regional minima areas at higher levels, and QFZ has the steepest climb close to the root. These differences could be perceived in their contour maps, where QFZ have the sharpest edges on the main contours (see Fig 1.4 for an example).

WATER-VOL, WATER-AREA, and WATER-PAR all have similar contour maps and similar distributions by attribute. The most discernable is WATER-PAR, with slightly more evenly distributed values. There are some similitudes in the ALPHA and WATER-VOL structures; for instance, both present most of their values at lower altitude levels, but their extinctions values are disparate. A closer look at the area attribute shows that most of the regions in ALPHA are either very small or very large, while in WATER-VOL, it increasingly accumulates towards larger areas. In fact, WATER-VOL has the smoothest distribution of area among its peers (WATER-PAR and WATER-AREA).

The experiments in the typical pipeline (Section 1.6) showed that individual inputs from different datasets and hierarchical types have very distinct structures to analyze with the horizontal cut strategy. However, when grouped in a single set, the different hierarchies present the same behavior for various attributes.

The observations in this section made for Birds could also fit the other two datasets. Table 5.3 shows a summary of the distribution of the values for all hierarchical types and considered attributes, and as one can see, despite their visual differences, the training set of data is very similar between datasets.

The BSDS500 is distinguished from the others by the number of unique values. Still, it also has three times more samples, and the distribution overall is similar by the mean and standard deviation. Another distinction regards the area attribute on the Sky dataset, in which the mean and deviation are considerably higher than the others, especially since the number of unique values is a par with the other similar-sized dataset. This indicates that the construction algorithms are grouping larger areas under the same node, which is

Table 5.3: The table presents distribution values for the topological approach for all datasets, hierarchical types, and considered attributes. It offers the number of distinct values (unique) , the mean, and the standard deviation (std) for a particular representation. Distinguished areas are highlighted.

Altitudes

	BSDS			Birds			Sky		
Hierarchy	Unique	Mean	Std	Unique	Mean	Std	Unique	Mean	Std
QFZ	6,218	0.458	0.272	2,195	0.427	0.257	2,359	0.418	0.254
WATER-DYN	5,594	0.414	0.259	1,362	0.432	0.263	1,605	0.414	0.258
ALPHA	3,154	0.235	0.146	1,395	0.273	0.169	1,516	0.271	0.170
WATER-VOL	163,014	0.005	0.029	38,425	0.005	0.030	37,982	0.002	0.018
WATER-AREA	45,967	0.018	0.057	11,383	0.015	0.056	10,656	0.009	0.039
WATER-PAR	23,295	0.039	0.085	5,674	0.029	0.074	5,713	0.029	0.071

Dynamics

	BSDS			Birds			Sky		
Hierarchy	Unique	Mean	Std	Unique	Mean	Std	Unique	Mean	Std
QFZ	4,825	0.371	0.235	1,632	0.321	0.199	1,822	0.328	0.206
WATER-DYN	5,593	0.414	0.259	1,361	0.432	0.263	1,604	0.414	0.258
ALPHA	5,033	0.515	0.332	2,237	0.508	0.311	2,410	0.508	0.314
WATER-VOL	163,013	0.005	0.029	38,424	0.005	0.030	37,981	0.002	0.018
WATER-AREA	45,966	0.018	0.057	11,382	0.015	0.056	10,655	0.009	0.039
WATER-PAR	23,294	0.039	0.085	5,673	0.029	0.074	5,712	0.029	0.071

Area

	BSDS			Birds			Sky		
Hierarchy	Unique	Mean	Std	Unique	Mean	Std	Unique	Mean	Std
QFZ	21,051	56,279	59,354	8,003	91,001	126,990	8,706	77,596	101,303
WATER-DYN	20,372	51,835	57,413	7,587	72,631	116,024	8,331	65,034	93,319
ALPHA	2,070	6,214	14,253	898	8,743	41,102	899	13,716	44,769
WATER-VOL	30,138	27,559	27,245	14,510	22,901	33,002	16,060	46,176	61,011
WATER-AREA	17,837	19,959	23,838	9,340	16,567	29,684	8,943	28,003	49,108
WATER-PAR	18,470	21,121	25,357	9,392	18,427	40,202	8,881	24,216	45,600

Volume

	BSDS			Birds			Sky		
Hierarchy	Unique	Mean	Std	Unique	Mean	Std	Unique	Mean	Std
QFZ	187,316	7,304	23,633	62,928	5,966	31,109	65,689	5,335	26,558
WATER-DYN	165,729	5,109	19,195	48,941	5,912	31,676	51,523	5,430	27,345
ALPHA	108,255	320	5,637	42,573	255	7,030	42,590	322	7,427
WATER-VOL	1,276,149	45	1,689	351,126	76	3,456	304,232	63	3,043
WATER-AREA	530,230	113	2,572	149,179	174	4,996	128,934	158	4,706
WATER-PAR	361,415	187	3,287	103,549	264	6,339	91,036	270	5,863

interesting giving the nature of the dataset task.

Overall, the quantitative analysis reinforces the visual one: (i) WATER-VOL, WATER-AREA, and WATER-PAR present small regions and variation (create various unique values with low mean and std), particularly for the attributes altitudes and dynamics; (ii) the three types have similar distribution among them, where WATER-VOL is more spread out than the others (considerably more unique values but similar mean and std); (iii) ALPHA creates fewer regions than the other hierarchical types (less unique) with abrupt changes on the tree levels (relatively large deviation); and (iv) the differences

between QFZ and WATER-DYN are subtle. One separate observation derived from the information in Table 5.3, is that the distribution for the attributes altitudes and dynamics are nearly identical for the region-oriented hierarchies.

Perhaps the most valuable analysis when preparing data for machine learning is correlation and co-variance, but given the size of the data, both in the number of samples and feature size, coupled with the variation on individual values, any meaningful analysis in those terms would be overwhelming. Opportunely, the RF model is lenient in its execution time and generalization capability, allowing further assessment through experimentation.

5.1.2 Experiments: Topological representation

This section shows the experiments with the topological approach on the two image tasks: edge detection in the BSDS500 dataset and segmentation in Birds and Sky. The pipeline takes the colored images in the datasets and computes the GIG gradient without any additional preprocessing of the images. Next, it constructs the graph with a $4-$adjacency and the Euclidean distance on the gradient for the weighting function.

The hierarchy construction explores the aforementioned hierarchies: QFZ, ALPHA, WATER-VOL, WATER-AREA, WATER-DYN, and WATER-PAR. It does not perform additional post-processing, such as filtering the hierarchies, realigning, or balancing the levels. The regular representation takes all the topological attributes considered for selection: altitudes, area, dynamics, and volume. The models are trained separately without a combination of hierarchical types or features. Like in GIG, the label attribution for the tasks is discrete and binary but at the leaf level instead of the vertex.

For the BSDS500, the pipeline uses a RF regressor as a model, where the average predictions are mapped back to the image domain for evaluation. It allows comparisons with other methods by using the proposed evaluation system for boundaries that takes an edge map and threshold the values in the range $[0, 1[$ with a 0.01 step computing the $F1-$score at all scales. Usually, the results are then presented in terms of optimal dataset scale, optimal image scale, and average precision. However, due to the number of variations considered and for clarity, the results presented in this section for the BSDS500 are only for the optimal dataset scale. It gives the score obtained in the threshold that best represents most images, averaging the predictions, which is the best to evaluate the overall performance and the most challenging.

For the segmentation datasets, the pipeline considers a RF classifier where predictions for each leaf on the binary segmentation labels are directly mapped back to the image

Table 5.4: Final parameters for the grid search results for the Random Forest : number of trees in the forest (est), the minimum number of samples to split an internal node (split), the minimum number of samples to be a leaf node (leaf), percent for the bootstrap sample size (samples), amount of sampled features for the split (feat - a function on the whole set of features), and the maximum depth of the trees (depth).

	Altitudes						**Area**					
Hierarchy	est	split	leaf	samples	feat	depth	est	split	leaf	samples	feat	depth
QFZ	40	5	4	10%	sqrt	None	500	10	2	10%	sqrt	100
ALPHA	20	10	20	50%	log2	10	300	10	20	10%	sqrt	10
WATER-VOL	300	10	20	10%	auto	100	300	2	20	10%	log2	100
WATER-AREA	50	10	1	10%	auto	10	80	2	2	75%	auto	10
WATER-DYN	100	2	4	10%	log2	50	70	5	10	50%	auto	None
WATER-PAR	500	5	20	All	sqrt	10	80	2	20	10%	sqrt	10

space for evaluation. In this case, the evaluation metric is the Jaccard score.

Random Forest parameters

For the Random Forest parameters, it performs a grid search using the Random Forest classifier for all hierarchies and the attributes: area and altitudes. The grid search evaluation takes the $F1-$score on the edge detection task in the BSDS500 dataset in the validation set. The RF parameters in consideration are: (i) number of trees in the forest; (ii) minimum number of samples to split an internal node; (iii) the minimum number of samples to be a leaf node; (iv) percent for the bootstrap sample size; (v) amount of sampled features for the split; and (vi) the maximum depth of the trees. Table 5.4 presents the final parameters for each representation.

Due to the number of variations and the similitudes in the data distribution, the final parameters for altitudes will also be used for dynamics, and the ones for the area be used for volumes. Also, the search contemplates only the BSDS500 because it is the most extensive set, with more unique values, and the only one provided with a validation set. The final parameters for BSDS500 will be used for the other datasets.

Quantitative analysis

Table 5.5 shows the results for all variations on the three datasets compared with the best scale on the typical trivial approach (cut by threshold on altitude levels and by the number of regions—Section 1.6) and the representations from graph attributes (from Section 4.4.3). The graph's results are the best with the GIG, the best with only the edge attributes (GIG-Edge), and the best with only color features (onlyColor).

Table 5.5: Quantitative comparison of the results obtained in all datasets for the graph representation, the typical, and the topological approach. $F1-$score for best dataset scale for the BSDS500 and average Jaccard score for Birds and Sky. Emphasizes the best scores per approach variation; a green highlight for the best values per dataset for the topological approach; and red emphasis on the best score among strategies per dataset. Perfect scores=1.

		BSDS				**Sky**				**Birds**			
Graph	GIG	**0.65**				**0.86**				**0.29**			
	GIG-Edge	0.61				0.78				0.27			
	onlyColor	0.64				0.85				0.28			
	Hierarchy	Threshold		Regions		Threshold		Regions		Threshold		Regions	
Typical	ALPHA	0.21		0.33		0.00		0.00		0.15		0.13	
	QFZ	0.26		0.28		0.45		0.01		0.30		0.05	
	WATER-DYN	**0.27**		0.42		0.79		0.13		0.29		0.15	
	WATER-VOL	0.20		**0.55**		0.89		**0.87**		**0.36**		**0.24**	
	WATER-AREA	0.24		0.53		0.83		0.83		0.30		0.22	
	WATER-PAR	0.24		0.53		**0.90**		0.86		0.32		0.24	
	Hierarchy	Alt	Area	Dyn	Vol	Alt	Area	Dyn	Vol	Alt	Area	Dyn	Vol
Topological	ALPHA	**0.63**	0.55	0.60	0.61	0.67	**0.81**	0.61	0.78	0.18	**0.18**	0.15	0.15
	QFZ	**0.60**	0.60	0.57	0.58	0.67	**0.90**	0.47	0.89	0.14	**0.37**	0.07	0.15
	WATER-DYN	**0.64**	0.62	0.62	0.63	0.67	**0.92**	0.47	0.90	0.17	**0.20**	0.06	0.11
	WATER-VOL	0.57	**0.58**	0.53	0.56	0.90	**0.96**	0.90	0.95	0.26	**0.37**	0.06	0.17
	WATER-AREA	**0.51**	0.50	0.46	0.48	**0.85**	0.82	0.84	0.82	0.26	**0.31**	0.18	0.18
	WATER-PAR	0.54	**0.54**	0.50	0.51	0.91	**0.95**	0.72	0.92	0.28	**0.41**	0.20	0.27

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

Regarding the individual hierarchical types, the most significant improvement from the typical is the ALPHA hierarchy in all datasets. Parsing and binarizing the ALPHA structures with horizontal cuts is very difficult. However, the regular representation with topological attributes captures enough information for the learning model to discriminate between classes. Furthermore, the number of padding values did not disturb the model performance in any hierarchical type. WATER-VOL with area attributes is even the best variation for the Sky dataset.

The attribute altitudes perform better on the edge detection and the area on the segmentation, which matches the task goals with the attributes' properties. Despite slightly worst results, the attributes dynamics and volume have similar values with their counter

attributes (altitudes and area, respectively).

Regarding execution times, the entire pipeline is very fast to compute. On average, creating the regular sets takes 500 seconds for all the +30 million leaves and the test instances on the BSDS500. Birds and Sky, with +10 million, takes 40 seconds on average. These time measures include the initial pass to retrieve the maximum depth in the dataset and the IO operations. In Algorithm 2, there are two nested for loops. However, they are included to facilitate the comprehension of the structure. In practice, all hierarchical implementations use the *Higra* python module [4], which provides fast functions that retrieve parents and attributes in real-time through efficient indexation.

The training time using the *scikit* [5] RF parallel implementation over 50 CPU cores is relative to the number of trees and representation size. On average, the training takes less than ten minutes for most variations. Nevertheless, the models with 300 or 500 can take as long as 2 hours to train. With a feature vector with 230 dimensions and 300 RF estimators, WATER-VOL takes almost 6 hours to complete the training. Above all else, the longest step on the pipeline is the external evaluation of the BSDS500. The benchmarked algorithm takes approximately 4 hours to process each variation.

The most critical aspect of the topological approach regards the computational resources required to process large sets of data. Most variations could be processed with a typical 8GB RAM machine, but the WATER-VOL representation with over 230 feature vector dimensions in all datasets and the BSDS500 with 30 million leaf samples regardless of the attribute (except ALPHA) demand over 20GB memory. WATER-VOL on BSDS500 requires over 30GB. Just storing the data on the CPU memory could be challenging. The RF random sampling makes it robust for this issue, but it may pose a problem for other models.

Final considerations on the topological approach

Representing the hierarchical structures by taking the entire set of parents of a leaf retains the semantical information embedded on the hierarchical trees without the need to filter or select a particular level for evaluation. Also, making the representation at the leaf level allows the discrete label attribution that does not demand considerations specific to a task. Furthermore, using only the topological attributes from the hierarchical tree structure allows for constructing a generic model for the media.

4. Perret B. et al. (2019). *Higra: hierarchical graph analysis.*
5. Pedregosa Fabian et al. (2011). *Scikit-learn: machine learning in Python.*

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. Particularly for the ALPHA representation, which in the trivial method had close to zero performance in all tasks and datasets on the topological approach, it is competitive with other contour-oriented types (with only a few dimensions).

Regarding the representation choices, the attributes altitudes and area presented the best results for the task most related to the information they portray. For the hierarchical types, overall, the contour-oriented representations give the best results in the edge detection task and WATER-VOL and WATER-PAR in the segmentation task.

The topological strategy constructs a regular representation that could be used in most available learning models. However, the dimensions of this representation could be challenging in terms of computational resources. The efficient implementations for hierarchical structures and the flexibility of the RF model allow working with these sizable structures. However, considering that most of the feature positions are filled with padding values, one may wonder if it is not possible to create another regular representation that also preserves the semantical structure but is reduced in size.

5.2 Regional attributes

The second strategy proposes using a set of regional attributes to represent the hierarchical structure instead of the topological ones. Procedurally, this approach is equivalent to performing horizontal cuts by altitude levels, but rather than creating a representation for each cut and evaluating them individually, the proposed method represents all of them systematically as a regular representation.

Equally to the topological approach, the regional strategy avoids any feature extracted from the media and only uses the information on the hierarchical tree and the conjoined graph. Similar methods in the literature (Section 2.1) use the region defined in the hierarchies to gather features from the media or even extract subparts of the input to bolster the learning model. By keeping the formulation on the structures, the proposed framework evades making any decision at the media level and relies on the already modeled data on the graphs and hierarchies.

Like before, the challenge in the framework is to create a regular representation that could be used by the learning algorithm while preserving the important information on the structures. The topological approach made a vectorial representation for each leaf

on the hierarchical tree, portraying its parents' topological attributes. In the regional process, it is proposed the same leaf-centered representation. However, each position on the feature vector portrays a selected characteristic of the region created by a horizontal cut on altitude levels that contains said leaf.

To be precise, 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 $G(V, \mathcal{F})$, denoted as the conjoined graph, has a set of nodes $\mathcal{N}$ and k levels. 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. $\mathcal{R}_{\mathcal{H}}$ is the set of regions of $\mathcal{H}$ and the union of all partitions of $\mathcal{H}$. A cut is a partition $\mathbb{P}$ of V made of regions of $\mathcal{H}$. A horizontal cut is a partition $\mathbb{P} = \mathbb{P}_i$ for $i \in \{0, \ldots, k\}$. 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 $\texttt{alt}_n > \sigma$.

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 $v \in V$. Consider β as a series of altitude levels to cut the hierarchy $\mathcal{H}$. 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 attribute selection corresponds to one of the following:

— **Area**: Represents the same concept as in the topological approach; however, the regional analysis defines the number of leaves on the region $\mathcal{R}_n$ created by the cut on node n.

$$\texttt{area}_{\mathcal{R}_n} = |\{\mathcal{L}_{\mathcal{R}_n}\}|, \text{ for } \mathcal{L}_{\mathcal{R}_n} = \{l \,|\, \forall\, l \subseteq \mathcal{R}_n\}, \mathcal{L}_{\mathcal{R}_n} \subseteq \mathcal{L}$$

— **Contour strength**: The contour of a node in the hierarchical tree ζ 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.

$$\texttt{contour}_{\mathcal{R}_n} = \frac{\sum\{\mathcal{F}(u,v) \,|\, \forall\, u, v \in \zeta\}}{|\zeta|}$$
$$\text{where: } \zeta = \{(u,v) \in E \,|\, \forall\, u \in \mathcal{R} \wedge v \in \mathcal{R}' \text{ and } \forall\, \mathcal{R}, \mathcal{R}' \subseteq \mathcal{R}_n\}$$

— **Inertia**: Computes Hu's first moment of inertia for the node n characterizing the shape of the region $\mathcal{R}_n$. This regional attribute can only be computed if the conjoint

graph is a 2D grid graph (for example, modeling an image or a video frame). In 2D grid graphs, each vertice could be associated with a pair of coordinates (x, y) that indicates its relative position in the 2D grid. Consider `coord` the set of coordinates of all vertices (corresp. leaves) in the region $\mathcal{R}_n$ defined by the node n. Consider also $(\bar{x}, \bar{y})$ as the coordinates of the centroid of the region $\mathcal{R}_n$. The first moment of inertia for $\mathcal{R}_n$ is defined as:

$$\begin{aligned}\texttt{inertia}_{\mathcal{R}_n} =& \frac{(\mu_{20} + \mu_{02})}{(\texttt{area}_{\mathcal{R}_n})^2} \text{ , where:} \\ \mu_{ij} =& \sum (x - \bar{x})^i (y - \bar{y})^j, \, \forall \, (x, y) \in \texttt{coord} \text{ and } i, j \in \{0, 2\}\end{aligned}$$

— **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 (where one could calculate the co-variance). Here, the leaf weights are the sum of the weights of the edges that comprise the vertice equivalent of the leaf.

$$\texttt{gaussian}_{\mathcal{R}_n} = [[\texttt{mean}_{\mathcal{R}_n}, \texttt{var}_{\mathcal{R}_n}]]$$

where:

$$\begin{aligned}\texttt{mean}_{\mathcal{R}_n} =& \frac{W_{\mathcal{R}_n}}{\texttt{area}_{\mathcal{R}_n}} \text{ for } W_{\mathcal{R}_n} = \sum \{\mathcal{F}(u, v) \,|\, \forall \, u \in \Gamma(v) \text{ and } v = l \in \mathcal{L}_{\mathcal{R}_n}\} \text{ , and} \\ \texttt{var}_{\mathcal{R}_n} =& \frac{(\texttt{mean}_{\mathcal{R}_n})^2}{\texttt{area}_{\mathcal{R}_n}} - (\texttt{mean}_{\mathcal{R}_n})^2\end{aligned}$$

In $\mathbf{R}_l$, the selected attribute is computed for all regions created by the cut $\sigma \in \beta$. Therefore, the ordered representation is preserved on the cut despite not representing every possible region in the hierarchy. Furthermore, the regional strategy is considerably easier to standardize than the topological approach. The topological approach took the maximal possible depth in all hierarchies in a dataset. Therefore, it created high-dimensional data for some types of hierarchies and, more often than not, multiple padding positions due to the multiform structures. For the current strategy, 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.

As the case may be, some altitude selections could benefit one representation over

another. For instance, as shown in Section 1.6, the hierarchies of watersheds by area, volume, or the number of parents may present an unbalanced distribution of regions for some datasets, where most are distinguished at lower levels on the structure. However, since the goal is to create a generic framework, the formulations of the cuts will take evenly distributed steps on the hierarchical levels and investigate through experimentation and data analysis if it conveys the necessary information for the learning model.

Definition 15: Regular representation of hierarchical regional attributes

The proposed regular representation on **regional attributes** $\mathbf{R}_{\mathcal{H}}$ of a hierarchical tree $\mathcal{T}_{\mathcal{H}}$ in the set $\mathbb{T}$ of all hierarchies in a dataset is: $\mathbf{R}_{\mathcal{H}} = ((\mathbf{R}_1, \mathbf{Y}_1), \ldots, (\mathbf{R}_{|\mathcal{L}|}, \mathbf{Y}_{|\mathcal{L}|}))$. In $\mathbf{R}_{\mathcal{H}}$, 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 the regional attribute $\texttt{reg} \in \{\texttt{area}, \texttt{contour}, \texttt{inertia}, \texttt{gaussian}\}$. The size of $\mathbf{R}_l$ is $|\,\beta\,|$.

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 on the dataset, 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. For the test instances, the procedure takes the regular representation of each hierarchy in the test set and individually subjects them to the RF estimations without the labels.

Algorithm 3 describes the steps to create and store the regular representation on regional attributes for both the training input $\mathcal{D}_r$ and the individual test instances. For clarity, the operations not detailed in Algorithm 3 are:

— **empty**[[nrows, ncols]]: allocates an array memory space of size number of rows by the number of columns.

— **getLabel**(leaf): gets the ground-truth label for a leaf node.

— *getParentLabel*(level, hierarchy): gets the parents' node labels for all leaves that satisfy the cut level condition.

— *getAttribute*(attribute, node list): computes the regional attribute for the node list in the cut.

— *append*(array): appends array vector to another array row-wise.

The following sections will investigate this representation and Section 5.2.1 shows some experiments, evaluating the proposed strategy qualitatively and quantitatively in the image tasks.

Algorithm 3: Regular representation regional attributes

Input : $\mathbb{T}$: a set of hierarchical trees computed for a dataset, a parameter `reg` $\in$ {`area`, `contour`, `inertia`, `gaussian`} indicating the regional attribute to be computed, and a list β with values indicating the altitude levels for the cut.

Output : $\mathcal{D}_r$: a regular training input for the learning framework, and a regular representation $\mathbf{R}_{\mathcal{H}}$ for all $\mathcal{T}_{\mathcal{H}}$ in the test set.

Function *getRegular*$(\mathcal{T}_{\mathcal{H}}, \beta)$**:**

$p_t = |\beta|$
if `reg` == `gaussian` **then** $\mathbf{R}_{\mathcal{H}} =$ **empty**$[[\,|\mathcal{L}|\,, (2 * p_t) + 1\,]]$
else $\mathbf{R}_{\mathcal{H}} =$ **empty**$[[\,|\mathcal{L}|\,, p_t + 1\,]]$
if $\mathcal{T}_{\mathcal{H}}$ *isTrainInstance* **then**
  $\mathbf{Y} =$ **getLabel**(l) **for all** $l \in \mathcal{L}$
  $\mathbf{R}_{\mathcal{H}}[[:, p_t + 1\,]] \leftarrow \mathbf{Y}$ `// leaves labels at added column at the end`
else $\mathbf{R}_{\mathcal{H}} = \mathbf{R}_{\mathcal{H}}[\,|\mathcal{L}|\,, p_t\,]$
for *cut* $\in \beta$ **do**
  leafLabel=*getParentLabel*(cut, $\mathcal{T}_{\mathcal{H}}$) `// gets node cuts for all leaves`
  regionAttribute=*getAttribute*(`reg`,leafLabel)
  $\mathbf{R}_l$[:, *cut*]$\leftarrow$ regionAttribute
end
return $\mathbf{R}_{\mathcal{H}}$

Main:

$\mathcal{D}_r = [[\,]]$
for $\mathcal{T}_{\mathcal{H}} \in \mathbb{T}$ **do**
  $\mathbf{R}_{\mathcal{H}}$=*getRegular*$(\mathcal{T}_{\mathcal{H}}, \beta)$
  if $\mathcal{T}_{\mathcal{H}}$ *isTrainInstance* **then**
    $\mathcal{D}_r \leftarrow$ *append*$(\mathbf{R}_{\mathcal{H}})$
  else *save*$(\mathbf{R}_{\mathcal{H}})$
end
save$(\mathcal{D}_r)$

5.2.1 Experiments: Regional representation

This section shows the experiments with the regional approach on the two image tasks: edge detection in the BSDS500 dataset and segmentation on Birds and Sky. The pipeline takes the same initial steps, colored images to GIG gradient without any additional pre-processing followed by the graph creation with a 4−adjacency and the Euclidean distance for the weighting function. The hierarchy construction explores the aforementioned hierarchies: QFZ, ALPHA, WATER-VOL, WATER-AREA, WATER-DYN, and WATER-PAR. It does not perform additional post-processing.

Table 5.6: Final parameters for the grid search results for the Random Forest: number of trees in the forest (est), the minimum number of samples to split an internal node (split), the minimum number of samples to be a leaf node (leaf), percent for the bootstrap sample size (samples), amount of sampled features for the split (feat - a function on the whole set of features), and the maximum depth of the trees (depth).

	Area						**Contour**					
Hierarchy	est	split	leaf	samples	feat	depth	est	split	leaf	samples	feat	depth
QFZ	200	2	10	10%	sqrt	100	250	5	4	50%	auto	50
ALPHA	100	2	10	All	log2	None	40	10	10	25%	log2	None
WATER-VOL	40	2	4	10%	log2	10	70	5	1	All	sqrt	10
WATER-AREA	1	5	2	All	log2	10	60	2	2	10%	auto	None
WATER-DYN	200	5	10	10%	sqrt	100	250	5	4	50%	sqrt	None
WATER-PAR	150	5	1	All	log2	10	500	5	1	25%	auto	10
	Gaussian						**Inertia**					
Hierarchy	est	split	leaf	samples	feat	depth	est	split	leaf	samples	feat	depth
QFZ	100	10	4	10%	sqrt	100	300	5	1	25%	log2	100
ALPHA	10	5	20	10%	auto	150	100	2	20	10%	auto	None
WATER-VOL	500	5	20	75%	log2	150	150	10	1	10%	sqrt	10
WATER-AREA	200	2	4	10%	sqrt	150	200	5	10	75%	sqrt	10
WATER-DYN	250	10	10	All	auto	None	40	10	2	10%	log2	100
WATER-PAR	500	10	2	10%	log2	50	500	10	4	10%	auto	10

The regular representation takes all the regional attributes considered for selection: area, contour, inertia, and Gaussian. For the cut step, it considers the makes cuts in the range $]0, 1[$ with a 0.1 step and adds the 0.01 and 0.99 for the extremal regions in the structure. This creates a regular representation with 11 dimensions. Except for the Gaussian attribute that has two values for each position, totaling 22 in feature length. The number of samples per dataset remains the same.

The models are trained separately without a combination of hierarchical types or attributes. The label attribution and evaluation metrics are the same as in the topological approach. Also, the same models are used: RF classifier for the segmentation and RF regressor for the edge detection.

Random Forest parameters

For the Random Forest parameters, it performs a grid search using the Random Forest classifier for all hierarchies and for all regional attributes. The grid search evaluation takes the $F1-$score on the edge detection task in the BSDS500 dataset evaluating the validation set. The RF parameters in consideration are: (i) number of trees in the forest (est); (ii) minimum number of samples to split an internal node (split); (iii) the minimum number of samples to be a leaf node (leaf); (iv) percent for the bootstrap sample size

Table 5.7: Quantitative comparison of the results obtained in all datasets for the graph representation, the typical, the topological, and the regional approach. $F1-$score for best dataset scale for the BSDS500 and average Jaccard score for Birds and Sky. Emphasizes the best scores per approach variation; a green highlight for the best values per dataset for the topological approach; and red emphasis on the best score among strategies per dataset. Perfect scores=1.

		BSDS				Sky				Birds			
Graph	GIG	**0.65**				**0.86**				**0.29**			
	GIG-Edge	0.61				0.78				0.27			
	onlyColor	0.64				0.85				0.28			
	Hierarchy	Threshold		Regions		Threshold		Regions		Threshold		Regions	
Typical	ALPHA	0.21		0.33		0.00		0.00		0.15		0.13	
	QFZ	0.26		0.28		0.45		0.01		0.30		0.05	
	WATER-DYN	**0.27**		0.42		0.79		0.13		0.29		0.15	
	WATER-VOL	0.20		**0.55**		0.89		**0.87**		**0.36**		**0.24**	
	WATER-AREA	0.24		0.53		0.83		0.83		0.30		0.22	
	WATER-PAR	0.24		0.53		**0.90**		0.86		0.32		0.24	
	Hierarchy	Alt	Area	Dyn	Vol	Alt	Area	Dyn	Vol	Alt	Area	Dyn	Vol
Topological	ALPHA	**0.63**	0.55	0.60	0.61	0.67	**0.81**	0.61	0.78	0.18	**0.18**	0.15	0.15
	QFZ	**0.60**	0.60	0.57	0.58	0.67	**0.90**	0.47	0.89	0.14	**0.37**	0.07	0.15
	WATER-DYN	**0.64**	0.62	0.62	0.63	0.67	**0.92**	0.47	0.90	0.17	**0.20**	0.06	0.11
	WATER-VOL	0.57	**0.58**	0.53	0.56	0.90	**0.96**	0.90	0.95	0.26	**0.37**	0.06	0.17
	WATER-AREA	**0.51**	0.50	0.46	0.48	**0.85**	0.82	0.84	0.82	0.26	**0.31**	0.18	0.18
	WATER-PAR	0.54	**0.54**	0.50	0.51	0.91	**0.95**	0.72	0.92	0.28	**0.41**	0.20	0.27
	Hierarchy	Area	Contour	Gaussian	Inertia	Area	Contour	Gaussian	Inertia	Area	Contour	Gaussian	Inertia
Regional	ALPHA	0.62	0.65	**0.67**	0.62	0.78	0.43	0.75	**0.95**	0.36	**0.55**	0.25	0.49
	QFZ	0.64	0.63	**0.66**	0.58	0.82	0.56	0.91	**0.96**	0.32	**0.53**	0.27	0.51
	WATER-DYN	0.64	0.65	**0.66**	0.65	**0.86**	0.61	0.80	0.82	**0.36**	0.27	0.25	0.26
	WATER-VOL	0.38	0.37	0.36	**0.44**	0.80	0.73	**0.96**	0.87	0.44	**0.69**	0.61	0.44
	WATER-AREA	**0.62**	0.62	0.55	0.65	0.76	0.56	**0.96**	0.68	0.32	0.54	**0.57**	0.24
	WATER-PAR	0.61	0.63	0.60	**0.65**	0.89	0.54	**0.95**	0.71	0.50	**0.71**	0.64	0.32

(samples); (v) amount of sampled features for the split (feat - a function on the whole set of features); and (vi) the maximum depth of the trees (depth). Table 5.4 presents the final parameters for each representation.

Quantitative analysis

Table 5.7 shows the results with the regional strategy for all variations on the three datasets. It is presented alongside with the results from the best scale on the typical trivial approach, the representations from graph attributes, and the variations on the topological approach.

As shown, the regional strategy presents 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. WATER-DYN is the only type with all scores below 0.5, yet it remains considerably better than the other strategies for this dataset.

In BSDS500, the regional features are overall better than the topological attributes, except for the WATER-VOL hierarchy. As seen in the data analysis (Section 5.1.1), WATER-VOL has the smoothest area distribution among its peers, where most regions

are concentrated at the lower end of the trees, particularly for the BSDS500 dataset. It is compelling to believe that the cut levels neglected most of the discernable values in this structure.

Regrettably, there is not a single attribute that one could point out as the best selection for the regional strategy, like the altitudes and area in the topological approach. However, 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. For instance, for the contour-oriented hierarchies, it is the best for edge detection, and for the region-oriented, the best for segmentation. Future applications of this strategy may consider the task at hand to select a hierarchical type that most agrees with the objectives and use the Gaussian attribute for the representation.

Regarding the execution time, the most costly step, the evaluation, remains unchanged since it is external to these proposals. However, the data creation takes approximately one-fifth of the time in the topological, namely 110 seconds for BSDS500 and 10 seconds for the others. The training time is also reduced, taking from 70 seconds to a maximum of 10 minutes depending only on the RF parameter (mainly the number of estimators and maximal depth). The computational resources demand is less than 4GB of RAM for all approaches.

5.3 Hierarchical attributes discussion

This chapter presented two strategies to process hierarchical data in a learning pipeline that creates the regular representation required by most machine learning models while preserving the ordered information embedded in the structures.

The first strategy represents all the hierarchical levels in the form of topological attributes of set parent nodes that traverse the hierarchical trees from leaves to root. The maximum depth in a dataset's entire set of hierarchies standardizes the feature vectors. This strategy not only improved the results obtained with a typical approach with the hierarchies but also provided an overview of the value distribution when all the hierarchical information is taken as a whole. Data analysis of this distribution showed that despite the differences between the types and individual constructions for the inputs, the aggregated values present similar characteristics.

The second strategy aims to preserve the ordered representation, but instead of repre-

senting each level, it presents the hierarchical structure as a set of regional attributes. This approach parallels a series of horizontal cuts by thresholding the hierarchy by altitude. Still, instead of creating and evaluating each partition, all the regions are presented as regional attributes on a path. This approach resulted in a more compact representation that captured the critical information on the hierarchies and improved the results in all experiments.

The strategy formulates both proposed representations using only the information on the hierarchical structure and its conjoined graph. Therefore, the media depiction is at the discretion of the graph modeling. Similarly, label attribution takes the labels directly from the leaf that represent the primary components on the graph, exempting any further considerations on the task.

Conclusion

The main goal of this thesis was to design a generic learning framework that could operate on hierarchical data. To do so, it must deal with the generalization challenges in media and tasks and place a strategy to conform the hierarchical data to a learning framework. It argued that it is possible to directly insert the hierarchical structures in a learning framework benefiting from the embedded information.

It is challenging because hierarchical structures are rich multiform representations of ordered data and learning models usually require systematic structures to operate. Also, no direct metric could indicate the quality of a hierarchical structure, and this process usually relies on performance measurements on a task.

This thesis presented the study in three parts, each assessing a crucial aspect of achieving the primary goal. The first part contextualized and delimitated the problem in theory and literature. The second one investigated the capabilities of an agnostic model using graphs as an anchor point between multimedia data and hierarchies. The last and final part gathered the information to propose the final framework using the hierarchical structure, inserted in a learning framework that relies solely on the hierarchical information to operate. The following sections will conclude each of these parts, highlighting the most significant points connected with the initial questions and hypothesis.

Conclusions Part I: Hierarchical data

The first part of this thesis comprised Chapter 1 and 2 assessing the question: **How do hierarchical methods model various media information, and what are the practical challenges faced when applying them to a learning framework?** With the hypothesis:

Hypothesis 1

Hierarchical representations contain useful information embedded in their structures for a generic learning framework, and the learning framework could assist in parsing the structure.

Chapter 1 contextualized the hierarchies as portrayed in mathematical morphology, which presents formulations with a solid theoretical foundation, efficient implementations, and guiding principles of order. 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. The formalisms are usually for image data, but in general, they characterize regions and could model any desired characteristics providing ordered representations in the visual domain.

Chapter 1 also introduced the theory of hierarchies as graphs, formalizing graph concepts and describing their components and terminologies. Furthermore, it outlined the different hierarchical types contemplated in the thesis, inserted in a typical framework presenting through illustrations and experiments the challenges and problems usually faced when applying the structures in a task.

It showed that the application requires a deep understanding of the media, the region connotation in the hierarchies, and the task. Additionally, parsing the structures must consider many aspects crucial for a good performance, which is even more critical considering that it relies on the task to evaluate its quality. Using the trivial approach with horizontal cuts, searching for an ideal partition for an application could be strenuous and neglect essential details present in the hierarchies.

Chapter 2 featured a systematic review of the literature on "Learning on hierarchies", which is the first on the theme. The review gathered the strategies that combine machine learning and hierarchical data on the same framework. The search retrieved 225 publications, and after filtering by the relevance for the scope of this work, it reviewed 64 methods grouped by the way the hierarchical information is inserted into the learning framework. Namely, methods that: (i) applies the hierarchies to a learning framework assessing how the structure assists on the task and how the authors format the hierarchical information for the application.; and (ii) applies the learning strategies to the hierarchical structures assessing how the learning helps improve the representation.

It also included some hand-engineered techniques for transforming the hierarchies into a more suitable representation for the tasks and some local-optimization strategies that are not framed as machine learning and, therefore, not retrieved by the search keys. The review also added two other categories to group the methods that did not fit the problem tackled in this thesis but whose purposes are relevant in the context. Namely, approaches for the non-hierarchical watershed and learning strategies inspired by the hierarchical construction algorithms.

The review assessed: (i) the types of media, how the authors model their representation both on the hierarchical structure and the task; (ii) the types of hierarchies and which role they play in the learning framework; and (iii) the machine learning methods and the reasons for choosing them.

The review found that hierarchies assisting the machine learning algorithms in performing a task define regions delimiting areas for feature extraction or represent masks applied on the media. Almost all methods rely on media features for the learning step and often require reducing the size of the hierarchical representations, either by filtering, compression, or hand-picked samples. The strategies in this category require a complete understanding of how the media's low-level components interact in the space and how they relate to the task. Most applications are classification, segmentation, or detection. There are many domains, but the most dominant is the aerial and medical analysis and generic image processing. Regarding the models, Random Forest, SVM, and neural networks are often the models of choice for their robustness and generalization capabilities.

Among the methods using machine learning applied in the hierarchical structure, the typical approach is the energy optimization strategy to identify regions of interest inside the hierarchical structure. Another common technique is to transfer the learning target to a parallel task that induces a response on the hierarchical nodes. Most of the methods present complex solutions or combinatorial analysis. Learning the hierarchical structure remains an active open research topic.

The majority of retrieved results in the review are for the non-hierarchical watershed. It is prevalent among medical applications that rely on coherent and consistent regions. A widespread problem among all methods using the classical watershed is over-segmentation. Many strategies rely on thorough preprocessing for successful applications, while others propose learning techniques to merge some regions or select areas of interest.

Answer 1: Hierarchies are rich structures that could model a myriad of data. It facilitates the analysis of complex problems in multiple domains. However,

they require careful consideration and parsing the structures can be challenging and limit their applications. There is great interest in the literature on integrating hierarchies and machine learning in the same framework. Still, it usually relies on media features and provides solutions not easily generalized for similar tasks. Learning the hierarchical structure remains an active open research topic.

Conclusions Part II: Learning on graphs

The second part of this thesis comprised Chapter 3 and 4 assessing the question: **Question 2: How to create a learning framework agnostic to media and tasks?** With the hypothesis:

Hypothesis 2

Using a selection of graph attributes as input to the learning framework allows a construction agnostic to the media, and modeling it at the graphs' components level allows assigning each entry with a task label without imposing assumptions on the data source.

Defining the graph representation is a modeling question with various connotations. They are used to represent generic objects, and the primary concern in graph theory is how these objects are interconnected. They can depict many data and carry information about the objects in their components, including from different domains. All deliberations in this work were centered on graph theory as they could provide generalization tools for: (i) the hierarchical structures depicted in a tree structure; (ii) the multimedia data; and (iii) a media-independent learning framework.

Chapter 3 presented a literature review of machine learning on graphs, exploring the motivations, strategy, and fundamental issues. It concentrated its analysis on the multimedia processing perspective and gathered information to advocate for the proposed framework choices.

In a graph representing digital media with arbitrary dimensions, the vertices may correspond to the media's units, such as pixels, voxels, or data points. This approach usually results in large sets of vertices but favors back-and-forth operations. Alternatively, the vertices could correspond to objects inferred from the data, such as superpixels, partitions,

and surfaces, creating a more concise representation but requiring complex mappings dependent on the grouping strategy.

Graph embedding methods are suitable for creating a systematic representation of the graphs that allow their utilization in multiple learning frameworks. But embeddings are very expensive in terms of computational resources and are prohibitive for large graphs. Deep learning methods on graphs are a contemporary solution to many tasks, primarily semantic and high-level analysis. Despite their improvements in inferring information, they impose limitations regarding the underlying graph and the modeling choices. Particularly when modeling media data. A general approach when handling deep learning on graphs in a multimedia context is to model the concepts and abstractions rather than the raw media data. Random Forest on graphs provides solutions for many computational problems, particularly medical applications and social network analysis. The most significant limitation of aggregating graphs and Random Forests is the systematic input required by the model. Careful graph parsing must take place, considering the type of graph, its proximity to the original data, and the expected results.

Chapter 4 presented the case study for a learning framework operating on a selection of graph attributes aggregated with the Random Forest model. Beyond a good performance in an application, the motivation relied on a proposal of a machine learning framework working on graphs that could later be exploited for the hierarchical structures. The study proposed to use edge-weighted graphs acting as a transformation filter based on local differences in images and the RF as a regularization process to mitigate some noise and reinforce desirable characteristics. Also, it described the graphs at the vertice's level, which allows training the model on the discrete space by associating each entry with a single label.

Dealing with graphs created from images has a unique modeling space. From the framework perspective, all attributes are just sets of values stored on the vertices and edges of the graph. But conceptually, the image graph creates a unique transformed space close to the spatial domain of the images, strengthened with relational aspects on the edges of the graph. Cognizant of this unique space, it investigated the relational factors of the graphs while pondering the particular conditions for image processing. It evaluated the impact of the characteristics on the results obtained on two different tasks and discoursed aspects that could not be generalized.

A quality assessment of the topology choices addressed the considerations about the type of graph and its proximity to the original media. The experimental investigation on

the attribute selection established that representing larger regions through the neighboring size and the number of connections with the adjacency relation translated into higher confidence values on the edges and less noise in the resulting images. Also, the weighting function must characterize similarities in the original data to be descriptive. The adjacency relation and the weighting function are modeling choices, conditioning the interaction between the data and the graph. The RF regularization mitigates most poor topology choices, except when the input is extremely noisy and correlated. The final assessment of the attribute selection regards the vertex attributes representing low-level descriptors of the image. Including the vertex attributes in the regular representation makes a direct reference to the media but results in less noise, stronger borders, and more details on the final image gradients, hence crucial to the practical applications assessed.

The main challenge in the framework concerned the regular representation required by most machine learning algorithms, which is inherently opposed to the unconstrained nature of graphs. Furthermore, the framework also considered the high-dimensional space usually presented with graphs representing digital media and the label attribution strategy that should not impose assumptions on the data source to later generalize to other tasks. The RF mechanics allows it to work with high-dimensional data, making it a fast, simple, and scalable method for the investigation. Also, the RF paired with the edge-weights gradient operator acts as a regularizer diminishing noise, accentuating strong connections, and mitigating any eventual poor topology choice. Furthermore, mapping the RF predictions back to the image space in the form of image gradients allows the evaluation of the results qualitatively and quantitatively.

The images obtained by mapping the predictions of the RF, trained on edge detection labels, and receiving the regular representation of the selected attributes of the edge-weighted graphs as input, presented characteristics of image gradients, referred to as GIG. GIG's gradients are generally very descriptive, with firm contours of the objects and other aspects such as minor components, textures, and large uniform regions. Gradients are commonly used as a preprocessing step in many applications because they are fast to compute and usually facilitate image analysis, particularly for the segmentation task.

Compared with other popular gradient strategies, GIG's gradients, as input for the watershed hierarchies segmentation method, produced better-segmented images than traditional gradient methods like SED, Sobel, and Laplace. Comparing it with more elaborated edge maps, like the ones made by deep approaches HED and RCF, demonstrated that the segmentation task's performance depends on the characteristics portrayed on

the gradient. Overall, better segmentations result from gradients with thick and sharp contours and additional details that contribute to identifying small objects, and information about uniform regions provides consistency. The strategy addressing the question of missing values in the regular representation, created by the vertices of the image's border, directly influences the feature connotation considering the RF mechanics. Therefore, addressing this aspect improved the results of the edge detection task.

Regarding the performance of edge detection (the task the model is trained on), the results could be better compared with the deep methods or SED. However, observing the outputs showed that the procedures that perform better on the task are the ones with thicker contours. This is a result of the evaluation method proposed for the dataset. The representations with a more significant margin on the contours are more likely to match the ground truth. Because GIG is centered on the analysis of the vertices, the more confident the RF is in distinguishing a vertice as a contour from its surrounding vertices, the more precise the predictions are, resulting in thinner contours. Nonetheless, the other aspects portrayed on the gradients, such as the large uniform regions and simplified patterns, could be considered a failure on the task, even if beneficial for other applications and descriptive of the properties relayed by the graphs.

Answer 2: Graphs are dynamic structures for modeling multimedia, but like hierarchies, they require thoughtful considerations when applied in a machine learning framework. Using the available information on the graph edges and vertices is a viable method to represent a graph in a learning framework. It allows controlling the representation size and selecting the information depicted considering the type of graph, its proximity to the original data, and the expected results. Furthermore, representing the graphs at the vertex level allows maintaining the analyses on the discrete space by assigning a single label for an entry. This assignment is particularly advantageous with the graph strategy since it represents an entire region on a single vertex, and the task makes no assumptions about the media.

Conclusions Part III: Learning on hierarchies

The final part of this thesis comprises Chapter 5, assessing the third and final question: **Question 3: Could the hierarchical structure provide useful information in an agnostic learning framework?** With the hypothesis:

Hypothesis 3

The topology of the hierarchical structures alone could be used in a learning framework to solve multiple tasks if it preserves their semantical arrangement.

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.

Chapter 5 presented the culmination of the proposals, expanding the concepts and strategies to the hierarchical data. It proposed two strategies for representing the hierarchical structures: (i) by the topological properties taking the hierarchical trees as inputs; and (ii) by regional features deduced from the hierarchical topology with their conjoined graph. Both were formulated using only the information on the hierarchical structure and its conjoined graph. Therefore, the media depiction is done at the discretion of the graph modeling. Furthermore, the task label attribution is performed at the leaf level at the bottom of the tree, where each leaf has a unique discrete label. While the assignment on the graph allowed representing an entire region on a single vertex, on the hierarchy, multiple regions that share a path on the tree are represented in a single leaf.

The topological approach proposed using the entire set of parents of a leaf node described by their topological attributes. Representing the hierarchical structures by sets of parents of a leaf retains the semantical information embedded on the hierarchical trees without the need to filter or select a particular level for evaluation. Understanding the distribution of the values using the topological representation was beneficial to guide the decisions regarding the learning step and to better comprehending the hierarchical structure.

The first assessment regarded the order of the regular representation on the leaves. The order could be ascending (from leaf to root) or descending (from root to leaf). Because of the multiformity in the hierarchies, different leaves have a variable amount of parents on the structure and may not be an alignment between the feature position in the regular representation and the parent position in the hierarchical tree. The case study in GIG

indicated that the RF performs better when there is a meaningful correspondence between the features and the position they assume in the feature vector. Therefore, to provide some clarity on the relation between the model and the order of the features on the training data, the analysis inspected the feature position importance on the decision nodes of the model and probed the values used for the split. Experiments showed that the ascending order provided more consistent distributions favored by the model.

Experiments with the topological approach showed that it contains crucial information about the hierarchies and not only improved the results obtained with a typical approach with the hierarchies but also provided an overview of the value distribution when all the hierarchical information is taken as a whole. Data analysis of this distribution showed that despite the differences between the types and individual constructions for the inputs, the aggregated values present similar characteristics. Regarding the representation choices, the attributes altitudes and area presented the best results for the task most related to the information they portray. The contour-oriented hierarchies give the best results in the edge detection task with the altitudes attribute and WATER-VOL and WATER-PAR in the segmentation task with the area.

The topological strategy constructed a regular representation that could be used in most available learning models. However, the dimensions of this representation could be challenging in terms of computational resources, particularly considering that most of the feature positions were filled with padding values. The efficient implementations for hierarchical structures and the flexibility of the RF model allowed us to work with sizable structures, but using this approach with a different model could be challenging.

The second strategy proposed to use of a set of regional attributes to represent the hierarchical structure. Procedurally, it is equivalent to performing horizontal cuts by altitude levels, but rather than creating a representation for each cut and evaluating them individually, the proposed method represented all of them systematically as a regular representation. The second strategy also aimed to preserve the ordered representation, but instead of representing each level, it presents the hierarchical structure as a set of regional attributes ordered on the feature position. This approach resulted in a more compact representation that captured the critical information on the hierarchies and improved the results in all experiments.

Regrettably, there was not a single attribute that one could point out as the best selection for the regional strategy, like the altitudes and area in the topological approach. However, the Gaussian presented, in general, superior results on the different tasks. Be-

cause the Gaussian attribute quantifies the region distribution on the hierarchical trees, it assimilates the representation with the task. For instance, for the contour-oriented hierarchies, it is the best for edge detection, and for the region-oriented, the best for segmentation. Future applications of this strategy may consider the task at hand to select a hierarchical type that most agrees with the objectives and use the Gaussian attribute for description.

Answer 3: This thesis demonstrated that it is possible to create a learning framework dependent only on the hierarchical data that performs well in multiple tasks with different models. It created and delivered a learning framework operating directly on the hierarchical structure, avoiding any feature extracted from the media and only using the information on the hierarchical tree and graph. Also, it did not select any particular region that better suited an application. Instead, the entire structure is represented in a vectorial form that preserves its semantical arrangement. Furthermore, label attribution takes the labels directly from the leaf that represent the primary components on the graph, exempting the framework from making any further considerations on the task.

Perspectives and future work

The applications and experiments developed in this thesis were all performed on the image space because it allows an easy visual inspection of the result's quality. However, all proposals are formulated on the structures of the graphs or hierarchies or both. Furthermore, the literature review showed that both subjects are extensively used to model many media types, particularly visual media.

The thesis proposals provide solutions to incorporate the structures in a learning framework representing the structure in a format supported by many machine learning models. Also, they remove the need for detailed scrutiny regarding selecting an appropriate level or region and provide a way to experiment with multiple hierarchical types without changing the considerations. Furthermore, the label attribution at the components level removes the task concerns such as binarization and foreground/background selection for evaluation.

An application in another media type could take the same considerations of attribute selection since once the media is modeled as a hierarchy or graph, they will all share

the same rules in that space. Particularly the hierarchical structures that additionally incorporate the notions of order and navigation in their rules in a non-gridded form.

If generalization is not a concern in future applications, one could use the proposed strategies to transpose the structure to the vectorial space while taking the appropriate measures to improve the results on a media-specific task. For instance, one could better model the media in the hierarchical or graph space using pre-processing techniques or even region selection (if the type of region of interest is known).

Another way to improve task performance using the proposed strategies is by selecting a different machine learning model. The Random Forest model was chosen in the thesis because it is fast, inspectable, and scalable. It allowed the execution of multiple experiments with the raw media data without the time cost or designing decisions to favor scalability that other models would require.

One possible direction for future work on the already proposed strategies is to combine the attributes selected from the graphs and hierarchies in the same or different categories of features, enriching the information presented to the machine learning model. Another possibility is to apply a strategy to reduce the structure prior to the attribute selection or employ a data reduction strategy after the features are in a vectorial form. However, one must always be mindful of the semantical arrangement and hierarchical rules when using such techniques.

Finally, the thesis demonstrated that creating a learning framework operating with hierarchical data that performs well in multiple tasks and is media agnostic is possible. The proposals work with any hierarchical type, including the ones not created from graphs. However, they do not improve the hierarchical structure, and learning a hypothetical ideal hierarchy remains an open problem.

Main contributions

— Learning framework on graph attributes for image processing (Published in 34th Conference on Graphics, Patterns, and Images - Awarded as best paper).

— Extended formalism on graphs attributes exploring more extensive input areas through region adjacency graphs and changes driven by the model mechanics (Published in Pattern Recognition Letters).

— Learning framework operating directly on the hierarchical data, focusing the formulations solely on the structural components of the hierarchies (submitted).

— Critical systematic review of the literature on "Learning on hierarchies", which is the first on the theme to the best of our knowledge.

Contribution to knowledge

The thesis demonstrates that it is possible to create a learning framework operating with hierarchical data that performs well in multiple tasks with different models.

BIBLIOGRAPHY

ACHANTA Radhakrishna et al. (2010). *Slic superpixels*. Tech. rep. 149300. École Polytechnique Fédérale de Lausanne, pp. 1–15 (cit. on p. 130).

ACUNA David et al. (2018). *Efficient interactive annotation of segmentation datasets with polygon-RNN++*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 859–868. DOI: 10.1109/CVPR.2018.00096 (cit. on pp. 96, 103).

ADÃO Milena M., GUIMARÃES Silvio Jamil F., and JR Zenilton K. G. Patrocínio (2020). *Learning to realign hierarchy for image segmentation*. In: *Pattern Recognition Letters* 133, pp. 287–294. DOI: 10.1016/j.patrec.2020.03.010 (cit. on pp. 41, 72, 77).

AKÇAY Gokhan and AKSOY Selim (2008). *Automatic detection of geospatial objects using multiple hierarchical segmentations*. In: *IEEE Transactions on Geoscience and Remote Sensing* 46, pp. 2097–2111. DOI: 10.1109/TGRS.2008.916644 (cit. on p. 73).

ALEXANDRE Eduardo Barreto (2017). *IFT-SLIC: geração de superpixels com base em agrupamento iterativo linear simples e transformada imagem-floresta*. PhD thesis. Universidade de São Paulo. DOI: 10.11606/D.45.2017.tde-24092017-235915 (cit. on p. 49).

ALMEIDA Raquel, KIJAK Ewa, et al. (2022). *Graph-based image gradients aggregated with random forests*. In: *Pattern Recognition Letters*. DOI: 10.1016/j.patrec.2022.08.015 (cit. on p. 26).

ALMEIDA Raquel, PATROCÍNIO JR. Zenilton K. G., et al. (2021). *Descriptive image gradient from edge-weighted image graph and random forests*. In: *34th Conference on Graphics, Patterns and Images*, pp. 338–345. DOI: 10.1109/SIBGRAPI54419.2021.00053 (cit. on p. 25).

ALONSO-GONZALEZ Alberto, LOPEZ-MARTINEZ Carlos, and SALEMBIER Philippe (2014). *Polsar time series processing with binary partition trees*. In: *IEEE Transactions on Geoscience and Remote Sensing* 52, pp. 3553–3567. DOI: 10.1109/TGRS.2013.2273664 (cit. on p. 31).

ALUSH Amir, GREENSPAN Hayit, and GOLDBERGER Jacob (2009). *Lesion detection and segmentation in uterine cervix images using an ARC-level MRF*. In: *IEEE Interna-*

tional Symposium on Biomedical Imaging: From Nano to Macro. IEEE, pp. 474–477. DOI: 10.1109/ISBI.2009.5193087 (cit. on pp. 79, 80, 83).

ALUSH Amir, GREENSPAN Hayit, and GOLDBERGER Jacob (2010). *Automated and interactive lesion detection and segmentation in uterine cervix images*. In: *IEEE Transactions on Medical Imaging* 29, pp. 488–501. DOI: 10.1109/TMI.2009.2037201 (cit. on pp. 79, 80, 83).

ALVES Wonder A. L., GOBBER Charles F., et al. (2020). *Image segmentation based on ultimate levelings: from attribute filters to machine learning strategies*. In: *Pattern Recognition Letters* 133, pp. 264–271. DOI: 10.1016/j.patrec.2020.03.013 (cit. on pp. 72, 74, 75).

ALVES Wonder A. L., HASHIMOTO Ronaldo F., and MARCOTEGUI Beatriz (2017). *Ultimate levelings*. In: *Computer Vision and Image Understanding* 165, pp. 60–74. DOI: 10.1016/j.cviu.2017.06.010 (cit. on p. 75).

ALVES Wonder Alexandre Luz and HASHIMOTO Ronaldo Fumio (2014). *Ultimate grain filter*. In: *IEEE International Conference on Image Processing*. IEEE, pp. 2953–2957. DOI: 10.1109/ICIP.2014.7025597 (cit. on p. 75).

ALVES Wonder Alexandre Luz, MORIMITSU Alexandre, et al. (2013). *Extraction of numerical residues in families of levelings*. In: *Conference on Graphics, Patterns and Images*. IEEE, pp. 349–356. DOI: 10.1109/SIBGRAPI.2013.55 (cit. on p. 75).

APTOULA Erhan, WEBER Jonathan, and LEFÈVRE Sébastien (2013). *Vectorial quasi-flat zones for color image simplification*. In: *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer Berlin Heidelberg, pp. 231–242. DOI: 10.1007/978-3-642-38294-9_20 (cit. on p. 39).

ARBELAEZ Pablo, MAIRE Michael, et al. (2009). *From contours to regions: An empirical evaluation*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2294–2301. DOI: 10.1109/CVPR.2009.5206707 (cit. on p. 52).

ARBELAEZ Pablo, PONT-TUSET Jordi, et al. (2014). *Multiscale combinatorial grouping*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 328–335. DOI: 10.1109/CVPR.2014.49 (cit. on pp. 41, 52).

ARBELÁEZ Pablo et al. (2011). *Contour detection and hierarchical image segmentation*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, pp. 898–916. DOI: 10.1109/TPAMI.2010.161 (cit. on pp. 63, 77).

Atwood James and Towsley Don (2016). *Diffusion-convolutional neural networks*. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates Inc., pp. 2001–2009 (cit. on p. 95).

Avci Murat (2000). *A hierarchical classification of landsat TM imagery for land cover mapping*. PhD thesis. Middle East Technical University (cit. on p. 68).

Awajan Arafat (2015). *Keyword extraction from arabic documents using term equivalence classes*. In: *ACM Transactions on Asian and Low-Resource Language Information Processing* 14, pp. 1–18. doi: 10.1145/2665077 (cit. on p. 104).

Ayotte Blaine et al. (2020). *Fast free-text authentication via instance-based keystroke dynamics*. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 2, pp. 377–387. doi: 10.1109/TBIOM.2020.3003988 (cit. on pp. 104, 105).

Bacciu Davide et al. (2020). *A gentle introduction to deep learning for graphs*. In: *Neural Networks* 129, pp. 203–221. doi: 10.1016/j.neunet.2020.06.006 (cit. on pp. 17, 94, 6).

Bai Min and Urtasun Raquel (2017). *Deep watershed transform for instance segmentation*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2858–2866. doi: 10.1109/CVPR.2017.305 (cit. on p. 85).

Bajaj Mohit, Wang Lanjun, and Sigal Leonid (2019). *G3raphGround: graph-based language grounding*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 4280–4289. doi: 10.1109/ICCV.2019.00438 (cit. on p. 98).

Barcelos Isabela Borlido et al. (2019). *Exploring hierarchy simplification for non-significant region removal*. In: *Conference on Graphics, Patterns and Images*. IEEE, pp. 100–107. doi: 10.1109/SIBGRAPI.2019.00022 (cit. on p. 41).

Bejar Hans H. C., Guimaraes Silvio Jamil Ferzoli, and Miranda Paulo A. V. (2020). *Efficient hierarchical graph partitioning for image segmentation by optimum oriented cuts*. In: *Pattern Recognition Letters* 131, pp. 185–192. doi: 10.1016/j.patrec.2020.01.008 (cit. on p. 41).

Belkin Mikhail and Niyogi Partha (2001). *Laplacian eigenmaps and spectral techniques for embedding and clustering*. In: *Advances in Neural Information Processing Systems*. MIT Press, pp. 585–591 (cit. on p. 93).

Bertrand Gilles et al. (2013). *Mathematical morphology: from theory to applications*. Ed. by Najman Laurent and Talbot Hugues. John Wiley & Sons, Inc., pp. 82–107. doi: 10.1002/9781118600788 (cit. on p. 109).

Beucher Serge (1979). *Use of watersheds in contour detection*. In: *International Workshop on Image Processing*. CCETT/IRISA, pp. 2.1–2.12 (cit. on p. 36).

— (1994). *Watershed, hierarchical segmentation and waterfall algorithm*. In: *Computational Imaging and Vision*. Vol. 2. Springer, pp. 69–76. doi: 10.1007/978-94-011-1040-2_10 (cit. on pp. 35, 36, 4).

Biau Gérard and Scornet Erwan (2016). *A random forest guided tour*. In: *TEST* 25, pp. 197–227. doi: 10.1007/s11749-016-0481-7 (cit. on pp. 103, 104, 112).

Bleau Andrè and Joshua Leon (2000). *Watershed-based segmentation and region merging*. In: *Computer Vision and Image Understanding* 77, pp. 317–370. doi: 10.1006/cviu.1999.0822 (cit. on p. 78).

Börold Axel et al. (2020). *Deep learning-based object recognition for counting car components to support handling and packing processes in automotive supply chains*. In: *IFAC World Congress* 53, pp. 10645–10650. doi: 10.1016/j.ifacol.2020.12.2828 (cit. on pp. 83, 84).

Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey*. In: *Journal of Imaging* 4, p. 33. doi: 10.3390/jimaging4020033 (cit. on pp. 16, 18, 28, 34, 5, 6).

Bouman Charles A. and Shapiro Michael (1994). *A multiscale random field model for Bayesian image segmentation*. In: *IEEE Transactions on Image Processing* 3, pp. 162–177. doi: 10.1109/83.277898 (cit. on p. 76).

Bouritsas Giorgos et al. (2019). *Neural 3D morphable models: spiral convolutional networks for 3D shape representation learning and generation*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 7212–7221. doi: 10.1109/ICCV.2019.00731 (cit. on p. 99).

Boykov Yuri, Veksler Olga, and Zabih Ramin (2001). *Fast approximate energy minimization via graph cuts*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, pp. 1222–1239. doi: 10.1109/34.969114 (cit. on p. 72).

Breiman Leo (2001). *Random forests*. In: *Machine Learning* 45, pp. 5–32. doi: 10.1023/A:1010933404324 (cit. on pp. 20, 65, 103, 109, 112, 9).

Bruna Joan et al. (2014). *Spectral networks and locally connected networks on graphs*. In: *2nd International Conference on Learning Representations*. DBLP (cit. on p. 95).

Cardelino Juan et al. (2013). *A contrario selection of optimal partitions for image segmentation*. In: *SIAM Journal on Imaging Sciences* 6, pp. 1274–1317. doi: 10.1137/11086029X (cit. on p. 71).

Chakravarthy Adithi D. et al. (2020). *A thrifty annotation generation approach for semantic segmentation of biofilms*. In: *International Conference on Bioinformatics and Bioengineering*. IEEE, pp. 602–607. DOI: 10.1109/BIBE50027.2020.00103 (cit. on pp. 78, 79, 84).

Challa Aditya et al. (2019). *Watersheds for semi-supervised classification*. In: *IEEE Signal Processing Letters* 26, pp. 720–724. DOI: 10.1109/LSP.2019.2905155 (cit. on p. 85).

— (2022). *Triplet-watershed for hyperspectral image classification*. In: *IEEE Transactions on Geoscience and Remote Sensing* 60, pp. 1–14. DOI: 10.1109/TGRS.2021.3113721 (cit. on p. 85).

Chen Cheng and Fan Guoliang (2010). *Coupled region-edge shape priors for simultaneous localization and figure-ground segmentation*. In: *Pattern Recognition* 43, pp. 2521–2531. DOI: 10.1016/j.patcog.2010.01.021 (cit. on pp. 82, 84).

Chen Da et al. (2020). *Hierarchical sequence representation with graph network*. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 2288–2292. DOI: 10.1109/ICASSP40776.2020.9054195 (cit. on p. 96).

Chen Gongwei et al. (2020). *Scene recognition with prototype-agnostic scene layout*. In: *IEEE Transactions on Image Processing* 29, pp. 5877–5888. DOI: 10.1109/TIP.2020.2986599 (cit. on p. 102).

Chen Siheng et al. (2019). *PCT: large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving*. In: *IEEE International Conference on Image Processing*. IEEE, pp. 4395–4399. DOI: 10.1109/ICIP.2019.8803525 (cit. on pp. 96, 99).

Chen Yanlin et al. (2020). *Automatic segmentation of individual tooth in dental CBCT images from tooth surface map by a multi-task FCN*. In: *IEEE Access* 8, pp. 97296–97309. DOI: 10.1109/ACCESS.2020.2991799 (cit. on p. 63).

Chen Yuhua et al. (2016). *Scale-aware alignment of hierarchical image segmentation*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 364–372. DOI: 10.1109/CVPR.2016.46 (cit. on pp. 41, 72, 76).

Chen Yuxin et al. (2020). *Graph convolutional network with structure pooling and joint-wise channel attention for action recognition*. In: *Pattern Recognition* 103, p. 107321. DOI: 10.1016/j.patcog.2020.107321 (cit. on pp. 96, 97).

Chierchia Giovanni and Perret Benjamin (2020). *Ultrametric fitting by gradient descent*. In: *Proceedings of the 33rd International Conference on Neural Information*

Processing Systems. Red Hook, NY, USA: Curran Associates Inc., p. 124004. DOI: 10.1088/1742-5468/abc62d (cit. on p. 41).

CHONG Hamilton Y., GORTLER Steven J., and ZICKLER Todd (2008). *A perception-based color space for illumination-invariant image processing*. In: *ACM Transactions on Graphics* 27, pp. 1–7. DOI: 10.1145/1360612.1360660 (cit. on p. 69).

CHUANG Ching-Yao et al. (2018). *Learning to act properly: predicting and explaining affordances from images*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 975–983. DOI: 10.1109/CVPR.2018.00108 (cit. on p. 102).

CLÉMENT Michaël, KURTZ Camille, and WENDLING Laurent (2018). *Learning spatial relations and shapes for structural object description and scene recognition*. In: *Pattern Recognition* 84, pp. 197–210. DOI: 10.1016/j.patcog.2018.06.017 (cit. on pp. 18, 66, 67, 7).

COMANICIU Dorin and MEER Peter (2002). *Mean shift: a robust approach toward feature space analysis*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, pp. 603–619. DOI: 10.1109/34.1000236 (cit. on p. 66).

CORDTS Marius et al. (2016). *The cityscapes dataset for semantic urban scene understanding*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (cit. on p. 86).

CORTES Corinna and VAPNIK Vladimir (1995). *Support-vector networks*. In: *Machine Learning* 20, pp. 273–297. DOI: 10.1007/BF00994018 (cit. on p. 67).

COUSTY Jean, BERTRAND Giles, et al. (2009). *Watershed cuts: minimum spanning forests and the drop of water principle*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, pp. 1362–1374. DOI: https://doi.org/10.1109/TPAMI.2008.173 (cit. on p. 37).

COUSTY Jean and NAJMAN Laurent (2011). *Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts*. In: *Mathematical Morphology and Its Applications to Image and Signal Processing*. Springer, pp. 272–283. DOI: 10.1007/978-3-642-21569-8_24 (cit. on pp. 35, 37, 125).

— (2014). *Morphological floodings and optimal cuts in hierarchies*. In: *IEEE International Conference on Image Processing*. IEEE, pp. 4462–4466. DOI: 10.1109/ICIP.2014.7025905 (cit. on pp. 41, 73).

COUSTY Jean, NAJMAN Laurent, KENMOCHI Yukiko, et al. (2018). *Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps*. In:

Journal of Mathematical Imaging and Vision 60, pp. 479–502. DOI: 10.1007/s10851-017-0768-7 (cit. on pp. 32, 35, 36, 38, 125).

Cousty Jean, Najman Laurent, and Perret Benjamin (2013). *Constructive links between some morphological hierarchies on edge-weighted graphs*. In: *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer Berlin Heidelberg, pp. 86–97. DOI: 10.1007/978-3-642-38294-9_8 (cit. on pp. 28, 32).

Cretu Ana-Maria and Payeur Pierre (2013). *Building detection in aerial images based on watershed and visual attention feature descriptors*. In: *International Conference on Computer and Robot Vision*. IEEE, pp. 265–272. DOI: 10.1109/CRV.2013.8 (cit. on pp. 81, 84).

Cucurull Guillem, Taslakian Perouz, and Vazquez David (2019). *Context-aware visual compatibility prediction*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 12609–12618. DOI: 10.1109/CVPR.2019.01290 (cit. on p. 98).

Cutler Adele, Cutler D. Richard, and Stevens John R. (2012). *Random forests*. In: *Ensemble Machine Learning*. Springer US, pp. 157–175. DOI: 10.1007/978-1-4419-9326-7_5 (cit. on pp. 103, 110, 112).

Dalal Navneet and Triggs Bill (2005). *Histograms of oriented gradients for human detection*. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 886–893. DOI: 10.1109/CVPR.2005.177 (cit. on p. 67).

Demir Fatih (2021). *DeepCoronet: a deep LSTM approach for automated detection of Covid-19 cases from chest X-ray images*. In: *Applied Soft Computing* 103, p. 107160. DOI: 10.1016/j.asoc.2021.107160 (cit. on pp. 78, 80, 84).

Demirci M. Fatih and Kacka Serdar (2016). *Object recognition by distortion-free graph embedding and random forest*. In: *IEEE Tenth International Conference on Semantic Computing*. IEEE, pp. 17–23. DOI: 10.1109/ICSC.2016.46 (cit. on p. 105).

Dempster Arthur P., Laird Nan M., and Rubin Donald B. (1977). *Maximum likelihood from incomplete data via the em algorithm*. In: *Journal of the Royal Statistical Society: Series B* 39, pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x (cit. on p. 70).

Derivaux S. et al. (2010). *Supervised image segmentation using watershed transform, fuzzy classification and evolutionary computation*. In: *Pattern Recognition Letters* 31.15, pp. 2364–2374. DOI: 10.1016/j.patrec.2010.07.007 (cit. on pp. 81, 82, 84).

Derivaux Sébastien et al. (2007). *On machine learning in watershed segmentation*. In: *IEEE Workshop on Machine Learning for Signal Processing*. IEEE, pp. 187–192. doi: 10.1109/MLSP.2007.4414304 (cit. on pp. 81, 82, 84).

De Souza Kleber Jacques et al. (2013). *Hierarchical video segmentation using an observation scale*. In: *Conference on Graphics, Patterns and Images*. IEEE, pp. 320–327. doi: 10.1109/SIBGRAPI.2013.51 (cit. on p. 31).

Díaz Gloria, González Fabio A., and Romero Eduardo (2009). *A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images*. In: *Journal of Biomedical Informatics* 42, pp. 296–307. doi: 10.1016/j.jbi.2008.11.005 (cit. on pp. 69, 70, 71).

Diego Ferran et al. (2013). *Automated identification of neuronal activity from calcium imaging by sparse dictionary learning*. In: *IEEE International Symposium on Biomedical Imaging*. IEEE, pp. 1058–1061. doi: 10.1109/ISBI.2013.6556660 (cit. on pp. 79, 83).

Dollar Piotr, Belongie Serge, and Perona Pietro (2010). *The fastest pedestrian detector in the west*. In: *Procedings of the British Machine Vision Conference*. British Machine Vision Association, pp. 68.1–68.11. doi: 10.5244/C.24.68 (cit. on pp. 119, 120, 121).

Dollar Piotr and Zitnick C. Lawrence (2015). *Fast edge detection using structured forests*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, pp. 1558–1570. doi: 10.1109/TPAMI.2014.2377715 (cit. on pp. 40, 113, 123, 20).

Domínguez Didier and Morales Roberto Rodriguez (2016). *Image segmentation: advances*. Vol. 1 (cit. on p. 123).

Dotsenko Viktor S. (1986). *Hierarchical model of memory*. In: *Physica A: Statistical Mechanics and its Applications* 140.1-2, pp. 410–415. doi: 10.1016/0378-4371(86)90248-7 (cit. on p. 16).

Eetha Sagar, Agrawal Sonali, and Neelam Srikanth (2018). *Zynq FPGA based system design for video surveillance with Sobel edge detection*. In: *IEEE International Symposium on Smart Electronic Systems*. IEEE, pp. 76–79. doi: 10.1109/iSES.2018.00025 (cit. on p. 123).

Elmoataz A., Lezoray O., and Bougleux S. (2008). *Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing*. In: *IEEE Transactions on Image Processing* 17, pp. 1047–1060. doi: 10.1109/TIP.2008.924284 (cit. on pp. 109, 111).

Everingham Mark et al. (2010). *The pascal visual object classes (VOC) challenge*. In: *International Journal of Computer Vision* 88, pp. 303–338. doi: 10.1007/s11263-009-0275-4 (cit. on p. 67).

Fan Jianping et al. (2017). *HD-MTL: hierarchical deep multi-task learning for large-scale visual recognition*. In: *IEEE Transactions on Image Processing* 26, pp. 1923–1938. doi: 10.1109/TIP.2017.2667405 (cit. on pp. 16, 62, 5).

Fan Lifeng et al. (2019). *Understanding human gaze communication by spatio-temporal graph reasoning*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 5723–5732. doi: 10.1109/ICCV.2019.00582 (cit. on pp. 101, 102).

Fehri Amin (2018). *Image characterization by morphological hierarchical representations*. PhD thesis. Université Paris sciences et lettres (cit. on p. 27).

Felzenszwalb Pedro F. and Huttenlocher Daniel P. (2004). *Efficient graph-based image segmentation*. In: *International Journal of Computer Vision* 59, pp. 167–181. doi: 10.1023/B:VISI.0000022288.19776.77 (cit. on p. 66).

Fogel I. and Sagi D. (1989). *Gabor filters as texture discriminator*. In: *Biological Cybernetics* 61, pp. 103–113. doi: 10.1007/BF00204594 (cit. on p. 110).

Fu Sichao, Yang Xinghao, and Liu Weifeng (2018). *The comparison of different graph convolutional neural networks for image recognition*. In: *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*. ACM Press, pp. 1–6. doi: 10.1145/3240876.3240915 (cit. on p. 97).

Gao Junyu and Xu Changsheng (2020). *CI-GNN: building a category-instance graph for zero-shot video classification*. In: *IEEE Transactions on Multimedia* 22, pp. 3088–3100. doi: 10.1109/TMM.2020.2969787 (cit. on p. 100).

Gao Junyu, Zhang Tianzhu, and Xu Changsheng (2021). *Learning to model relationships for zero-shot video classification*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, pp. 3476–3491. doi: 10.1109/TPAMI.2020.2985708 (cit. on p. 100).

Gao Yali et al. (2018). *Graph mining-based trust evaluation mechanism with multidimensional features for large-scale heterogeneous threat intelligence*. In: *IEEE International Conference on Big Data*. IEEE, pp. 1272–1277. doi: 10.1109/BigData.2018.8622111 (cit. on pp. 104, 105).

Garnier Mickaël, Hurtut Thomas, and Wendling Laurent (2012). *Object description based on spatial relations between level-sets*. In: *International Conference on Digital*

Image Computing Techniques and Applications. IEEE, pp. 1–7. DOI: 10.1109/DICTA.2012.6411730 (cit. on p. 66).

GEORGE Yasmeen Mourice et al. (2014). *Remote computer-aided breast cancer detection and diagnosis system based on cytological images*. In: *IEEE Systems Journal* 8, pp. 949–964. DOI: 10.1109/JSYST.2013.2279415 (cit. on pp. 78, 80, 83).

GÉRARD Biau, DEVROYE Luc, and LUGOSI Gábor (2008). *Consistency of random forests and other averaging classifiers*. In: *Journal of Machine Learning Research* 9, pp. 2015–2033 (cit. on pp. 104, 112).

GIDARIS Spyros and KOMODAKIS Nikos (2019). *Generating classification weights with GNN denoising autoencoders for few-shot learning*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 21–30. DOI: 10.1109/CVPR.2019.00011 (cit. on pp. 96, 100).

GOBBER Charles, ALVES Wonder A. L., and HASHIMOTO Ronaldo F. (2018). *Ultimate leveling based on Mumford-Shah energy functional applied to plant detection*. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer International Publishing, pp. 220–228. DOI: 10.1007/978-3-319-75193-1_27 (cit. on p. 75).

GOLDBERG Yoav and LEVY Omer (2014). *Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. In: *arXiv* (cit. on p. 93).

GONG Jibing et al. (2020). *Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view*. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 79–88. DOI: 10.1145/3397271.3401057 (cit. on p. 98).

GONZALEZ Rafael (2009). *Digital image processing*. 3rd ed. Pearson Education, pp. 126–220 (cit. on p. 121).

GRADY Leo (2006). *Random walks for image segmentation*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, pp. 1768–1783. DOI: 10.1109/TPAMI.2006.233 (cit. on p. 65).

GROSSIORD Eloise et al. (2017). *Automated 3D lymphoma lesion segmentation from PET/CT characteristics*. In: *International Symposium on Biomedical Imaging*. IEEE, pp. 174–178. DOI: 10.1109/ISBI.2017.7950495 (cit. on pp. 64, 65, 67, 146).

GROSSIORD Éloïse et al. (2020). *Shaping for PET image analysis*. In: *Pattern Recognition Letters* 131, pp. 307–313. DOI: 10.1016/j.patrec.2020.01.017 (cit. on p. 74).

Grover Aditya and Leskovec Jure (2016). *Node2vec: scalable feature learning for networks*. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 855–864. doi: 10.1145/2939672.2939754 (cit. on p. 93).

Guigues Laurent, Cocquerez Jean Pierre, and Men Hervé Le (2006). *Scale-sets image analysis*. In: *International Journal of Computer Vision* 68, pp. 289–317. doi: 10.1007/s11263-005-6299-0 (cit. on pp. 27, 34, 41, 73).

Guillame-Bert Mathieu and Dubrawski Artur (2017). *Classification of time sequences using graphs of temporal constraints*. In: *The Journal of Machine Learning Research* 18, pp. 4370–1277 (cit. on p. 105).

Guimarães Silvio et al. (2017). *Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity*. In: *Mathematical Morphology - Theory and Applications* 2. doi: 10.1515/mathm-2017-0004 (cit. on p. 77).

Guo Xin et al. (2020). *Graph neural networks for image understanding based on multiple cues: group emotion recognition and event recognition as use cases*. In: *IEEE Winter Conference on Applications of Computer Vision*. IEEE, pp. 2910–2919. doi: 10.1109/WACV45572.2020.9093547 (cit. on p. 102).

He Xin, Liu Qiong, and Yang You (2020). *MV-GNN: multi-view graph neural network for compression artifacts reduction*. In: *IEEE Transactions on Image Processing* 29, pp. 6829–6840. doi: 10.1109/TIP.2020.2994412 (cit. on p. 99).

Hernandez Jorge and Marcotegui Beatriz (2009). *Filtering of artifacts and pavement segmentation from mobile LiDAR Data*. In: *ISPRS Workshop Laserscanning*. Paris, France (cit. on p. 68).

Herzig Roei et al. (2019). *Spatio-temporal action graph networks*. In: *IEEE/CVF International Conference on Computer Vision Workshop*. IEEE, pp. 2347–2356. doi: 10.1109/ICCVW.2019.00288 (cit. on pp. 96, 100).

Hirata Roberto et al. (2000). *Color image gradients for morphological segmentation*. In: *Proceedings 13th Brazilian Symposium on Computer Graphics and Image Processing*. IEEE Comput. Soc, pp. 316–326. doi: 10.1109/SIBGRA.2000.883928 (cit. on p. 122).

Honeycutt Wesley T. and Bridge Eli S. (2021). *UnCanny: exploiting reversed edge detection as a basis for object tracking in video*. In: *Journal of Imaging* 7, p. 77. doi: 10.3390/jimaging7050077 (cit. on p. 123).

Hu Zhongwen, Shi Tiezhu, et al. (2021). *Scale-sets image classification with hierarchical sample enriching and automatic scale selection*. In: *International Journal of Applied*

Earth Observation and Geoinformation 105, p. 102605. DOI: 10.1016/j.jag.2021.102605 (cit. on pp. 65, 67, 146).

Hu Zhongwen, Wu Zhaocong, et al. (2013). *A spatially-constrained color–texture model for hierarchical VHR image segmentation*. In: *IEEE Geoscience and Remote Sensing Letters* 10, pp. 120–124. DOI: 10.1109/LGRS.2012.2194693 (cit. on p. 66).

Huang Jiashuang et al. (2020). *A novel node-level structure embedding and alignment representation of structural networks for brain disease analysis*. In: *Medical Image Analysis* 65, pp. 101755–110767. DOI: 10.1016/j.media.2020.101755 (cit. on pp. 96, 97).

Huang Ziling et al. (2019). *DoT-GNN: domain-transferred graph neural network for group re-identification*. In: *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, pp. 1888–1896. DOI: 10.1145/3343031.3351027 (cit. on p. 101).

Ilin Roman, Watson Thomas, and Kozma Robert (2017). *Abstraction hierarchy in deep learning neural networks*. In: *International Joint Conference on Neural Networks*. Anchorage, AK, USA: IEEE, pp. 768–774. DOI: 10.1109/IJCNN.2017.7965929 (cit. on pp. 16, 62, 5).

Jeong Hyeonwoo, Choi Ye-Chan, and Choi Kang-Sun (2021). *Parallelization of levelset-based text baseline detection in document images*. In: *International Conference on Artificial Intelligence in Information and Communication*. IEEE, pp. 48–51. DOI: 10.1109/ICAIIC51459.2021.9415268 (cit. on p. 123).

Ji Wei et al. (2020). *Context-aware graph label propagation network for saliency detection*. In: *IEEE Transactions on Image Processing* 29, pp. 8177–8186. DOI: 10.1109/TIP.2020.3002083 (cit. on pp. 96, 97).

Jiao Licheng et al. (2010). *Natural and remote sensing image segmentation using memetic computing*. In: *IEEE Computational Intelligence Magazine* 5, pp. 78–91. DOI: 10.1109/MCI.2010.936307 (cit. on pp. 82, 84).

Jimenez-Sanchez Daniel, Ariz Mikel, and Ortiz-de-Solorzano Carlos (2020). *Unsupervised learning of contextual information in multiplex immunofluorescence tissue cytometry*. In: *IEEE 17th International Symposium on Biomedical Imaging*. IEEE, pp. 1275–1279. DOI: 10.1109/ISBI45749.2020.9098352 (cit. on p. 100).

Jing Ya et al. (2020). *Relational graph neural network for situation recognition*. In: *Pattern Recognition* 108, p. 107544. DOI: 10.1016/j.patcog.2020.107544 (cit. on p. 102).

Junejo Aisha Zahid et al. (2018). *Brain tumor segmentation using 3D magnetic resonance imaging scans*. In: *1st International Conference on Advanced Research in Engineering Sciences*. IEEE, pp. 1–6. DOI: 10.1109/ARESX.2018.8723285 (cit. on p. 123).

Karunaratne Thashmee, Bostrom Henrik, and Norinder Ulf (2010). *Pre-processing structured data for standard machine learning algorithms by supervised graph propositionalization: a case study with medicinal chemistry datasets*. In: *9th International Conference on Machine Learning and Applications*. IEEE, pp. 828–833. DOI: 10.1109/ICMLA.2010.128 (cit. on p. 105).

Karunaratne Thashmee and Boström Henrik (2009). *Graph propositionalization for random forests*. In: *International Conference on Machine Learning and Applications*. IEEE, pp. 196–201. DOI: 10.1109/ICMLA.2009.113 (cit. on p. 105).

Khalimsky Efim, Kopperman Ralph, and Meyer Paul R. (1990). *Computer graphics and connected topologies on finite ordered sets*. In: *Topology and its Applications* 36, pp. 1–17. DOI: 10.1016/0166-8641(90)90031-V (cit. on p. 114).

Kim Eunji et al. (2022). *Deep learning-based phenotypic assessment of red cell storage lesions for safe transfusions*. In: *IEEE Journal of Biomedical and Health Informatics* 26, pp. 1318–1328. DOI: 10.1109/JBHI.2021.3104650 (cit. on p. 63).

Kim Jaehwan and Lee Junsuk (2020). *Adaptive directional walks for pose estimation from single body depths*. In: *IEEE International Conference on Multimedia and Expo*. IEEE, pp. 1–6. DOI: 10.1109/ICME46284.2020.9102922 (cit. on p. 105).

Kim Jongmin et al. (2019). *Edge-labeling graph neural network for few-shot learning*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 11–20. DOI: 10.1109/CVPR.2019.00010 (cit. on p. 100).

Kiran B. Ravi and Serra Jean (2014). *Global–local optimizations by hierarchical cuts and climbing energies*. In: *Pattern Recognition* 47, pp. 12–24. DOI: 10.1016/j.patcog.2013.05.012 (cit. on pp. 41, 71).

Kiran Bangalore Ravi and Serra Jean (2015). *Braids of partitions*. In: *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer International Publishing, pp. 217–228. DOI: 10.1007/978-3-319-18720-4_19 (cit. on pp. 31, 41).

Koepfler Georges, Lopez Christian, and Morel Jean-Michel (1994). *A multiscale algorithm for image segmentation by variational method*. In: *SIAM Journal on Numerical Analysis* 31, pp. 282–299. DOI: 10.1137/0731015 (cit. on p. 73).

Krishnammal Perumal Muthu et al. (2022). *Wavelets and convolutional neural networks-based automatic segmentation and prediction of MRI brain images*. In: *IOT with Smart Systems*. Singapore: Springer, pp. 229–241. doi: 10.1007/978-981-16-3945-6_23 (cit. on p. 4).

Kruskal Joseph B. (1956). *On the shortest spanning subtree of a graph and the traveling salesman problem*. In: *Proceedings of the American Mathematical Society* 7, p. 48. doi: 10.2307/2033241 (cit. on p. 35).

Kuo Jay (2016). *Understanding convolutional neural networks with a mathematical model*. In: *Journal of Visual Communication and Image Representation* 41, pp. 406–413. doi: 10.1016/j.jvcir.2016.11.003 (cit. on pp. 17, 6).

Kurtz Camille, Naegel Benoit, and Passat Nicolas (2014). *Connected filtering based on multivalued component-trees*. In: *IEEE Transactions on Image Processing* 23, pp. 5152–5164. doi: 10.1109/TIP.2014.2362053 (cit. on p. 39).

Kurtz Camille, Passat Nicolas, et al. (2012). *Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology*. In: *Pattern Recognition* 45, pp. 685–706. doi: 10.1016/j.patcog.2011.07.017 (cit. on pp. 72, 73).

Kurtz Camille, Stumpf André, et al. (2014). *Hierarchical extraction of landslides from multiresolution remotely sensed optical images*. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 87, pp. 122–136. doi: 10.1016/j.isprsjprs.2013.11.003 (cit. on pp. 72, 73).

Kurzweil Ray (2013). *How to create a mind: the secret of human thought revealed*. Ed. by Giffords Bruce and Ottewell Roland. Penguin Books, pp. 34–41 (cit. on pp. 15, 4).

Lakshmi M. Muthu and Chitra P. (2020). *Tooth decay prediction and classification from X-ray images using deep CNN*. In: *International Conference on Communication and Signal Processing*. IEEE, pp. 1349–1355. doi: 10.1109/ICCSP48568.2020.9182141 (cit. on p. 123).

Levner Ilya and Zhang Hong (2007). *Classification-driven watershed segmentation*. In: *IEEE Transactions on Image Processing* 16, pp. 1437–1445. doi: 10.1109/TIP.2007.894239 (cit. on pp. 83, 84).

Li Jinghui and Fang Peiyu (2019). *FVGNN: a novel GNN to finger vein recognition from limited training data*. In: *IEEE 8th Joint International Information Technology and*

Artificial Intelligence Conference. IEEE, pp. 144–148. DOI: 10.1109/ITAIC.2019.8785512 (cit. on p. 103).

Li Kaiyue, Ding Guangtai, and Wang Haitao (2018). *L-FCN: A lightweight fully convolutional network for biomedical semantic segmentation*. In: *IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, pp. 2363–2367. DOI: 10.1109/BIBM.2018.8621265 (cit. on pp. 78, 84).

Li Ruiyu et al. (2017). *Situation recognition with graph neural networks*. In: *IEEE International Conference on Computer Vision*. IEEE, pp. 4183–4192. DOI: 10.1109/ICCV.2017.448 (cit. on p. 102).

Li Wanhua et al. (2020). *Graph-based kinship reasoning network*. In: *IEEE International Conference on Multimedia and Expo*. IEEE, pp. 1–6. DOI: 10.1109/ICME46284.2020.9102823 (cit. on pp. 96, 100).

Li Yaoyu et al. (2019). *Adaptive feature fusion via graph neural network for person re-identification*. In: *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, pp. 2115–2123. DOI: 10.1145/3343031.3350982 (cit. on p. 101).

Li Yujia et al. (2016). *Gated graph sequence neural networks*. In: *4th International Conference on Learning Representations*. DBLP (cit. on p. 95).

Li Zongmin et al. (2019). *Graph attention neural networks for point cloud recognition*. In: *IEEE International Conference on Multimedia and Expo*. IEEE, pp. 387–392. DOI: 10.1109/ICME.2019.00074 (cit. on pp. 96, 99).

Liang Jianheng and Huang Dong (2019). *Laplacian-weighted random forest for high-dimensional data classification*. In: *IEEE Symposium Series on Computational Intelligence*. IEEE, pp. 748–753. DOI: 10.1109/SSCI44817.2019.9003067 (cit. on p. 105).

Lin Tsung-Yi et al. (2017). *Feature pyramid networks for object detection*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA: IEEE, pp. 936–944. DOI: 10.1109/CVPR.2017.106 (cit. on pp. 16, 62, 5).

Lindsay Grace W. (2021). *Convolutional neural networks as a model of the visual system: Past, present, and future*. In: *Journal of cognitive neuroscience* 33.10, pp. 2017–2031. DOI: 10.1162/jocn_a_01544 (cit. on p. 16).

Liu Bingbin et al. (2020). *Spatiotemporal relationship reasoning for pedestrian intent prediction*. In: *IEEE Robotics and Automation Letters* 5, pp. 3485–3492. DOI: 10.1109/LRA.2020.2976305 (cit. on pp. 101, 102).

Liu Hongying et al. (2019). *A novel deep framework for change detection of multi-source heterogeneous images*. In: *International Conference on Data Mining Workshops*. IEEE, pp. 165–171. DOI: 10.1109/ICDMW.2019.00034 (cit. on p. 100).

Liu Jinde et al. (2020). *Kinematic skeleton graph augmented network for human parsing*. In: *Neurocomputing* 413, pp. 457–470. DOI: 10.1016/j.neucom.2020.07.002 (cit. on p. 96).

Liu Tao, Im Jungho, and Quackenbush Lindi J. (2015). *A novel transferable individual tree crown delineation model based on Fishing Net Dragging and boundary classification*. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 110, pp. 34–47. DOI: 10.1016/j.isprsjprs.2015.10.002 (cit. on pp. 81, 84).

Liu Ting et al. (2013). *Watershed merge forest classification for electron microscopy image stack segmentation*. In: *IEEE International Conference on Image Processing*. IEEE, pp. 4069–4073. DOI: 10.1109/ICIP.2013.6738838 (cit. on pp. 78, 83).

Liu Weibo et al. (2017). *A survey of deep neural network architectures and their applications*. In: *Neurocomputing* 234, pp. 11–26. DOI: 10.1016/j.neucom.2016.12.038 (cit. on pp. 17, 6).

Liu Yun et al. (2019). *Richer convolutional features for edge detection*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, pp. 1939–1946. DOI: 10.1109/TPAMI.2018.2878849 (cit. on pp. 16, 50, 62, 129, 133, 5, 20).

Lopez-Fandino Javier et al. (2018). *Stacked autoencoders for multiclass change detection in hyperspectral images*. In: *IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 1906–1909. DOI: 10.1109/IGARSS.2018.8518338 (cit. on pp. 82, 84).

Lu Yi et al. (2021). *CNN-G: convolutional neural network combined with graph for image segmentation with theoretical analysis*. In: *IEEE Transactions on Cognitive and Developmental Systems* 13, pp. 631–644. DOI: 10.1109/TCDS.2020.2998497 (cit. on p. 102).

Luo Wu et al. (2019). *Improving action recognition with the graph-neural-network-based interaction reasoning*. In: *IEEE Visual Communications and Image Processing*. IEEE, pp. 1–4. DOI: 10.1109/VCIP47243.2019.8965768 (cit. on pp. 101, 102).

Ma Cong et al. (2019). *Deep Association: end-to-end graph-based learning for multiple object tracking with conv-graph neural network*. In: *Proceedings of the International Conference on Multimedia Retrieval*. ACM, pp. 253–261. DOI: 10.1145/3323873.3325010 (cit. on p. 100).

Ma Jiangtao et al. (2018). *De-anonymizing social networks with random forest classifier*. In: *IEEE Access* 6, pp. 10139–10150. DOI: 10.1109/ACCESS.2017.2756904 (cit. on p. 104).

Ma Wenping et al. (2012). *Image segmentation based on a hybrid Immune Memetic Algorithm*. In: *IEEE Congress on Evolutionary Computation*. IEEE, pp. 1–8. DOI: 10.1109/CEC.2012.6256422 (cit. on pp. 83, 84).

Maia Deise Santana et al. (2021). *Classification of remote sensing data with morphological attribute profiles: a decade of advances*. In: *IEEE Geoscience and Remote Sensing Magazine* 9, pp. 43–71. DOI: 10.1109/MGRS.2021.3051859 (cit. on p. 31).

Makarov Ilya et al. (2021). *Survey on graph embeddings and their applications to machine learning problems on graphs*. In: *PeerJ Computer Science* 7. DOI: 10.7717/peerj-cs.357 (cit. on p. 93).

Makrogiannis Sokratis et al. (2021). *A system for spatio-temporal cell detection and segmentation in time-lapse microscopy*. In: *IEEE International Conference on Bioinformatics and Biomedicine*. Houston, TX, USA: IEEE, pp. 2266–2273. DOI: 10.1109/BIBM52615.2021.9669421 (cit. on p. 4).

Maninis Kevis-Kokitsi et al. (2018). *Convolutional oriented boundaries: from image segmentation to high-level tasks*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, pp. 819–833. DOI: 10.1109/TPAMI.2017.2700300 (cit. on pp. 52, 63).

Mansilla Lucy A. C. and Miranda Paulo A. V. (2016). *Oriented image foresting transform segmentation: connectivity constraints with adjustable width*. In: *Conference on Graphics, Patterns and Images*. IEEE, pp. 289–296. DOI: 10.1109/SIBGRAPI.2016.047 (cit. on p. 49).

Mao Kelong et al. (2020). *Item tagging for information retrieval: a tripartite graph neural network based approach*. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 2327–2336. DOI: 10.1145/3397271.3401438 (cit. on p. 98).

Mao Xueyue, Xiao Xiao, and Lu Yilong (2022). *PolSAR data-based land cover classification using dual-channel watershed region-merging segmentation and bagging-ELM*. In: *IEEE Geoscience and Remote Sensing Letters* 19, pp. 1–5. DOI: 10.1109/LGRS.2020.3018162 (cit. on pp. 81, 84).

Marr David (1982). *Vision: a computational investigation into the human representation and processing of visual information*. Ed. by Acock Malcolm. MIT Press, pp. 8–37. DOI: 10.7551/mitpress/9780262514620.001.0001 (cit. on pp. 15, 4).

Martin D. et al. (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*. IEEE Comput. Soc, pp. 416–423. DOI: 10.1109/ICCV.2001.937655 (cit. on pp. 48, 114).

Massaro Alessandro (2021). *Image vision advances*. In: *Electronics in Advanced Research Industries: Industry 4.0 to Industry 5.0 Advances*. Ed. by Massaro Alessandro. Vol. 1. Wiley. Chap. 7, pp. 301–340. DOI: 10.1002/9781119716907.ch7 (cit. on pp. 15, 4).

Matejek Brian et al. (2019). *Biologically-constrained graphs for global connectomics reconstruction*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2084–2093. DOI: 10.1109/CVPR.2019.00219 (cit. on pp. 79, 84).

MATLAB version 7.10.0 (R2010a) (2010). The Mathworks, Inc. Natick, Massachusetts: The MathWorks Inc. (cit. on p. 116).

Merciol Franlois and Lefevre Sébastien (2012). *Fast image and video segmentation based on alpha-tree multiscale representation*. In: *International Conference on Signal Image Technology and Internet Based Systems*. IEEE, pp. 336–342. DOI: 10.1109/SITIS.2012.56 (cit. on p. 39).

Meyer Fernand (1996). *The dynamics of minima and contours*. In: *Mathematical Morphology and its Applications to Image and Signal Processing*. Springer, pp. 329–336. DOI: 10.1007/978-1-4613-0469-2_38 (cit. on pp. 35, 36).

— (1998). *From connected operators to levelings*. In: *Proceedings of the Fourth International Symposium on Mathematical Morphology and Its Applications to Image and Signal Processing*. Amsterdam, The Netherlands: Kluwer Academic Publishers, pp. 191–198 (cit. on p. 68).

— (2001). *Hierarchies of partitions and morphological segmentation*. In: *Scale-Space and Morphology in Computer Vision*. Vol. 2106. Springer Berlin Heidelberg, pp. 161–182. DOI: 10.1007/3-540-47778-0_14 (cit. on pp. 18, 7).

Meyer Fernand and Beucher Serge (1990). *Morphological segmentation*. In: *Journal of Visual Communication and Image Representation* 1, pp. 21–46. DOI: 10.1016/1047-3203(90)90014-M (cit. on pp. 15, 39, 78, 4).

Micheli A. (2009). *Neural network for graphs: a contextual constructive approach*. In: *IEEE Transactions on Neural Networks* 20.3, pp. 498–511. DOI: 10.1109/tnn.2008.2010350 (cit. on pp. 94, 95).

Mittal Himanshu et al. (2022). *A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets*. In: *Multimedia Tools*

and Applications 81, pp. 35001–35026. DOI: 10.1007/s11042-021-10594-9 (cit. on p. 123).

MOLINA Angel et al. (2021). *Automatic identification of malaria and other red blood cell inclusions using convolutional neural networks*. In: *Computers in Biology and Medicine* 136, p. 104680. DOI: 10.1016/j.compbiomed.2021.104680 (cit. on pp. 79, 80, 84).

MONASSE Pascal and GUICHARD Frédéric (2000). *Scale-space from a level lines tree*. In: *Journal of Visual Communication and Image Representation* 11, pp. 224–236. DOI: 10.1006/jvci.1999.0441 (cit. on p. 65).

MONTAVON Grégoire, SAMEK Wojciech, and MÜLLER Klaus-Robert (2018). *Methods for interpreting and understanding deep neural networks*. In: *Digital Signal Processing* 73, pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011 (cit. on pp. 17, 6).

MUMFORD David and SHAH Jayant (1985). *Boundary detection by minimizing functionals*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 17. San Francisco, pp. 137–154 (cit. on p. 71).

NAEGEL Benoît et al. (2007). *Segmentation using vector-attribute filters: Methodology and application to dermatological imaging*. In: *International Symposium on Mathematical Morphology*. Vol. 1. Rio de Janeiro, Brazil: INPE, pp. 239–250 (cit. on p. 75).

NAGODA Nadeesha and RANATHUNGA Lochandaka (2018). *Rice sample segmentation and classification using image processing and support vector machine*. In: *International Conference on Industrial and Information Systems*. IEEE, pp. 179–184. DOI: 10.1109/ICIINFS.2018.8721312 (cit. on pp. 83, 84).

NAJMAN Laurent and COUPRIE Michel (2006). *Building the component tree in quasi-linear time*. In: *IEEE Transactions on Image Processing* 15, pp. 3531–3539. DOI: 10.1109/TIP.2006.877518 (cit. on pp. 38, 69).

NAJMAN Laurent and COUSTY Jean (2014). *A graph-based mathematical morphology reader*. In: *Pattern Recognition Letters* 47, pp. 3–17. DOI: 10.1016/j.patrec.2014.05.007 (cit. on pp. 28, 29, 31, 32).

NAJMAN Laurent, COUSTY Jean, and PERRET Benjamin (2013). *Playing with Kruskal: algorithms for morphological trees in edge-weighted graphs*. In: *International Symposium on Mathematical Morphology*. Springer Berlin Heidelberg, pp. 135–146. DOI: 10.1007/978-3-642-38294-9_12 (cit. on pp. 35, 38).

NAJMAN Laurent and SCHMITT Michel (1996). *Geodesic saliency of watershed contours and hierarchical segmentation*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, pp. 1163–1173. DOI: 10.1109/34.546254 (cit. on p. 4).

Najman Laurent and Talbot Hugues (2013a). *Mathematical morphology: from theory to applications*. Ed. by Najman Laurent and Talbot Hugues. 1st ed. John Wiley & Sons, Inc., pp. 5–33. doi: 10.1002/9781118600788 (cit. on pp. 15, 16, 27, 4, 5).

— (2013b). *Mathematical morphology: from theory to applications*. Ed. by Najman Laurent and Talbot Hugues. 1st ed. Complete work. John Wiley & Sons, Inc. doi: 10.1002/9781118600788 (cit. on pp. 31, 32).

Nandy Kaustav et al. (2011). *Supervised learning framework for screening nuclei in tissue sections*. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEEE*. Boston, MA, USA: IEEE, pp. 5989–5992. doi: 10.1109/IEMBS.2011.6091480 (cit. on pp. 18, 79, 80, 83, 7).

Naveen P. and Sivakumar P. (2021). *Adaptive morphological and bilateral filtering with ensemble convolutional neural network for pose-invariant face recognition*. In: *Journal of Ambient Intelligence and Humanized Computing* 12, pp. 10023–10033. doi: 10.1007/s12652-020-02753-x (cit. on p. 123).

Neggers J. et al. (2016). *On image gradients in digital image correlation*. In: *International Journal for Numerical Methods in Engineering* 105, pp. 243–260. doi: 10.1002/nme.4971 (cit. on p. 122).

Nguyen Tin T. et al. (2019). *Feature extraction and clustering analysis of highway congestion*. In: *Transportation Research Part C: Emerging Technologies* 100, pp. 238–258. doi: 10.1016/j.trc.2019.01.017 (cit. on pp. 18, 31, 7).

Niepert Mathias, Ahmed Mohamed, and Kutzkov Konstantin (2016). *Learning convolutional neural networks for graphs*. In: *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, pp. 2014–2023 (cit. on p. 95).

Nistér David and Stewénius Henrik (2008). *Linear time maximally stable extremal regions*. In: *Computer Vision*. Springer Berlin Heidelberg, pp. 183–196. doi: 10.1007/978-3-540-88688-4_14 (cit. on p. 65).

O'Mahony Niall et al. (2019). *Deep learning vs. traditional computer vision*. In: *Advances in Intelligent Systems and Computing*. Vol. 943. Springer International Publishing, pp. 128–144. doi: 10.1007/978-3-030-17795-9_10 (cit. on pp. 16, 6).

Otiniano-Rodríguez Karla et al. (2019). *Hierarchy-based salient regions: a region detector based on hierarchies of partitions*. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Vol. 11401. Springer International Publishing, pp. 444–452. doi: 10.1007/978-3-030-13469-3_52 (cit. on pp. 40, 124).

Ou Mingdong et al. (2016). *Asymmetric transitivity preserving graph embedding*. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1105–1114. DOI: 10.1145/2939672.2939751 (cit. on pp. 93, 105).

Oztan Basak et al. (2011). *Classification of breast cancer grades through quantitative characterization of ductal structure morphology in three-dimensional cultures*. In: *ACM Conference on Bioinformatics, Computational Biology and Biomedicine*. ACM Press, pp. 153–161. DOI: 10.1145/2147805.2147822 (cit. on pp. 79, 83).

Pachaury Shruti et al. (2018). *Link prediction method using topological features and ensemble model*. In: *11th International Conference on Contemporary Computing*. IEEE, pp. 1–6. DOI: 10.1109/IC3.2018.8530624 (cit. on pp. 104, 105).

Padilla Francisco Javier Alvarez et al. (2021). *Random walkers on morphological trees: a segmentation paradigm*. In: *Pattern Recognition Letters* 141, pp. 16–22. DOI: 10.1016/j.patrec.2020.11.001 (cit. on pp. 64, 65, 67, 146).

Paiva Katrine et al. (2022). *Performance evaluation of segmentation methods for assessing the lens of the frog Thoropa miliaris from synchrotron-based phase-contrast micro-CT images*. In: *Physica Medica* 94, pp. 43–52. DOI: 10.1016/j.ejmp.2021.12.013 (cit. on p. 4).

Paris Sylvain and Durand Fredo (2007). *A topological approach to hierarchical segmentation using mean shift*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8. DOI: 10.1109/CVPR.2007.383228 (cit. on p. 41).

Parisot Sarah et al. (2018). *Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer's disease*. In: *Medical Image Analysis* 48, pp. 117–130. DOI: 10.1016/j.media.2018.06.001 (cit. on p. 98).

Pedregosa Fabian et al. (2011). *Scikit-learn: machine learning in Python*. In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on pp. 119, 161).

Perozzi Bryan, Al-Rfou Rami, and Skiena Steven (2014). *Deepwalk: online learning of social representations*. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 701–710. DOI: 10.1145/2623330.2623732 (cit. on p. 93).

Perret B. et al. (2019). *Higra: hierarchical graph analysis*. In: *SoftwareX* 10, p. 100335. DOI: 10.1016/j.softx.2019.100335 (cit. on pp. 116, 161).

Perret Benjamin and Collet Christophe (2015). *Connected image processing with multivariate attributes: an unsupervised Markovian classification approach*. In: *Computer*

Vision and Image Understanding 133, pp. 1–14. DOI: 10.1016/j.cviu.2014.09.008 (cit. on pp. 72, 74, 76).

Perret Benjamin, Cousty Jean, Guimaraes Silvio Jamil F., et al. (2018). *Evaluation of hierarchical watersheds*. In: *IEEE Transactions on Image Processing* 27, pp. 1676–1688. DOI: 10.1109/TIP.2017.2779604 (cit. on pp. 15, 34, 40, 42, 52, 124, 4).

Perret Benjamin, Cousty Jean, Guimarães Silvio Jamil Ferzoli, et al. (2019). *Removing non-significant regions in hierarchical clustering and segmentation*. In: *Pattern Recognition Letters* 128, pp. 433–439. DOI: 10.1016/j.patrec.2019.10.008 (cit. on pp. 40, 41, 124).

Pinto Tiago W. et al. (2014). *Image segmentation through combined methods: watershed transform, unsupervised distance learning and Normalized Cut*. In: *Southwest Symposium on Image Analysis and Interpretation*. IEEE, pp. 153–156. DOI: 10.1109/SSIAI.2014.6806052 (cit. on pp. 72, 77).

Pont-Tuset Jordi and Marques Ferran (2012). *Supervised assessment of segmentation hierarchies*. In: *European Conference on Computer Vision*. Springer Berlin Heidelberg, pp. 814–827. DOI: 10.1007/978-3-642-33765-9_58 (cit. on p. 40).

— (2013). *Measures and meta-measures for the supervised evaluation of image segmentation*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2131–2138. DOI: 10.1109/CVPR.2013.277 (cit. on pp. 127, 130).

Pouyan Maziyar Baran and Nourani Mehrdad (2017). *Clustering single-cell expression data using random forest graphs*. In: *IEEE Journal of Biomedical and Health Informatics* 21, pp. 1172–1181. DOI: 10.1109/JBHI.2016.2565561 (cit. on p. 104).

Prabaharan L. and Raghunathan A. (2021). *An improved convolutional neural network for abnormality detection and segmentation from human sperm images*. In: *Journal of Ambient Intelligence and Humanized Computing* 12, pp. 3341–3352. DOI: 10.1007/s12652-020-02773-7 (cit. on p. 123).

Priya Michael Mary Adline and Jawhar Joseph (2020). *Advanced lung cancer classification approach adopting modified graph clustering and whale optimisation-based feature selection technique accompanied by a hybrid ensemble classifier*. In: *IET Image Processing* 14, pp. 2204–2215. DOI: 10.1049/iet-ipr.2019.0178 (cit. on p. 104).

Qi Xiaojuan et al. (2017). *3D graph neural networks for RGBD semantic segmentation*. In: *IEEE International Conference on Computer Vision*. IEEE, pp. 5209–5218. DOI: 10.1109/ICCV.2017.556 (cit. on pp. 96, 99).

Randrianasoa Jimmy Francky et al. (2018). *Binary partition tree construction from multiple features for image segmentation*. In: *Pattern Recognition* 84, pp. 237–250. doi: 10.1016/j.patcog.2018.07.003 (cit. on p. 31).

Renton Guillaume et al. (2019). *Graph neural network for symbol detection on document images*. In: *International Conference on Document Analysis and Recognition Workshops*. IEEE, pp. 62–67. doi: 10.1109/ICDARW.2019.00016 (cit. on p. 103).

Ronneberger Olaf, Fischer Philipp, and Brox Thomas (2015). *U-net: convolutional networks for biomedical image segmentation*. In: *Medical Image Computing and Computer-Assisted Intervention*. Vol. 9351. Springer International Publishing, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28 (cit. on p. 79).

Ronse Christian (2014). *Ordering partial partitions for image segmentation and filtering: merging, creating and inflating blocks*. In: *Journal of Mathematical Imaging and Vision* 49, pp. 202–233. doi: 10.1007/s10851-013-0455-2 (cit. on pp. 27, 31, 33).

Roweis Sam T. and Saul Lawrence K. (2000). *Nonlinear dimensionality reduction by locally linear embedding*. In: *Science* 290, pp. 2323–2326. doi: 10.1126/science.290.5500.2323 (cit. on p. 93).

Roy Rukhmini, Mazumdar Suparna, and Chowdhury Ananda S. (2020). *MDL-IWS: multi-view deep learning with iterative watershed for pulmonary fissure segmentation*. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 1282–1285. doi: 10.1109/EMBC44109.2020.9175310 (cit. on pp. 79, 84).

Sakaguchi Aiki, Wu Renjie, and Kamata Sei-ichiro (2019). *Fundus image classification for diabetic retinopathy using disease severity grading*. In: *Proceedings of the 9th International Conference on Biomedical Engineering and Technology*. ACM Press, pp. 190–196. doi: 10.1145/3326172.3326198 (cit. on p. 97).

Salembier Philippe and Garrido Luis (2000a). *Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval*. In: *IEEE Transactions on Image Processing* 9, pp. 561–576. doi: 10.1109/83.841934 (cit. on pp. 35, 39).

— (2000b). *Connected operators based on region-tree pruning strategies*. In: *Proceedings 15th International Conference on Pattern Recognition*. IEEE Comput. Soc, pp. 367–370. doi: 10.1109/ICPR.2000.903561 (cit. on p. 73).

Salembier Philippe, Liesegang Sergi, and Lopez-Martinez Carlos (2019). *Ship detection in SAR images based on maxtree representation and graph signal processing*.

In: *IEEE Transactions on Geoscience and Remote Sensing* 57, pp. 2709–2724. DOI: 10.1109/TGRS.2018.2876603 (cit. on p. 41).

Salembier Philippe, Oliveras Albert, and Garrido Luis (1998). *Antiextensive connected operators for image and sequence processing*. In: *IEEE Transactions on Image Processing* 7, pp. 555–570. DOI: 10.1109/83.663500 (cit. on pp. 65, 69).

Sawatzky Johann et al. (2019). *What object should I use?: task driven object detection*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 7597–7606. DOI: 10.1109/CVPR.2019.00779 (cit. on p. 102).

Scarselli F. et al. (2009). *The graph neural network model*. In: *IEEE Transactions on Neural Networks* 20.1, pp. 61–80. DOI: 10.1109/TNN.2008.2005605 (cit. on pp. 94, 95).

Schroeder Brigit, Tripathi Subarna, and Tang Hanlin (2019). *Triplet-aware scene graph embeddings*. In: *IEEE/CVF International Conference on Computer Vision Workshop*. IEEE, pp. 1783–1787. DOI: 10.1109/ICCVW.2019.00221 (cit. on p. 103).

Schulter Samuel et al. (2017). *Deep network flow for multi-object tracking*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2730–2739. DOI: 10.1109/CVPR.2017.292 (cit. on p. 100).

Scornet Erwan (2016). *Random forests and kernel methods*. In: *IEEE Transactions on Information Theory* 62, pp. 1485–1500. DOI: 10.1109/TIT.2016.2514489 (cit. on p. 104).

Selvan Raghavendra et al. (2020). *Graph refinement based airway extraction using mean-field networks and graph neural networks*. In: *Medical Image Analysis* 64, p. 101751. DOI: 10.1016/j.media.2020.101751 (cit. on p. 97).

Serna Andrés and Marcotegui Beatriz (2014). *Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning*. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 93, pp. 243–255. DOI: 10.1016/j.isprsjprs.2014.03.015 (cit. on pp. 68, 70, 71).

Serra Jean (2006). *A lattice approach to image segmentation*. In: *Journal of Mathematical Imaging and Vision* 24, pp. 83–130. DOI: 10.1007/s10851-005-3616-0 (cit. on pp. 16, 27, 31, 33, 5).

Shao Huikai and Zhong Dexing (2019). *Few-shot palmprint recognition via graph neural networks*. In: *Electronics Letters* 55, pp. 890–892. DOI: 10.1049/el.2019.1221 (cit. on p. 103).

Shao Jingzhi et al. (2019). *Emotion recognition by edge-weighted hypergraph neural network*. In: *IEEE International Conference on Image Processing*. IEEE, pp. 2144–2148. DOI: 10.1109/ICIP.2019.8803207 (cit. on p. 102).

Sharad Kumar and Danezis George (2014). *An automated social graph de-anonymization technique*. In: *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, pp. 47–58. DOI: 10.1145/2665943.2665960 (cit. on pp. 104, 105).

Sharifi M., Fathy M., and Mahmoudi M.T. (2002). *A classified and comparative study of edge detection algorithms*. In: *International Conference on Information Technology: Coding and Computing*. IEEE Comput. Soc, pp. 117–120. DOI: 10.1109/ITCC.2002.1000371 (cit. on p. 122).

Shi Jianbo and Malik J. (2000). *Normalized cuts and image segmentation*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, pp. 888–905. DOI: 10.1109/34.868688 (cit. on pp. 63, 78).

Shi Lei et al. (2019). *Skeleton-based action recognition with directed graph neural networks*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 7904–7913. DOI: 10.1109/CVPR.2019.00810 (cit. on p. 97).

Shin Seung Yeon et al. (2019). *Deep vessel segmentation by learning graphical connectivity*. In: *Medical Image Analysis* 58, p. 101556. DOI: 10.1016/j.media.2019.101556 (cit. on p. 100).

Si Chenyang et al. (2020). *Skeleton-based action recognition with hierarchical spatial reasoning and temporal stack learning network*. In: *Pattern Recognition* 107, p. 107511. DOI: 10.1016/j.patcog.2020.107511 (cit. on pp. 96, 97).

Sidorov Oleksii and Hardeberg Jon Yngve (2019). *Craquelure as a graph: application of image processing and graph neural networks to the description of fracture patterns*. In: *IEEE/CVF International Conference on Computer Vision Workshop*. IEEE, pp. 1429–1436. DOI: 10.1109/ICCVW.2019.00180 (cit. on p. 100).

Simonyan Karen and Zisserman Andrew (2015). *Very deep convolutional networks for large-scale image recognition*. In: *3rd International Conference on Learning Representations*. San Diego, CA, USA (cit. on p. 85).

Sinchuk Yuriy et al. (2021). *Geometrical and deep learning approaches for instance segmentation of CFRP fiber bundles in textile composites*. In: *Composite Structures* 277, p. 114626. DOI: 10.1016/j.compstruct.2021.114626 (cit. on p. 63).

Singh Ajeet Kumar et al. (2019). *From strings to things: knowledge-enabled VQA model that can read and reason*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 4601–4611. DOI: 10.1109/ICCV.2019.00470 (cit. on p. 102).

Soille Pierre (2008). *Constrained connectivity for hierarchical image partitioning and simplification*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, pp. 1132–1145. DOI: 10.1109/TPAMI.2007.70817 (cit. on pp. 35, 36, 39, 73).

Soille Pierre and Najman Laurent (2012). *On morphological hierarchical representations for image processing and spatial data clustering*. In: *Applications of Discrete Geometry and Mathematical Morphology*. Springer Berlin Heidelberg, pp. 43–67. DOI: 10.1007/978-3-642-32313-3_4 (cit. on p. 31).

Sokal Robert R. and Rohlf James (1962). *The comparison of dendrograms by objective methods*. In: *TAXON* 11, pp. 33–40. DOI: 10.2307/1217208 (cit. on p. 32).

Soni Akanksha and Rai Avinash (2021). *Automatic cataract detection using Sobel and morphological dilation operation*. In: *Advances in Intelligent Systems and Computing*. Vol. 1355. Springer, pp. 267–276. DOI: 10.1007/978-981-16-1543-6_25 (cit. on p. 123).

Suhail Mohammed and Sigal Leonid (2019). *Mixture-kernel graph attention network for situation recognition*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 10362–10371. DOI: 10.1109/ICCV.2019.01046 (cit. on p. 102).

Sun Lei et al. (2015). *A robust approach for text detection from natural scene images*. In: *Pattern Recognition* 48, pp. 2906–2920. DOI: 10.1016/j.patcog.2015.04.002 (cit. on pp. 69, 70, 71).

Taylor Camillo Jose (2013). *Towards fast and accurate segmentation*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1916–1922. DOI: 10.1109/CVPR.2013.250 (cit. on p. 52).

Temir Askhat, Artykbayev Kamalkhan, and Demirci M. Fatih (2020). *Image classification by distortion-free graph embedding and KNN-random forest*. In: *17th Conference on Computer and Robot Vision*. IEEE, pp. 33–38. DOI: 10.1109/CRV50864.2020.00013 (cit. on pp. 104, 105).

Tkalcic Marko and Tasic Jurij (2003). *Colour spaces: perceptual, historical and applicational background*. In: *The IEEE Region 8 EUROCON 2003. Computer as a Tool*. IEEE, pp. 304–308. DOI: 10.1109/EURCON.2003.1248032 (cit. on p. 119).

Tochon Guillaume et al. (2018). *Advances in utilization of hierarchical representations in remote sensing data analysis*. In: *Reference Module in Earth Systems and Environ-*

mental Sciences. Elsevier, pp. 77–107. DOI: 10.1016/B978-0-12-409548-9.10340-9 (cit. on p. 31).

TU Zhuowen et al. (2008). *Brain anatomical structure segmentation by hybrid discriminative/generative models*. In: *IEEE Transactions on Medical Imaging* 27, pp. 495–508. DOI: 10.1109/TMI.2007.908121 (cit. on p. 123).

TZIMIROPOULOS G et al. (2010). *Robust FFT-based scale-invariant image registration with image gradients*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, pp. 1899–1906. DOI: 10.1109/TPAMI.2010.107 (cit. on p. 122).

UZUNBAS Mustafa Gokhan, CHEN Chao, and METAXAS Dimitris (2016). *An efficient conditional random field approach for automatic and interactive neuron segmentation*. In: *Medical Image Analysis* 27, pp. 31–44. DOI: 10.1016/j.media.2015.06.003 (cit. on pp. 72, 74).

VACHIER Corinne and MEYER Fernand (1995). *Extinction value: a new measurement of persistence*. In: *IEEE Workshop on nonlinear signal and image processing*. Vol. 1. Neos Marmaras Greece, pp. 254–257 (cit. on p. 37).

VINCENT Luc and SOILLE Pierre (1991). *Watersheds in digital spaces: an efficient algorithm based on immersion simulations*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, pp. 583–598. DOI: 10.1109/34.87344 (cit. on p. 78).

WANG Daixin, CUI Peng, and ZHU Wenwu (2016). *Structural deep network embedding*. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1225–1234. DOI: 10.1145/2939672.2939753 (cit. on p. 93).

WANG Dezhao et al. (2021). *Combining progressive rethinking and collaborative learning: a deep framework for in-loop filtering*. In: *IEEE Transactions on Image Processing* 30, pp. 4198–4211. DOI: 10.1109/TIP.2021.3068638 (cit. on p. 31).

WANG Jinguang et al. (2020). *Multimodal graph convolutional networks for high quality content recognition*. In: *Neurocomputing* 412, pp. 42–51. DOI: 10.1016/j.neucom.2020.04.145 (cit. on p. 98).

WANG Junbo et al. (2020). *Learning visual relationship and context-aware attention for image captioning*. In: *Pattern Recognition* 98, p. 107075. DOI: 10.1016/j.patcog.2019.107075 (cit. on p. 97).

WANG Wenguan et al. (2019). *Zero-shot video object segmentation via attentive graph neural networks*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 9235–9244. DOI: 10.1109/ICCV.2019.00933 (cit. on p. 100).

Wang Xiaochan et al. (2019). *Predicting gene-disease associations from the heterogeneous network using graph embedding*. In: *IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, pp. 504–511. DOI: 10.1109/BIBM47256.2019.8983134 (cit. on p. 105).

Ward Joe H. (1963). *Hierarchical grouping to optimize an objective function*. In: *Journal of the American Statistical Association* 58, p. 236. DOI: 10.2307/2282967 (cit. on p. 66).

Wei Yinwei et al. (2019). *MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video*. In: *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, pp. 1437–1445. DOI: 10.1145/3343031.3351034 (cit. on p. 98).

Whitney Jon et al. (2022). *Quantitative nuclear histomorphometry predicts molecular subtype and clinical outcome in medulloblastomas: preliminary findings*. In: *Journal of Pathology Informatics* 13, p. 100090. DOI: 10.1016/j.jpi.2022.100090 (cit. on pp. 78, 79, 80, 84).

Wu Jiaxin, Zhong Sheng-hua, and Liu Yan (2020). *Dynamic graph convolutional network for multi-video summarization*. In: *Pattern Recognition* 107, p. 107382. DOI: 10.1016/j.patcog.2020.107382 (cit. on p. 99).

Wu Le et al. (2020). *Learning to transfer graph embeddings for inductive graph based recommendation*. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 1211–1220. DOI: 10.1145/3397271.3401145 (cit. on pp. 96, 98).

Wu Yiming et al. (2020). *Adaptive graph representation learning for video person re-identification*. In: *IEEE Transactions on Image Processing* 29, pp. 8821–8830. DOI: 10.1109/TIP.2020.3001693 (cit. on p. 100).

Wyner Abraham et al. (2017). *Explaining the success of adaboost and random forests as interpolating classifiers*. In: *The Journal of Machine Learning Research* 18, pp. 1558–1590 (cit. on pp. 104, 110, 112).

Xie Lipeng et al. (2020). *Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images*. In: *Neurocomputing* 376, pp. 166–179. DOI: 10.1016/j.neucom.2019.09.083 (cit. on pp. 78, 84).

Xie Saining and Tu Zhuowen (2017). *Holistically-nested edge detection*. In: *International Journal of Computer Vision* 125, pp. 3–18. DOI: 10.1007/s11263-017-1004-z (cit. on pp. 129, 133, 20).

Xin Hai et al. (2011). *Human head-shoulder segmentation*. In: *IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, pp. 227–232 (cit. on pp. 83, 84).

Xu Chenliang, Whitt Spencer, and Corso Jason J. (2013). *Flattening supervoxel hierarchies by the uniform entropy slice*. In: *IEEE International Conference on Computer Vision*. IEEE, pp. 2240–2247. DOI: 10.1109/ICCV.2013.279 (cit. on pp. 40, 72, 77).

Xu Chenliang, Xiong Caiming, and Corso Jason J. (2012). *Streaming hierarchical video segmentation*. In: *European Conference on Computer Vision*. Springer Berlin Heidelberg, pp. 626–639. DOI: 10.1007/978-3-642-33783-3_45 (cit. on p. 31).

Xu Yongchao, Geraud Thierry, and Najman Laurent (2016). *Connected filtering on tree-based shape-spaces*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, pp. 1126–1140. DOI: 10.1109/TPAMI.2015.2441070 (cit. on p. 41).

Xu Yongchao, Géraud Thierry, and Najman Laurent (2016). *Hierarchical image simplification and segmentation based on Mumford–Shah-salient level line selection*. In: *Pattern Recognition Letters* 83, pp. 278–286. DOI: 10.1016/j.patrec.2016.05.006 (cit. on pp. 72, 77).

Yang Fengting et al. (2020). *Superpixel segmentation with fully convolutional networks*. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 13961–13970. DOI: 10.1109/CVPR42600.2020.01398 (cit. on p. 131).

Yang Guang et al. (2020). *Graph-based neural networks for explainable image privacy inference*. In: *Pattern Recognition* 105, p. 107360. DOI: 10.1016/j.patcog.2020.107360 (cit. on p. 102).

Yoon Ji-Seok et al. (2019). *Automated integrated system for stained neuron detection: An end-to-end framework with a high negative predictive rate*. In: *Computer Methods and Programs in Biomedicine* 180, p. 105028. DOI: 10.1016/j.cmpb.2019.105028 (cit. on pp. 78, 80, 84).

Yu Jing et al. (2020). *Cross-modal knowledge reasoning for knowledge-based visual question answering*. In: *Pattern Recognition* 108, p. 107563. DOI: 10.1016/j.patcog.2020.107563 (cit. on p. 102).

Zeiler Matthew D. and Fergus Rob (2014). *Visualizing and understanding convolutional networks*. In: *Computer Vision*. Vol. 8689. Springer International Publishing, pp. 818–833. doi: 10.1007/978-3-319-10590-1_53 (cit. on pp. 17, 6).

Zhang Zhen and Lee Wee Sun (2019). *Deep graphical feature learning for the feature matching problem*. In: *IEEE/CVF International Conference on Computer Vision*. IEEE, pp. 5086–5095. doi: 10.1109/ICCV.2019.00519 (cit. on p. 100).

Zheng Wenfeng et al. (2021). *Improving visual reasoning through semantic representation*. In: *IEEE Access* 9, pp. 91476–91486. doi: 10.1109/ACCESS.2021.3074937 (cit. on p. 63).

Zhou Shuang et al. (2019). *LncRNA-miRNA interaction prediction from the heterogeneous network through graph embedding ensemble learning*. In: *IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, pp. 622–627. doi: 10.1109/BIBM47256.2019.8983044 (cit. on pp. 104, 105).

Zwettler Gerald and Backfrieder Werner (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*. Chennai, India: IEEE, pp. 12–18. doi: 10.1109/ICCCT2.2015.7292712 (cit. on pp. 18, 79, 80, 83, 7).

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Apprentissage sur les graphes et les hiérarchies

Mot clés : graphes, hiérarchies morphologiques, apprentissage automatique, forêt aléatoire

Résumé : Les hiérarchies, telles que décrites dans la morphologie mathématique, représentent des régions d'intérêt imbriquées et fournissent des mécanismes pour créer des concepts et une organisation cohérente des données. Elles facilitent l'analyse de haut niveau et la gestion de grandes quantités de données. Représentées sous forme d'arbres hiérarchiques, elles ont des formalismes croisés avec la théorie des graphes, et des applications qui peuvent être facilement généralisées. En raison des algorithmes déterministes, des représentations multiformes et distinctes, et de l'absence d'un moyen direct d'évaluer la qualité de la représentation hiérarchique, il est difficile d'insérer des informations hiérarchiques dans un cadre d'apprentissage et de bénéficier des avancées récentes dans le domaine. Les chercheurs s'attaquent généralement à ce problème en affinant les hiérarchies pour un média spécifique et en évaluant leur qualité pour une tâche particulière. L'inconvénient de cette approche est qu'elle dépend de l'application et que les formulations limitent la généralisation à des données similaires. Ce travail vise à créer un cadre d'apprentissage qui peut fonctionner avec des données hiérarchiques et qui est agnostique à l'entrée et à l'application. L'idée est d'étudier les moyens de transformer les données en une représentation régulière requise par la plupart des modèles d'apprentissage tout en préservant la richesse de l'information dans la structure hiérarchique. Il propose d'étudier et de formaliser les concepts sous forme de graphes, un point commun pour les hiérarchies et le multimédia, et un sujet de grand intérêt pour l'apprentissage automatique. Les méthodes proposées dans cette étude utilisent des graphes d'images pondérés par des arêtes et des arbres hiérarchiques comme entrée, et évaluent différentes propositions sur les tâches de détection des contours et de segmentation. Le modèle principal est la forêt aléatoire, une méthode rapide, verifiable et extensible, adaptée au travail avec des données de grandes dimensions. Malgré les médias, les tâches et les choix de modèle , il concentre les formulations sur des graphes et des arbres hiérarchiques, et n'utilise les tâches que pour évaluer la réponse produite par différentes caractéristiques. Il donne les résultats en termes quantitatifs et qualitatifs et propose des analyses statistiques de la distribution et de la dimensionnalité des données, évaluant ainsi leur impact sur l'apprentissage. En outre, il fournit une revue systématique de la literature sur des propositions qui intègrent l'apprentissage automatique et les hiérarchies. Il démontre qu'il est possible de créer un cadre d'apprentissage dépendant uniquement des données hiérarchiques qui fonctionne dans plusieurs tâches.

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

Title: Learning on graphs and hierarchies

Keywords: graphs, morphological hierarchies, machine learning, random forest

Abstract: Hierarchies, as described in mathematical morphology, represent nested regions of interest and provide mechanisms to create concepts and 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 applications that can be conveniently generalized. Due to the deterministic algorithms, the multiform and distinct representations, and the absence of a direct way to evaluate the hierarchical representation quality, it is hard to insert hierarchical information into a learning framework and benefit from the recent advances in the field. 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 the application. The idea is to study ways to transform the data to a regular representation required by most learning models while preserving the rich information in the hierarchical structure. It proposes to study and formalize the concepts as graphs, a common point for hierarchies and multimedia, and a topic of great interest for machine learning. The methods proposed in this study use edge-weighted image graphs and hierarchical trees as input, and it evaluates different proposals on the edge detection and segmentation tasks. The primary model is the Random Forest, a fast, inspectable, and scalable method suited to work with high-dimensional data. Despite the media, tasks, and model choices, it focuses the formulations on graphs and hierarchical trees and only uses the tasks to evaluate the response produced by different characteristics. It gives the results in quantitative and qualitative terms and offers statistical analyses of the data distribution and dimensionality, assessing their impact on learning. Furthermore, it provides a critical systematic review of proposals in the literature that integrates machine learning and hierarchies. It demonstrates that it is possible to create a learning framework dependent only on the hierarchical data that performs well in multiple tasks.

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par

Raquel PEREIRA DE ALMEIDA

Apprentissage sur les graphes et les hiérarchies

Thèse présentée et soutenue à Rennes, le 27 mars 2023
Unité de recherche : Institut de Recherche en Informatique et Systèmes Aléatoires

Rapporteurs avant soutenance :

Davide BACCIU — Professeur associé, Université de Pise, Italie
Alexandre FALCÃO — Professeur, Unicamp, Brésil

Composition du Jury :

Président :	Laurent NAJMAN	Professeur, Université Gustave Eiffel, France
Examinateurs :	Yukiko KENMOCHI	Chargé de recherche DR CNRS, Greyc, France
	Zenilton PATROCÍNIO	Professeur, PUC-Minas, Brésil
	Davide BACCIU	Professeur associé, Université de Pise, Italie
	Alexandre FALCÃO	Professeur, Unicamp, Brésil
Dir. de thèse :	Laurent AMSALEG	Directeur de recherche CNRS, Université de Rennes 1, France
Co-dir. de thèse :	Silvio GUIMARÃES	Professeur, PUC-Minas, Brésil

Invité(s) :

Ewa KIJAK — Maître de Conférence, Université de Rennes 1, France
Simon MALINOWSKI — Maître de Conférence, Université de Rennes 1, France

RÉSUMÉ

Les hiérarchies, telles que décrites dans la morphologie mathématique, représentent des régions d'intérêt imbriquées et fournissent des mécanismes pour créer des concepts et une organisation cohérente des données. Elles facilitent l'analyse de haut niveau et la gestion de grandes quantités de données. Représentées sous forme d'arbres hiérarchiques, elles ont des formalismes croisés avec la théorie des graphes, et des applications qui peuvent être facilement généralisées. En raison des algorithmes déterministes, des représentations multiformes et distinctes, et de l'absence d'un moyen direct d'évaluer la qualité de la représentation hiérarchique, il est difficile d'insérer des informations hiérarchiques dans un cadre d'apprentissage et de bénéficier des avancées récentes dans le domaine. Les chercheurs s'attaquent généralement à ce problème en affinant les hiérarchies pour un média spécifique et en évaluant leur qualité pour une tâche particulière. L'inconvénient de cette approche est qu'elle dépend de l'application et que les formulations limitent la généralisation à des données similaires. Ce travail vise à créer un cadre d'apprentissage qui peut fonctionner avec des données hiérarchiques et qui est agnostique à l'entrée et à l'application. L'idée est d'étudier les moyens de transformer les données en une représentation régulière requise par la plupart des modèles d'apprentissage tout en préservant la richesse de l'information dans la structure hiérarchique. Il propose d'étudier et de formaliser les concepts sous forme de graphes, un point commun pour les hiérarchies et le multimédia, et un sujet de grand intérêt pour l'apprentissage automatique. Malgré les médias, les tâches et les choix de modèle , il concentre les formulations sur des graphes et des arbres hiérarchiques, et n'utilise les tâches que pour évaluer la réponse produite par différentes caractéristiques. Il donne les résultats en termes quantitatifs et qualitatifs et propose des analyses statistiques de la distribution et de la dimensionnalité des données, évaluant ainsi leur impact sur l'apprentissage. En outre, il fournit une revue systématique de la literature sur des propositions qui intègrent l'apprentissage automatique et les hiérarchies. Il démontre qu'il est possible de créer un cadre d'apprentissage dépendant uniquement des données hiérarchiques qui fonctionne dans plusieurs tâches.

INTRODUCTION

Les hiérarchies sont une propriété inhérente qui compose plusieurs éléments de la vie réelle, liés à la façon dont nous percevons naturellement les motifs, les scènes et les mouvements [1]. Selon KURZWEIL (2013) [2], il existe un identifiant de modèle au cœur de notre perception visuelle, fonctionnant de manière hiérarchique pour reconnaître simultanément des parties, des objets et des concepts abstraits. La hiérarchie perceptive est difficile à traduire en modèles informatiques imitant notre capacité à percevoir la nature intrinsèque de la réalité. Mais, dans le traitement des médias visuels, la morphologie mathématique a un avantage dans la définition, la création et la manipulation des hiérarchies.

Les méthodes hiérarchiques, formulées en morphologie mathématique [3], fournissent des structures sémantiquement organisées de régions imbriquées faciles à naviguer, à interpréter, et restent très populaires depuis leur création [4] [5] [6] [7] [8] [9] [10]. Cependant, elles sont difficiles à évaluer et à insérer dans les frameworks d'apprentissage pour bénéficier des avancées récentes dans le domaine [11].

Cette thèse centre son étude sur les hiérarchies, visant à créer un cadre d'apprentissage qui pourrait opérer sur les structures hiérarchiques du point de vue du traitement des médias. Cette introduction présente le contexte définissant les hiérarchies, dé-

1. MARR David (1982). *Vision: a computational investigation into the human representation and processing of visual information.*
2. KURZWEIL Ray (2013). *How to create a mind: the secret of human thought revealed.*
3. NAJMAN Laurent and TALBOT Hugues (2013a). *Mathematical morphology: from theory to applications.*
4. MEYER Fernand and BEUCHER Serge (1990). *Morphological segmentation.*
5. BEUCHER Serge (1994). *Watershed, hierarchical segmentation and waterfall algorithm.*
6. NAJMAN Laurent and SCHMITT Michel (1996). *Geodesic saliency of watershed contours and hierarchical segmentation.*
7. KRISHNAMMAL Perumal Muthu et al. (2022). *Wavelets and convolutional neural networks-based automatic segmentation and prediction of MRI brain images.*
8. PAIVA Katrine et al. (2022). *Performance evaluation of segmentation methods for assessing the lens of the frog Thoropa miliaris from synchrotron-based phase-contrast micro-CT images.*
9. MAKROGIANNIS Sokratis et al. (2021). *A system for spatio-temporal cell detection and segmentation in time-lapse microscopy.*
10. MASSARO Alessandro (2021). *Image vision advances.*
11. PERRET Benjamin, COUSTY Jean, GUIMARAES Silvio Jamil F., et al. (2018). *Evaluation of hierarchical watersheds.*

taille la problématique de leur insertion dans un cadre d'apprentissage, et énonce les objectifs de cette étude, en établissant quelques hypothèses et questions auxquelles répondre. Enfin, elle présente l'organisation de la thèse pour faciliter la navigation dans le document.

Contextualisation des hiérarchies

Les hiérarchies sont largement définies dans la littérature et pourraient représenter différents concepts. Par exemple, dans la littérature les hiérarchies sont présentés comme une abstraction de méthode [12], une description des architectures de modèles [13] et une forme pour organiser les fonctionnalités [14] ou les concepts [15]. Cette définition large renforce l'idée que les hiérarchies sont la forme d'organisation naturelle des données, en particulier les données visuelles dans le multimédia.

Les hiérarchies morphologiques utilisent des transformations non linéaires pour recueillir des informations en fonction de la réaction qu'elles produisent [16]. En ce sens, elles définissent les principes hiérarchiques comme des transformations obtenues en appliquant les opérateurs appropriés. Les opérateurs et concepts présentent un formalisme mathématique solide utilisant l'espace géométrique non linéaire pour représenter les formulations et généraliser la théorie des ensembles des réseaux complets [17].

Leurs méthodes représentent des régions d'intérêt imbriquées qui facilitent la navigation et les opérations de fusion pour créer des objets sémantiquement plus significatifs à partir d'instances de niveaux inférieurs. Dans le traitement multimédia, la délimitation de la région prend en compte les éléments constitutifs du média, tels que les pixels, les voxels et la fréquence [18]. Dans le même temps, les hiérarchies produisent des représentations multiformes, leurs algorithmes sont principalement déterministes et il n'existe aucun moyen direct d'évaluer leurs qualités.

12. Ilin Roman, Watson Thomas, and Kozma Robert (2017). *Abstraction hierarchy in deep learning neural networks.*
13. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*
14. Lin Tsung-Yi et al. (2017). *Feature pyramid networks for object detection.*
15. Fan Jianping et al. (2017). *HD-MTL: hierarchical deep multi-task learning for large-scale visual recognition.*
16. Najman Laurent and Talbot Hugues (2013a). *Mathematical morphology: from theory to applications.*
17. Serra Jean (2006). *A lattice approach to image segmentation.*
18. Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey.*

Récemment, les architectures d'apprentissage en profondeur ont radicalement changé le paradigme de calcul pour les tâches visuelles [19]. Le principal avantage de la méthodologie d'apprentissage en profondeur est qu'elle ne nécessite pas de modèle d'ingénierie pour fonctionner, ce qui signifie qu'elle peut apprendre les fonctionnalités pour représenter les données et les modèles pour les décrire [20]. Le succès de ces approches repose sur une hiérarchie de concepts appris via le réseau [21]. Par exemple, dans la tâche de reconnaissance d'objets, les pixels bruts sur la couche d'entrée sont compris comme des segments et des parties jusqu'à la composition du concept d'objet aux dernières couches.

L'approche typique d'apprentissage en profondeur est loin d'être idéale, car elle impose une structure rigide pour l'entrée, ce qui limite ses capacités de généralisation pour les données multiformes [22]. De plus, même avec les progrès récents dans l'explication et l'inspection des réseaux, le raisonnement derrière les inférences reste obscur [23] [24] et doit être plus empirique que formel.

Dans les deux domaines — hiérarchies des partitions et apprentissage en profondeur — la hiérarchie est la réaction créée par les opérations appliquées. Les hiérarchies de partitions font partie intégrante des structures, mais les méthodes déterministes produisant des données hétérogènes sont difficiles à améliorer à l'aide de l'apprentissage automatique. En revanche, l'apprentissage en profondeur présente des concepts hiérarchiques implicites, mais la généralisation et le raisonnement sont limités.

Formulation du problème

Dans les applications pratiques, les hiérarchies morphologiques aident à effectuer des tâches sémantiques dans le traitement des données visuelles, telles que la proposition d'objet, le contour sémantique et la segmentation sémantique [25]. Cependant, ils

19. O'Mahony Niall et al. (2019). *Deep learning vs. traditional computer vision.*
20. Liu Weibo et al. (2017). *A survey of deep neural network architectures and their applications.*
21. Zeiler Matthew D. and Fergus Rob (2014). *Visualizing and understanding convolutional networks.*
22. Bacciu Davide et al. (2020). *A gentle introduction to deep learning for graphs.*
23. Kuo Jay (2016). *Understanding convolutional neural networks with a mathematical model.*
24. Montavon Grégoire, Samek Wojciech, and Müller Klaus-Robert (2018). *Methods for interpreting and understanding deep neural networks.*
25. Bosilj Petra, Kijak Ewa, and Lefèvre Sébastien (2018). *Partition and inclusion hierarchies of images: a comprehensive survey.*

nécessitent un prétraitement minutieux des données [26] [27] et des stratégies pour traiter des problèmes tels que le sur/sous-partitionnement de l'espace [28] [29] ou la sélection d'un nombre idéal de régions [30]. Par conséquent, il est difficile de généraliser une approche réussie à d'autres médias et tâches.

Pour une généralisation en termes de médias, la plupart des défis concernent la caractérisation de l'information, principalement : les données des médias présentant des caractéristiques différentes, et les blocs de construction des médias composant les régions qui ont des connotations différentes. Ces différences de formes et de connotations finissent par devenir des facteurs limitants. Les modèles créés pour résoudre un problème ne pouvaient traiter que ce type de données particulières, malgré leurs éventuelles similitudes. En termes de tâche, la généralisation est difficile en raison de l'absence d'une mesure évaluant la qualité d'une hiérarchie, ce qui nécessite un raffinement empirique à travers une série d'ajustements par essais et erreurs pour une application particulière.

De plus, la création d'un cadre pour opérer sur des hiérarchies présente des défis supplémentaires considérables en plus du problème de généralisation, à savoir : (i) le produit des hiérarchies est multiforme, ce qui signifie qu'elles ont des tailles, des composants et des interprétations différentes ; et (ii) les mêmes données pourraient créer plusieurs structures hiérarchiques en fonction des opérateurs hiérarchiques et des contraintes. Par conséquent, l'application des hiérarchies morphologiques dans un cadre d'apprentissage agnostique nécessite une stratégie pour surmonter les aspects déterministes, l'évaluation de la qualité et les aspects hétérogènes.

26. Clément Michaël, Kurtz Camille, and Wendling Laurent (2018). *Learning spatial relations and shapes for structural object description and scene recognition.*
27. Nguyen Tin T. et al. (2019). *Feature extraction and clustering analysis of highway congestion.*
28. Nandy Kaustav et al. (2011). *Supervised learning framework for screening nuclei in tissue sections.*
29. Zwettler Gerald and Backfrieder Werner (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.*
30. Meyer Fernand (2001). *Hierarchies of partitions and morphological segmentation.*

Énoncé de thèse

Cette thèse soutient qu'il est possible d'insérer directement les structures hiérarchiques dans un cadre d'apprentissage, et de bénéficier des informations intégrées pour créer un modèle généralisable pour les tâches visuelles qui est agnostique au média et à la tâche.

Objectifs et questions

L'objectif principal de cette thèse est de concevoir un cadre d'apprentissage qui peut fonctionner sur des données hiérarchiques et qui est agnostique au média et à la tâche. Ce faisant, il doit faire face aux défis de la généralisation et mettre en place une stratégie pour conformer la hiérarchie des informations à un cadre d'apprentissage. Par conséquent, l'étude d'investigation dans cette thèse vise à répondre à trois questions principales :

Question 1 : Comment les méthodes hiérarchiques modélisent-elles diverses informations médiatiques et quels sont les défis pratiques rencontrés lors de leur application à un cadre d'apprentissage ?

Dans l'étude hiérarchique, une compréhension critique est la façon dont les blocs de construction des médias se rapportent au niveau inférieur pour les regrouper dans des régions homogènes. Les données visuelles, telles que les images et les vidéos, sont des structures de données organisées, et des informations telles que la couleur, la distance spatiale ou la variance définissent l'homogénéité. Et bien que la définition des régions homogènes et leurs connotations soient particulières à chaque média, la stratégie de regroupement et leur stockage dans la structure hiérarchique suivent les mêmes règles.

Compte tenu de ces considérations, cette thèse étudie les hiérarchies dans le contexte multimédia et inspecte leurs forces et leurs limites. Elle propose également une revue systématique de la littérature sur « l'apprentissage des hiérarchies », qui s'interroge sur l'insertion des hiérarchies dans un cadre d'apprentissage. Elle évalue les avantages de l'information hiérarchique dans le processus d'apprentissage et l'amé-

lioration que l'apprentissage automatique peut apporter à la représentation hiérarchique.

Hypothèse 1

Les représentations hiérarchiques contiennent des informations précieuses intégrées dans leurs structures pour un cadre d'apprentissage générique, et ce cadre d'apprentissage pourrait aider au traitement de la structure.

Question 2 : Comment créer un cadre d'apprentissage indépendant des médias et des tâches ?

Répondre à cette question nécessite de définir une représentation appropriée, idéalement partagée entre la plupart des types de médias et dotée de la capacité de retenir l'information présentée dans le média d'origine. De plus, la définition de la tâche ne doit pas imposer d'hypothèses sur la source de données.

Les graphes sont des structures utilisées pour représenter des objets, et la principale préoccupation de la théorie des graphes est de savoir comment ces objets sont interconnectés. Ils peuvent représenter de nombreuses données et transporter des informations sur les objets dans leurs composants, y compris dans différents domaines, tels que numérique, textuel et logique. En ce sens, malgré leurs différences, les données multimédia partagent les mêmes règles une fois modélisées sous forme de graphes. De plus, une façon de représenter les données hiérarchiques consiste à utiliser des arbres hiérarchiques. Par conséquent, les graphes et les hiérarchies ont des formalismes qui se croisent avec la théorie des graphes, et des applications qui peuvent être facilement généralisées.

Compte tenu de ces considérations, cette thèse propose de prendre des représentations graphes pour modéliser le cadre d'apprentissage. Pour être explicite, la thèse ne présente pas d'application multimédia. Cependant, les formulations et considérations se concentrent sur les structures de graphes comme point commun entre les hiérarchies et la modélisation multimédia. En outre, la thèse suggère d'utiliser les forêts aléatoires [31], un modèle rapide, simple et évolutif capable de traiter des données de grande dimension et d'obtenir des résultats satisfaisants dans plusieurs tâches.

31. Breiman Leo (2001). *Random forests.*

Le principal défi de cette proposition concerne la représentation régulière requise par la plupart des algorithmes d'apprentissage automatique, y compris les forêts aléatoires. La représentation régulière est intrinsèquement opposée à la nature non contrainte des graphes. Ainsi, la stratégie proposée est de représenter les composantes du graphe comme des vecteurs d'attributs sélectionnés et d'évaluer sa capacité à retenir l'information modélisée dans les graphes tout en restant discriminant pour une tâche.

Hypothèse 2

L'utilisation d'une sélection d'attributs de graphe comme entrée dans le cadre d'apprentissage permet la formulation d'un modèle indépendant des médias, et la diffusion des informations au niveau des composants des graphes permet d'attribuer à chaque entrée une étiquette de tâche sans imposer d'hypothèses sur la source de données.

Question 3 : La structure hiérarchique pourrait-elle fournir des informations utiles dans un cadre d'apprentissage agnostique ?

Selon les choix de modélisation des graphes, elle peut créer un espace structuré particulier appelé graphes grilles proche du domaine spatial du média. Présumer une généralisation sur un graphe en grille peut être trompeur, de plus que les informations structurelles peuvent être nécessaires pour une représentation discriminative. Cependant, la modélisation des graphes à partir de la structure hiérarchique fournit une caractérisation non régulière des régions avec des notions d'ordre et de navigation.

Pour répondre à cette question, il faut tenir compte de l'arrangement sémantique au sein des hiérarchies, et toute proposition doit conserver les structures et les relations d'ordonnancement conformes aux principes hiérarchiques. Ainsi, comme il n'existe aucun moyen direct d'évaluer la qualité d'une hiérarchie, le modèle d'apprentissage doit faciliter la navigation entre les tâches pour évaluer divers aspects par l'expérimentation. En outre, le cadre doit s'appuyer sur autre chose que des stratégies pour préparer adéquatement les données pour une tâche spécifique ou affiner les structures pour une application.

Compte tenu de ces considérations, cette thèse propose d'utiliser les caractéristiques topologiques et régionales de la structure hiérarchique, transposées à une représentation ordonnée qui respecte leur disposition d'origine.

Hypothèse 3

La topologie des structures hiérarchiques seule pourrait être utilisée dans un cadre d'apprentissage pour résoudre plusieurs tâches si elle préserve leur arrangement sémantique.

Organisation de la thèse

Compte tenu de ces objectifs et de ces questions, cette thèse est organisée en trois parties principales, chacune abordant une question et corroborant une hypothèse. Spécifiquement :

Partie 1 : comprend les chapitres 1 et 2, évaluant la première question. Le chapitre 1 contextualise les hiérarchies morphologiques, présente les principaux facteurs faibles et forts des méthodes hiérarchiques, et décrit les différents types hiérarchiques utilisés dans la thèse. Il explique et formalise également les graphes et les hiérarchies sur la notation partagée décrivant leurs composants et leurs terminologies, suivi d'une discussion dans un cadre type délimitant le problème cible de cette thèse. Le chapitre 2 présente une revue systématique de la littérature sur « l'apprentissage des hiérarchies », qui est la première sur le thème à notre connaissance. La revue de littérature vise à rassembler les stratégies d'apprentissage appliquées aux structures hiérarchiques et à mettre en évidence les approches les plus prometteuses et pertinentes pour ce travail.

Partie 2 : comprend les chapitres 3 et 4, évaluant la deuxième question. Le chapitre 3 fournit quelques considérations sur les graphes considérés comme essentiels au développement de ce travail, et présente une revue de la littérature sur l'apprentissage automatique sur graphes, en explorant les motivations, la stratégie et les principaux problèmes. Il passe en revue l'apprentissage profond sur graphes pour en formuler la pertinence et identifier les limites, en se concentrant sur la perspective du traitement multimédia. Le chapitre 4 présente l'étude de cas d'un framework d'apprentissage opérant sur une sélection d'attributs de graphe, établissant le framework pour les structures hiérarchiques. Il évalue le problème de représentation régulière et contient des expériences d'investigation, des résultats et des analyses.

Partie 3 : comprend le chapitre 5, évaluant la troisième et dernière question. Le chapitre 5 présente l'aboutissement des propositions, élargissant les concepts et les stratégies aux données hiérarchiques. Il crée et délivre un cadre d'apprentissage opérant directement sur les données hiérarchiques, focalisant les formulations uniquement sur les composants structurels des hiérarchies.

Le chapitre **Conclusions par partie** (inclus dans ce résumé) présente une discussion considérant les différents aspects de l'investigation expérimentale, résume les propriétés observées et tire des conclusions pour guider les travaux futurs dans l'étude hiérarchique.

La figure ci-dessous présente un aperçu graphique de l'organisation du document original complet, indiquant l'association entre les sections principales et leur sujet.

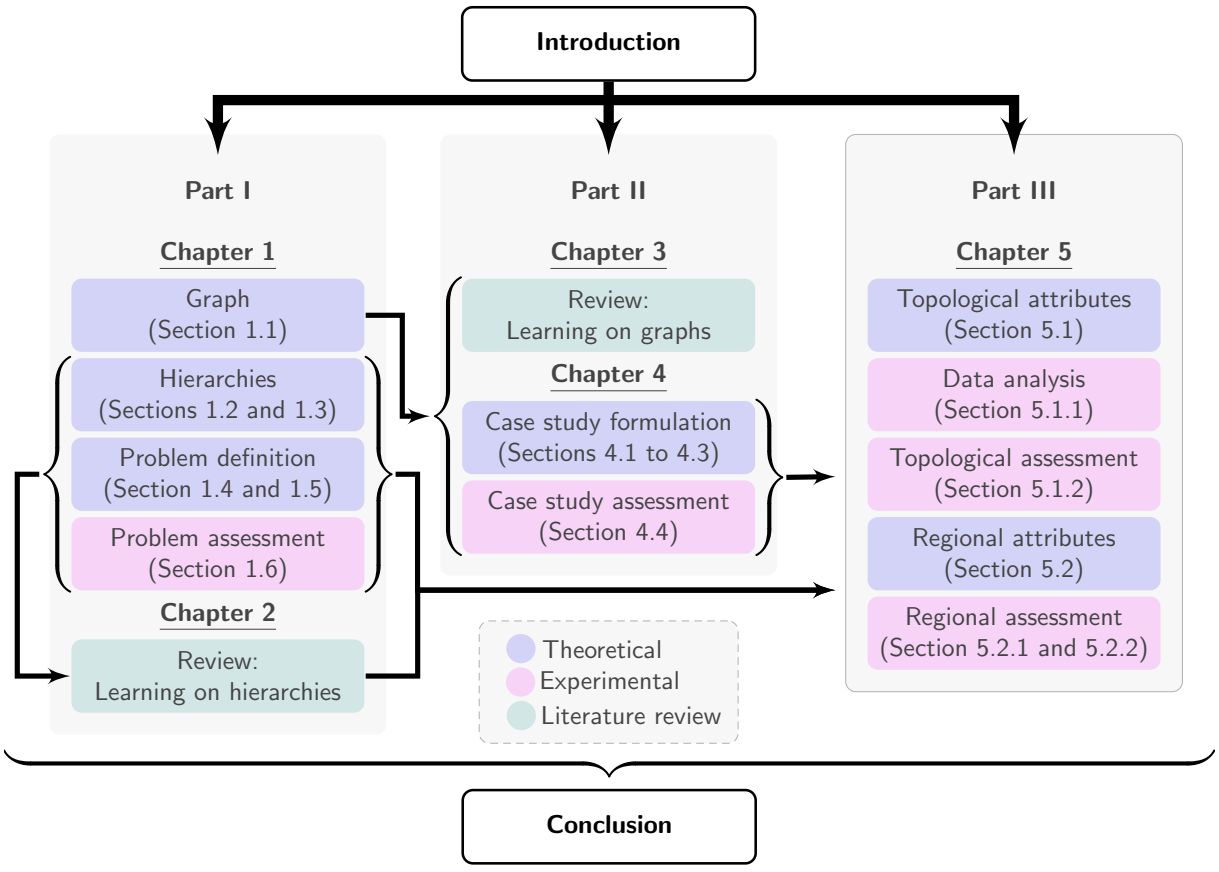


Figure présentant un aperçu graphique du document original complet regroupé par parties et affichant les principales sections. Les couleurs indiquent le thème (théorique, expérimental et revue de la littérature) et les flèches montrent la dépendance conceptuelle entre les chapitres et les sections. Toutes les sections d'une partie sont interdépendantes.

CONCLUSIONS PAR PARTIE

Énoncé de thèse

Cette thèse soutient qu'il est possible d'insérer directement les structures hiérarchiques dans un cadre d'apprentissage, et de bénéficier des informations intégrées pour créer un modèle généralisable pour les tâches visuelles qui est agnostique au média et à la tâche.

L'objectif principal de cette thèse était de concevoir un cadre d'apprentissage générique pouvant fonctionner sur des données hiérarchiques. Pour ce faire, il doit faire face aux défis de la généralisation dans les médias et les tâches, et mettre en place une stratégie pour conformer les données hiérarchiques à un cadre d'apprentissage. Il a fait valoir qu'il est possible d'insérer directement les structures hiérarchiques dans un cadre d'apprentissage bénéficiant de l'information embarquée.

C'est difficile parce que les structures hiérarchiques sont des représentations multiformes riches de données ordonnées et que les modèles d'apprentissage nécessitent généralement des structures systématiques pour fonctionner. De plus, aucune meusure directe ne peut indiquer la qualité d'une structure hiérarchique, et ce processus repose généralement sur des mesures de performance sur une tâche.

Cette thèse a présenté l'étude en trois parties, chacune évaluant un aspect crucial de la réalisation de l'objectif principal. La première partie a contextualisé et délimité le problème en théorie et en littérature. La seconde a étudié les capacités d'un modèle agnostique utilisant des graphes comme point d'ancrage entre les données multimédias et les hiérarchies. La dernière partie a rassemblé les informations pour proposer le cadre final utilisant la structure hiérarchique, insérée dans un cadre d'apprentissage qui s'appuie uniquement sur les informations hiérarchiques pour fonctionner. Les sections suivantes concluront chacune de ces parties en soulignant les points les plus significatifs liés aux questions et hypothèses initiales.

Conclusions Partie I : Données hiérarchiques

La première partie de cette thèse comprenait les chapitres 1 et 2 évaluant la question : **Comment les méthodes hiérarchiques modélisent-elles diverses informations médiatiques et quels sont les défis pratiques rencontrés lors de leur application à un cadre d'apprentissage ?** Avec l'hypothèse :

Hypothèse 1

Les représentations hiérarchiques contiennent des informations utiles intégrées dans leurs structures pour un cadre d'apprentissage générique, et le cadre d'apprentissage pourrait aider à analyser la structure.

Le chapitre 1 a contextualisé les hiérarchies telles qu'elles sont décrites dans la morphologie mathématique qui présente des formulations avec une base théorique solide, des implémentations efficaces et des principes directeurs d'ordre. Une structure pourrait être définie comme une hiérarchie si elle suit deux **principes hiérarchiques :** (i) le principe de causalité : un élément particulier à un niveau hiérarchique doit être présent à n'importe quel niveau consécutif ; et (ii) le principe de localité : les régions doivent être stables lors de la création ou de la suppression de partitions. Les formalismes sont généralement pour les données d'image, mais en général, ils caractérisent les régions et pourraient modéliser toutes les caractéristiques souhaitées fournissant des représentations ordonnées dans le domaine visuel.

Le chapitre 1 a également introduit la théorie des hiérarchies sous forme de graphes, formalisant les concepts de graphes et décrivant leurs composants et terminologies. En outre, il a décrit les différents types hiérarchiques envisagés dans la thèse, insérés dans un cadre typique présentant à travers des illustrations et des expériences les défis et les problèmes généralement rencontrés lors de l'application des structures dans une tâche. Il a montré que l'application nécessite une compréhension approfondie des médias, de la connotation de la région dans les hiérarchies et de la tâche.

De plus, l'analyse des structures doit prendre en compte de nombreux aspects cruciaux pour une bonne performance, ce qui est d'autant plus critique qu'elle s'appuie sur la tâche pour évaluer sa qualité. En utilisant l'approche triviale avec des coupes horizontales, la recherche d'une partition idéale pour une application peut être ardue

et négliger des détails essentiels présents dans les hiérarchies.

Le chapitre 2 a présenté une revue systématique de la littérature sur « l'apprentissage des hiérarchies », qui est la première sur le thème. La revue de littérature a rassemblé les stratégies qui combinent l'apprentissage automatique et les données hiérarchiques sur le même cadre. La recherche a récupéré 225 publications et, après filtrage par pertinence par rapport à la portée de ce travail, elle a passé en revue 64 méthodes regroupées selon la manière dont les informations hiérarchiques sont insérées dans le cadre d'apprentissage. À savoir, les méthodes qui : (i) appliquent les hiérarchies à un cadre d'apprentissage en évaluant comment la structure aide à la tâche et comment les auteurs formatent les informations hiérarchiques pour l'application ; et (ii) appliquent les stratégies d'apprentissage aux structures hiérarchiques en évaluant comment l'apprentissage contribue à améliorer la représentation.

Les techniques conçues à la main pour transformer les hiérarchies en une représentation plus adaptée aux tâches et certaines stratégies d'optimisation locale qui ne sont pas encadrées par l'apprentissage automatique et, par conséquent, non récupérées par les clés de recherche, ont également été inclus dans la revue. En plus, deux autres catégories ont été ajoutés pour regrouper les méthodes qui ne correspondaient pas au problème abordé dans cette thèse mais dont les finalités sont pertinentes dans le contexte. A savoir, des approches pour le bassin versant non-hiérarchique et des stratégies d'apprentissage inspirées des algorithmes de construction hiérarchique.

La revue de littérature a évalué : (i) les types de médias, comment les auteurs modèlent leur représentation tant sur la structure hiérarchique que sur la tâche ; (ii) les types de hiérarchies et quel rôle elles jouent dans le cadre d'apprentissage ; et (iii) les méthodes d'apprentissage automatique et les raisons de les choisir. Elle a révélé que les hiérarchies aidant les algorithmes d'apprentissage automatique à effectuer une tâche définissent des régions délimitant des zones pour l'extraction de caractéristiques ou représentent des masques appliqués sur les médias. Presque toutes les méthodes reposent sur des fonctionnalités multimédias pour l'étape d'apprentissage et nécessitent souvent de réduire la taille des représentations hiérarchiques, soit par filtrage, compression ou échantillons sélectionnés à la main. Les stratégies de cette catégorie nécessitent une compréhension complète de la manière dont les composants de bas niveau des médias interagissent dans l'espace et de leur relation avec la tâche. La plupart des applications sont la classification, la segmentation ou la détection. Les domaines sont nombreux, mais les plus dominants sont l'analyse aérienne et médicale

et le traitement d'images génériques. En ce qui concerne les modèles, la forêt aléatoire, les SVM et les réseaux de neurones sont souvent les modèles de choix pour leur robustesse et leurs capacités de généralisation.

Parmi les méthodes utilisant l'apprentissage automatique appliquées dans la structure hiérarchique, l'approche typique est la stratégie d'optimisation énergétique pour identifier les régions d'intérêt à l'intérieur de la structure hiérarchique. Une autre technique courante consiste à transférer la cible d'apprentissage vers une tâche parallèle qui induit une réponse sur les nœuds hiérarchiques. La plupart des méthodes présentent des solutions complexes ou une analyse combinatoire. L'apprentissage de la structure hiérarchique reste un sujet de recherche actif et ouvert.

La majorité des résultats extraits de la revue de littérature concernent le bassin versant non hiérarchique. Il est répandu parmi les applications médicales qui reposent sur des régions cohérentes et homogènes. Un problème répandu parmi toutes les méthodes utilisant le bassin versant classique est la sur-segmentation. De nombreuses stratégies reposent sur un prétraitement approfondi pour des applications réussies, tandis que d'autres proposent des techniques d'apprentissage pour fusionner certaines régions ou sélectionner des domaines d'intérêt.

Réponse 1 : les hiérarchies sont des structures riches qui peuvent modéliser une myriade de données. Elles facilitent l'analyse de problèmes complexes dans de multiples domaines. Cependant, elles nécessitent une attention particulière, et l'analyse des structures peut être difficile et peut limiter leurs applications. Il y a un grand intérêt dans la littérature sur l'intégration des hiérarchies et sur l'apprentissage automatique dans le même cadre. Pourtant, cet apprentissage s'appuie généralement sur des fonctionnalités multimédias et fournit des solutions difficiles à généraliser pour des tâches similaires. L'apprentissage de la structure hiérarchique reste un sujet de recherche actif et ouvert.

Conclusions Partie II : Apprentissage sur graphes

La deuxième partie de cette thèse comprenait les chapitres 3 et 4 évaluant la question : **Question 2 : Comment créer un cadre d'apprentissage agnostique aux médias et aux tâches ?** Avec l'hypothèse :

Hypothèse 2

L'utilisation d'une sélection d'attributs de graphe comme entrée dans le cadre d'apprentissage permet la formulation d'un modèle indépendant des médias, et la diffusion des informations au niveau des composants des graphes permet d'attribuer à chaque entrée une étiquette de tâche sans imposer d'hypothèses sur la source de données.

La définition de la représentation en graphe est une question de modélisation aux connotations diverses. Ils sont utilisés pour représenter des objets génériques, et la principale préoccupation de la théorie des graphes est de savoir comment ces objets sont interconnectés. Ils peuvent représenter de nombreuses données et transporter des informations sur les objets dans leurs composants, y compris de différents domaines. Toutes les délibérations de ce travail ont été centrées sur la théorie des graphes car elle pourrait fournir des outils de généralisation pour : (i) les structures hiérarchiques représentées dans un arbre ; (ii) les données multimédia ; et (iii) un cadre d'apprentissage indépendant des médias.

Le chapitre 3 a présenté une revue de la littérature sur l'apprentissage automatique sur graphes, en explorant les motivations, la stratégie et les problèmes fondamentaux. Il a concentré son analyse sur la perspective du traitement multimédia et a recueilli des informations pour plaider en faveur des choix de cadre proposés.

De cette perspective, le graphe représentant un média numérique avec des dimensions arbitraires, les sommets peuvent correspondre aux unités du média, telles que des pixels, des voxels ou des points de données. Cette approche se traduit généralement par de grands ensembles de sommets, mais favorise les opérations de va-et-vient. Alternativement, les sommets pourraient correspondre à des objets déduits des données, tels que des superpixels, des partitions et des surfaces, créant une représentation plus concise mais nécessitant des mappages complexes dépendant de la stratégie de regroupement.

Les méthodes de plongement des graphes conviennent à la création d'une représentation systématique qui permettent leurs utilisations dans plusieurs cadres d'apprentissage. Mais les plongements sont très coûteux en termes de ressources de calcul et sont prohibitifs pour les grands graphes. Les méthodes d'apprentissage en pro-

fondeur sur les graphes sont une solution contemporaine à de nombreuses tâches, principalement l'analyse sémantique et de haut niveau. Malgré leurs améliorations dans l'inférence d'informations, ils imposent des limitations concernant le graphe sous-jacent et les choix de modélisation. En particulier lors de la modélisation de données multimédias. Une approche générale lors de la gestion de l'apprentissage en profondeur sur des graphes dans un contexte multimédia consiste à modéliser les concepts et les abstractions plutôt que les données multimédias brutes. La forêt aléatoire en graphes fournit des solutions à de nombreux problèmes de calcul, en particulier pour les applications médicales et l'analyse des réseaux sociaux. La limitation la plus importante dans l'agrégation des graphes et des forêts aléatoires est l'entrée systématique requise par le modèle. Une analyse minutieuse du graphe doit avoir lieu, en tenant compte du type de graphe, de sa proximité avec les données d'origine et des résultats attendus.

Le chapitre 4 a présenté l'étude de cas d'un cadre d'apprentissage fonctionnant sur une sélection d'attributs de graphe agrégés avec le modèle des forêts aléatoires. Au-delà d'une bonne performance dans une application, la motivation reposait sur une proposition d'un framework d'apprentissage automatique travaillant sur des graphes qui pourraient être exploités ultérieurement pour les structures hiérarchiques. L'étude a proposé d'utiliser des graphes pondérés par les arêtes agissant comme un filtre de transformation basé sur les différences locales dans les images. La forêt aléatoire fonctionne comme un processus de régularisation pour atténuer certains bruits et renforcer les caractéristiques souhaitables. De plus, l'étude de cas décrit les graphes au niveau du sommet, ce qui permet d'entraîner le modèle sur l'espace discret en associant chaque entrée à une seule étiquette.

Traiter des graphes créés à partir d'images dispose d'un espace de modélisation unique. Du point de vue du framework, tous les attributs ne sont que des ensembles de valeurs stockées sur les sommets et les arêtes du graphe. Mais conceptuellement, le graphe d'images crée un espace transformé unique proche du domaine spatial des images, renforcé par des aspects relationnels sur les arêtes du graphe. Consciente de cet espace unique, la thèse a investigué les facteurs relationnels des graphes tout en s'interrogeant sur les conditions particulières de traitement des images. Elle a aussi évalué l'impact des caractéristiques sur les résultats obtenus sur deux tâches différentes et traitait d'aspects non généralisables.

Une évaluation de la qualité des choix de topologie a abordé les considérations sur

le type de graphe et sa proximité avec le media d'origine. L'étude expérimentale sur la sélection d'attributs a établi que la représentation de régions plus grandes à travers la taille voisine et le nombre de connexions avec la relation de contiguïté se traduisait par des valeurs de confiance plus élevées sur les bords de l'objet et moins de bruit dans les images résultantes. De plus, la fonction de pondération doit caractériser les similitudes dans les données d'origine pour être descriptive. La relation d'adjacence et la fonction de pondération sont des choix de modélisation, conditionnant l'interaction entre les données et le graphe. La régularisation de la forêt aléatoire atténue la plupart des mauvais choix de topologie, sauf lorsque l'entrée est extrêmement bruyante et corrélée. L'évaluation finale de la sélection d'attributs a concernée les attributs de sommet représentant des descripteurs de bas niveau de l'image. L'inclusion des attributs de vertex dans la représentation régulière fait une référence directe au média mais entraîne moins de bruit, des bordures plus fortes et plus de détails sur les gradients d'image finaux, donc cruciaux pour les applications pratiques évaluées.

Le principal défi du cadre concernait la représentation régulière requise par la plupart des algorithmes d'apprentissage automatique, qui s'oppose par nature à la nature non contrainte des graphes. En outre, le cadre a également pris en compte l'espace de grande dimension généralement présenté avec des graphes représentant les médias numériques et la stratégie d'attribution d'étiquettes qui ne devrait pas imposer d'hypothèses sur la source de données pour généraliser ultérieurement à d'autres tâches. La mécanique la forêt aléatoire lui permet de travailler avec des données de grande dimension, ce qui en fait une méthode d'investigation rapide, simple et évolutive. En outre, la forêt aléatoire associée au graphe agit comme un régulateur diminuant le bruit, accentuant les connexions fortes et atténuant tout éventuel mauvais choix de topologie. De plus, les mappages des prédictions de la forêt aléatoire dans à l'espace image sous la forme de gradients d'image permet l'évaluation qualitative et quantitative des résultats.

Les images obtenues en mappage des prédictions de la forêt aléatoire, créé sur les étiquettes de détection de bords de l'objet, et recevant la représentation régulière des attributs sélectionnés des graphes pondérés par les arêtes en entrée, présentaient les caractéristiques des gradients d'image, appelées dans la thèse « graph image gradient» (GIG). Les gradients créés par GIG sont généralement très descriptifs, avec des contours fermes des objets et d'autres aspects tels que des composants mineurs, des textures et de grandes régions uniformes. Les dégradés sont couramment utilisés

comme étape de prétraitement dans de nombreuses applications car ils sont rapides à calculer et facilitent généralement l'analyse d'image, en particulier pour la tâche de segmentation.

Par rapport à d'autres stratégies de gradient populaires, les gradients de GIG, en tant qu'entrées pour la méthode de segmentation des hiérarchies des bassins versants, ont produit des images mieux segmentées que les méthodes de gradient traditionnelles comme SED [32], Sobel et Laplace. La comparaison avec des cartes des bords de l'image plus élaborées, comme celles réalisées par les approches profondes HED [33] et RCF [34], a démontré que les performances de la tâche de segmentation dépendent des caractéristiques représentées sur le gradient. Dans l'ensemble, de meilleures segmentations résultent de gradients avec des contours épais et nets et des détails supplémentaires qui contribuent à l'identification de petits objets, et les informations sur les régions uniformes assurent la cohérence.

En ce qui concerne les performances de détection des bords (la tâche sur laquelle le modèle est enseigné), les résultats pourraient être meilleurs par rapport aux méthodes profondes ou SED. Cependant, l'observation des résultats a montré que les procédures les plus performantes sur la tâche sont celles avec des contours plus épais. Ceci est le résultat de la méthode d'évaluation proposée pour l'ensemble des données. Les représentations avec une marge plus importante sur les contours sont plus susceptibles de correspondre à les données réelles de référence. Parce que GIG est centré sur l'analyse des sommets, plus le la forêt aléatoire est confiant dans la distinction d'un sommet en tant que contour de ses sommets environnants, plus les prédictions sont précises, ce qui donne des contours plus fins. Néanmoins, les autres aspects représentés sur les gradients, tels que les grandes régions uniformes et les motifs simplifiés, pourraient être considérés comme un échec à la tâche, même s'ils sont bénéfiques pour d'autres applications et descriptifs des propriétés relayées par les graphes.

Réponse 2 : Les graphes sont des structures dynamiques pour la modélisation multimédia, mais comme les hiérarchies, ils nécessitent des considérations réfléchies lorsqu'ils sont appliqués dans un cadre d'apprentissage automatique. L'utilisation des informations disponibles sur les arêtes et les sommets du graphe est une méthode viable pour représenter un graphe dans un

32. Dollar Piotr and Zitnick C. Lawrence (2015). *Fast edge detection using structured forests.*
33. Xie Saining and Tu Zhuowen (2017). *Holistically-nested edge detection.*
34. Liu Yun et al. (2019). *Richer convolutional features for edge detection.*

cadre d'apprentissage. Elle permet de contrôler la taille de la représentation et de sélectionner les informations représentées en tenant compte du type de graphe, de sa proximité avec les données d'origine et des résultats attendus. De plus, la représentation des graphes au niveau des sommets permet de maintenir les analyses sur l'espace discret en attribuant un seul label à une entrée. Cette affectation est particulièrement avantageuse avec la stratégie des graphes car elle représente une région entière sur un seul sommet et la tâche ne fait aucune hypothèse sur le média.

Conclusions Partie III : Apprentissage des hiérarchies

La dernière partie de cette thèse comprend le chapitre 5, évaluant la troisième et dernière question : **Question 3 : La structure hiérarchique pourrait-elle fournir des informations utiles dans un cadre d'apprentissage agnostique ?** Avec l'hypothèse :

Hypothèse 3

La topologie des structures hiérarchiques seule pourrait être utilisée dans un cadre d'apprentissage pour résoudre plusieurs tâches si elle préserve leur arrangement sémantique.

Selon les choix de modélisation des graphes, ils peuvent créer un espace structuré particulier appelé graphes grilles proche du domaine spatial du média. Présumer une généralisation sur un graphe en grille peut être trompeur, de plus, les informations structurelles peuvent être nécessaires pour une représentation discriminative. Cependant, la modélisation des graphes à partir de la structure hiérarchique fournit une caractérisation non régulière des régions avec des notions d'ordre et de navigation.

Le chapitre 5 a présenté l'aboutissement des propositions, élargissant les concepts et les stratégies aux données hiérarchiques. Il a proposé deux stratégies pour représenter les structures hiérarchiques : (i) par les propriétés topologiques prenant les arbres hiérarchiques comme entrées ; et (ii) par des traits régionaux déduits de la topologie hiérarchique avec leur graphe conjoint. Les deux stratégies ont été formulés en utilisant uniquement les informations sur la structure hiérarchique et son graphe conjoint. Par conséquent, la représentation des médias se fait au gré de la modélisation graphe. De plus, l'attribution de l'étiquette de tâche est effectuée au niveau de

la feuille au bas de l'arbre, où chaque feuille a une étiquette discrète unique. Alors que l'affectation sur le graphe permettait de représenter une région entière sur un seul sommet, sur la hiérarchie, plusieurs régions partageant un chemin sur l'arbre et sont représentées dans une seule feuille.

L'approche topologique a proposé d'utiliser l'ensemble des parents d'un nœud feuille décrit par leurs attributs topologiques. La représentation des structures hiérarchiques par des ensembles de parents d'une feuille conserve les informations sémantiques intégrées dans les arbres hiérarchiques sans qu'il soit nécessaire de filtrer ou de sélectionner un niveau particulier pour l'évaluation. Consevoir la distribution des valeurs à l'aide de la représentation topologique a été bénéfique pour guider les décisions concernant l'étape d'apprentissage et pour mieux comprendre la structure hiérarchique.

La première évaluation de la topologie hiérarchique portait sur l'ordre de la représentation régulière sur les feuilles. L'ordre peut être ascendant (de la feuille à la racine) ou descendant (de la racine à la feuille). En raison de la multiformité dans les hiérarchies, différentes feuilles ont un nombre variable de parents sur la structure et peuvent ne pas être un alignement entre la position de l'entité dans la représentation régulière et la position du parent dans l'arbre hiérarchique. L'étude de cas dans GIG a indiqué que le la forêt aléatoire fonctionne mieux lorsqu'il existe une correspondance significative entre les caractéristiques et la position qu'elles prennent dans le vecteur de caractéristiques.

Par conséquent, pour clarifier la relation entre le modèle et l'ordre des caractéristiques sur les données d'apprentissage, une analyse a inspecté l'importance de la position des caractéristiques sur les nœuds de décision du modèle et a sondé les valeurs utilisées pour le chemin divisé dans l'arbre de la forêt aléatoire. Les expériences ont montré que l'ordre croissant fournissait des distributions plus cohérentes favorisées par le modèle.

Des applications avec l'approche topologique ont montré qu'elle contient des informations cruciales sur les hiérarchies, et non seulement améliorait les résultats obtenus par une approche typique avec les hiérarchies, mais fournissait également un aperçu de la distribution des valeurs lorsque toutes les informations hiérarchiques sont prises dans leur ensemble. L'analyse des données de cette distribution a montré que malgré les différences entre les types et les constructions individuelles pour les entrées, les valeurs agrégées présentent des caractéristiques similaires.

En ce qui concerne les choix topologiques représentés, les attributs altitude et superficie ont présenté les meilleurs résultats pour la tâche la plus liée à l'information

qu'ils dépeignent. Par example, les hiérarchies orientées contour donnent de meilleurs résultats dans la tâche de détection des contours avec l'attribut topologique altitude, et les hiérarchies orientées région pour la tâche de segmentation avec l'attribut topologique superficie.

La stratégie topologique a construit une représentation régulière qui pourrait être utilisée dans la plupart des modèles d'apprentissage disponibles. Cependant, les dimensions de cette représentation pourraient être difficiles en termes de ressources de calcul, en particulier compte tenu du fait que la plupart des positions des caractéristiques étaient remplies de valeurs de rembourrage. Les implémentations efficaces pour les structures hiérarchiques et la flexibilité du modèle la forêt aléatoire ont permis de travailler avec des structures importantes, mais l'utilisation de cette approche avec un modèle différent pourrait être difficile.

La deuxième stratégie proposait d'utiliser un ensemble d'attributs régionaux pour représenter la structure hiérarchique. Du point de vue procédural, cela équivaut à effectuer des coupes horizontales par niveaux d'altitude, mais plutôt que de créer une représentation pour chaque coupe et de les évaluer individuellement, la méthode proposée les représentait toutes systématiquement sous forme de représentation régulière. La deuxième stratégie visait également à préserver la représentation ordonnée, mais au lieu de représenter chaque niveau, elle présente la structure hiérarchique comme un ensemble d'attributs régionaux ordonnés sur la position de l'entité. Cette approche a abouti à une représentation plus compacte qui a capturé les informations critiques sur les hiérarchies et amélioré les résultats dans toutes les expériences.

Malheureusement, il n'y avait pas un seul attribut régional unique que l'on puisse désigner comme la meilleure sélection pour la stratégie régionale, comme les altitudes et la superficie dans l'approche topologique. Cependant, la gaussienne a présenté, en général, des résultats supérieurs sur les différentes tâches. Comme l'attribut gaussien quantifie la distribution des régions sur les arbres hiérarchiques, il assimile la représentation à la tâche. Par exemple, pour les hiérarchies orientées contours, c'est la meilleure pour la détection des contours, et pour les hiérarchies orientées région, la meilleure pour la segmentation. Les applications futures de cette stratégie peuvent considérer la tâche à accomplir pour sélectionner un type hiérarchique qui correspond le mieux aux objectifs et utiliser l'attribut gaussien pour la description.

Réponse 3 : Cette thèse a démontré qu'il est possible de créer un cadre d'apprentissage dépendant uniquement des données hiérarchiques qui fonctionne

bien dans plusieurs tâches avec différents modèles. Elle a créé et livré un cadre d'apprentissage opérant directement sur la structure hiérarchique, évitant toute fonctionnalité extraite du média et n'utilisant que les informations de l'arbre hiérarchique et du graphe. De plus, elle n'a pas sélectionné de région particulière mieux adaptée à une application, en préservant ainsi un aspect agnostique. Au lieu de cela, la hierarchie entière est représentée sous une forme vectorielle qui qui a retenu le l'arrangement sémantique de la structure hiérarchique. De plus, l'attribution d'étiquettes prend les étiquettes directement de la feuille représentent les principaux composants des hiérarchies, exemptant ainsi le framework de toute autre considération sur la tâche.

Perspectives et travaux futurs

Les applications et expérimentations développées dans cette thèse ont toutes été réalisées sur l'espace image, car cela permet une inspection visuelle aisée de la qualité du résultat. Cependant, toutes les propositions sont formulées sur les structures des graphes ou des hiérarchies ou les deux. De plus, la revue de la littérature a montré que les deux sujets sont largement utilisés pour modéliser de nombreux types de médias, en particulier les médias visuels.

Les propositions de thèse fournissent des solutions pour incorporer les structures dans un cadre d'apprentissage représentant la structure dans un format pris en charge par de nombreux modèles d'apprentissage automatique. En outre, ils suppriment le besoin d'un examen détaillé concernant la sélection d'un niveau ou d'une région appropriée et offrent un moyen d'expérimenter plusieurs types hiérarchiques sans modifier les considérations. De plus, l'attribution d'étiquettes au niveau des composants supprime les problèmes de tâche tels que la binarisation et la sélection de premier plan/arrière-plan pour l'évaluation.

Une application dans un autre type de média pourrait prendre les mêmes considérations de sélection d'attribut puisqu'une fois que le média est modélisé comme une hiérarchie ou un graphique, ils partageront tous les mêmes règles dans cet espace. Notamment les structures hiérarchiques qui intègrent en plus les notions d'ordre et de navigation dans leurs règles sous une forme non maillée.

Si la généralisation n'est pas une préoccupation dans les applications futures, on pourrait utiliser les stratégies proposées pour transposer la structure à l'espace vec-

toriel tout en prenant les mesures appropriées pour améliorer les résultats sur une tâche spécifique au média. Par exemple, on pourrait mieux modéliser les médias dans l'espace hiérarchique ou graphique en utilisant des techniques de prétraitement ou même une sélection de région (si le type de région d'intérêt est connu).

Une autre façon d'améliorer les performances des tâches à l'aide des stratégies proposées consiste à sélectionner un modèle d'apprentissage automatique différent. La forêt aléatoire a été choisi comme modèle dans la thèse, car elle est rapide, inspectable et évolutif. Elle a permis l'exécution de plusieurs expériences avec les données brutes des médias sans le coût en temps ni les décisions de conception favorisant l'évolutivité que d'autres modèles exigeraient.

Une direction possible pour les travaux futurs sur les stratégies déjà proposées est de combiner les attributs sélectionnés à partir des graphiques et des hiérarchies dans la même ou différentes catégories de caractéristiques, enrichissant les informations présentées au modèle d'apprentissage automatique. Une autre possibilité est d'appliquer une stratégie pour réduire la structure avant la sélection des attributs ou d'employer une stratégie de réduction des données une fois que les entités sont sous forme vectorielle. Cependant, il faut toujours garder à l'esprit l'arrangement sémantique et les règles hiérarchiques lors de l'utilisation de ces techniques.

Enfin, la thèse a démontré qu'il est possible de créer un cadre d'apprentissage fonctionnant avec des données hiérarchiques, performant dans plusieurs tâches et indépendant des médias. Les propositions fonctionnent avec n'importe quel type hiérarchique, y compris celles qui ne sont pas créées à partir de graphiques. Cependant, ils n'améliorent pas la structure hiérarchique et l'apprentissage d'une hypothétique hiérarchie idéale reste un problème ouvert.

Principales contributions

— Cadre d'apprentissage sur les attributs de graphes pour le traitement d'images (Publié à la 34e Conference on Graphics, Patterns and Images [35]. Récompensé comme meilleur article de la conference).

— Formalisme étendu sur les attributs de graphes explorant des zones d'entrée plus

35. Almeida Raquel, Patrocínio Jr. Zenilton K. G., et al. (2021). *Descriptive image gradient from edge-weighted image graph and random forests.*

étendues à travers des graphes de contiguïté de régions et des changements entraînés par la mécanique du modèle (publié dans Pattern Recognition Letters [36]).

— Cadre d'apprentissage opérant directement sur les données hiérarchiques, focalisant les formulations uniquement sur les composantes structurelles des hiérarchies (article soumis).

— Revue systématique critique de la littérature sur « l'apprentissage des hiérarchies », qui est la première sur le thème à notre connaissance.

Contribution à la connaissance :

La thèse a démontre qu'il est possible de créer un cadre d'apprentissage avec des données hiérarchiques qui fonctionne bien dans plusieurs tâches et est agnostique aux médias.

36. Almeida Raquel, Kijak Ewa, et al. (2022). *Graph-based image gradients aggregated with random forests.*