



Universidade Estadual de Campinas
Instituto de Computação



Rafael de Oliveira Werneck

Learning Graph-based Representations and Matching in Classification Tasks

Aprendizado de Representações e Correspondências
baseadas em Grafos para Tarefas de Classificação

CAMPINAS
2019

Rafael de Oliveira Werneck

**Learning Graph-based Representations and Matching in
Classification Tasks**

**Aprendizado de Representações e Correspondências baseadas em
Grafos para Tarefas de Classificação**

Tese apresentada ao Instituto de Computação
da Universidade Estadual de Campinas como
parte dos requisitos para a obtenção do título
de Doutor em Ciência da Computação.

Dissertation presented to the Institute of
Computing of the University of Campinas in
partial fulfillment of the requirements for the
degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Ricardo da Silva Torres

Este exemplar corresponde à versão final da
Tese defendida por Rafael de Oliveira
Werneck e orientada pelo Prof. Dr. Ricardo
da Silva Torres.

CAMPINAS
2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Werneck, Rafael de Oliveira, 1989-
W495L Learning graph-based representations and matching in classification tasks /
Rafael de Oliveira Werneck. – Campinas, SP : [s.n.], 2019.

Orientador: Ricardo da Silva Torres.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de
Computação.

1. Aprendizado de máquina. 2. Representação multimodal. 3.
Correspondência de grafos (Teoria dos grafos). 4. Aprendizado de custos. 5.
Classificação multi-classe. I. Torres, Ricardo da Silva, 1977-. II. Universidade
Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Aprendizado de representações e correspondências baseadas em grafos para tarefas de classificação

Palavras-chave em inglês:

Machine learning
Multimodal representation
Graph matching (Graph theory)
Cost learning
Multi-class classification

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Ricardo da Silva Torres [Orientador]

Luciano Rebouças de Oliveira

David Menotti Gomes

Alexandre Mello Ferreira

Fábio Luiz Usberti

Data de defesa: 31-05-2019

Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-8217-7250>

- Currículo Lattes do autor: <http://lattes.cnpq.br/4130404789443948>



Universidade Estadual de Campinas
Instituto de Computação



Rafael de Oliveira Werneck

Learning Graph-based Representations and Matching in Classification Tasks

Aprendizado de Representações e Correspondências baseadas em Grafos para Tarefas de Classificação

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres
Instituto de Computação - UNICAMP
- Prof. Dr. Luciano Rebouças de Oliveira
Universidade Federal da Bahia
- Prof. Dr. David Menotti Gomes
Universidade Federal do Paraná
- Dr. Alexandre Mello Ferreira
Instituto de Computação - UNICAMP
- Prof. Dr. Fábio Luiz Usberti
Instituto de Computação - UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 31 de maio de 2019

*We don't make mistakes,
just happy little accidents.*

(Bob Ross)

Acknowledgements

First and foremost, my gratitude to God, who blessed me with this opportunity to conclude one more stage in my life, and to give me strength to face every challenge arisen on the way.

I would like to thank my supervisor Professor Dr. Ricardo da Silva Torres for believing in my work; for his dedication, and for being a save haven for his students, being empathetic when we need.

I also thank Professor Dr. Antoine Tabbone, for collaborating and supervising me during my PhD internship in LORIA, Université de Lorraine. It was my first time abroad, and he helped me to adapt to a new environment.

I am very grateful to my parents, Edna Maria Fajardo de Oliveira Werneck and Carlos Henrique Vargas Werneck, who were with me all this time, supporting and cheering me when necessary. I am also grateful to my siblings, Gustavo and Carolina, and the rest of my family, who understood my absence in family gatherings.

I would like to thank my friends from RECOD, IC, Facebook, school, all of them. You helped me thorough a lot. Whether talking about work, amenities, or relaxing the mind. You guys rock!

I would like to thank CAPES (#1458623), CNPq (#141584/2016 – 5), and the São Paulo Research Foundation (FAPESP) for the grants #2016/18429–1 and #2017/16453–5, which made possible the development of this research.

This work was also supported by CNPq, São Paulo Research Foundation – FAPESP (grants #2014/12236-1, #2015/24494-8, #2016/50250-1, and #2017/20945-0) and the FAPESP-Microsoft Virtual Institute (grants #2013/50155-0 and #2014/50715-9). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Resumo

Muitas situações do mundo real podem ser modeladas por meio de objetos e seus relacionamentos, como, por exemplo, estradas conectando cidades em um mapa. Grafo é um conceito derivado da abstração dessas situações. Grafos são uma poderosa representação estrutural que codifica relações entre objetos e entre seus componentes em um único formalismo. Essa representação é tão poderosa que é aplicada em uma ampla gama de aplicações, de bioinformática a redes sociais. Dessa maneira, diversos problemas de reconhecimento de padrões são modelados para utilizar representações baseadas em grafos. Em problemas de classificação, os relacionamentos presentes entre objetos ou entre seus componentes são explorados para obter soluções efetivas e/ou eficientes.

Nesta tese, nós investigamos o uso de grafos em problemas de classificação. Nós propomos duas linhas de pesquisa na tese: 1) uma representação baseada em grafos associados a objetos multi-modais; e 2) uma abordagem baseada em aprendizado para identificar correspondências entre grafos.

Inicialmente, nós investigamos o uso do método Sacola de Grafos Visuais para representar regiões na classificação de imagens de sensoriamento remoto, considerando a distribuição espacial de pontos de interesse dentro da imagem. Quando é feita a combinação de representações de cores e textura, nós obtivemos resultados efetivos em duas bases de dados da literatura (Monte Santo e Campinas). Em segundo lugar, nós propomos duas novas extensões do método de Sacola de Grafos para a representação de objetos multi-modais. Ao utilizar essas abordagens, nós combinamos visões complementares de diferentes modalidades (por exemplo, descrições visuais e textuais). Nós validamos o uso dessas abordagens no problema de detecção de enchentes proposto pela iniciativa Media-Eval, obtendo 86,9% de acurácia nos 50 primeiros resultados retornados.

Nós abordamos o problema de correspondência de grafos ao propor um arcabouço original para aprender a função de custo no método de distância de edição de grafos. Nós também apresentamos algumas implementações utilizando métodos de reconhecimento em cenário aberto e medidas de redes complexas para caracterizar propriedades locais de grafos. Até onde sabemos, nós fomos os primeiros a tratar o processo de aprendizado de custo como um problema de reconhecimento em cenário aberto e os primeiros a explorar medidas de redes complexas em tais problemas. Nós obtivemos resultados efetivos, que são comparáveis a diversos métodos da literatura em problemas de classificação de grafos.

Abstract

Many real-world situations can be modeled through objects and their relationships, like the roads connecting cities in a map. Graph is a concept derived from the abstraction of these situations. Graphs are a powerful structural representation, which encodes relationship among objects and among their components into a single formalism. This representation is so powerful that it is applied to a wide range of applications, ranging from bioinformatics to social networks. Thus, several pattern recognition problems are modeled to use graph-based representations. In classification problems, the relationships among objects or among their components are exploited to achieve effective and/or efficient solutions.

In this thesis, we investigate the use of graphs in classification problems. Two research venues are followed: 1) proposal of graph-based multimodal object representations; and 2) proposal of learning-based approaches to support graph matching.

Firstly, we investigated the use of the recently proposed Bag-of-Visual-Graphs method in the representation of regions in a remote sensing classification problem, considering the spatial distribution of interest points within the image. When we combined color and texture representations, we obtained effective results in two datasets of the literature (Monte Santo and Campinas). Secondly, we proposed two new extensions of the Bag-of-Graphs method to the representation of multimodal objects. By using these approaches, we can combine complementary views of different modalities (e.g., visual and textual descriptions). We validated the use of these approaches in the flooding detection problem proposed by the MediaEval initiative, achieving 86.9% of accuracy at the Precision@50.

We addressed the graph matching problem by proposing an original framework to learn the cost function in a graph edit distance method. We also presented a couple of formulations using open-set recognition methods and complex network measurements to characterize local graph properties. To the best of our knowledge, we were the first to conduct the cost learning process as an open-set recognition problem and to exploit complex network measurements in such problems. We have achieved effective results, which are comparable to several baselines in graph classification problems.

List of Figures

1.1	Example of graph in the bioinformatics domain.	14
1.2	Graphical representation of the research areas covered in this thesis.	15
1.3	Examples of regions in remote sensing images and their graph-based descriptions.	16
1.4	Examples of multi-modality and cross-modality scenarios.	17
1.5	Examples of flood (left) and non-flood (right) images, with associated tags.	18
1.6	Representation of matching with a Graph Edit Distance.	19
2.1	Concept map of the Bag-of-Graphs model.	25
2.2	Illustration of the Bag-of-Graphs approach with the same steps of the BoW.	26
2.3	Representation of the reduction of the graph matching problem in a bipartite graph matching problem.	32
3.1	Overview the Chapter 3.	34
3.2	Visual representation of the steps of Bag of Visual Graphs.	36
3.3	Remote sensing images of the datasets selected in this chapter.	39
3.4	Statistical analysis of the experiments in the Table 3.5 using Student's t-test.	44
3.5	Statistical analysis of the experiments in the Table 3.6 using Student's t-test.	45
4.1	Overview the Chapter 4.	48
4.2	Bag of KNN Graphs.	51
4.3	Bag of Cluster Graphs.	52
5.1	Overview the Chapter 5.	58
5.2	Representation of the label assignment to a distance vector.	62
5.3	Proposed SVM approach to compute the edit cost matrix.	62
5.4	Classification of the training set for the Letter LOW dataset.	66
6.1	Overview the Chapter 6.	69
6.2	Schematic overview of the Graph Distance Learning framework.	72
6.3	Illustration of the creation of a distance vector based on node properties of four graphs.	73
6.4	Differences in the classification of the "X"-shaped test graph from the closed set (upper) and open-set (bottom) approaches.	77
6.5	Differences between OSNN1 and OSNN2 open-set recognition approaches when selecting training neighbors	78
6.6	Evaluation of the different weight learning strategies with regard to the use of normalization procedures and different training set sizes.	81

List of Tables

2.1	Caption	24
2.2	Examples of initiatives on graph-based multimodal representations.	28
2.3	Examples of hash-based initiatives on multimodal representations.	29
2.4	Examples of initiatives on multimodal representations using deep learning.	31
3.1	Summarization of the datasets.	38
3.2	Comparison of global descriptors in the Monte Santo dataset.	40
3.3	Parameter settings evaluation for the BoVG approach using BIC descriptors in the Monte Santo dataset.	42
3.4	Parameter settings evaluation for the BoVG approach using the best result from Table 3.3.	42
3.5	Experiments results for different descriptors in the Monte Santo dataset. .	43
3.6	Several experiments results for different descriptors in the Campinas dataset. .	43
4.1	Details on the Disaster Image Retrieval from Social Media dataset.	53
4.2	Average precision for the baselines in the validation.	54
4.3	Results of the concatenation of all visual features and our best modalities features as baseline.	54
4.4	Bag of KNN Graphs (BoKG) results.	55
4.5	Bag of Cluster Graphs (BoCG) results.	55
4.6	Results of the Relation Network deep approach.	55
5.1	Graph matching learning approaches.	59
5.2	Information about the datasets.	64
5.3	Accuracy results for HEOM distance and random population of the cost matrix in the graph matching problem (in %).	64
5.4	Mean accuracy and standard deviation (in %) for the HEOM distance and SVM multi-class approach in the graph matching problem. The best results for each dataset are show in bold.	65
5.5	Accuracy scores for four datasets (in %).	65
6.1	Details of the datasets used in these experiments.	79
6.2	Best results observed for the different weight learning strategies in terms of normalized accuracy. In all cases, 10 graphs are used for training.	82
6.3	Best results observed for the different weight learning strategies in terms of normalized accuracy, considering the use of complex network measurements in the characterization of graph local properties. In all cases, 10 graphs are used for training.	82
6.4	Comparison of our approach with the same evaluation protocol defined in [48] using the MAO dataset.	83

6.5 Mean runtimes of each iteration in the MAO dataset with the Leave-One-Out protocol.	84
---	----

Contents

1	Introduction	14
1.1	Motivation	15
1.1.1	Graph-based Image Representation	16
1.1.2	Graph-based Multimodal Representation	16
1.1.3	Learning Graph Matching	18
1.2	Objectives	18
1.3	Research Questions	19
1.4	Contributions	20
1.5	Thesis Organization	21
2	Background & Related Work	22
2.1	Bag of Words and Bag of Visual Words	22
2.1.1	BoVW Related Work	23
2.2	Bag of Graphs	24
2.3	Multimodal Representations	26
2.3.1	Graph-based Approaches	26
2.3.2	Hash-based Multimodal Representations	29
2.3.3	Deep Learning Multimodal Methods	30
2.4	Graph Matching	31
3	A Bag-of-Visual-Graphs Approach for Remote Sensing Images	33
3.1	Introduction	33
3.2	Bag of Visual Graphs in Remote Sensing Images	35
3.3	Material and Methods	38
3.3.1	Datasets	38
3.3.2	Experimental Protocol	39
3.3.3	Baselines	40
3.3.4	Bag of Visual Graphs	40
3.4	Experiments and Results	41
3.4.1	<i>What are the best parameter settings for the proposed method?</i> . . .	41
3.4.2	<i>Does the proposed Bag-of-Visual-Graphs approach yield better results than other methods in the literature?</i>	43
3.5	Conclusions	46
4	Graph-based Early-fusion for Flood Detection	47
4.1	Introduction	47
4.2	Graph-based Early-Fusion Methods	49
4.2.1	Bag of KNN Graphs	49
4.2.2	Bag of Cluster Graphs	50

4.3	Experiments and Results	52
4.3.1	Dataset	52
4.3.2	Features and Baselines	53
4.3.3	Evaluation	53
4.3.4	Results	53
4.4	Conclusions	55
5	Learning Cost Functions for Graph Matching	57
5.1	Introduction	57
5.2	Proposed Approach	60
5.2.1	Local Description	60
5.2.2	HEOM Distance	60
5.2.3	SVM-based Node Dissimilarity Learning	61
5.2.4	Graph Classification	63
5.3	Experimental Results	63
5.3.1	Datasets	63
5.3.2	Experimental Protocol	63
5.3.3	Results	64
5.4	Conclusions	67
6	Learning Cost Function for Graph Classification with Open-Set Methods	68
6.1	Introduction	68
6.2	Graph Distance Learning Framework	71
6.2.1	Local descriptor	72
6.2.2	Distance vector	72
6.2.3	Distance learning component	73
6.3	Graph Distance Learning Implementation	74
6.3.1	Local Descriptor	74
6.3.2	Distance vector	75
6.3.3	Distance Learning Component	76
6.4	Experiments	78
6.4.1	Datasets	78
6.4.2	Research Questions and Experimental Protocol	79
6.5	Results and Analysis	80
6.5.1	Q1: Impact of normalization and the size of training sets	80
6.5.2	Q2: Identification of the best learning methods	80
6.5.3	Q3: Comparison with state-of-the-art baselines	82
6.5.4	Computational complexity and runtimes	82
6.6	Conclusions	84
7	Conclusions	85
7.1	Summary of Contributions	85
7.2	Future Work	86
7.3	Research Outcomes	87
	Bibliography	89

Chapter 1

Introduction

We can model many real-world situations by means of *objects* and their *relationships*. Examples include friendship among people and roads connecting multiple cities. The concept of graphs is derived from the mathematical abstraction of these situations [15].

A **graph** $G = (\mathcal{V}, \mathcal{E})$ is a tuple composed of a set of vertices \mathcal{V} (object representation) and a set of edges \mathcal{E} , which encode relationships among pairs of vertices in \mathcal{V} [15, 115].

Graph is a structural method used to represent complex information among objects [77]. Due to its powerful representation, which allows the combination of objects' components and their relationships into a single formalism, they are used in a wide range of applications. Common applications include bioinformatics [65] (see Figure 1.1), chemistry [59], social networks [74], databases [82, 102], among others.

Given the widespread use of graphs in several domains and applications, several pattern recognition problems have been modeled in such a way that the use of graph-based representations is a key element for their solution [30, 105]. This is especially true for digital object classification problems, for which relations among objects or even among their components are explored in the definition of effective and/or efficient solutions. Classification is a process according to which an object is assigned a label based on a set of labeled training examples [73], i.e., given a set \mathcal{L} of possible labels of an object \mathcal{O} , a classification function f assigns a label in \mathcal{L} to \mathcal{O} : $f(\mathcal{O}) \rightarrow \mathcal{L}$.

Effective and efficient classification services depend on the use of suitable objects'

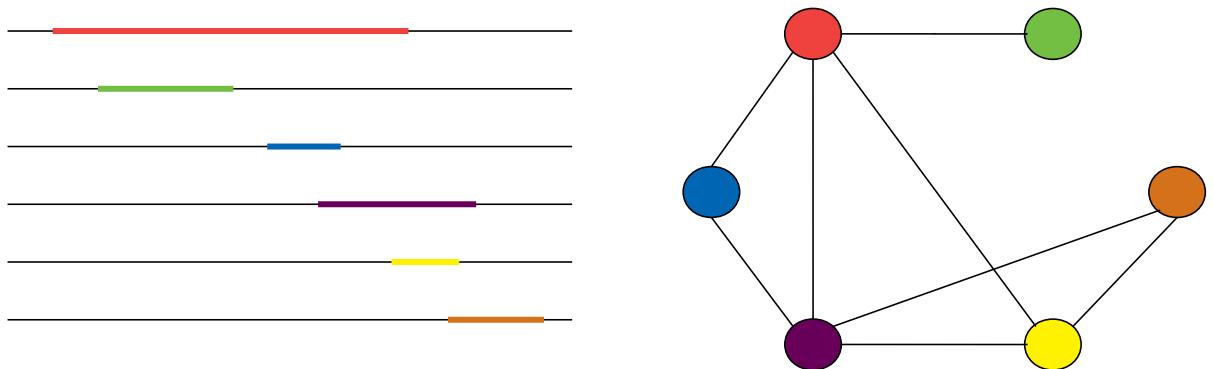


Figure 1.1: **Example of graph in bioinformatics.** In this example, a graph is constructed considering the overlap of intervals in linear genes [60].

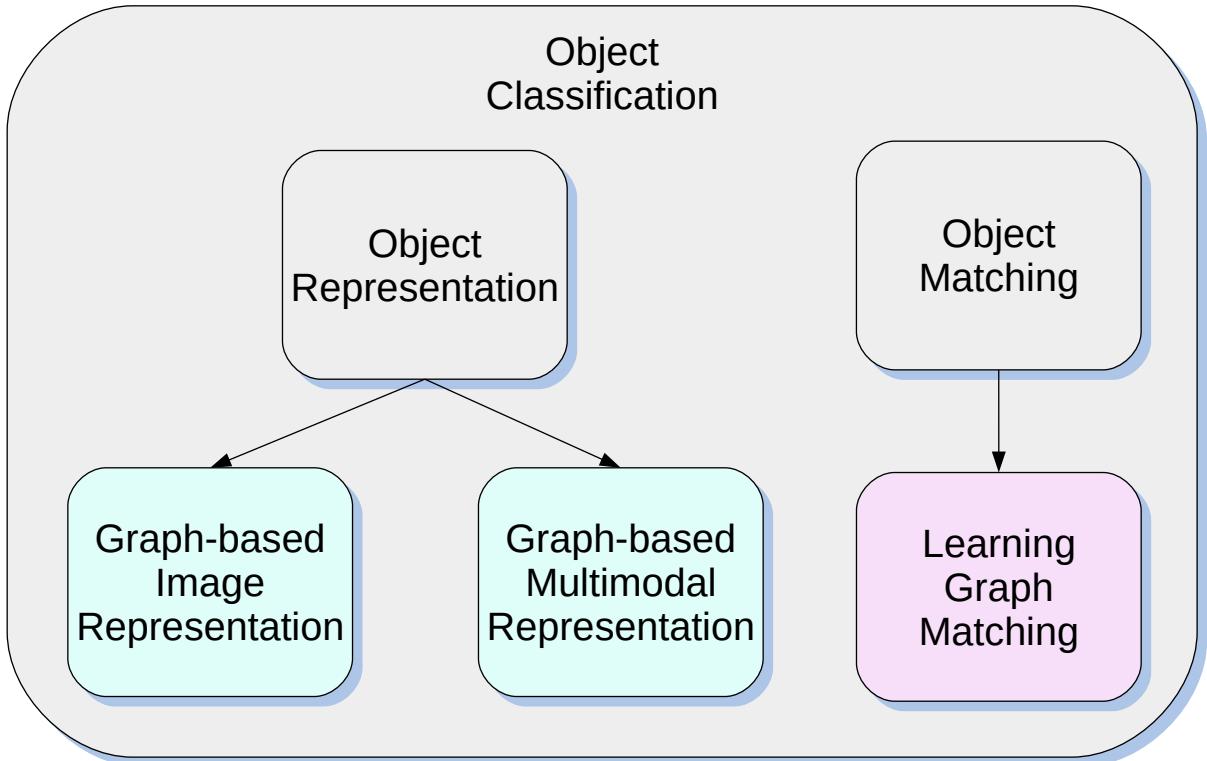


Figure 1.2: **Graphical representation of the research areas covered in this thesis.** We proposed in this thesis a spatial exploitation of visual properties to generate image representations, two new graph-based schemes to combine different object modalities, and a novel framework to include the use of learning schemes to support graph matching in object classification tasks.

representations and metrics to assess if representations extracted from objects are *similar* enough. In this thesis, we investigate the use of graphs for object representation and object matching in object classification problems. In object representation, two problems are considered: how to exploit the spatial distribution of visual properties encoded in graphs with the goal of generating effective image representations; and how to use a graph-based embedding schemes to combine complementary views provided by different object modalities (e.g., visual and textual properties). Furthermore, we address the object matching problem in classification tasks through a graph formulation. We particularly investigate learning schemes that could be used to graph matching, in special an open-set approach. The research venues followed in our work are summarized in Figure 1.2.

1.1 Motivation

In this section, we present motivational aspects of our work, discussing scenarios in which we can apply our graph-based studies.

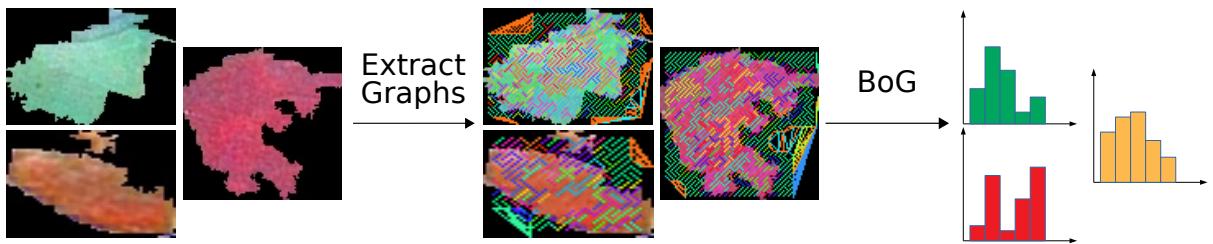


Figure 1.3: Examples of regions in remote sensing images and their graph-based descriptions. We can extract a graph-based representation of an image by first finding interest points in the image, and then connecting these points to create a graph. This graph represents the spatial distribution of these points in the remote sensing image. Later, graphs are transformed into vectors, using the Bag of Graphs (BoG) method [115].

1.1.1 Graph-based Image Representation

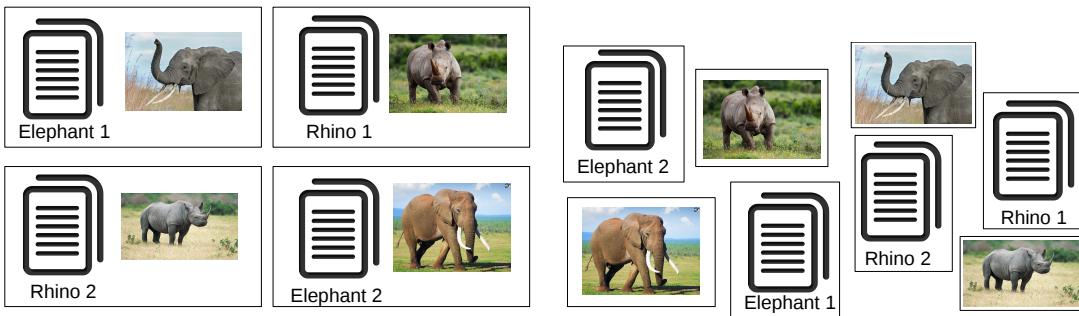
Remote Sensing Image (RSI) analysis plays an important role in several applications, ranging from agriculture and urban planning to sophisticated analysis of the impact of environmental conditions on ecosystems [78, 90]. One common usage of RSI refers to the support of the decision-making process, through the production of thematic maps about the regions of interest. The identification of regions of interest is often modeled as a classification problem based on pixel visual properties [34]. In fact, there are several works in the literature that describe remote sensing regions, usually considering color and texture descriptors [27, 39, 42, 151, 152]. On the other hand, few research initiatives [28, 76] have been considering the spatial relationship between regions in the RSI classification problem. We address this gap in this thesis.

Figure 1.3 illustrates this scenario. In the example, a graph-based model is used to encode the spatial relationship among objects. Later, a vector representation, referred to as Bag of Graphs (BoG) [115], is used to encode graphs. This graph model is expected to provide a richer representation of image regions.

1.1.2 Graph-based Multimodal Representation

Nowadays, due to the wide availability of media capturing devices, we can follow the rapid growth of multimedia content available in the World Wide Web. These devices, along with the adoption of online publishing platforms, turn casual users into media producers [99]. Users publish multimedia content related to events or discussions with which they are involved. In such cases, they can publish text, images, audio, video, and any combination of the previous data.

These multimedia objects represent a concept under different perspectives (e.g., text, images, or videos). These perspectives are the multi-modal components of the multimedia object. By exploiting the different descriptions provided by these modalities, we can have a better understanding of the context described by the object [150]. As we want to provide a representation combining different modalities, it is necessary to identify and characterize the relationship between these modalities. The challenge of this problem is to find a way to define and correlate these modalities and learn from this correlation [93, 128].



(a) Objects in a multi-modal scenario. (b) Objects in a cross-modality scenario.

Figure 1.4: Examples of multi-modality and cross-modality scenarios. In the multi-modality scenario, each multimedia object contains multiple modalities (e.g., a Wikipedia entry). However, in the cross-modality scenario, each object has only one modality. In this thesis, we will focus on the multi-modal scenario.

The literature presents two main approaches to explore the information of more than one modality in classification and retrieval tasks [143]: multi-modality or cross-modality. Xie et al. [143] define multi-modality as a natural extension of a unimodal approach, in which the objects are composed of more than one modality. This extension can be performed by combining the feature representations of each modality into a single feature vector [108], or by a late-fusion approach, combining their respective classification/retrieval results [68]. Otherwise, in the cross-modality approach, we have a set of modalities, but each object has only one modality. In this case, it is necessary to construct a correlation model to establish a connection between different modalities. The main challenge here relies on finding a way to address the semantic gap between modalities. A traditional solution for this problem relies on projecting the representation of different modalities into a common subspace, focusing on minimizing the distance between two semantic similar objects, and maximizing the distance between two semantic dissimilar objects [143]. Figure 1.4 presents examples of both scenarios.

This multi-modality content can be found and applied to several scenarios, such as classification and retrieval tasks. Examples of applications include tasks in remote sensing (in this case, we can consider each image band as different modality) [118, 35, 50], social media [89, 147, 103], pornography detection (considers both static and motion information) [97, 83, 3], among others.

One application in which we can take advantage of multiple modalities is the detection of natural disasters. We can obtain information about natural disasters from a wide range of sources, ranging from remote sensing images to news in social media [14]. In this scenario, authorities use information from these sources to propose strategies for damage control and victims' assistance. One of the focus of this thesis is on the detection of flooding events in social media content, by creating a graph-based joint representation for both visual and textual descriptions. Figure 1.5 shows two examples of social media content related to the problem of flooding detection.



Tags: "flood", "river", "thames"

Tags: "bridge", "holme bridge", "river nairn"

Figure 1.5: Examples of flood (left) and non-flood (right) images, with associated tags.

1.1.3 Learning Graph Matching

In pattern recognition, objects are often represented using two main approaches: statistical or structural [20]. In the former, objects are represented as points in an n -dimensional space; while in the latter, objects are represented through data structures, which encode their components and relationships. The literature related to classification and retrieval tasks encompasses many more statistical representations [38].

When considering the structural representation of objects, one of the most used is *graph*. The graph representation is powerful, as it describes in a single formalism both the object components and their relations. However, as said, the literature on structural representation is limited when compared to the statistical representation, and yet graph comparison is a high-complexity problem.

Usually, the graph matching is performed using the Graph Edit Distance (GED), an error-tolerant paradigm, which considers the minimum number of operations necessary to transform one graph into the other [19]. Figure 1.6 shows an example of operations to transform a graph A into graph F .

These operations have a cost, and this cost is usually manually designed and domain dependent. However, little research has been made to design cost functions automatically. In this thesis, we investigate alternative learning schemes to support graph matching in classification tasks.

1.2 Objectives

In this thesis, we first propose an application of the Bag-of-Graphs approach [115] in the remote sensing classification scenario, in which we want to classify regions of an image instead of the whole image. Our selected formulation, named Bag of Visual Graphs (BoVG), considers the spatial distribution of interest points found within RSI regions. We validated the BoVG in two application scenarios related to the classification of coffee regions in Monte Santo de Minas (Brazil) and urban areas in Campinas (Brazil).

We also present in this thesis a proposal to create a statistical representation of a

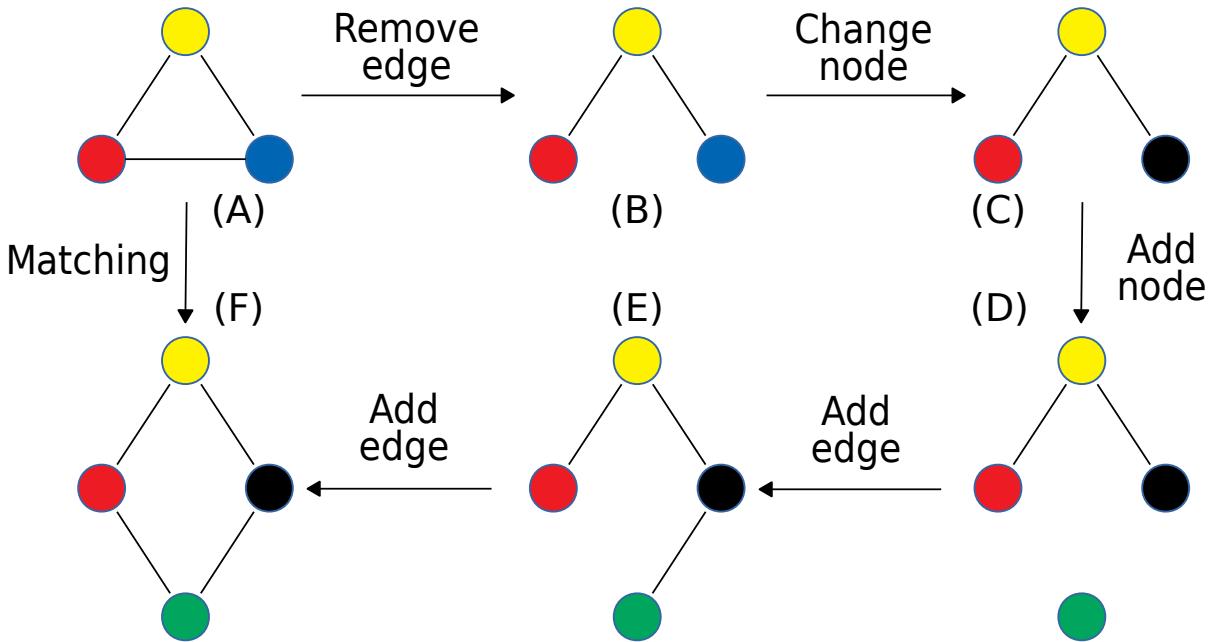


Figure 1.6: **Representation of matching with a Graph Edit Distance.** Graph A is expected to match Graph F . According to the Graph Edit Distance paradigm, operations to transform one graph into the other graph are determined. In this example, the distance between those two graphs are associated with the following operations: edge removal (A to B), change of node value (B to C), addition of a node (C to D), and addition of two new edges (D to E and E to F).

multimedia object through a bag-of-words approach, combining complementary views provided by multiple modalities. For this objective, we propose two new approaches to model the relationship between different modalities of a multimedia object using graphs. This graph representation of an object is the input of a bag-of-words framework model that generates the statistical representation of the multimedia object. We applied those two approaches in a challenge proposed by the MediaEval initiative in 2017, related to the detection of flooding events based on multimedia data posted on social media.

Lastly, for the graph matching part of this thesis, we address the difficult task of graph comparison. Graph comparison has high complexity, often an NP-hard problem [47, 149]. We address this task by proposing a novel framework to determine the cost functions in a graph edit distance method. We present two formulations for learning graph matching, a closed-set and an open-set formulation. We apply those two formulations in graph datasets, considering classification problems. In these problems, we first learn the edit distance costs between the test graph and training graphs. Once calculated the distance to all training graphs, we can classify the test graph according to the classes of the training set.

1.3 Research Questions

We guide the elaboration of this thesis by addressing some research questions. We approach each research question in the following chapters. The proposed research questions

are:

1. *Do Bag of Visual Graphs lead to an improvement in the accuracy of Remote Sensing Image classification problem? How can this representation take effectively advantage of the relations among regions encoded in their spatial distribution? How does this representation perform in this scenario?*
 - (a) *What are the best parameter settings for the proposed method?*
 - (b) *Does the proposed Bag-of-Visual-Graphs approach yield better results than other methods in the literature?*
2. *Would a combination of different features and/or modalities using a graph-based approach create a better representation of a multimedia object? How can we combine these different representations into a single one? How does our graph-based approaches perform in the flooding detection problem?*
 - (a) *Do our proposed approaches yield effective results for the flooding detection problem?*
 - (b) *How do our proposed approaches perform compared to a neural network, which infers relationships between objects?*
3. *Does the use of learning approaches improve graph matching results? Is it possible to learn intrinsic cost functions without a specialist? Does our approach yield effective results in the scenario of graph classification?*
 - (a) *How does our proposed approach compare with baselines from the literature?*
 - (b) *How can we improve our results avoiding misclassification of graphs from different classes?*
4. *Do open-set learning methods improve our proposal (3) for learning cost functions for graph classification? How does the open-set approach behave when compared to the state of the art?*
 - (a) *What is the impact of the training set size and normalization procedures in the effective performance of the evaluated learning methods?*
 - (b) *Which learning method leads to better effectiveness performance?*
 - (c) *How effective are the proposed methods when compared to state-of-the-art solutions?*

1.4 Contributions

This study provides a set of contributions to the domains of pattern recognition and graph classification. We can summarize them as follow:

1. Application of the Bag-of-Visual-Graphs approach to the scenario of remote sensing images, in order to describe both interest points and its spatial distribution (Chapter 3);
2. Two new approaches to create a joint representation of multiple modalities of multimedia objects, and their validation in the flood detection scenario (Chapter 4);
3. Original approach to learn cost functions to match nodes of two graphs (Chapter 5);
4. A generic framework to learn discriminative costs for a bipartite graph edit distance computation between two graphs, with two implementations on different classification paradigms (Chapter 6);
5. Investigation of complex network measurements on the characterization of graph local properties (Chapter 6).

1.5 Thesis Organization

This thesis is organized as follow: In Chapter 2, we present some backgrounds and a general related work, which embraces all research questions. Later, in each chapter, we present just the related work associated with the objective of the chapter. In Chapter 3, we present an application of the Bag of Visual Graphs in the scenario of Remote Sensing Images classification. In Chapter 4, we propose two new approaches for multimedia object representation considering multiple modalities using graphs, and we assess these approaches in the scenario of flooding detection. Chapter 5 introduces our first approach to the problem of learning cost function for graph matching. Chapter 6 goes beyond and proposes a framework for the problem of learning cost function for graph matching, in which we apply an open-set formulation for this problem. Chapter 7 presents our conclusions about the presented work and points out possible research venues for future work.

Chapter 2

Background & Related Work

This chapter discusses upon some backgrounds and related work of this thesis. Section 2.1 presents the Bag-of-Words and the Bag-of-Visual-Words approaches and some related work. Section 2.2 describes the Bag-of-Graphs approach and its formalization. Section 2.3 presents some related work on multimodal representation. Finally, Section 2.4 shows the relation between the Hungarian Algorithm and a bipartite graph matching problem. This section presents works which use graph-based encoding, hash-based methods, and approaches that exploit deep learning approaches.

2.1 Bag of Words and Bag of Visual Words

Several objects are not associated with a semantic meaning that easily identifies their content. To determine the similarity between two objects, a possible method is to identify similar patterns or local structures within them. Thus, representing objects by their local structures can lead to effective solutions in several tasks (e.g., classification or retrieval).

An effective approach to represent objects as aforementioned is based on bags, which describe the object by the frequency of occurrence of object features. The Bag-of-Words (BoW) [8] approach was designed originally to create a vectorial representation to describe documents based on the frequencies of word occurrences, being a simple and efficient form of representation to compute objects' similarities.

Later the BoW was adapted for the image context as the Bag of Visual Words (BoVW) [119]. The BoVW describes an image based on the global appearance of its local visual patterns, being more general than local descriptors and more discriminative than global descriptors. The procedures for computing a BoVW is the same used for BoW. These steps are discussed next.

Low-level feature extraction

Instead of words, the BoVW is based on the bags of low-level features, so the first step is to extract feature vectors from images. A common approach relies on detecting points of interest within the image and on extracting features that represent the region surrounding that point. The interest point detection can be applied either by a sparse sampling, in which interest points are detected in regions with difference of contrast, or by a dense

sampling, in which a dense grid divides the image, and every region of this grid has its features extracted. The sampled image is described by image descriptors, which characterize visual properties of the image, such as color (e.g., BIC [121]), texture (e.g., SASI [22, 23]), or shape (e.g., GIST [92]).

Feature space quantization

The BoW approach describes a document by a set of words, while the BoVW describes an image by visual words, by taking into account their occurrence. These visual words are encoded in a dictionary, induced by a quantization of the feature space. Each region of the quantized feature space is a visual word. The most popular method to quantize the feature space is the K-Means. However, it can suffer from the curse of dimensionality in high dimensional spaces. An alternative approach exploits random dictionaries, whose effectiveness performance is demonstrated to be no worse than K-Means [63].

Word assignment (coding)

After creating the dictionary, it is necessary to assign each image description to the quantized feature space to be possible to compare different images. This step, called coding, can be made by assigning an image local description to only one visual word (therefore called hard assignment), or by assigning the local descriptor to a set of regions in the quantized space, according to its activation of the region (procedure known as soft assignment).

Pooling

The pooling step is responsible for compiling the image local descriptions coded to the quantized space into a single feature vector. There are different operations for pooling, such as sum pooling, which sums the assignment for each visual word; average pooling, which calculates the average assignment value of each visual word; and max pooling, which considers only the maximum activation of a visual word.

2.1.1 BoVW Related Work

Perronnin et al. [98] proposed the application of Fisher kernels to image categorization, in which it combines the strengths of discriminative and generative approaches. The Fisher kernels characterize a signal with a gradient vector from probability density function, modeling the process of the signal, which can be used as input to a classifier. The main advantage of the Fisher kernel over the traditional BoW is that the gradient representation of the Fisher kernel has a higher dimensionality than histograms for the same vocabulary size.

Jégou et al. [64] proposed a method to aggregate local image descriptors into a compact vector, termed Vector of Locally Aggregated Descriptors (VLAD). VLAD accumulates, for each visual word, the difference between local descriptions that are the nearest neighbor of the visual word. Then the vector is normalized with a L_2 norm. The advantage of the

Table 2.1: Caption

Related Work	Year	Approach
Perronnin et al. [98]	2007	Fisher kernels with higher dimensionality
Jégou et al. [64]	2010	Compact vector which aggregates local image descriptors (VLAD)
Avila et al. [7]	2013	Vector considering the distribution of the descriptor around each visual word
Penatti et al. [95]	2014	Exploits spatial relationship, dividing the image space into quadrants, and counting the occurrence of visual words in each quadrant
Torii et al. [129]	2015	Modified weights of repeated visual words

VLAD method is that it add more discriminative power in the final feature vector than the traditional BoW and it is cheap to compute.

Avila et al. [7] introduced the BossaNova representation, based on a new pooling formalism, which keeps more information than the traditional BoW approach. In BossaNova, the distribution of the descriptors around each visual word is estimated by computing a histogram of distances between the local descriptor and the visual word. After computing the histograms for all visual words, the BossaNova approach concatenates them to form the image representation.

Penatti et al. [95] presented a spatial pooling approach, named Word Spatial Arrangement (WSA), which exploits the spatial relationship of visual words into the feature vector. Their approach generates feature vectors more compact than other methods that exploit spatial relationship. The WSA method divides the image space into quadrants, and at each interest point, counts the occurrence of each visual word in each quadrant. The feature vector is then created, in which each visual word has 4 dimensions, concatenated from top-right to bottom-right in counterclockwise direction.

Torii et al. [129] developed a representation for large-scale matching of repeated structures. They first detected repeated structures by finding groups of visual words with similar appearance. Then, they modified the weights of repeated visual words where multiple occurrences of repeated elements provide a natural soft-assignment. Also, the contribution of the repetitive structures is controlled to avoid dominating the matching scores. It demonstrated significant gains in recognition against the traditional Bag-of-Visual-Words approach.

Table 2.1 summarizes the aforementioned related work.

2.2 Bag of Graphs

The Bag of Graphs (BoG) is an extension of the Bag of Words (BoW) in the context of graphs, which uses a graph-based vocabulary to represent graphs as histograms [117]. For this approach, we present the following definitions and their relations that we will use on our thesis [115]:

We first define a **graph** as $G = (\mathcal{V}, \mathcal{E})$, in which \mathcal{V} is a set of vertices and \mathcal{E} is a set of

edges. Each edge $e = (v_i, v_j) \in \mathcal{E}$ represents a link between the vertices v_i and v_j of \mathcal{V} .

Next, the **vertex descriptor** is a tuple $d_V = (\epsilon_V, \delta_V)$, where $\epsilon_V : \mathcal{V} \rightarrow \mathcal{T}$ is a function that associates a vertex v of \mathcal{V} with an element of \mathcal{T} , called a vertex attribute, and $\delta_V : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ is a function that computes the similarity between the attributes of a pair of vertices. \mathcal{T} is defined as a set of vertex and edge attributes.

Similar to the vertex descriptor, an **edge descriptor** is a tuple $d_E = (\epsilon_E, \delta_E)$, where $\epsilon_E : \mathcal{E} \rightarrow \mathcal{T}$ is a function that associates an edge e of \mathcal{E} with an element of \mathcal{T} , called an edge attribute, and $\delta_E : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ is the similarity function between a pair of edges.

A **word** is an element $w \in \mathcal{T}$ that represents the prototype of a graph. A **codebook**, or dictionary, is a set of words representing different prototypes of the set of graph.

Coding is a vector containing the activation value for each graph in pair with each element from the codebook.

Given a coding C , **pooling** is a function that summarizes all element assignments, defined in a coding C , into a numerical vector.

Finally, **bag extraction** is a function, which associates a graph with a vector in \mathbb{R}^N .

We related each definition to each other as follow (Figure 2.1):

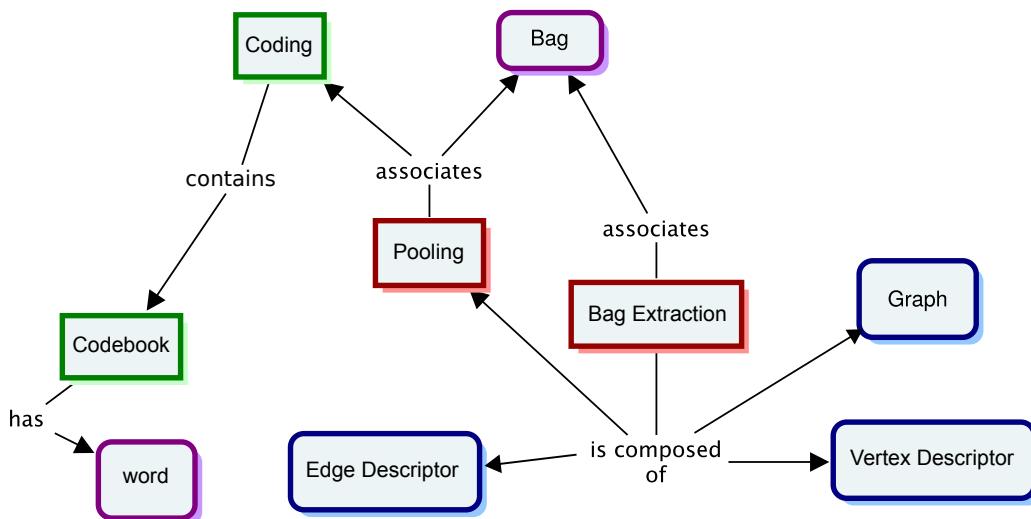


Figure 2.1: **Concept map of the Bag-of-Graphs model.** The colors of the squares represent the type of the concept: blue to the definition of tuples, red to functions, green to sets, while purple corresponds to specific representation elements.

In our Bag-of-Graphs approach, we describe each graph vertex by node signatures. One formulation for the node signature is:

$$NS(v_i) = AV_i; D; AE_{i1}, AE_{i2}, \dots, AE_{iD} \quad (2.1)$$

where AV_i is the attributes of the vertex v_i , D is the degree of the vertex, and AE_{ij} is the attributes of each edge linked to vertex v_i [117]. That way, each graph is then defined as a bag of node signatures of its vertices.

The steps of the BoG approach are the same of the Bag of Words, as shown in Figure 2.2. First, a node signature is associated with every vertex of the collection of graphs.

Then, a dictionary is created using a clustering method. Therefore, the histograms are created coding a set of graphs to the vocabulary, and then pooling them.

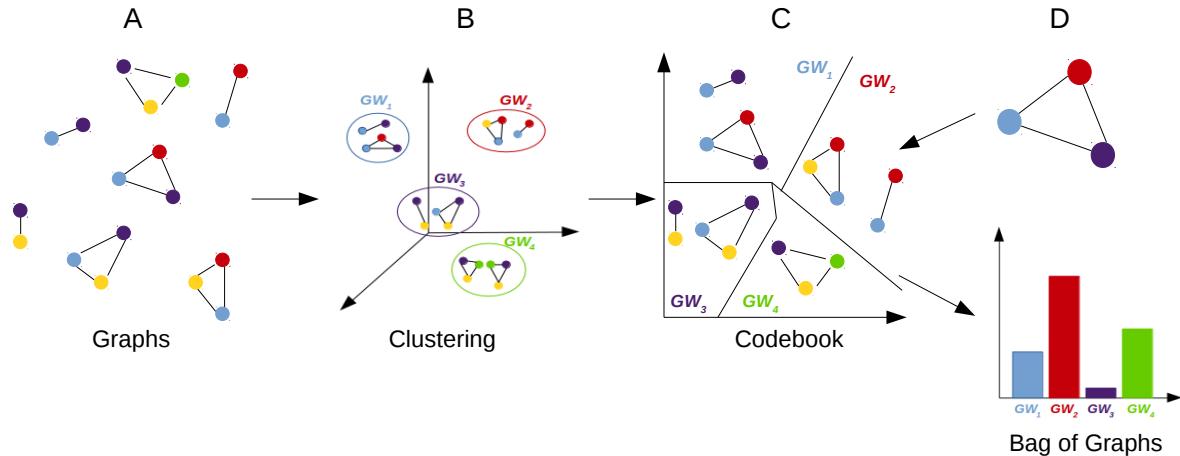


Figure 2.2: Illustration of the Bag-of-Graphs approach with the same steps of the BoW.

Silva et al. [117] use the *Heterogeneous Euclidean Overlap Metric* [61] as the dissimilarity measure between node signatures that are calculated in the clustering and coding step, because it handles numeric and symbolic attributes. The *Heterogeneous Euclidean Overlap Metric* for two node signatures ($NS(V_i)$ and $NS(V_j)$) is defined as follows:

$$d(NS(V_i), NS(V_j)) = \sqrt{\sum_{k=1}^N \sigma(A_{ik}, A_{jk})^2} \quad (2.2)$$

$$\sigma(A_{ik}, A_{jk}) = \begin{cases} \frac{|A_{ik} - A_{jk}|}{\text{range}(A_k)} & \text{if } A_k \text{ is numeric} \\ \tau(A_{ik}, A_{jk}) & \text{if } A_k \text{ is symbolic} \\ 1 & \text{if } A_{ik} \text{ or } A_{jk} \text{ is missing} \end{cases} \quad (2.3)$$

where $\text{range}(A_k)$ is the range of values of A_k .

The advantages of the BoG approach are: It is independent of the number of graph vertices, and it supports the computation of graph similarity taking advantage of widely used vector-based distance functions.

2.3 Multimodal Representations

We organize the related work on multimodal representation into three main families of methods: graph-based, hash-based, and deep-learning-based.

2.3.1 Graph-based Approaches

This section presents related work that uses a graph-based representation when dealing with different modalities.

Pan et al. [93] addressed the problem of auto-captioning using a cross-modal correlation discovery. They proposed a Mixed Media Graph (MMG) that represents the objects and their attributes as vertices of a graph. Each graph contains $m + 1$ layers, being m the number of attributes, and the left layer is a vertex representing the object. They found the correlation of an object with a determined caption through a random walk with restarts.

Tong et al. [128] studied a graph point of view to learn from multi-modal features. Each feature is represented as an individual graph, and the learning is made by inferring from the constraints in every graph. They proposed two different fusion schemes for semi-supervised learning, and indicated that this method can be easily extended to unsupervised learning. The first fusion scheme is a linear one, weighting the constraints of the optimization equation, and the second scheme is sequential, in which the optimization for each modality is made separately.

Wang et al. [133] presented the Optimized Multigraph-Based Semi-Supervised Learning (OMG-SSL), which integrates multiple graphs representing each modality, and graphs representing the temporal consistency in the object. Next, they performed a semi-supervised learning in the fused graph, which is equivalent to integrate multiple graphs to explore their complementarity.

Jia et al. [57] proposed a Markov Random Fields model named Multi-modal Document Random Field (MDRF) that are not restricted to problems with words describing visual object, or problems with full correspondence between modalities. The proposed model learns a set of topics across the different modalities, based on a similarity graph.

Zhai et al. [150] performed a cross-modality retrieval in the Wikipedia dataset. They proposed a novel cross-modality correlation propagation (CMCP) using a k -NN graph that considers both the positive and negative correlation between media objects. The learning process relies on the propagation of known labels. The propagation is made first to cover one modality, and then the other modalities.

Wang et al. [134] proposed a web image search re-ranking approach to explore multiple modalities in a graph-based learning, named Multimodal graph-based learning (MGL). This work integrates the learning of relevance scores, weights of modalities, distance metric, and scaling. A Gaussian function converts the distance between data into similarity. The proposed approach is based on the normalized Laplacian graph using k -nearest neighbor and squared loss.

Zhou et al. [153] proposed a generative latent variable model (LVM) to provide a compact representation of visual speech data. The model is generated from latent speaker variable (visual appearance) and latent utterance variable (variations caused by uttering) modeled by a path graph, and incorporating the structure information through a low-dimensional curve embedded with the graph. They learn this model to make accurate predictions within the low dimensional latent variable space.

Petkos et al. [99] presented a multimodal clustering method that applies a denominated Same Event (SE) model that predicts whether two items belong to the same cluster. The items are organized in a graph, where each image is a node, and a link between two nodes is determined by a positive prediction in the SE model, computed in terms of the nearest neighbors of an image. Finally, a community detection algorithm is applied, either as a batch or as a incremental community detection.

Table 2.2: Examples of initiatives on graph-based multimodal representations.

Related Work	Year	Modalities	Applications
Pan et al. [93]	2004	Images and captions	Automatic image captioning
Tong et al. [128]	2005	Plain and anchor text; Images (different features)	Classification and image retrieval
Wang et al. [133]	2009	Videos and images	Video annotation and person identification
Jia et al. [57]	2011	Text and images	Image retrieval using text queries
Zhai et al. [150]	2012	Text and images	Cross-modality retrieval
Wang et al. [134]	2012	Images (multiple features)	Re-ranking for web image search
Zhou et al. [153]	2014	Images and sound	Visual speech recognition
Petkos et al. [99]	2014	Images (multiple features)	Social event detection
Li et al. [72]	2017	Images and text	Detection and tracking of news topics
Zadeh et al. [148]	2018	Language, vision, and acoustic	Sentiment analysis and emotion recognition

Li et al. [72] presented a method for detecting and tracking news topics from multimodal TV data. They create an And-Or Graph which jointly combines images and texts in a hierarchical structure. This graph model balances the syntactic representation of the natural language processing and the simplistic BoW representation.

Zadeh et al. [148] proposed a novel multimodal fusion method denominated Dynamic Fusion Graph (DFG). DFG has some desired properties, such as it explicitly models n -modals interactions; it has an efficient number of parameters; and it can alter itself based on the importance of each n -modal. They also introduced the largest dataset on sentiment analysis and emotion recognition, on which they evaluate the proposed technique.

Table 2.2 summarizes the aforementioned related work.

While the initiatives [150, 128, 133, 134] create graphs based only on information of the same modality, our approaches seek to create a correlation link between objects from different modalities. Our approaches are similar to the methods discussed in [93, 57, 153, 99, 72, 148]. However, we do not rely on random walks as [93] or a Markov Random Field formulation as [57]. Instead, we exploit the bag-of-words (BoW) model to create a vector representation of the object. One of our proposed approaches is similar to [99, 72] regarding the use of clustering methods. In our case, however, clustering is used as a preprocessing step of the BoW model. The main advantage of using a BoW model is that this model generates a single vectorial representation of an object. When representing a complex object with a single vectorial representation, we can use the large quantity of methods available in the literature to different applications (e.g., retrieval and classification).

2.3.2 Hash-based Multimodal Representations

Bronstein et al. [16] approached the problem of cross-modality by means of embedding incommensurable data into a common metric space. They extended the similarity-sensitive hashing to multiple modalities (MMSSH). They showed that this learning can be solved using boosting techniques.

Song et al. [120] proposed an inter-media hashing model to achieve efficient multimedia retrieval. They discovered a Hamming space in which different types of data has inter-media and intra-media consistency. Their approach learned a set of hash functions for each data type. The inter-media consistency is achieved using the available tags within the objects so that similar semantics among objects are linked properly. The intra-media consistency is achieved by computing an affinity matrix for each data type using k -nearest neighbors.

Xie et al. [144] presented the Multi-graph Cross-modal Hashing (MGCMH), an unsupervised method that unifies multi-graph and hash function learning. They formulated a joint multi-graph framework that learns the weights of each modality, and learned a hash function that maps all modalities to a unified hash space. Xie et al. [144] created graphs of each modality such as [150, 128, 133, 134], but used a hash function to map all modalities into a unified space.

Li et al. [71] presented an approach to generate hash codes by ranking linear subspaces. They learn two groups of subspaces jointly, one for each modality, with the objective of aligning the rank ordering in one subspace with the other subspace. They also presented a probabilistic relaxation of the problem, so it can be flexible for different loss functions and be efficiently solved by using stochastic gradient descended algorithms.

Jin et al. [58] proposed a semantic neighbor graph hashing for the problem of nearest neighbor search. The hashing method aims to preserve fine-grained similarity based on the semantic graph, constructed by pursuing semantic supervision and local structure at the same time. Later, they defined a function based on the local similarity to encode intra-class and inter-class variation.

Table 2.3 summarizes the above research initiatives.

Table 2.3: Examples of hash-based initiatives on multimodal representations.

Related Work	Year	Modalities	Applications
Bronstein et al. [16]	2010	Shapes and medical images	Retrieval of non-rigid shapes and alignment of medical images
Song et al. [120]	2013	Images and web documents	Inter-media retrieval
Xie et al. [144]	2016	Images and text	Image or text retrieval
Li et al. [71]	2017	Images and text	Image or text retrieval
Jin et al. [58]	2018	Images and text	Nearest neighbor search

Our approaches does not use hash functions, however it also considers that the created graphs connect the different modalities into a hyper-space. Also, as our method generates a vectorial representations of the objects, we can also take advantage of hash-based solutions to speed up the processing time.

2.3.3 Deep Learning Multimodal Methods

This section discusses related work that handles deep-learning-based methods to establish correlations between multiple modalities.

Ngiam et al. [88] modeled a mid-level relationship between audio and video. Their work considered the learning step divided into three phases: feature learning, supervised training, and testing. They use three learning settings: multimodal fusion, cross-modality learning, and shared representation learning. The multimodal fusion method, in which data from all modalities are available for all phases, is applied to a bimodal deep belief network model to learn the correlation between modalities. The cross-modality learning, in which during the training and testing, only one modality is available, applies a deep autoencoder, that is trained to reconstruct both modalities with only one available and discover the correlation between the modalities. The shared representation learning, in which a modality present in training is different from the modality of the testing, also uses a deep autoencoder model.

Wu et al. [140] proposed a framework of online multimodal deep similarity learning, which learns a flexible nonlinear similarity function of multimodal features and learns the optimal combination of multiple modalities simultaneously. This work learns a multimodal distance metric from side information in form of triplets constraints, then a Exponential Gradient learning is used to learn the similarity functions of the triplet constraints.

Feng et al. [46] presented a correspondence autoencoder (Corr-AE) to solve the problem of cross-modal retrieval. Corr-AE learns the representation and the correlation between multi-modal objects into a single process.

Wei et al. [136] proposed a deep semantic method to solve cross-modal retrieval problem. They perform a deep network for each modality to learn to map its modality into a common semantic space. The retrieval is made by a deep semantic matching approach, in which a deep network with multiple no-linear transformations produces a probability distribution over classes.

Yang et al. [146] proposed a new model for multimodal fusion of temporal inputs, called CorrRNN. This model is based on an Encoder-Decoder framework that learns the joint representations of multimodal inputs by exploiting the correlation among modalities. They also introduced a dynamic weighting which allows the encoder to modify the contribution of each modality in the computation of the feature representation.

Shahroudy et al. [113] developed a novel deep learning framework for the recognition of human actions in videos using RGB and depth sequences inputs. Each layer of the network factorizes the multimodal input into a common modality-specific parts. They also proposed a structured sparsity-based classifier, which utilizes mixed norms to apply component and layer selection for a proper fusion of feature components.

Table 2.4 summarizes related work focusing on the use of deep-learning approaches.

The initiatives described in [88, 140, 46, 136, 146, 113] propose deep learning methods to find the correlation between the different modalities. Our approach in principle does not use deep-learning methods, yet it can be extended to determine the relationship between multiple modalities. We can aggregate deep-learning methods to our proposed approach either by describing the features of each modality, or by describing the correlation between

Table 2.4: Examples of initiatives on multimodal representations using deep learning.

Related Work	Year	Modalities	Applications
Ngiam et al. [88]	2011	Audio and video	Audio and/or video classification
Wu et al. [140]	2013	Global and local features	Image retrieval
Feng et al. [46]	2014	Images and text	Images or text retrieval
Wei et al. [136]	2016	Images and text	Images or text retrieval
Yang et al. [146]	2017	Video-sensor; Audio-video	Video/sensor activity classification; Audiovisual speech recognition
Shahroudy et al. [113]	2018	RGB and depth sequences	Human action recognition

the modalities.

2.4 Graph Matching

To perform the graph matching between two graphs, we choose the approach of reducing the problem into a problem of bipartite graph matching [62]. In the bipartite graph matching problem, we collect the nodes from each graph and create a complete bipartite graph with the nodes of the first graph in one side and the nodes of the second graph on the other, and then, we connect all nodes from one side to all the nodes in the other side. Figure 2.3 shows this problem reducing.

In the bipartite graph matching problem, we associate each node from each graph to its low-level description, and aim to obtain the costs to transform the nodes from one graph into the other, i.e. the costs of the edges between the nodes in the bipartite graph. With the costs to transform each node into the nodes of the other graph, we populate a cost matrix.

Then, we use the Hungarian algorithm in this matrix, to find to existing minimum cost path. Let $G = (Na, Nb, E)$ be this bipartite graph, with Na and Nb the nodes from the graphs A and B , respectively, and E the edges connecting these nodes, with the cost present in the matrix. Let $y : (Na \cup Nb) \rightarrow \mathbb{R}$ called potential if $y(i) + y(j) \leq e(i, j)$, for $i \in Na, j \in Nb, e(i, j) \in E$. The algorithm starts with a cover M empty, $R_{Na} \subseteq Na$ and $R_{Nb} \subseteq Nb$ not covered by M . Let Z be the set of nodes reachable from R_{Na} . If $R_{Nb} \cup Z$ is nonempty, the algorithm increases corresponding matching by 1. Otherwise, let $\Delta = \min\{e(i, j) - y(i) - y(j) \mid i \in Z \cap R_{Na}, j \in R_{Nb} \setminus Z\}$. The algorithm increases y by Δ on nodes of $Z \cap R_{Na}$ and decreases y by Δ on nodes of $Z \cap R_{Nb}$. This process is repeated until M is a perfect matching. Thus, with this matching, we can find the minimum cost assignment.

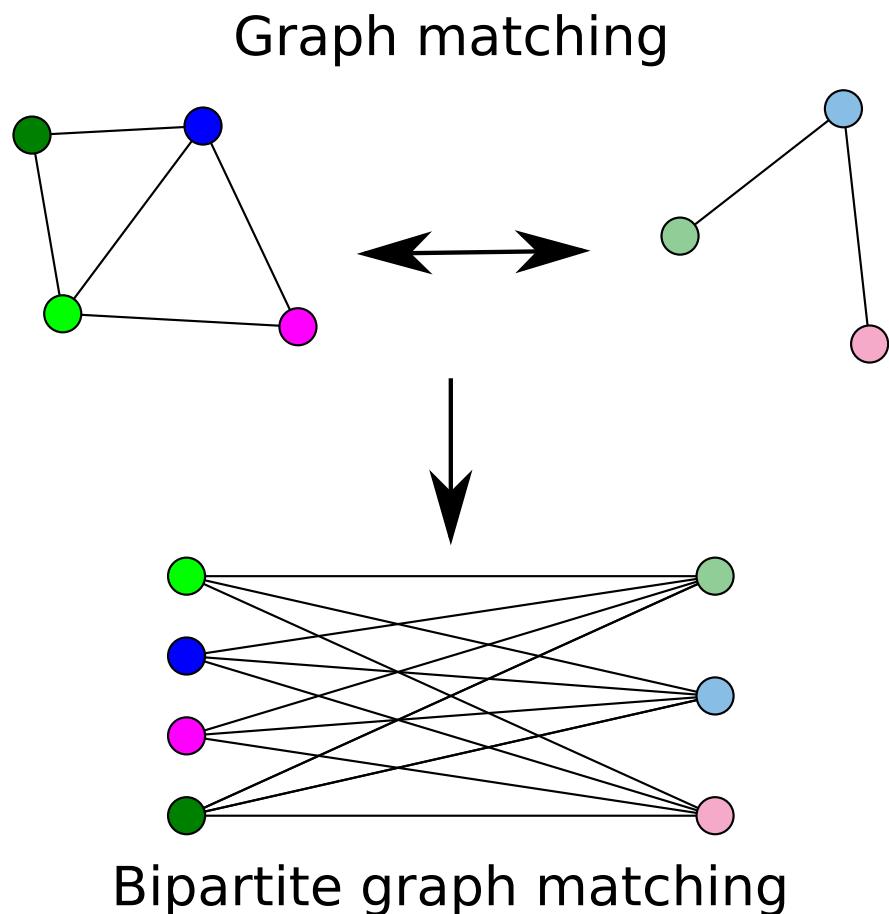


Figure 2.3: **Representation of the reduction of the graph matching problem in a bipartite graph matching problem.** In the reduction, the nodes from one graph are connected to all the nodes of the second graph, forming the bipartite graph. Therefore, we consider that the weight of the edge between two nodes in the bipartite graph is the cost to transform one node into the other. Thus, we can populate a cost matrix between the nodes, and then apply the Hungarian Algorithm to find the matching with the minimum cost between the two graphs.

Chapter 3

A Bag-of-Visual-Graphs Approach for Remote Sensing Images

This chapter, denominated *A Bag-of-Visual-Graphs Approach for Remote Sensing Images* refers to the work published in the *Pattern Recognition* journal under the paper *Graph-based bag-of-words for classification*¹ [115]. This work represents our graph-based image representation approach, using a Bag-of-Words model to encode in graphs local structures of an object, denominated Bag of Graphs. In this chapter, we propose to use the Bag of Visual Graphs (BoVG) method to represent spatial relationships between interest points within images in tasks of RSI classification. We perform experiments with two datasets: Campinas and Monte Santo. Conducted experiments demonstrate that BoVG yield effective results when combining color and texture representations, being superior than the traditional Bag of Visual Words (BoVW). Figure 3.1 summarizes the main concepts handled in this chapter.

3.1 Introduction

A wide range of studies uses Remote Sensing Images (RSI), such as agriculture [145], disaster monitoring, and urban planning [12] just to cite a few. These images are typically used to provide information to support the decision-making process, as they aim to produce thematic maps from the classified regions in the images. With regard to characterizing RSI visual content, several methods are applied. Most of these methods consider color, texture, or shape properties [27, 39, 42, 151, 152], or they exploit information using multiple-scale regions [39, 40, 41]. However, they usually do not consider the spatial information present in the image, i.e., intrinsic spatial relationships among local properties. In this chapter, we propose to work with the Bag of Visual Graphs [116, 115] to encode spatial information within the images aiming to support RSI classification tasks.

There are, in the literature, several studies which characterize the content of RSI using traditional descriptors. Santos et al. [42] evaluated the effectiveness of seven textures and twelve color descriptors in RSI retrieval and classification tasks. Joint Auto-Correlogram,

¹Reprinted from Pattern Recognition, 74, Fernanda B. Silva, Rafael de O. Werneck, Siome Goldenstein, Salvatore Tabbone, and Ricardo da S. Torres, “Graph-based bag-of-words for classification”, 266-285, Copyright (2018), with permission from Elsevier.

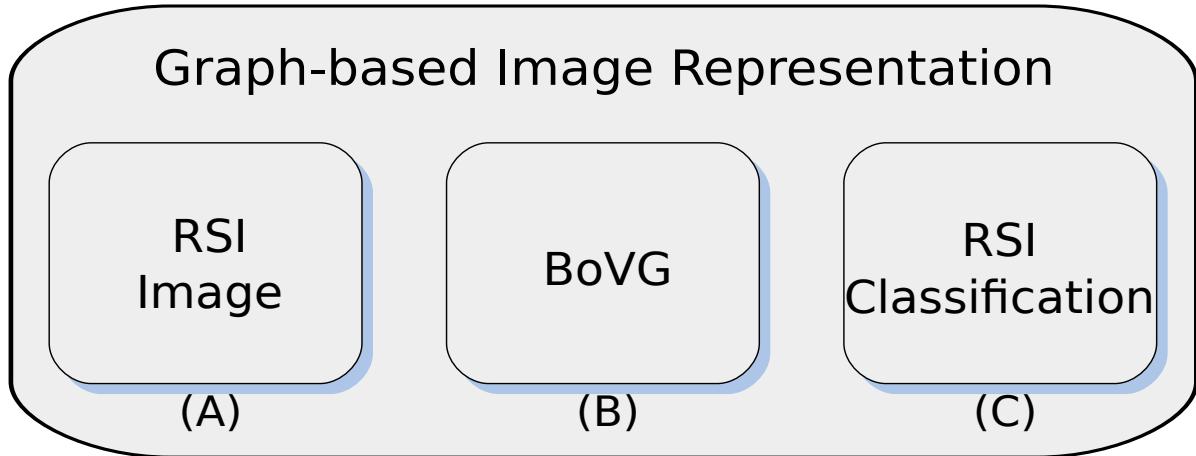


Figure 3.1: **Overview of the Chapter 3.** We have (A) the object of our work, which are RSI images; in (B), we have our proposed approach, the Bag of Visual Graphs (BoVG); for which we want to address the problem (C) of RSI classification based on the similarity of graph occurrence.

Color Bitmap, and Steerable Pyramid Decomposition were the descriptors with best results. Chen et al. [27], in turn, compared the performance of thirteen descriptors in the remote sensing scenario, describing either structure, texture, or color features. They selected the best descriptor from each approach and combined them, obtaining improved results for two classifiers: *k*-Nearest Neighborhood and Support Vector Machine (SVM).

Santos et al. [40] also proposed a propagation strategy, in which features from interest points from a finer scale are propagated to a coarse scale using the Bag-of-Visual-Words method. They also investigated the impact of assigning zero to pixels outside the regions. Their BOW-Propagation improved classification results when compared to global descriptors with the zero-padding. Santos et al. [39] studied how the use of multiple scales improves the classification of remote sensing images. They also studied how four color descriptors and three texture descriptors contribute in the characterization of regions from these scales. They showed that coarse scales have a great power of describing the image, as the finer scales improve the classification by detailing the segmentation.

Zhang et al. [151] also explored multiple features in the classification scenario. They introduced a framework to combine multiple features based on manifold learning and path alignment, generating a final low-dimensional feature with a meaningful combination. They show that their framework outperformed a method based on the selection of the best feature, feature concatenation methods, and different dimensional reduction approaches. Santos et al. [41] addressed the problem of the definition of a representation scale of the data. They proposed two approaches to exploit relationships between different scales, namely H-Propagation, which propagates the histograms of the regions in one scale to the corresponding region in the next scale, and BoW-Propagation, which uses the Bag-of-Words (BoW) model to propagate features between the scales. They showed that the BoW-Propagation with SIFT (Scale-Invariant Feature Transform) descriptor yields very promising results.

Our proposed approach not only characterizes RSIs considering their color or texture

properties, similarly to [42, 27, 151], but also defines a relationship between regions within the images. Our method differs from [41] as they do not consider the spatial relationship between regions of a scale in its propagation.

Only a few works in the literature consider the use of the spatial information of the image in its characterization. Plaza et al. [101] described two spatial/spectral data processing techniques, one improved from morphological techniques, that can use spatial and spectral information simultaneously; and Markov random fields that use a neuro-fuzzy classifier, whose output is fed to a spatial analysis stage. Zheng et al. [152] proposed a local feature named local self-similarity (LSS) that integrates geometric information, using self-similarities of color, edges, patterns, and textures, in a bag-of-visual-words approach. This process achieved good performance in the classification of areas for Quickbird satellite images.

Sun et al. [123] developed a detection framework based on a spatial sparse coding bag-of-words (SSCBOW). The spatial mapping, which is invariant to rotation, maps all parts of a target into a polar coordinate system, which is used to detect the regions of interest. They avoided reconstruction errors by using sparse coding. The results of the SSCBOW were better in both precision and recall than the traditional BoW. Fauvel et al. [45] proposed a kernel-based formulation to join the spatial and spectral information provided by a remote sensing image. They defined the spatial neighborhood as a connected set of pixels, resulting in a self-complementary area filter. Extracted features were fed to an SVM classifier in the dual formulation, being used in a classification task using the One-Vs-All approach.

Our approach differs from [123, 45] as they only represent the spatial relationship between the same visual word or pixels with the same neighborhood. Our method also differs from [101] as they do not consider spatial relationship between its pixels and regions, only spatial information, such as size, orientation, and local contrast. The method of Zheng et al. [152] differs from our method as they describe the spatial relationship within a region, while our approach describes the relationship among regions.

The Bag of Visual Graphs was proposed by Silva et al. [116] as an extension of the Bag of Visual Words that encodes spatial relationships of visual words as graphs. First, a traditional Bag of Visual Words is applied in the image. Next, edges are defined between the interest points of the previous step using the Delaunay triangulation [114] to generate connected graphs. Another BoW approach is now applied to these graphs, resulting in a final histogram that encodes the presence of these graphs in the image. This approach obtained a high accuracy score in two traditional datasets, Caltech-101 and Caltech-256.

3.2 Bag of Visual Graphs in Remote Sensing Images

The Bag of Visual Graphs (BoVG) is an extension of the Bag of Visual Words (BoVW) using a graph-based approach to include spatial information to the final descriptor, improving its discriminative power. The BoVG approach combines the spatial information of interest points with their labels defined by the traditional visual-word codebook to build graphs, which are used in the Bag of Visual Graphs.

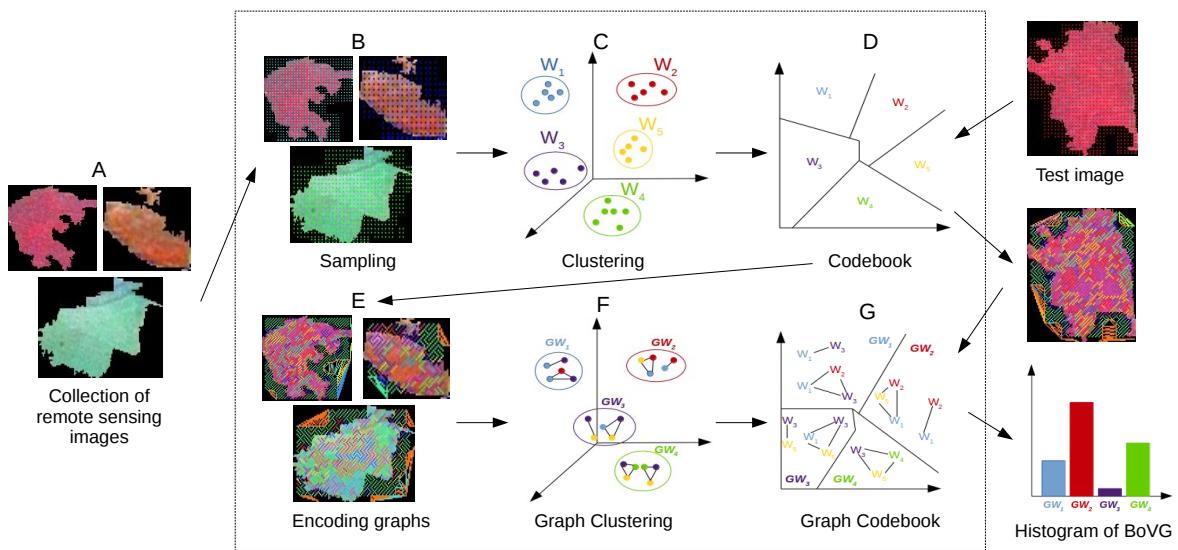


Figure 3.2: Visual representation of the steps of Bag of Visual Graphs. From a set of images from a collection (A), interest points are sampled (B) and the features of these points are clustered in the feature space (C). These clusters define the visual-word codebook used in the Bag-of-Visual-Words approach (D). Next, a set of connected graphs are created with the information of the codebook, and a Delaunay triangulation on its points to represent the image (E). Another clustering step (F) selects new words for the next codebook (G), represented as visual graphs. To generate the final descriptor using the Bag-of-Visual-Graphs approach, an image uses this graph codebook to generate a histogram, which encodes the frequency of the visual graphs present within the image.

Figure 3.2 shows the steps to generate the feature vector using the Bag-of-Visual-Graphs approach in the remote sensing scenario. These steps are described in the following.

First, it is necessary to define a remote sensing image (A) in terms of local features. A common approach relies on the definition of a set of interest points. These points (B) can be obtained through the use of interest point detectors, or through a dense sampling of the image, which is selecting a grid of interest points. Then these points are described by a local descriptor, such as Scale-Invariant Feature Transform (SIFT) [75] and Border/Interior pixel Classification (BIC) [121].

Next, a quantization of the feature space is performed to create a visual codebook, clustering (C) these interest points into codewords that will be used to describe the image. This codebook (D) can be created using a clustering method or a random selection of features. After this, each image from the collection is described using the codebook. This is done by assigning each interest point of the image to a region in the quantized feature space, selecting the closest region (as the hard assignment) or a set of regions (soft assignment). The next step relies on aggregating these points in the feature space to a unique vector representing the image using a pooling method, such as Max Pooling, which computes the maximum of each assignment for each region, and Sum Pooling, which computes the sums the assignment of each region.

These steps are the same used in the Bag-of-Visual-Words approach. The Bag of Visual Graphs extends this approach by creating connected graphs based on the interest points (E). The method we use to define the edges between the interest points was proposed by Hashimoto and Cesar [53], that also uses the Delaunay Triangulation to define edges. Edges are pruned according to their weights defined in terms of the distance among points, as lower weights encode close points and higher weights describe non-local features. The prune strategy can also be defined by the user.

Next, the graph-based codebook is generated in the same process made for the visual codebook, except for the fact that this time the features will be represented by connected graphs. Again, a clustering method or a random selection can be used to select the codewords of the codebook.

For the clustering method (F), it is necessary to define a graph matching function to calculate the distance between two graphs. Being a graph G_I defined by its vertices V and edges E as $G_I = (V, E)$, the distance function of two graphs G_1 and G_2 is defined as in [116]:

$$D(G_1, G_2) = \frac{\bar{C}}{|C|} + ||G_1| - |G_2||,$$

where $|G_i|$ is the number of vertices in G_i , \bar{C} is the cost of the optimum graph matching, which is computed on C_1 and C_2 , two distance matrices where each element corresponds to the distance between a vertex in G_1 to a vertex in G_2 ; and $|C|$ is the number of matching operation of C . C_1 and C_2 matrices differ in how to compute the vertex signature distance, as in C_1 the sequence of edges attributes are considered counterclockwise direction of vertices, and in C_2 , clockwise. The edge signature is defined in this work as the Local Binary Patterns (LBP) [91] descriptor or the Border/Interior pixel Classification (BIC) [121] descriptor.

Table 3.1: Summarization of the datasets.

Dataset	Location	Year	Sattelite	Bands	Size (px)
Monte Santo	Monte Santo de Minas	2005	SPOT	IR-R-G	1000 × 1000
Campinas	Campinas	2003	Quickbird	R-G-B	9079 × 9486

In this work, we use the function using vertex signature described in [116]. The signature of a vertex $v_i \in V$ is

$$S(v_i) = \{l_i, \text{degree}(v_i), e_{i1}, e_{i2}\},$$

where l_i is the label of the vertex, $\text{degree}(v_i)$ is the vertex degree, and e_{ij} is the texture-based signature of the edges linked to the vertex v_i .

The distance between two vertices is computed as the overlap distance between these vertex labels, and the distance of two edges signatures is calculated with the normalized Manhattan distance, using the HEOM distance presented in Equation 2.2.

After the creation of the graph-based codebook (G), the assignment of the graphs to a codeword is made using the hard or soft method, also using the graph matching function. Then, each assignment is pooled to generate the final feature vector, either by computing the maximum of each assignment for each region, or computing the sum of the assignments of each region. The final feature vector represents the remote sensing image as a Bag of Visual Graphs, which contains the distribution of visual graphs in the image.

3.3 Material and Methods

This section describes the datasets used in this work, as well as the evaluation protocol. Section 3.3.1 describes two datasets, Monte Santo and Campinas, Section 3.3.2 presents the adopted protocol, Section 3.3.3 presents the baselines with which we compare our work, and Section 3.3.4 describes the parameters of the Bag of Visual Graphs.

3.3.1 Datasets

The first dataset used is a composition of scenes of Monte Santo de Minas county, Brazil. These images were obtained in 2005 by a SPOT sensor. The dataset consists of the *red*, *infrared*, and *green* bands, with a total size of 1000×1000 pixels. This area presents a coffee cultivation, and was divided into 3 region masks that comprehends the whole image.

The second dataset is an image of Campinas, Brazil. Taken by Quickbird satellite in 2003, this image is composed of the three visible bands (*red*, *green*, and *blue*). This image size is 9079×9486 pixels, with 0.62m of spatial resolution. The entire image is divided into seven masks of interest, labeled as: bare soil, building, forest, houses, mixed field, road and parking, and sugar cane. Table 3.1 shows a summarization of the datasets and Figure 3.3 shows the images of the datasets.

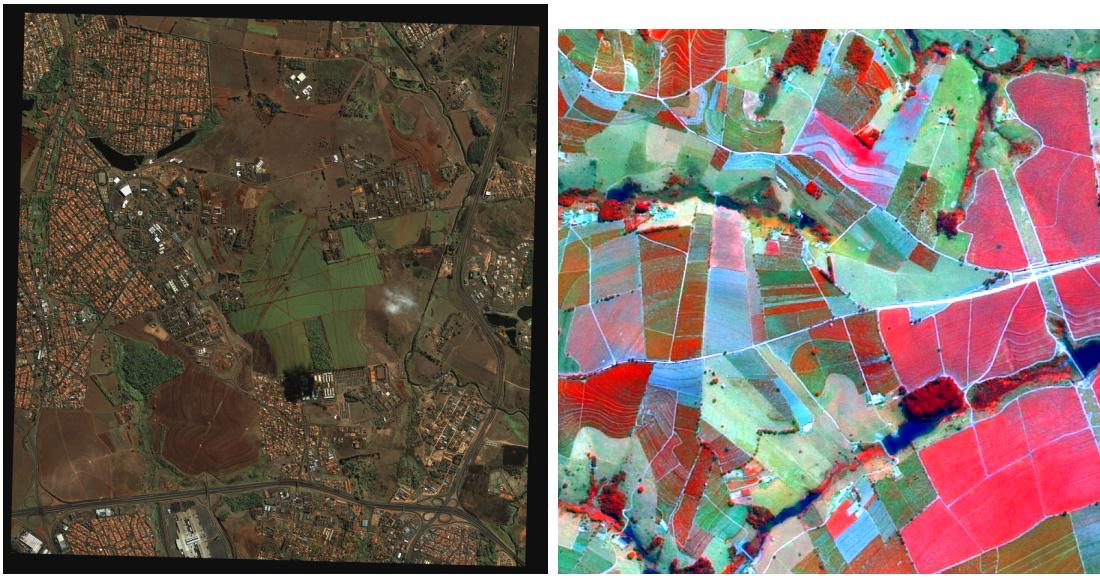


Figure 3.3: Remote sensing images of the datasets selected in this chapter.

In order to obtain the set of regions, we segmented the image and associated each segmented region with a label. We selected the Simple Linear Iterative Clustering (SLIC) [1] algorithm to group pixels into perceptually meaningful regions, because this method is fast and memory efficient.

The SLIC algorithm has only one parameter, which is the number of desired equally-sized superpixels. We used 300 superpixels for the Monte Santo dataset. To assign a label to each superpixel region, we considered the intersection of each region with the masks, and a region is assigned the label of the mask with more than 60% of pixels in it. After that, we cropped the superpixels into separated files, obtaining 203 region images. We applied the same protocol to the Campinas dataset, but using 900 superpixels. We selected this greater number of superpixels, because the unclassified mask of the Campinas dataset takes a larger region of the remote sensing image. Considering the 900 superpixels, we can crop the superpixels labeled with the classes of interest and obtain a total of 246 image regions to classify.

3.3.2 Experimental Protocol

We selected the stratified k -fold cross-validation protocol for our experiments. This protocol splits the dataset into k folds preserving the class proportion of objects of the dataset. We performed experiments with $k = 5$, in which we used one fold for testing, when training with the remaining four. We performed the classifications using a linear SVM from libSVM 3.17 with default parameters.

We present our results using the follow evaluation measures: global accuracy, which considers the fraction of correct predictions over all predictions; and balanced accuracy, which is the mean value of the accuracy for each class. For evaluation of our results, we present the agreement between the classification and the ground-truth with Cohen's Kappa, and a statistical analysis with Student's t-test and Wilcoxon test to confirm

Table 3.2: Comparison of global descriptors in the Monte Santo dataset.

Global Descriptor	Norm. Accuracy	Kappa
BIC	94.20%	0.8987
GCH	90.89%	0.8288
QCCH	89.16%	0.8255
Unser	56.37%	0.3758

that our approach yields results, which are significantly different from the ones of other methods.

3.3.3 Baselines

In order to compare our proposed feature extraction methodology with the literature, we chose four global descriptors, being two color and two texture descriptors, and the traditional Bag of Visual Words as baselines.

We have chosen the Border/Interior pixel Classification (BIC) [121], because it achieved a better overall effectiveness in RSI classification tasks [42] and in a web retrieval scenario [96], and Global Color Histogram (GCH) [124], it is a popular descriptor and constant baseline. The chosen texture descriptors are Quantized Compound Change Histogram (QCCH) [55], because of the simplicity of its extraction algorithm and compact feature vector, and Unser [130], which has a compact feature vector and lower complexity [96].

First, we performed experiments in the Monte Santo dataset to select the global descriptor baseline with the best performance. Table 3.2 shows our results in the normalized accuracy score and Kappa index. BIC achieved the best result in this dataset, so we picked it for our next experiments.

The BoVW approach has the following parameters: a dense-sampling of 16 pixels, with an overlapping of 50% of regions, described with BIC or SIFT descriptors; K-Means clustering to select 200 words to form the codebook; and hard assignment and sum pooling. These parameters were selected according to the literature [27].

3.3.4 Bag of Visual Graphs

For the Bag of Visual Graphs, we performed a study aiming to evaluate the best parameter settings. We used a dense sampling with a space of 6, 10, and 30 pixels with overlapping to the selection the interest points in the images, and described then using the BIC descriptor [121] and SIFT descriptor [75]. We performed experiments selecting 100, 200, 1000, and 10000 words to compose the codebook, selected randomly, as it archives a similar quality as the K-means clustering algorithm with a lower cost and avoiding the curse of dimensionality [63]. To assign each point of interest to a codeword, we used either the hard approach as the soft approach, being that last with a sigma parameter of 60. After that, we used two pooling methods, the max pooling and sum pooling.

The Bag of Visual Graphs also has the edge descriptor as an important parameter. We used the LBP and BIC descriptors, with BIC with two quantizations, with 128 bins

and with 64 bins. We selected these two descriptors to consider information both from colors and from texture. We made some modifications in the size of the Region of Interest (ROI) described by the BIC descriptor. We proposed two sizes of ROI: a smaller size, it is defined by the two interest points being its extremities; and a bigger size, which is the double of the smaller size. We also experimented pruning the graph edges, limiting the minimum distance for a graph edge to be four pixels, avoiding edges between closer points.

3.4 Experiments and Results

The experiments were performed, aiming to address three research questions:

- *What are the best parameter settings for the proposed method?*
- *Does the proposed Bag-of-Visual-Graphs approach yield better results than other methods from the literature?*

3.4.1 *What are the best parameter settings for the proposed method?*

To address this question, we used the BIC descriptor for both the point descriptor and the edge descriptor, and the hard-assignment procedure. These experiments were performed in the Monte Santo dataset. The parameters related to the density of the sampling, the size of codebook, and the pooling method were defined based on these experiments. Table 3.3 shows the results in the Monte Santo dataset.

In this parametric evaluation, we considered a dense sampling of overlapping regions with radius ranging in the set $\{6, 10, 30\}$. Experiments with radius equal to 30 led to the worst results because the largest regions do not emphasize the small local visual properties of the image, losing its discriminative power.

We also varied the codebook size in the range $\{100, 200, 1000, 10000\}$. All experiments had approximately the same accuracy, except for the experiment of dense sampling with radius equal to 30. In this case, almost all words of the codebook are used, which leads to a less enriched representation and to overfit the classification results for the second class.

We also performed experiments changing the pooling method between max pooling and average pooling. Observed results were very close to each other.

Next, we conducted the same study, but considering different point descriptors (BIC and SIFT), and edge descriptor (LBP, BIC with 128 bins, and BIC with 64 bins). We selected the best result with 6-radius dense sampling obtained in Table 3.3.

We can see in Table 3.4 that the best results were observed combining color and texture information of the image, both in the vertex and edge descriptors.

With the best parameters discovered, we can perform several experiments in our two datasets. The best results for the Monte Santo dataset are shown in Table 3.5. The results for the Campinas dataset, in turn, are shown in Table 3.6.

Table 3.5 shows that the BoVG approach, with either the color or texture descriptor, had a similar accuracy to the global BIC descriptor. Indeed, the results for all our BoVG

Table 3.3: Parameter settings evaluation for the BoVG approach using BIC descriptors in the Monte Santo dataset.

Dense Sampling	Codebook Size	Pooling	Norm. Accuracy	Kappa
6	100	Max	90.12%	0.8275
		Avg	90.16%	0.8400
	200	Max	91.40%	0.8510
		Avg	89.30%	0.8247
	1000	Max	93.54%	0.8956
		Avg	93.27%	0.8889
	10000	Max	93.00%	0.8966
		Avg	92.00%	0.8648
10	100	Max	86.95%	0.7924
		Avg	91.34%	0.8506
	200	Max	93.57%	0.8966
		Avg	93.18%	0.8958
	1000	Max	93.05%	0.8896
		Avg	93.00%	0.8886
	10000	Max	87.61%	0.8369
		Avg	77.82%	0.6731
30	100	Max	86.38%	0.7781
		Avg	85.60%	0.7613
	200	Max	88.48%	0.8143
		Avg	88.53%	0.8031
	1000	Max	80.91%	0.7355
		Avg	84.78%	0.7664
	10000	Max	36.30%	0.0503
		Avg	33.33%	0.0

Table 3.4: Parameter settings evaluation for the BoVG approach using the best result from Table 3.3.

Local descriptor	Edge descriptor	ROI Size	Norm. Accuracy	Kappa
BIC	BIC 128 bins	Smaller ROI	93.54%	0.8956
		Bigger ROI	93.59%	0.9023
	BIC 64 bins	Smaller ROI	92.99%	0.8883
		Bigger ROI	94.25%	0.9121
SIFT	LBP SQUARE REGION	—	95.14%	0.9275
	BIC 128 bins	Smaller ROI	91.70%	0.8798
		Bigger ROI	81.93%	0.7300
	BIC 64 bins	Smaller ROI	93.31%	0.8966
		Bigger ROI	88.14%	0.8229
	LBP SQUARE REGION	—	65.71%	0.4801

Table 3.5: Experiments results for different descriptors in the Monte Santo dataset.

Descriptor	Normalized Acc.	Global Acc.	Kappa
Global BIC	94.20%	93.63%	0.8987
BoVW-BIC	88.81%	87.74%	0.8046
BoVW-SIFT	57.93%	63.12%	0.3924
BoVG-BIC _{LBP}	95.14%	96.10%	0.9275
BoVG-SIFT _{BIC64}	93.31%	93.56%	0.8966

Table 3.6: Several experiments results for different descriptors in the Campinas dataset.

Descriptor	Normalized Acc.	Global Acc.	Kappa
Global BIC	80.41%	86.99%	0.8351
BoVW-BIC	88.64%	90.15%	0.8762
BoVW-SIFT	73.20%	83.39%	0.7901
BoVG-BIC _{LBP}	87.71%	92.73%	0.9068
BoVG-SIFT _{BIC64}	49.34%	63.11%	0.5318

approach surpasses by far the accuracy obtained by the BoVW of the literature. The combination of a different descriptor (color) for the edge features of the BoVG-SIFT_{BIC64} led to a great improvement in the accuracy from the BoVW-SIFT. This confirms that the combination of two types of descriptions (color and texture) is a good strategy when dealing with RSIs. The results obtained in Table 3.6 are consistent with the results shown in Table 3.5. The methods, which describe the image using the BIC color descriptor, achieved better results than the methods using SIFT descriptor.

3.4.2 *Does the proposed Bag-of-Visual-Graphs approach yield better results than other methods in the literature?*

To address this question, we performed some statistical analyses to compare our obtained results. We selected the Student's t-test and Wilcoxon test (5% level of significance) to compare the results. We present the comparison between the methods using the Students t-test for the normalized accuracy, in which we compare our best approach (BoVG-BIC_{LBP}) with every other method. If the comparison is above the zero line, the BoVG-BIC_{LBP} is better and statistically different from the compared method. If the comparison cross the zero line, we can not assure their difference. Figure 3.4 shows the Students t-test analysis for the Monte Santo dataset, confirming that our BoVG-BIC_{LBP} is statistically different from the Bag-of-Visual-Words approaches. The Wilcoxon test confirms the results obtained with the Student's t-test.

Figure 3.5 presents the same Student's t-test, however, applied to the Campinas dataset. Our method achieved the best results in this case, tied with the BoVW approaches, only when the normalized accuracy measure is considered. However, as it can be observed in Table 3.6, in all the other evaluation measures, our proposed method achieved the overall best result.

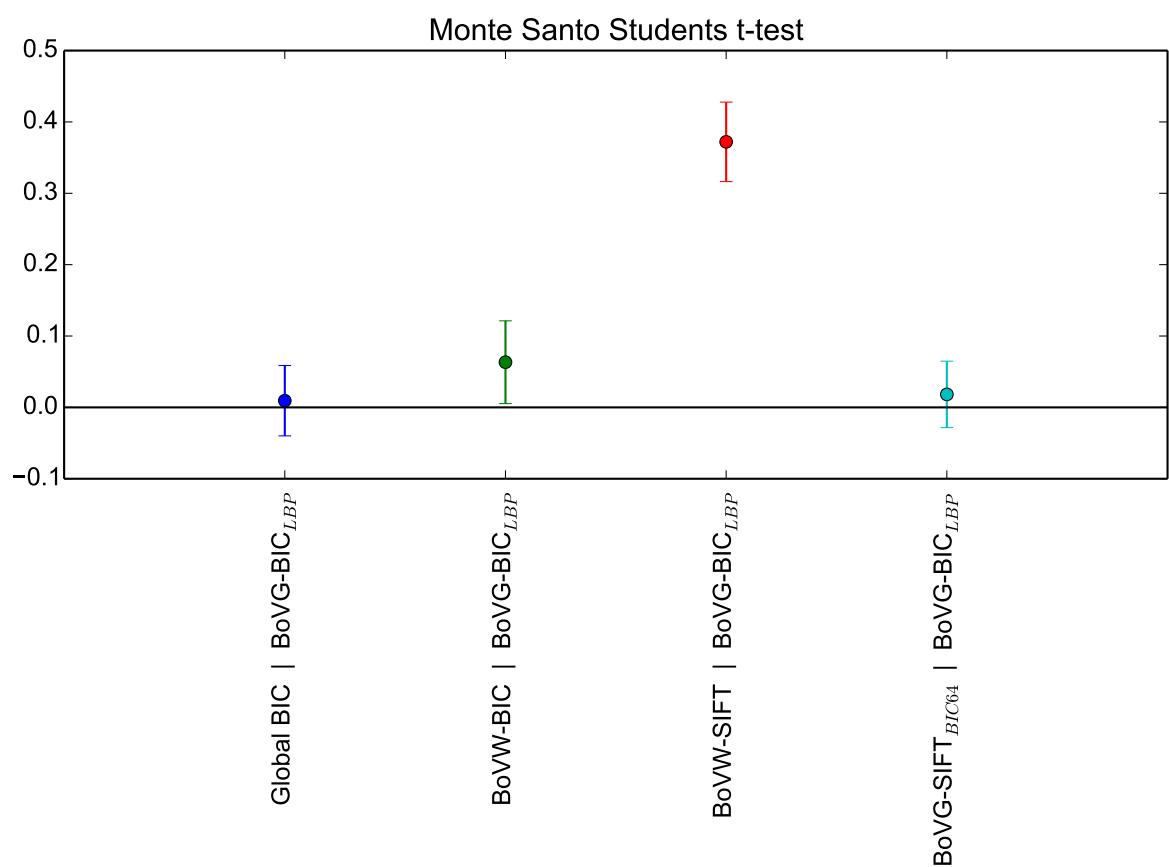


Figure 3.4: Statistical analysis of the experiments in the Table 3.5 using Student's t-test.

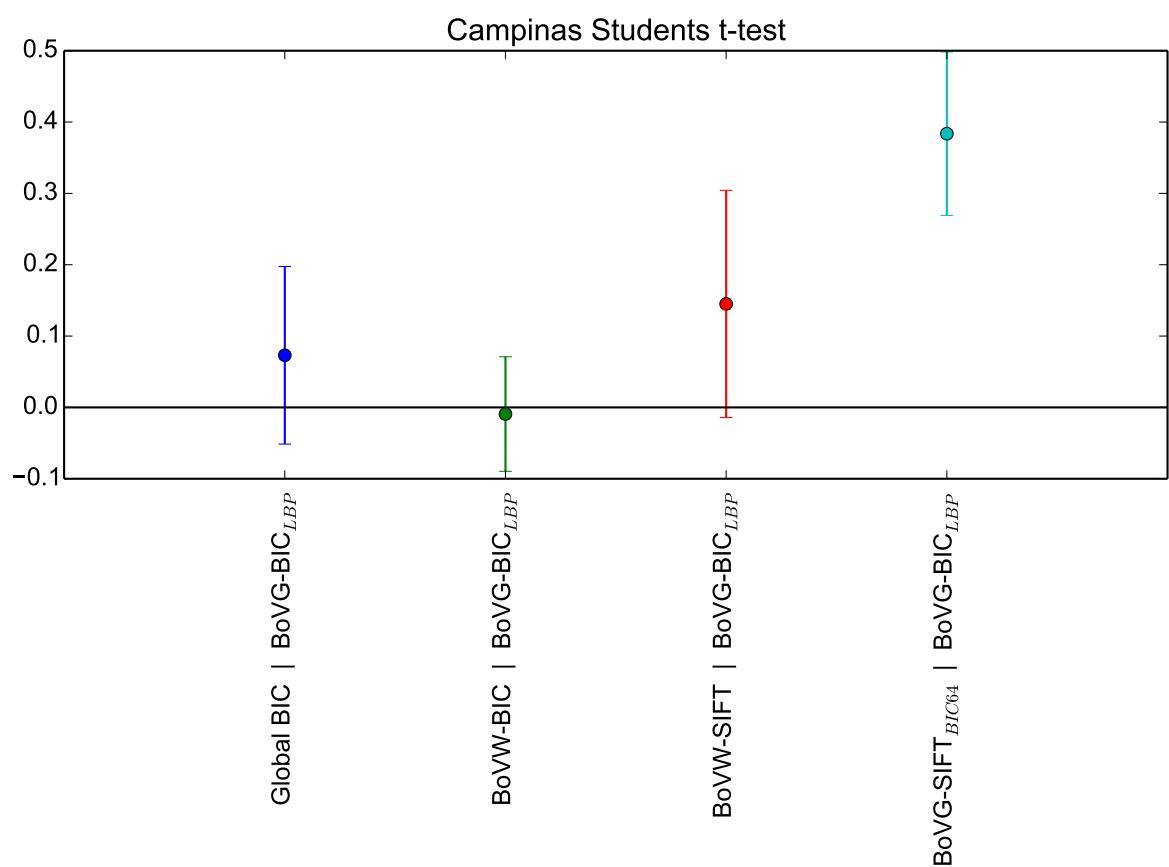


Figure 3.5: Statistical analysis of the experiments in the Table 3.6 using Student's t-test.

3.5 Conclusions

In this work we have applied the Bag-of-Visual-Graphs approach to the scenario of remote sensing images. This approach considers both the description of interest points, and their spatial distribution within RSIs.

We applied the Bag of Visual Graphs in two different datasets, Monte Santo and Campinas, and also performed a study of its parameter settings. We have shown that the BoVG approach achieved effective results in the Monte Santo dataset, and its results in the Campinas dataset were consistent with the obtained in the Monte Santo dataset.

We also performed statistical analysis with Student's t-test and Wilcoxon test using the normalized accuracy. For the Monte Santo dataset, the BoVG approach was statistically different from the BoVW approach, as in the Campinas dataset, the BoVG approach was tied with the BoVW-BIC approach, however, our approach yielded better results considering other evaluation measures.

For future work, we plan to increase the number of methods to compare with the BoVG approach, such as the BoW-Propagation, and as well develop a BoVG-Propagation. We also plan to use this approach on different remote sensing datasets.

Chapter 4

Graph-based Early-fusion for Flood Detection

This chapter refers to the paper¹ [137] published in the proceedings of the *2018 IEEE International Conference on Image Processing* (ICIP 2018), which took place in Athens, Greece.

In this chapter, we present our graph-based multimodal representation approach, and discuss its use in the context of the task related to the detection of flooding events. Flooding is one of the most harmful natural disasters, as it poses danger to both buildings and human lives. Therefore, it is fundamental to monitor these disasters to define prevention strategies and help authorities in damage control. With the wide use of portable devices (e.g., smartphones), there is an increase of the documentation and communication of flood events in social media. However, the use of these data in monitoring systems is not straightforward and depends on the creation of effective recognition strategies. In this chapter, we propose a fusion-based recognition system for detecting flooding events in images extracted from social media. We propose two new graph-based early-fusion methods, which consider multiple descriptions and modalities to generate an effective image representation. Our results demonstrate that the proposed methods yield better results than a traditional early-fusion method and a specialized deep neural network fusion solution. Figure 4.1 shows the main concepts addressed in this chapter.

4.1 Introduction

Natural disasters caused 306 billion dollars in damage in the United States of America in 2017,² and it may rise with global warming, increasing the intensity of heavy rainstorms [4]. In this scenario, it is fundamental to create monitoring systems that help authorities define appropriate strategies for damage control prevention and for victims' assistance. Among the different natural disasters, flooding is one of the most harmful and

¹© 2018 IEEE. Reprinted, with permission, from R. De O. Werneck, I. C. Dourado, S. G. Fadel, S. Tabbone and R. Da S. Torres, "Graph-Based Early-Fusion for Flood Detection," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018.

²<https://www.nytimes.com/2018/01/08/climate/2017-weather-disasters.html> (As of Jan. 2018).

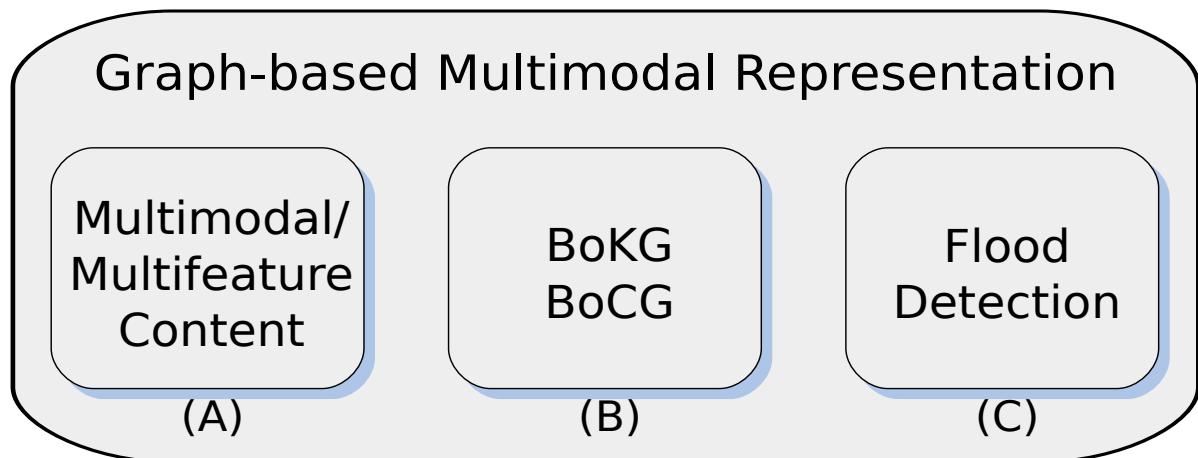


Figure 4.1: **Overview of the Chapter 4.** In (A) we have our object of studies, which are multimodal content; our proposed approaches for dealing with these objects is presented in (B); and (C) shows the flood detection problem which is our target in this chapter.

costly, as it destroys buildings, devastates agricultures, and threatens human lives [80].

However, traditional hydrological monitoring systems during floods have limited use in emergency response, due to, among other factors, ground inaccessibility or lack of aerial information [37]. Meanwhile, smartphones can provide an increase of documentation, dissemination, and communication of flooding events in social media streams. This new source of information may provide a much denser coverage of the natural disaster, and also document the impact of the disaster on human lives [127]. Also, handling multiple and complementary data modalities (e.g., text, images, videos) can help in the interpretation of flooding events. However, the accuracy and validity of these data may be questionable [109].

The literature considers the use of social media in the detection of natural events from different perspectives. Basnyat et al. [11] investigated a multi-modal approach using Twitter text and images to assess flood impacts. Twitter text was clustered using Latent Semantic Analysis into three clusters (help, damage, and casualties), and images were processed using Discrete Cosine Transformations to be classified into *water*, *nowater*, and *others*. Wang et al. [132] explored computer vision to classify natural events. They combined text content, based on a codebook containing the 1000 most frequent tags, for which each image has a vector indicating the presence or absence of the tag, with image content features learned using a Convolutional Neural Network (CNN).

The MediaEval initiative in 2017 also payed attention to this challenging detection problem. It proposed a task related to the retrieval of multimedia content from social media streams that are associated with flooding events (Disaster Image Retrieval from Social Media) [14]. One of the studies developed in the context of MediaEval 2017 refers to the work of Bischke et al. [13]. They proposed to extract visual features using CNNs and metadata features trained in a Word2Vec, with weights defined in terms of TF-IDF. They also concatenated the above representations for multi-modality-based experiments. Ahmad et al. [2] also proposed the use of CNNs. They extracted eight feature vectors, that were fed into ensembles of Support Vector Machines. For textual metadata, they

used a Random Tree classifier. In the multimodal approach, the classification scores were combined for both methods using Induced Ordered fusion scheme and Particle Swarm Optimization. Avgerinakis et al [6] proposed a CNN framework, using the GoogLeNet architecture to classify visual features only. To detect flooding using textual data, they adapted the DBpedia Spotlight, followed by a disambiguation algorithm using Jaccard similarities. They also performed a late fusion method to combine both modalities with a non-linear graph-based technique. Nogueira et al. [89] also employed CNNs (ResNet [54] and GoogLeNet architectures) to classify visual data. They used a Relation Network (RN) to learn the co-occurrence of words in the metadata, and also a ranked solution, in which they used a rank aggregation technique on the best three pairs of text representation models and distance functions. Finally, they concatenated the RN and the CNN to devise a multimodal approach.

In this chapter, we present two graph-based early-fusion approaches that combine different features and/or modalities, and apply them in a scenario of flooding detection in social media streams. We provide a joint representation of the image considering different modality descriptions, completing each others view. A graph representation is used to encode existing relations among representations in multiple feature/modality spaces. This graph is projected into a graph codebook, generating a final joint vector representation. These approaches can be applied for any feature extraction framework that provides multiple representations associated with the same or different modalities. Experiments show that the graph-based representation is more effective than traditional baselines from the literature, e.g. concatenation fusion. Moreover, in some cases, our approaches perform better than a recent deep neural network approach where feature description pairs are learned.

4.2 Graph-based Early-Fusion Methods

Our motivation is based on previous works [115] where we propose a discriminant and efficient representation based on local structures of an image combining graphs with the BoW model. We introduced two Bag-of-Graphs (BoG)-based models that generate a meaningful vocabulary describing the main local patterns of a set of objects. We presented formal definitions, introducing concepts and rules that make these models flexible and adaptable for classification problems.

In this perspective, we propose in this chapter two graph-based early-fusion methods which extend the BoG approach to create a joint representation of multiple descriptions and/or modalities: Bag of KNN Graphs and Bag of Cluster Graphs. The fusion scheme aims to encode existing relationships between different features of objects.

4.2.1 Bag of KNN Graphs

Bag of KNN Graphs (BoKG) considers multiple features or modalities originated from a same object. This approach first builds a graph, where a vertex represents the object and edges connect their multiple representations associated with different feature spaces. In the following, this graph is enriched by adding edges that connect each object with its

k -nearest neighbors according to each representation. The weights of edges connecting vertices within the same feature space are defined as the similarity score among object features. The weights of edges among vertices of different feature space, in turn, are based on the identification of the k nearest neighbors of each vertex, and on the use of a ranked-list-based similarity function.

Figure 4.2 illustrates this process. Given the graphs defined for each object, we apply the bag approach to describe the object represented by this graph. First, a collection of objects (A) is described by different description schemes (B). For each object, its k -nearest neighbors are determined for each description and a graph is created connecting vertices associated with objects and their neighbors (C). Each vertex is an object, and its feature its the description in the feature space. We define the edge weight as the distance between the two vertices connected by it. Then, we connect the different features (i.e., points in different feature spaces) of an object with an edge. The weight of the edge is defined by the similarity between the ranked lists of the objects connected by the edge (D). Next, we extract node signatures from all object graphs. We use the same definition of node signature as [115], which is composed of the feature of the vertex, its degree, and the features of its adjacent edges. These node signatures are used to create the codebook of the bag approach (E), either by a random selection or a clustering approach. For a new object (F), the same process is repeated. It is characterized by the description approaches (G), and has its graph created, considering as the nearest neighbors the objects in the collection (H). Edge weights are again computed by the similarity of ranked lists (I). And finally, we extract all the node signatures from this object graph to perform the coding and pooling steps of the bag approach (J) and thus generate its final vector representation (K).

4.2.2 Bag of Cluster Graphs

We also proposed another extension for the Bag-of-Graphs approach, a Bag of Cluster Graphs (BoCG). In this extension, given multiple representations, a unique graph is created. In this graph, objects represented within the same feature space are first clustered into n clusters. Cluster centroids represent the vertices of the final graph. Next, for each object in the collection, we find the clusters in the different feature spaces to which this object representation is assigned. Later, edges are created, connecting centroids of clusters to which the object belongs. The edge weight is defined as the ratio of the number of objects belonging to the two vertices of the edge, by the total number of objects in the collection.

Figure 4.3 illustrates this approach. First, a collection of objects (A) is described using two or more description methods (e.g., D_1 and D_2). Then, we create clusters of these features (B) and use their centroids to represent the vertices of the final graph (C). Next, for each object in the collection, we find the clusters to which each object is assigned. Then, we connect the clusters that are associated with the same object (C), e.g., the triangle object belongs to cluster w_1 in D_1 and w_2 in D_2 , so we connect the vertices w_1D_1 and w_2D_2 in our graph. Later, we extract node signatures from the graph created. These node signatures are also clustered to construct the codebook of the Bag-of-Graphs

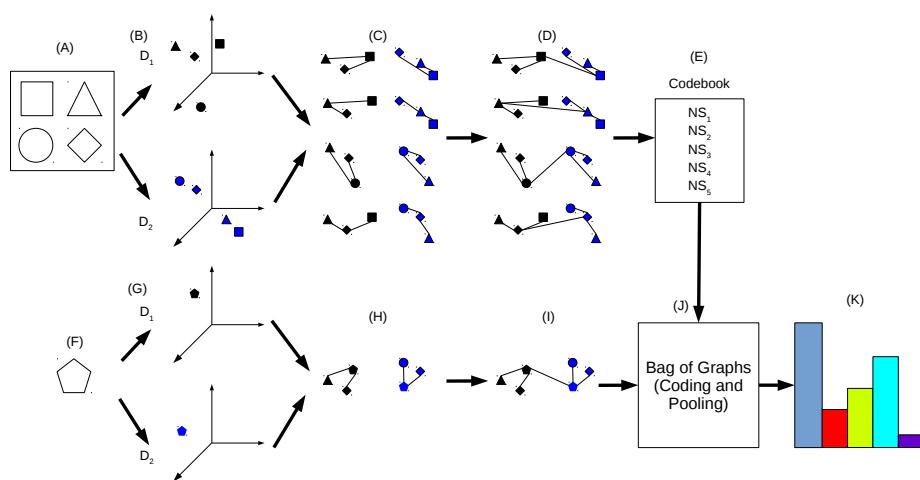


Figure 4.2: Bag of KNN Graphs. In this method, a collection of objects (A) is described by two or more different description schemes (B). In the feature space of each scheme, we find the k -nearest neighbors of each object, and a star graph is created connecting the objects and their neighbors (C). After that, we connect the different descriptions of each feature space of the same object with an edge (D). As they are the same object, just in different spaces, we can connect them. Having the graphs for each object, we can extract the node signatures from all objects, which are used to create our codebook (E). At the arrival of a new object (F), we perform the same steps to extract its node signatures (G, H, and I). Finally, we obtain the object node signatures to perform the coding and polling steps of the bag approach (J), and thus generate the vector representation of the new object (K).

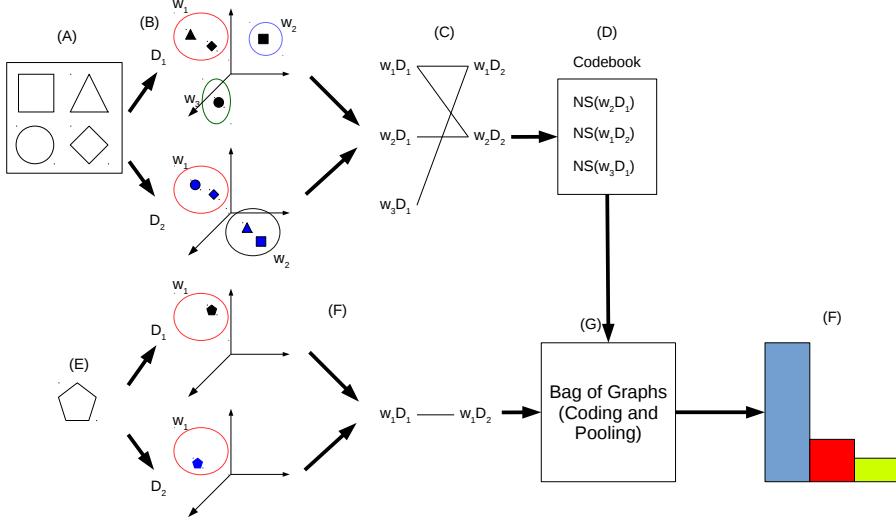


Figure 4.3: Bag of Cluster Graphs. In this approach, a collection of objects (A) is described by two or more different description schemes (B). In each feature space, we perform a clustering method on the objects, and use the obtained centroids to represent the vertices of the final graph. Next, we connect with an edge the centroids which contains the same object in different feature spaces (C). Later, we extract the node signatures from this graph, which we will use in the codebook (D). Given a new object (E), we apply the same steps as before. First we describe the object and find in which clusters it belongs to, then have the graph representing this object (F). Later, we extract the node signatures from the object graph, and apply the coding and pooling steps of the bag approach (G) to obtain the final feature representing the new object (H).

approach (D), in which each object is represented by the node signatures of each description. Given a new object (E), we apply the same steps to generate the node signatures. We predict in which clusters the new object features are, following the edges between these cluster vertices, and extract the node signatures from this object graph (F). Finally, coding and pooling methods (G) are applied to generate the final feature.

4.3 Experiments and Results

4.3.1 Dataset

The dataset used in this work was the one developed for the Multimedia Satellite Task at MediaEval 2017 in its first subtask: Disaster Image Retrieval from Social Media. This dataset is composed of 6,600 Flickr images extracted from the YFCC100M-Dataset [126], in which the images with *flooding* tags were selected and refined by human annotators. Both images and metadata were available in the challenge. The images were divided into two separated sets, development and test, with 5,280 and 1,320 images, respectively.

To evaluate our proposals before testing it, we also split the development set into training and validation sets, with a ratio of 80/20 (4,224 in the training set and 1,056 in the validation set). Thus, we could train our proposed methods, and select their best configuration to test, following the evaluation protocol adopted in the MediaEval

Table 4.1: Details on the Disaster Image Retrieval from Social Media dataset.

	# images	# images dev	# images test
DIRSM MediaEval 2017	6,600	4,224 training + 1,056 validation	1,320

competition. Table 4.1 summarizes the information on the dataset.

4.3.2 Features and Baselines

We used the visual features provided by the organization of the challenge. These visual features were extracted with the LIRE library³ with default parameters. The provided visual features are: AutoColorCorrelogram (ACC) [56]; EdgeHistogram (EH)⁴; Color and Edge Directivity Descriptor (CEDD) [24]; ColorLayout (CL)⁴; Fuzzy Color and Texture Histogram (FCTH) [25]; Joint Composite Descriptor (JCD) [26]; Gabor [36]; Scalable-Color (SC)⁴; and Tamura [125]. For the textual data provided, we use 2GRAMS with Term Frequency, followed by a PCA for reducing the dimensionality.

We also included the concatenation of the provided features as a baseline early-fusion method to compare with the graph-based early-fusion methods proposed in this chapter.

4.3.3 Evaluation

The MediaEval 2017 contest proposed the use of Average Precision@ X , with several cut-offs (50, 100, 250, and 480), for the correctness of retrieved images in the experiments. This metric scores the proportion of relevant images among the top- X retrieved images, also taking their order into account. Here we present our results considering the top-50 retrieved images. For the baselines and the proposed approaches, we performed experiments with a two-class SVM classifier (with linear kernel and $C = 1$).

4.3.4 Results

Baselines

First, we computed the results considering all features provided, using the validation set. Table 4.2 shows the Average Precision @ 50 (AP@50). EdgeHistogram obtained the best AP for the provided image features, with a precision of 69.03%, and 2GRAMS_TF (PCA), a text descriptor, obtained an AP of 81.47%.

As baselines, we also considered the concatenation of these features. Table 4.3 shows the results considering the concatenation of the provided features normalized between 0.0 and 1.0, and the concatenation of the best provided image features with the textual one. The visual features introduced noise when concatenated with textual features, leading to worse precision scores.

³<http://www.lire-project.net/> (As of Nov. 2017).

⁴<https://mpeg.chiariglione.org/standards/mpeg-7/visual> (As of Nov. 2017).

Table 4.2: Average precision for the baselines in the validation.

Feature	AP@50 (%)
ACC	50.55
CEDD	58.17
CL	47.26
EH	69.03
FCTH	59.53
Gabor	24.84
JCD	60.58
SC	5.63
Tamura	15.11
2GRAMS_TF (PCA)	81.47

Table 4.3: Results of the concatenation of all visual features and our best modalities features as baseline.

Concatenation	AP@50 (%)
ACC + CEDD + CL + EH + FCTH + Gabor + JCD + SC + Tamura	82.25
EH + 2GRAMS_TF (PCA)	68.27

The best baseline result, considering the concatenation of features, attained an AP@50 of 82.25%. Once the most promising features were identified, we evaluated the proposed approaches and compared with these baselines in the scenario of the challenge, i.e., using the validation and test sets.

Evaluation of Proposed Approaches

First, we used the validation set with the aim of identifying the best parameters for our approaches. The Bag of KNN Graphs uses two sets of nearest neighbors (with 10 and 20 neighbors), a Cosine similarity metric, a codebook of 500 node signatures randomly selected, the intersection of ranked lists as similarity function, hard assignment, and max pooling. For the Bag of Cluster Graphs, which has less parameters, we selected 1000 random features for each modal cluster and 2000 random node signatures. We also used hard assignment and max pooling. These parameters were selected as they provided the best results in experiments with the validation set.

Table 4.4 shows the results for the Bag of KNN Graphs compared with the baselines. As we can see, our BoKG approach performed similarly as the concatenation considering all provided features but, for the multiple modality combination, it performed better than baselines, showing that our proposed approach can provide an effective joint representation by combining different modalities (text and visual features) of the same object.

Table 4.5 presents the results obtained with the Bag of Cluster Graphs. This table shows that, although it did not perform better than the BoKN, our results for the multiple modalities joint representation also outperformed the baseline based on early-fusion concatenation. The results of the BoCG are below the one of BoKG because of its sparse final

Table 4.4: Bag of KNN Graphs (BoKG) results.

Features	AP@50 (%)
ACC + CEDD + CL + EH + FCTH + Gabor + JCD + SC + Tamura	81.11
EH + 2GRAMS_TF (PCA)	86.90

Table 4.5: Bag of Cluster Graphs (BoCG) results.

Features	AP@50 (%)
ACC + CEDD + CL + EH + FCTH + Gabor + JCD + SC + Tamura	47.94
EH + 2GRAMS_TF (PCA)	73.85

vector representation, as this approach uses less node signatures in the coding and pooling steps than the BoKG. The sparse features provided less information for the classifier to train a separation model between considered classes.

Comparison with the Relation Network Approach

Relation Network (RN) [110] is a recently proposed neural network, which learns to infer relationships between objects and produce decisions over them. The RN is composed of two neural networks (denominated f and g) whose parameters are learned together. The function g is used to encode the relationship between pairs of objects, while the function f takes the sum of all encodings as input and produces a decision over the entire collection. We used a Relation Network as a baseline, finding the relationship between the objects representations. Table 4.6 shows the Average Precision @ 50 for this deep network. The experiments with the RN used the parameters suggested by the authors: 128 epochs and a learning rate of $2.5 \cdot 10^{-4}$, and we also used the same training set of the proposed approaches. The provided results were not better than BoKG (see Table 4.4), as our approach enriched the final representation when considering the neighborhood of objects in different feature spaces.

4.4 Conclusions

In this chapter, we presented two new approaches based on Bag of Graphs to create a joint representation of multiples modalities and/or descriptions: Bag of KNN Graphs and Bag of Cluster Graphs. We validate these approaches in the flood detection scenario, proposed by the MediaEval 2017 contest. In this scenario, we show that our early-fusion approach

Table 4.6: Results of the Relation Network deep approach.

Features	AP@50 (%)
ACC + CEDD + CL + EH + FCTH + Gabor + JCD + SC + Tamura	79.63
EH + 2GRAMS_TF (PCA)	75.55

outperforms the traditional concatenation fusion when dealing with multiple modalities. Our experiments also showed that our approach has better performance than a deep neural network (RN). For future work, we propose to compare our methods with other early-fusion methods, as well as other deep learning approaches that use early-fusion. We also plan to include deep features as input features/modalities of our approaches.

Chapter 5

Learning Cost Functions for Graph Matching

In this chapter, we present an original approach to learn how to perform matching between graphs. This approach, described in the paper “Learning Cost Functions for Graph Matching”¹ [138], was presented in the proceedings of the *IAPR Joint International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition* (S+SSPR 2018) in Beijing, China.

During the last decade, several approaches have been proposed to address detection and recognition problems, by using graphs to represent the content of images. Graph comparison is a key task in those approaches and usually is performed by means of graph matching techniques, which aim to find correspondences between elements of graphs. Graph matching algorithms are highly influenced by cost functions between nodes or edges. In this perspective, we propose an approach to learn the matching cost functions between graphs’ nodes. Our method is based on the combination of distance vectors associated with node signatures and a classifier, which is used to learn discriminative node dissimilarities. Experimental results on different datasets compared to a learning-free method are promising. An overview of this chapter is presented in Figure 5.1.

5.1 Introduction

In the pattern recognition domain, we can represent objects using two methods: statistical or structural [20]. In structural, objects are represented by a data structure (e.g., graphs, trees), which encodes their components and relationships; and in statistical, objects are represented by means of feature vectors. Most methods for classification and retrieval in the literature are limited to statistical representations [38]. However, structural representation are more powerful, as the object components and their relations are described in a single formalism [115]. Graphs are one of the most used structural representations.

¹Reprinted by permission from Springer Nature Terms and Conditions for RightsLink Permissions Springer Nature Customer Service Centre GmbH: Springer Structural, Syntactic, and Statistical Pattern Recognition. “Learning Cost Functions for Graph Matching”, Rafael de O. Werneck, Romain Raveaux, Salvatore Tabbone, Ricardo da S. Torres, COPYRIGHT (2018)

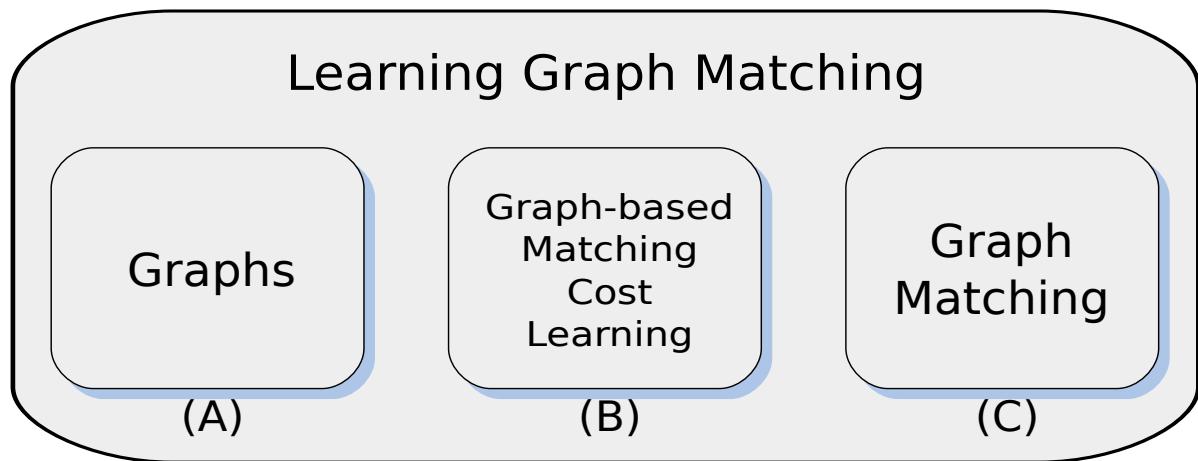


Figure 5.1: **Overview of the Chapter 5.** In this chapter, we approach the problem of graph matching (C) with a new approach to learn the matching cost (B) between two graphs (A).

Unfortunately, graph comparison suffers from high complexity, which, at the moment, does not have a polynomial solution for the problem.

One of the widely used method for graph matching is the graph edit distance (GED). GED is an error-tolerant graph matching paradigm that defines the similarity of two graphs by the minimum number of edit operations necessary to transform one graph into another [19]. A sequence of edit operations that transforms one graph into another is called edit path between two graphs. To quantify the modifications implied by an edit path, a cost function is defined to measure the changes proposed by each edit operation. Consequently, we can define the edit distance between graphs as the edit path with minimum computational cost.

The possible edit operations are: node substitution, edge substitution, node deletion, edge deletion, node insertion, and edge insertion. The cost function is of first interest and can change the problem being solved. In [17, 18], a particular cost function for the GED is introduced, and it was shown that under this cost function, the GED computation is equivalent to the maximum common subgraph problem. Neuhaus and Bunke [87], in turn, showed that if each elementary operation satisfies the criteria of a metric distance (separability, symmetry, and triangular inequality) then the GED is also a metric.

Usually, cost functions are manually designed and are domain-dependent. Domain-dependent cost functions can be tuned by learning weights associated with them. In Table 5.1, published papers dealing with edit cost learning are tabulated. Two criteria are optimized in the literature, the matching accuracy between graph pairs or an error rate on a classification task (classification level). In [86], learning schemes are applied on the GED problem while in [70, 21], other matching problems are addressed. In [70], the learning strategy is unsupervised as the ground truth is not available. In another research venue, different optimization algorithms are used. In [85], Self-Organizing Maps (SOMs) are used to cluster substitution costs in such a way that the node similarity of graphs from the same class is increased, whereas the node similarity of graphs from different classes is decreased. In [86], Expectation Maximization algorithm (EM) is used for the

Table 5.1: Graph matching learning approaches.

Ref.	Graph matching problem	Supervised	Criterion	Optimization method
[85]	GED	Yes	Recognition rate	SOM
[86]	GED	Yes	Recognition rate	EM
[31, 32]	GED	Yes	Matching accuracy	Quadratic programming
[21]	Other	Yes	Matching accuracy	Bundle
[29]	Other	Yes	Matching accuracy	SSVM
[70]	Other	No	Matching accuracy	Bundle

same purpose. An assumption is made on attribute types. In [29], the learning problem is mapped to a regression problem and a structured support vector machine (SSVM) is used to minimize it. In [31], a method to learn scalar values for the insertion and deletion costs on nodes and edges is proposed. An extension to substitution costs is presented in [32]. The contribution presented in [106] is the closest work related to our proposal. In that work, the node assignment is represented as a vector of 24 features. These numerical features are extracted from a node-to-node cost matrix that is used for the original matching process. Then, the assignments derived from exact graph edit distance computation is used as ground truth. On this basis, each node assignment computed is labeled as correct or incorrect. This set of labeled assignments is used to train an SVM endowed with a Gaussian kernel in order to classify the assignments computed by the approximation as correct or incorrect. This work operates at the matching level. All prior works rely on predefined cost functions adapted to fit an objective of matching accuracy. Few researches has been carried out to automatically design generic cost functions in a classification context.

In this chapter, we propose to learn a discriminative cost function between nodes with no restriction on graph types nor on labels for a classification task. On a training set of graphs, a feature vector is extracted from each node of each graph thanks to a node signature that describes local information in graphs. Node dissimilarity vectors are obtained by pairwise comparison of the feature vectors. Node dissimilarity vectors are labeled according to the node pair belonging to graphs of the same class or not. On this basis, an SVM classifier is trained. At the decision stage, two graphs are compared, a new node pair is given as an input of the classifier, and the class membership probability is outputted. These adapted costs are used to fill a node-to-node similarity matrix. Based on these learned matching costs, we approximate the matching graph problem as a Linear Sum Assignment Problem (LSAP) between the nodes of two graphs. The LSAP aims at finding the maximum weight matching between the elements of two sets and this problem can be solved by the Hungarian algorithm [67] in cubic time.

The chapter is organized as follow: Section 5.2 presents our approach for local description of graphs, and the proposed approaches to populate the cost matrix for the Hungarian algorithm. Section 5.3 details the datasets and the adopted experimental protocol, as well as presents the results and discussions about them. Finally, Section 5.4 is devoted to our

conclusions and perspectives for future work.

5.2 Proposed Approach

In this section, we present our proposal to solve the graph matching problem as a bipartite graph matching using local information.

5.2.1 Local Description

In this work, we use node signatures to obtain local descriptions of graphs. In order to define the signature, we use all information of the graph and the node. Our node signature is represented by the node attributes, node degree, attributes of incident edges, and degrees of the nodes connected to the edges. Given a general graph $G = (V, E)$, we can define the node signature extraction process and representation, respectively, as:

$$\Gamma(G) = \{\gamma(n) | \forall n \in V\} \quad (5.1)$$

$$\gamma(n) = \{\alpha_n^G, \theta_n^G, \Delta_n^G, \Omega_n^G\} \quad (5.2)$$

where α_n^G is the attributes of the node n , θ_n^G is the degree of the node n , Δ_n^G is the set of degrees of adjacent nodes to n , and Ω_n^G is a set of attributes of the incident edges of n . This node signature was selected as it was successfully used in other works [115, 137].

5.2.2 HEOM Distance

One of our approaches to perform graph matching consists on finding the minimum distance to transform the node signatures from one graph into the node signatures from another graph. To calculate the distance between two node signatures, we need a distance metric capable of dealing with numeric and symbolic attributes. We selected the *Heterogeneous Euclidean Overlap Metric* [139] (HEOM) and we provided an adaptation for our graph local description.

The HEOM distance is defined as:

$$HEOM(i, j) = \sqrt{\sum_{a=0}^n \delta(i_a, j_a)^2}, \quad (5.3)$$

where a is each attribute of the vector, and $\delta(i_a, j_a)$ is defined as:

$$\delta(i_a, j_a) = \begin{cases} 1 & \text{if } i_a \text{ or } j_a \text{ is missing,} \\ 0 & \text{if } a \text{ is symbolic and } i_a = j_a, \\ 1 & \text{if } a \text{ is symbolic and } i_a \neq j_a, \\ \frac{|i_a - j_a|}{\text{range}_a} & \text{if } a \text{ is numeric.} \end{cases} \quad (5.4)$$

In our approach, we define the distance between two node signatures as follows. Let $A = (V_a, E_a)$ and $B = (V_b, E_b)$ be two graphs and $n_a \in V_a$ and $n_b \in V_b$ be two nodes from these graphs. Let $\gamma(n_a)$ and $\gamma(n_b)$ be the signature of these nodes, that is:

$$\gamma(n_a) = \{\alpha_{n_a}^A, \theta_{n_a}^A, \Delta_{n_a}^A, \Omega_{n_a}^A\}$$

and

$$\gamma(n_b) = \{\alpha_{n_b}^B, \theta_{n_b}^B, \Delta_{n_b}^B, \Omega_{n_b}^B\}.$$

The distance ϵ between two node signatures is:

$$\begin{aligned} \epsilon(\gamma(n_a), \gamma(n_b)) &= HEOM(\alpha_{n_a}^A, \alpha_{n_b}^B) + HEOM(\theta_{n_a}^A, \theta_{n_b}^B) + \\ &HEOM(\Delta_{n_a}^A, \Delta_{n_b}^B) + \frac{\sum_{i=1}^{|\Omega_{n_a}^A|} HEOM(\Omega_{n_a}^A(i), \Omega_{n_b}^B(i))}{|\Omega_{n_a}^A|} \end{aligned} \quad (5.5)$$

5.2.3 SVM-based Node Dissimilarity Learning

We propose a SVM approach to learn the graph edit distance between two graphs. In this approach, we first define a *distance vector* ϵ' between two node signatures. Function ϵ' is derived from ϵ , but instead of summing up the distance related to all structures, the function considers each structure distance score as a value of a bin of the vector. This distance vector is composed of the HEOM distance between each structure of the node signature, i.e., the distance between the node attribute, node degree, degrees of the nodes connected to the edges, and attributes of incident edges are components of the vector, i.e.,

$$\begin{aligned} \epsilon'(\gamma(n_a), \gamma(n_b)) &= [HEOM(\gamma(n_a)_i, \gamma(n_b)_i)], \forall i \in \{0, \dots, |\gamma(n)|\} | \\ &\gamma(n)_i \text{ is a component of } \gamma(n) \text{ and} \\ &|\gamma(n)| \text{ is the cardinality of } \gamma(n). \end{aligned}$$

To each distance vector ϵ' , a label is assigned. These labels guide the SVM learning process. We propose the following formulation to assign labels to distance vectors. Let $Y = \{y_1, y_2, \dots, y_l\}$ be the set of l labels associated with graphs. In our formulation, denominated multi-class, distance vectors, which are associated with node signatures extracted from graphs of the same class (say y_i), are labeled as y_i . Otherwise, a new label y_{l+1} is used, representing that the distance vectors were computed from node signatures belonging to graphs from different classes. Figure 5.2 shows how this labeling is performed.

Figure 5.3 illustrates the main steps of our approach. Given a set of training graphs (step A in the figure), we first extract the node signatures from all graphs (B), and compute the pairwise distance vectors (C). We then use the labeling procedure described above to assign labels to the distance vectors defined by node signatures extracted from graphs of the training set and use these labeled vectors to train a SVM classifier (D).

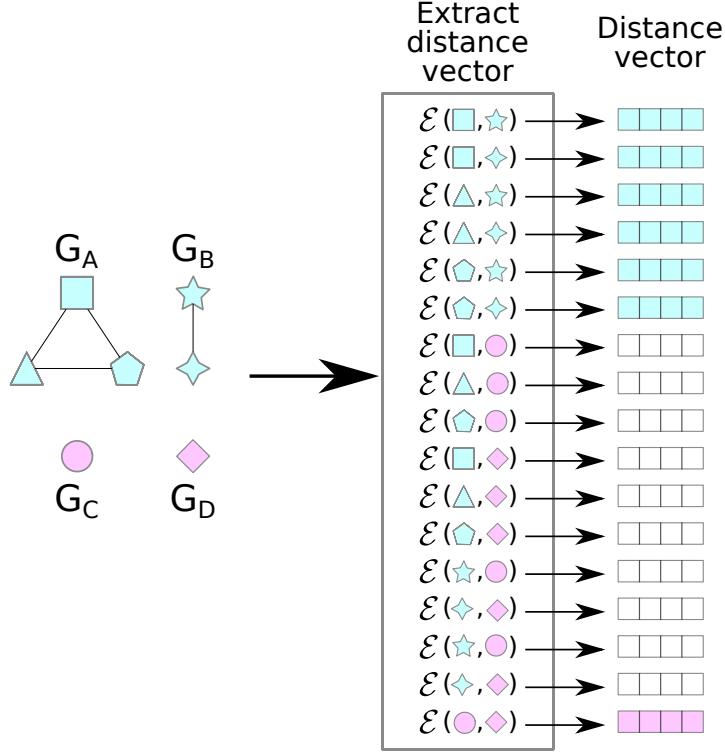


Figure 5.2: **Representation of the label assignment to a distance vector.** When the nodes belong to graphs of the same class (same color – blue and pink distance vectors – in the figure), the distance vector receives the same label. Alternatively, when the nodes belong to graphs of different classes, the distance vector is labeled as the new label, or “different” (white distance vectors)

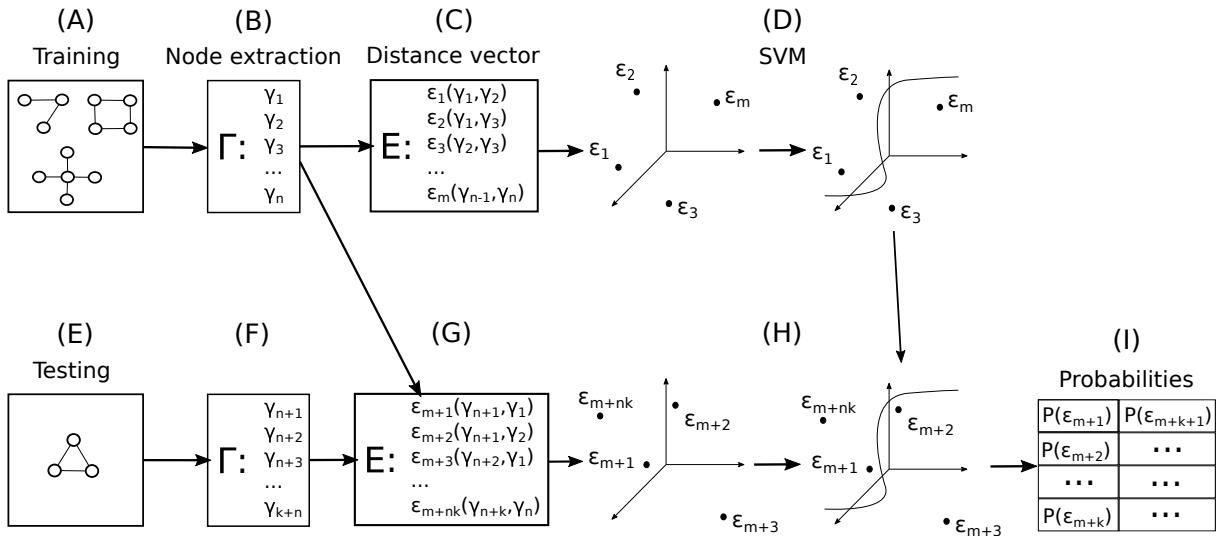


Figure 5.3: **Proposed SVM approach to compute the edit cost matrix.** Given a training set of graphs (A), we extract their node signatures (B), and, combining them pair to pair to obtain the distance vectors (C). Then we train a SVM classifier on these vectors. Next, for a new testing graph (E), we extract its node signatures (F), perform a pairwise combination with the signatures from the training set (G), and classify them using the trained SVM (H). Finally, we can obtain the probabilities of each vector belonging to the training class, to populate our Hungarian Matrix.

5.2.4 Graph Classification

At testing stage, each one of the graphs from the test set (E) has its node signatures extracted (F). Again, distance vectors are computed, now considering node signatures from the test and from the training set (G). With the distance vectors, we can project them into the learned feature space and obtain the probability of a test sample that belongs to the training set classes considering the SVM hyperplane of separation (H). These probabilities are used to populate a cost matrix for each graph in the training set (I), in such a way that, for each node signature from the test graph (row) and each node signature from the training graph (column), we create a matrix of probabilities for each combination of test and training graphs. This matrix is later used in the Hungarian algorithm. As the resulting cost matrices encodes probabilities, we compute the maximum cost path using the Hungarian algorithm instead of the minimum. The test sample classification is based on the k-nearest neighbor (kNN) graphs found in the training set, where graph similarity is defined by the Hungarian algorithm.

5.3 Experimental Results

In this section, we describe the datasets used in the experiments, we present our experimental protocol, and how our method was evaluated. At the end, we present our results and discuss them.

5.3.1 Datasets

In our chapter, we perform experiments in three labeled datasets from the IAM graph database [105]: Letter, Mutagenicity, and GREC. Their details are summarized in Table 5.2.

The **Letter** database compromises 15 classes of distorted letter drawings. Each letter is represented by a graph, in which the nodes are ending points of lines, and edges are the lines connecting ending points. The attribute of the node is its position. This dataset has three sub-datasets, considering different distortions (low distortion, medium distortion, and a high distortion).

Mutagenicity is a database of 2 classes representing molecular compounds. In this database, the nodes are the atoms and the edges the valence of the linkage.

GREC database consists of symbols from architectural and electronic drawings represented as graphs. Ending points are represented as nodes and lines and arcs are the edges connecting these ending points. It is composed of 22 classes.

5.3.2 Experimental Protocol

Considering that the complexity and computational time to calculate the distance vectors for the SVM method is soaring, we decide to perform preliminary experiments where we randomly selected two graphs of each class from the training set to be our training, and for our test, we selected 10% of the testing graphs from each class. As we are selecting

Table 5.2: Information about the datasets.

	Datasets				
	Letter-LOW	Letter-MED	Letter-HIGH	Mutagenicity	GREC
# graphs	750	750	750	1500	286
# classes	15	15	15	2	22
# graphs per class	50	50	50	830/670	13
# graphs in learning	30	30	30	4	44
# distance vectors	$\approx 10,000$	$\approx 10,000$	$\approx 10,000$	$\approx 14,000$	$\approx 130,000$
# graphs in testing	75	75	75	129/104	44

Table 5.3: Accuracy results for HEOM distance and random population of the cost matrix in the graph matching problem (in %).

Approach	Datasets				
	Letter-LOW	Letter-MED	Letter-HIGH	Mutagenicity	GREC
Random	0.53 ± 0.73	1.60 ± 2.19	1.60 ± 1.12	54.85 ± 4.22	1.36 ± 2.03
HEOM distance	40.53 ± 11.72	15.73 ± 3.70	10.93 ± 3.70	49.44 ± 10.69	52.27 ± 7.19

randomly the training and testing sets, we need to perform more experiments to obtain an average result, to avoid any bias a unique experiment selecting training and testing sets can have. Thus, we performed each experiment 5 times to obtain our results. To evaluate our approach, we present the mean accuracy score and the standard deviation of a k -NN classifier ($k = 3$).

5.3.3 Results

In our first experiments, we provide two baselines: 1) A random baseline, in which we populated the cost matrix with random values between 0 and 1, and 2) we performed the graph matching using the HEOM distance function between the node signatures to populate the cost matrix. Table 5.3 shows these results for the chosen datasets. The HEOM distance approach shows improvement over a simple random selection of values.

As we can see in Table 5.3, the HEOM distance presents a better result than the random assignment of weights, except for the Mutagenicity dataset, which is the only dataset with two classes. In this case, the obtained results are similar, considering the standard deviation of the executions (± 4.22 for Random approach, and ± 10.69 for the HEOM approach).

Next, we ran experiments using the proposed multi-class SVM approach to compare with the results obtained using the HEOM distance in the cost matrix. We used default parameters for the SVM for the training step (RBF kernel, $C = 0$). We also present results of experiments in which we normalize the distance vector, using min-max (normalizing between 0 and 1) and zscore (normalization using the mean and standard deviation) normalizations. Table 5.4 shows the mean accuracy of the experiments made.

Table 5.4 shows us that the SVM approach is promising, obtaining better results for three of the five datasets considered. The improvement in the Mutagenicity dataset

Table 5.4: Mean accuracy and standard deviation (in %) for the HEOM distance and SVM multi-class approach in the graph matching problem. The best results for each dataset are show in bold.

Method	Norm.	Datasets			
		Letter-LOW	Letter-MED	Letter-HIGH	Mutagenicity
HEOM distance		40.53 ± 11.72	15.73 ± 3.70	10.93 ± 3.70	49.44 ± 10.69
SVM Multi-class	min-max	30.67 ± 5.50	28.00 ± 9.80	18.93 ± 5.77	71.24 ± 29.50
	zscore	33.33 ± 7.12	20.27 ± 6.69	14.40 ± 5.02	63.26 ± 15.61
		37.87 ± 9.83	21.87 ± 1.52	20.27 ± 8.56	20.00 ± 7.43
				64.12 ± 7.68	30.91 ± 2.59

Table 5.5: Accuracy scores for four datasets (in %).

Modification	Multi-class	Datasets			
		Letter-LOW	Letter-MED	Letter-HIGH	GREC
Without “different” class		37.87 ± 5.88	34.13 ± 9.78	29.07 ± 4.36	38.18 ± 8.86
	min-max	30.13 ± 6.34	30.13 ± 9.31	27.47 ± 7.92	35.45 ± 2.03
	zscore	44.80 ± 5.94	25.87 ± 0.73	29.07 ± 5.99	41.82 ± 7.11

was above 20 percentage points from the HEOM distance baseline. As for the other cases, the Letter-LOW dataset had similar results for the HEOM distance and SVM approach (standard deviation of the HEOM is ± 11.72 and for the SVM is ± 9.83). The GREC dataset was the only dataset with a distant results from the HEOM approach. We discuss that it is because the dataset has more classes than the others, so its “different” class contains more distance vectors combining node signatures of different classes. With this imbalanced distribution, the “different” class shadows the other classes in the SVM classification.

Table 5.4 also shows that a normalization step can help separate the classes in the SVM, being successful in improving the result of three of five approaches used, specially the zscore normalization, that considers the mean and standard deviation of the vectors.

To better understand our results, we also calculated the accuracy of the SVM classification for the same training used in it. Our experiments shows that the “different” class does not help the learning, especially in the datasets with more classes, as this “different” class overlook the other classes, preventing the classification as the correct class. It also shows the necessity of a bigger training and a validation set to tune the parameters of the SVM. Figure 5.4 shows a confusion matrix of a classification of the training data in the Letter-LOW dataset.

To improve our results, we propose to ignore the “different” class in the training set. Table 5.5 shows the accuracy for this new proposal.

As we can see in Table 5.5, our proposed modifications improved the results obtained in our experimental protocol. The dataset Letter-LOW achieved the best result when we do not consider the “different” class in the training step, avoiding misclassification as “different” class. With this, we show that our proposed approach to learn the cost to match nodes are very promising.

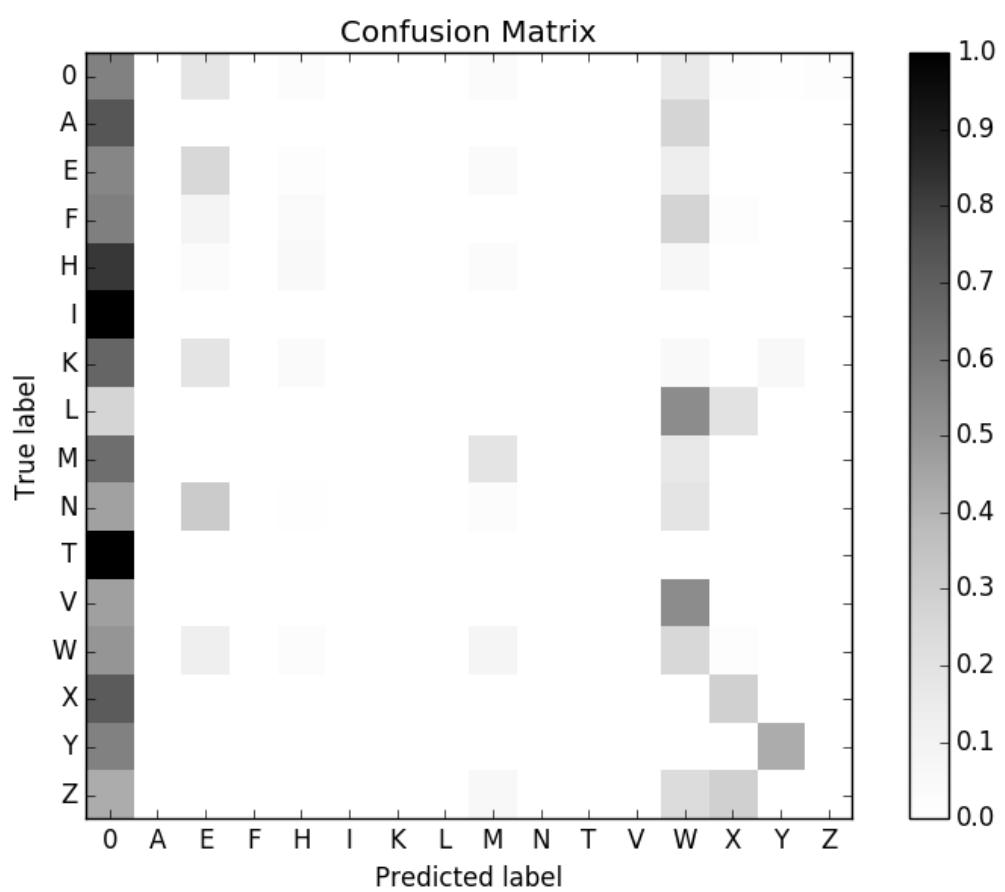


Figure 5.4: Classification of the training set for the Letter LOW dataset.

5.4 Conclusions

In this chapter, we presented an original approach to learn the costs to match nodes belonging to different graphs. These costs are later used to compute a dissimilarity measurement between graphs. The proposed learning scheme combines a node-signature-based distance vector and an SVM classifier to produce a cost matrix, based on which the Hungarian algorithm computes graph similarities. Performed experiments considered the graph classification problem, using k-NN classifiers built based on graph similarities. Promising results were observed for widely used graph datasets. These results suggest that our approach can also be extended to use similar methods based on local vectorial embeddings and can be exploited to compute probabilities as estimators of matching costs.

For future work, we want to perform experiments considering all training and testing sets to compare with our results presented in this chapter, and also make a complete study on the minimum training set necessary to achieve a good performance not only in classification, but also in retrieval tasks.

Chapter 6

Learning Cost Function for Graph Classification with Open-Set Methods

In this chapter, we present a generic framework to learn discriminative costs for a bipartite graph edit distance computation. This framework is described in the paper *Learning Cost Function for Graph Classification with Open-Set Methods*, submitted to the *Virtual Special Issue on “Recent Advances in Statistical, Structural and Syntactic Pattern Recognition”* of the *Pattern Recognition Letters* (PRL) journal.

In several pattern recognition problems, effective graph matching is of paramount importance. In this chapter, we introduce a novel framework to learn discriminative cost functions. These cost functions are embedded into a graph matching-based classifier. The learning algorithm is based on an open-set recognition approach. An open-set recognition describes a problem formulation in which the training process does not have access to labeled samples of all classes that may show up during the test phase. We also investigate a set of measures to characterize local graph properties. Performed experiments considering widely used datasets demonstrate that our solution leads to better or comparable results to those observed for several state-of-the-art baselines. Figure 6.1 shows an overview of this chapter.

6.1 Introduction

In several pattern recognition tasks, objects are often represented by means of two main approaches [20]: statistical or structural. In the former, objects are represented as points in n-dimensional space; while in the latter, objects are represented through data structures, which encode their components and relationships. The literature related to classification and retrieval tasks encompasses many more statistical representations. However, structural representations are more powerful, as they provide a single formalism on components and their relations [115]. In this work, we use graphs, one of the most adopted structural representation. In the field of structural pattern recognition, the graph comparison problem is of first importance. Unfortunately, due to the wide variability of patterns, the graph comparison problem is not a trivial task, as it often turns into an error-tolerant graph matching problem. The error-tolerant graph matching problem [19], in turn, is an

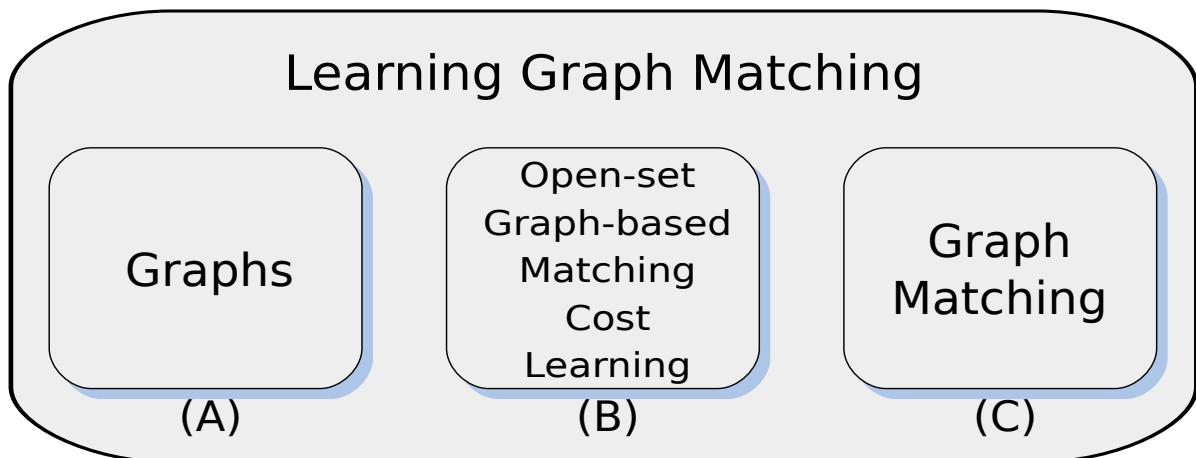


Figure 6.1: **Overview of the Chapter 6**, in which we present a novel framework to learn the discriminative cost functions based on an open-set approach (B) in the problem of graph matching (C) between two graphs (A).

NP-hard problem [47, 149]. Therefore, there are no exact methods that guarantee to solve the problem in polynomial time.

One successful tool to model the error-tolerant graph matching problem relies on the graph edit distance (GED) [104]. GED is an error-tolerant paradigm to define the similarity between two graphs through the minimum number of edit operations necessary to transform one graph into the other. A sequence of edit operations is called edit path between two graphs. To quantify the modifications implied by an edit path, a cost function is defined to measure the changes proposed by each operation. Consequently, we can define the edit distance between graphs as the edit path with minimum cost. Usually, cost functions are manually designed for each problem, being domain-dependent. Domain-dependent cost functions can be tuned by learning weights associated with them. In this chapter, we tackle a more general problem. What can we learn if the cost functions are not given by an expert? Can we extract information from the data to fit a specific goal given by the user?

Different papers address the edit cost learning problem. The contribution presented in [106] is the most related to our proposal. In their work, the authors represent the node assignment as a vector of 24 features. These features are extracted from a node-to-node cost matrix that is used for the original matching process. Then, the assignments derived from the exact graph edit distance computation is used as ground truth. Each node assignment computed is labeled as correct or incorrect where an SVM with a Gaussian kernel classify the assignments computed by the approximation as correct or incorrect. This work operates at the matching level. All prior works rely on predefined cost functions adapted to fit an objective of matching accuracy. Little research has been focusing on automatically designing generic cost functions in a classification context.

Recent initiatives have been focusing on the proposal of graph representation based on heat-kernel embeddings [142, 141], deep-learning methods [9], quantum walk [10], and generative models [52]. Some of them are detailed below.

Xiao et al. [142] proposed the characterization of the properties of a graph by means of

the flow of information across edges. The rate of flow is computed through the Laplacian of the graph. They explored three approaches computed from the heat kernel matrix: zeta function of the heat kernel trace, derivative of the zeta function, and heat-content invariants. Xiao et al. [141] also exploited a heat-kernel formulation based on the Laplacian graph transformation. They presented an embedding scheme to construct a generative model for graph structure. They mapped the nodes of the graphs as points in a vector space, and then computed the correspondence matrix between these points with the Scott and Longuet-Higgins alignment algorithm. Later, they captured any variations in the graph structure through a covariance matrix of the embedding points, so they can construct a point-distribution model using the eigenvalues and eigenvectors of this matrix. This model can be used to measure the distance between a pair of graphs.

Bai et al. [10] developed new graph kernels where the graph structure is examined by means of discrete-time quantum walk. They simulated the evolution of the quantum-walk on each graph, computing their associated density matrix. Later, for a pair of graph, they compute the kernel by the negative exponential of Jensen-Shannon of their density matrix, using a minimum spanning tree of a sparser version of the original graph. Han et al. [52] focused on the problem of representing graphs by edge connectivity. They aimed to learn a generative model to describe the distribution of structural variations present in graphs. Their proposal learns a generative supergraph by the probability distribution over the occurrence of nodes and edges. They encoded the complexity measurement using a Von Neumann entropy, and later they used an EM algorithm to minimize the criterion of correspondence between graphs.

Bai et al. [9] proposed a work to combine graph complexity measures and deep learning networks. Their goal is to compute a representation for each vertex. Later, a single graph feature vector is computed by averaging vertices' representations. For that, they first decompose the graph structure into a family of expansions subgraphs rooted at a vertex, and measured the entropy-based complexities, which is used to build the complexity trace, i.e., the depth-based representation of the root vertex. Next, they perform a clustering using k-means to find prototype representations, which are used to train a deep neural network.

In this chapter, we propose to learn a discriminative cost function between the nodes of graphs with no restriction on the graph type, nor on labels for a classification task. On a training set of graphs, a feature vector is extracted from each node of each graph, describing local information on the nodes. Node dissimilarity vectors are obtained by comparing pairs of feature vectors and labeled according to the node pair belonging to graphs of the same class or not. On this basis, a classifier is trained on these node dissimilarity vectors. At the decision stage, when comparing two graphs, a new pair of nodes is given as an input of the classifier, and the class membership probability is output. We use these adapted costs to fill a node-to-node similarity matrix, which encodes our learned matching costs. Based on these costs, we reduce the graph matching problem to a Linear Sum Assignment Problem (LSAP) between the nodes of two graphs. The LSAP aims at finding the maximum weight matching between the elements of two sets and this problem can be solved by the Hungarian algorithm [67] in $O(n^3)$ time. Instead of dealing with the graphs as a whole, we exploit their elements (e.g., their node attributes) to guide

the weight learning process. Thus, as we increase the number of elements that we use for learning, we can take advantage of only a few graphs in the training process. Our method is, therefore, suitable for problems, which handle small-size training sets, either because they are difficult to obtain, or hard to label.

This chapter extends the work presented in [138], by providing a theoretical overview of the introduced graph distance learning framework, as well as by detailing performed experiments related to the parametric evaluation of the proposed approach. We also present an original approach based on an open-set recognition problem formulation, in which the training step does not contain all classes because they are ill-sampled or unknown [112]. The goal is to learn the costs to match nodes from different graphs. The method is based on node-signatures, dissimilarities between node-signatures, a classifier to determine a cost matrix, and a Hungarian algorithm to compute similarities between graphs. Furthermore, this chapter presents and discusses for the first time experiments related to the use of open-set classifiers in weight-learning problems associated with graph-classification tasks. To the best of our knowledge, this is the first work to perform such evaluation in the open-set scenario. Finally, another novelty of this work relies on the investigation of complex network measurements in the characterization of local properties of graphs.

Open-set scenario, differently from the closed-set scenario, does not have, *a priori*, training samples for all classes, as these classes might appear in the testing step [81]. Open-set classifiers consider that not all classes are known *a priori* at training time. Therefore, a test sample can belong to a class from the training or it can belong to a class not “seen” during training, i.e., this sample can be considered as “*unknown*.” In this chapter, we take advantage of this formulation by mapping the distance vector related to nodes belonging to different classes as “*unknown*.” By doing that, learned cost functions are expected to encode more properly existing relations among nodes of vertices of the same class, leading to more discriminant graph matching.

6.2 Graph Distance Learning Framework

We propose a new framework to learn a discriminative cost function for computing the bipartite graph edit distance between two graphs. In our method, we describe each graph from a training set using a local descriptor. We extract feature vectors from each node of each graph. Next, we compute node dissimilarity vectors pair-wisely, generating feature vectors. These node dissimilarity vectors are then labeled according to the pair of nodes. If the pair of nodes belongs to the same graph class, the dissimilarity vector received the same label; if not, it is labeled as belonging to an “*unknown*” class. Later, a distance learning classifier is trained according to the distance vectors. At the decision phase, a graph from the testing set is compared to a graph from the training set. All its nodes are described by a local descriptor and the dissimilarity vector is computed between test and training samples. These vectors are the input to the distance learning classifier, which returns the class membership probability. These probabilities are the adapted costs used to fill a node-to-node similarity matrix between the two graphs. We use these learned matching costs to approximate the problem of matching graphs as a

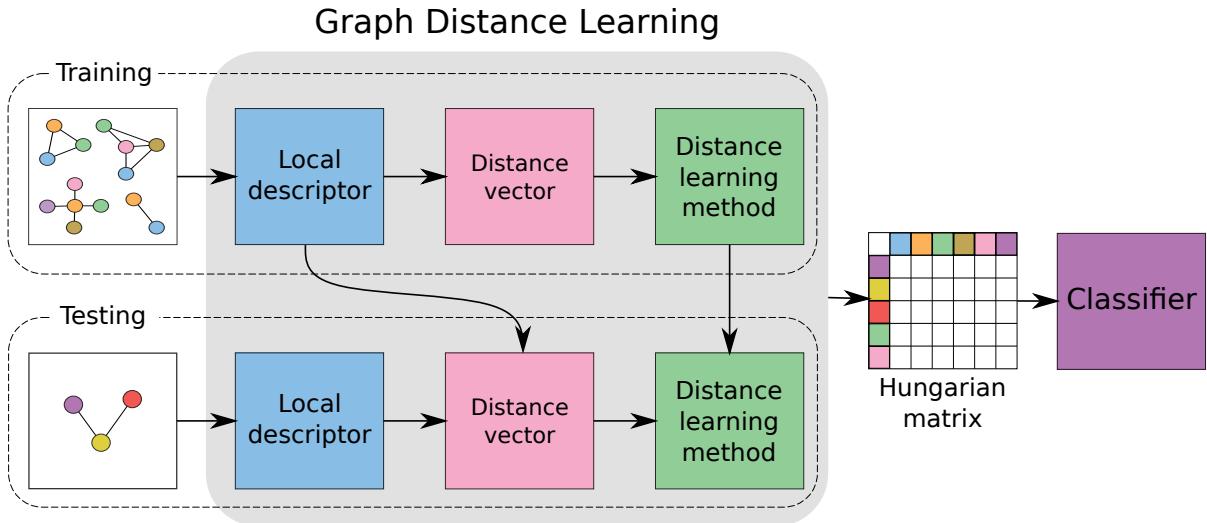


Figure 6.2: **Schematic overview of the Graph Distance Learning framework.** In our framework, the graphs goes through a local descriptor, following a distance vector step, and then a distance learning method, which will perform the learning step to populate the Hungarian matrix.

Linear Sum Assignment Problem (LSAP) between the nodes of two graphs. The LSAP, which aims to find the minimum cost matching between elements of two sets, can be solved by the Hungarian Algorithm [67] in $O(n^3)$ time. Figure 6.2 shows a schematic view of the proposed Graph Distance Learning framework. In the following, we describe each component of this framework.

6.2.1 Local descriptor

To describe the graphs of the training and testing sets, we propose the use of local descriptors to characterize local properties of all graph nodes. Then, we can compare them pair by pair, and calculate the matching cost to transform a set of nodes from one graph to the set of nodes of the other graph.

Given a general graph $G = (V, E)$, a local description is defined as:

$$\Gamma(G) = \{\gamma(v) \mid \forall v \in V\}, \quad (6.1)$$

where $\gamma(v)$ is a local descriptor which encodes local properties of vertex v into a vector.

6.2.2 Distance vector

Our proposed approach for graph matching consists in finding a minimum distance to transform a local description from one graph into a local description from another graph. To perform that, we use a function to calculate the distance between two local descriptors.

Let G_I and G_J be two graphs, v_i and v_j two nodes from these graphs, and $\gamma(v_i)$ and $\gamma(v_j)$ be two local descriptions of these nodes. We define a function \mathcal{E} that, using $\gamma(v_i)$ and $\gamma(v_j)$ as inputs, returns a feature vector (d) representing the distance between these

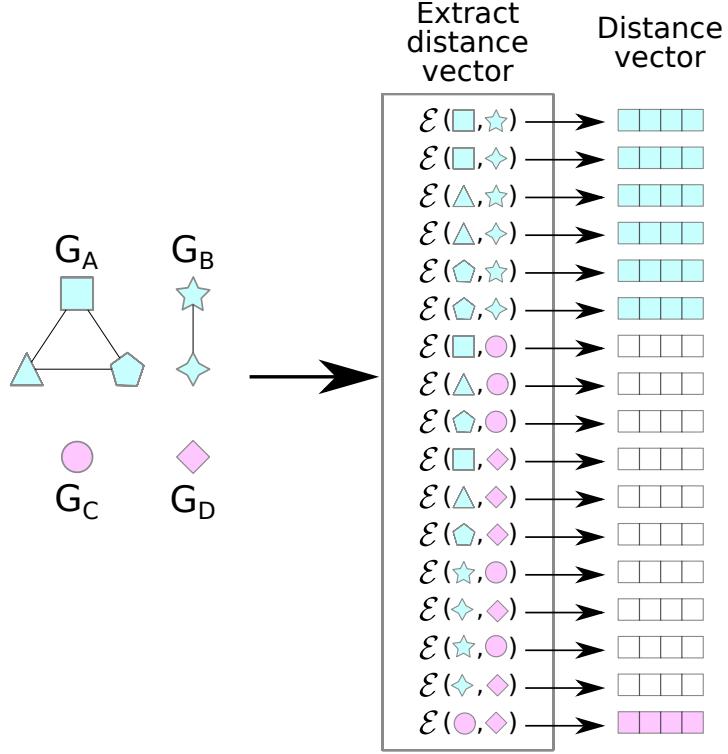


Figure 6.3: **Illustration of the creation of a distance vector based on node properties of four graphs.** When the nodes belong to graphs of the same class (same color – blue and pink distance vectors – in the figure), the distance vector receives the same label. Alternatively, when the nodes belong to graphs of different classes, the distance vector is labeled as “unknown” (white distance vectors).

two local descriptions.

$$\mathcal{E}(\gamma(v_i), \gamma(v_j)) = d_{ij} \quad (6.2)$$

To each distance vector d_{ij} , we assign a class label defined in set \mathcal{L} . The set containing possible labels (classes) is defined as:

$$\mathcal{L}(d_{ij}) \subset \mathcal{L}(G_I) \cup \mathcal{L}(G_J) \cup \{\text{unknown}\} \quad (6.3)$$

Figure 6.3 illustrates the computation of distance vectors based on the properties of vertices belonging to four graphs (graphs G_A , G_B , G_C , and G_D in the figure) and their labeling process.

6.2.3 Distance learning component

This component of our Graph Distance Learning framework is responsible for learning a cost value related to each distance vector received as input. We propose this component as a function \mathcal{F} , in which we obtain the probability of the desired class:

$$\mathcal{F} : \mathcal{D} \rightarrow \mathbb{R}^{|\mathcal{L}|} \quad (6.4)$$

where \mathcal{D} is the set of all distance vectors computed from vertices of two input graphs.

Hungarian matrix and classification

After we obtain the cost output from the distance learning method, we use these values to populate a cost matrix relative to the combination of each testing graph with each graph from the training set. The cost matrix contains the local description from one testing graph in the rows and the local description from one training graph in the columns. Thus, each entry of the matrix is the cost to transform the description from the row to the description of the column. Thus, the Hungarian algorithm finds the minimum cost assignment between the two sets of signatures.

Finally, the test sample is classified using the k-nearest neighbor (kNN), where the similarity between two graphs is defined by the Hungarian algorithm.

6.3 Graph Distance Learning Implementation

In this section, we provide an instantiation of the proposed framework, detailing implementation choices.

6.3.1 Local Descriptor

To describe local information of the graphs in this work, we use information of a graph and their nodes following the node signature:

$$\gamma(v) = \{\alpha_v^G, \theta_v^G, \Delta_v^G, \Omega_v^G\}, \quad (6.5)$$

where $G = (V, E)$ is a graph defined by vertices in V and edges in E , $v \in V$, and α_v^G , θ_v^G , Δ_v^G , and Ω_v^G are, respectively, the attributes of the node v , the degree of node v , the set of degrees of adjacent nodes to v , and a set of attributes of the incident edges of v [138, 62].

In this chapter, we also investigate the use of complex network measurements in the characterization of graph local properties. We selected the following complex network measurements:

- Vulnerability (V_n), which presents the difference in performance when the node is removed from the graph [51]: $V_v = \frac{E-E_v}{E}$, where E is the global efficiency of the graph, and E_v is the global efficiency after the removal of node v ;
- Clustering coefficient (C_v), which is the fraction of possible triangles that exist including the node [135]: $C_v = \frac{N_\Delta(v)}{N_3(v)}$, where $N_\Delta(v)$ is the number of triangles with node v and $N_3(v)$ is the number of connected triples with v as central node;
- Cyclic coefficient (Θ_v), which measures how cyclic a graph is, defined by the average of the inverse of the sizes of the smallest cycles formed by the node and its neighbors [66]: $\Theta_v = \frac{2}{n_v(n_v-1)} \sum_{w>u} \frac{1}{S_{uvw}} a_{uv} a_{vw}$, where n_v is the number of neighbors of node v , S_{uvw} is the size of smallest circle that passes through nodes u, v, w , and a_{uv} are the elements of adjacency matrix;

- Subgraph centrality (SC_v), which considers the number of subgraphs that constitute a closed walk starting and ending at the given node [44]: $SC_v = \sum_{k=0}^{\infty} \frac{(A^k)_{vv}}{k!}$, where $(A^k)_{vv}$ is the v th diagonal element of the k th power of adjacency matrix A , and $k!$ assures the convergence of the sum and that smaller subgraphs have more weight;
- the average neighbor degree [94]. The degree of a vertex v is the number of edges incident to v .

6.3.2 Distance vector

Our proposed approach for graph matching consists in finding a minimum distance to transform a node signature from one graph into a node signature from another graph. To perform that, we first need to define a function to calculate the distance between two node signatures, and in our case, a function that is capable of dealing with both numeric and symbolic attributes. We selected the *Heterogeneous Euclidean Overlap Metric* (HEOM) [139] which deals with these attributes, and adapted for our graph local descriptor.

The default HEOM distance function is defined as follow:

$$\text{HEOM}(d_i, d_j) = \sqrt{\sum_a \delta(d_{ia}, d_{ja})^2} \quad (6.6)$$

for d_i and d_j two heterogeneous feature vectors, where a is each attribute of the vector. $\delta(d_{ia}, d_{ja})$ is also defined as:

$$\delta(d_{ia}, d_{ja}) = \begin{cases} 1 & \text{if } d_{ia} \text{ or } d_{ja} \text{ is missing,} \\ 0 & \text{if } a \text{ is symbolic and } d_{ia} = d_{ja}, \\ 1 & \text{if } a \text{ is symbolic and } d_{ia} \neq d_{ja}, \\ \frac{|d_{ia} - d_{ja}|}{\text{range}_a} & \text{if } a \text{ is numeric} \end{cases} \quad (6.7)$$

Considering the node signature local descriptor, we define the HEOM distance between two signatures as follow. Considering $A = (V_a, E_a)$ and $B = (V_b, E_b)$ two graphs, $v_a \in V_a$ and $v_b \in V_b$ nodes from these graphs. According to Equation 6.5, the node signatures of these nodes are: $\gamma(v_a) = \{\alpha_{v_a}^A, \theta_{v_a}^A, \Delta_{v_a}^A, \Omega_{v_a}^A\}$ and $\gamma(v_b) = \{\alpha_{v_b}^B, \theta_{v_b}^B, \Delta_{v_b}^B, \Omega_{v_b}^B\}$. Then, the distance ϵ between two node signatures is:

$$\begin{aligned} \epsilon(\gamma(v_a), \gamma(v_b)) = & \text{HEOM}(\alpha_{v_a}^A, \alpha_{v_b}^B) + \text{HEOM}(\theta_{v_a}^A, \theta_{v_b}^B) + \\ & \text{HEOM}(\Delta_{v_a}^A, \Delta_{v_b}^B) + \\ & \frac{\sum_{i=1}^{|\Omega_{v_a}^A|} \text{HEOM}(\Omega_{v_a}^A(i), \Omega_{v_b}^B(i))}{|\Omega_{v_a}^A|} \end{aligned} \quad (6.8)$$

The goal of the Graph Matching Learning framework is to learn the edit distance between two graphs. For that, we need to define the *distance vector* that will be used in the cost learning process [138]. The function \mathcal{E} , which defines the *distance vector*, is

based on the ϵ function. Instead of summing the distance of all attributes, \mathcal{E} considers each attribute distance as a bin of the vector. Therefore, we can present the function \mathcal{E} as:

$$\begin{aligned} \mathcal{E}(\gamma(v_a), \gamma(v_b)) &= [HEOM(\gamma(v_a)_i, \gamma(v_b)_i)], \\ \forall i \in \{0, \dots, |\gamma(v)|\} \mid \gamma(v)_i &\text{ is a attribute of } \gamma(v). \end{aligned} \quad (6.9)$$

Using complex network measures, the node signature is defined as:

$$\gamma(v_a) = \{\alpha_{v_a}^A, \theta_{v_a}^A, \Delta_{v_a}^A, \Omega_{v_a}^A, V_{v_a}^A, C_{v_a}^A, \Theta_{v_a}^A, SC_{v_a}^A, AVG_{v_a}^A\},$$

and Equation 6.9 is adapted accordingly.

Later, we label these distance vectors to guide our learning process. We proposed the following formulation [138]. Let $Y = \{y_1, y_2, \dots, y_l\}$ be a set of l labels associated with the graphs according to the target graph classification problem. In this formulation, a label y_i is assigned to each distance vector built based on the node signatures of graphs belonging to the same class y_i . On the other hand, when a distance vector is built from node signatures of graphs belonging to different classes, an “unknown” label (e.g., y_{i+1}) is adopted (see Figure 6.3).

6.3.3 Distance Learning Component

In this chapter, we present two proposals for learning the graph edit distance between two graphs, using closed-set and open-set formulations.

Figure 6.4 illustrates graph classification tasks from both the closed-set and open-set perspectives. The test sample is the “X”-shaped graph. From the closed-set perspective, the test graph is labeled as belonging to the purple class, i.e., all test samples will receive one of the labels considering at the training stage. On the other hand, from the open-set perspective, the same test set is labeled as “unknown”, i.e., test samples, which are not “close” enough to labeled samples seen at training stage, are considered to belong to an “unknown” category.

Closed-set Formulation

The first approach, the closed-set one, aims to learn how to classify the distance vectors obtained in the previous step. For that, after obtaining the pairwise distance vectors, the vectors from the training set are used to learn a classifier. In this work, we learn the Support Vector Machine (SVM) margin that separates samples of the training set from different classes.

With the margin, we can predict the classes of the graphs in the testing set. First, we extract the local descriptor of each graph of the testing set. Next, we compute the distance vectors considering the node signatures from the test graph with the node signatures from the graphs of the training set. These vectors are projected into the learned feature space and we obtain the probability of a test sample belongs to the training set classes

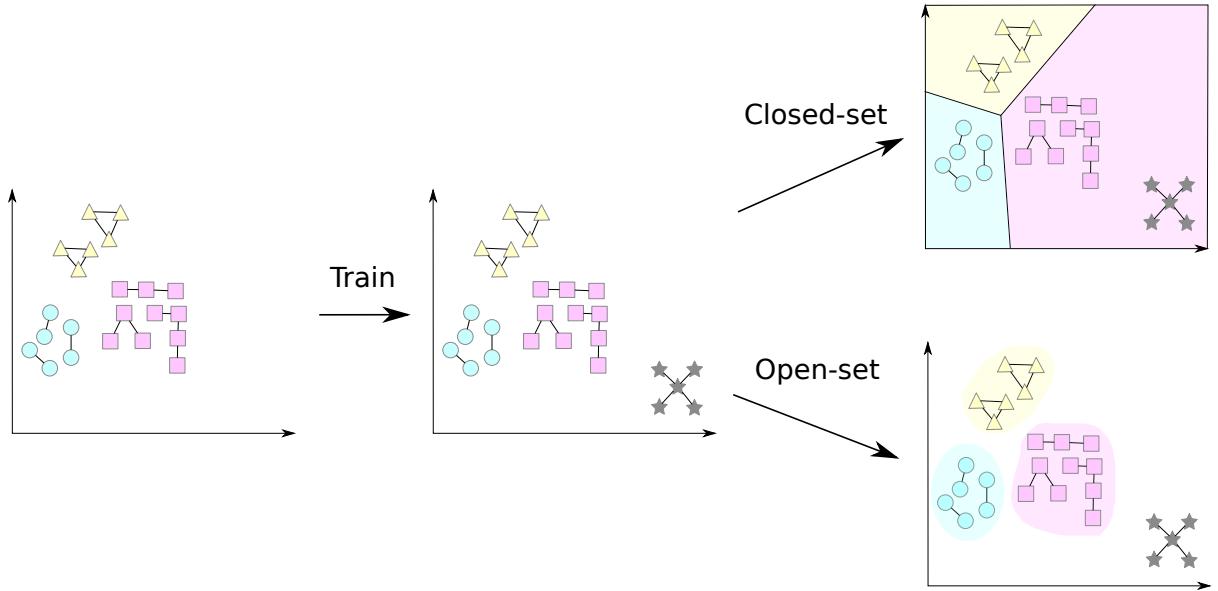


Figure 6.4: **Differences in the classification of the “X”-shaped test graph from the closed set (upper) and open-set (bottom) approaches.** From the closed-set perspective, the test graph is labeled as belonging to the purple class. For the open-set perspective, the same test set is labeled as “unknown”.

considering the SVM separation hyperplane.

Open-set Formulation

Our second approach is based on an open-set formulation, in which we can classify as “unknown” samples that do not belong to the different class available during the training step.

Scheirer et al. [112] presented a formalization for recognition problems from the open-set perspective. This formalization aims to find a function f , which minimizes the combination of the open space risk $R_{\mathcal{O}}$ and the empirical risk $R_{\mathcal{E}}$, the later regularized by a constant λ_r :

$$\operatorname{argmin}\{R_{\mathcal{O}}(f) + \lambda_r R_{\mathcal{E}}(f)\} \quad (6.10)$$

In this chapter, we investigate the use of two recently proposed open-set-based learning methods [81]: Open-Set Nearest Neighbors 1 (OSNN1) and Open-Set Nearest Neighbors 2 (OSNN2).

In the OSNN1 method, during the prediction phase, the two training-set nearest neighbors (s and u) of an input test sample t are selected. If they have the same label, this label is assigned to the test sample, otherwise, the test sample is unknown, i.e., to the test sample the *unknown* label is assigned.

The OSNN2, in turn, labels an input test sample t as follows: it first finds the two training-set nearest neighbors of different labels (s and v , being s the nearest), and then

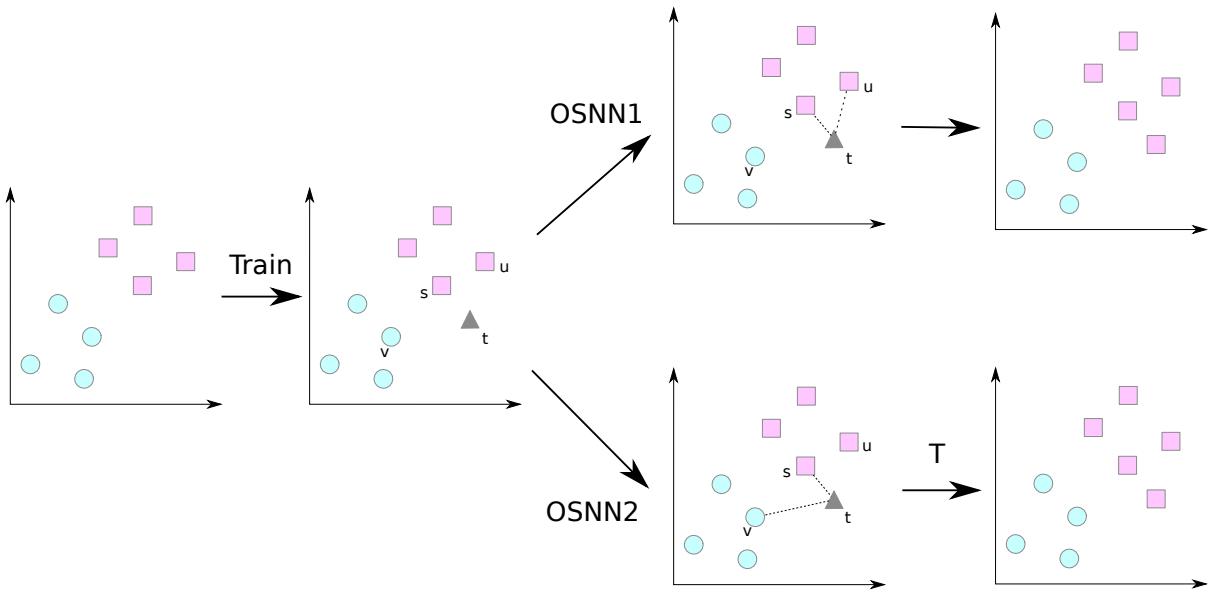


Figure 6.5: Differences between OSNN1 and OSNN2 open-set recognition approaches when selecting training neighbors. The OSNN1 approach considers the two closest training neighbors. If they are from the same class, the test sample (in black) is labeled as belonging to this class, otherwise, as “unknown”. For the OSNN2 approach, the two nearest neighbors from different classes are selected, and if the ratio of the distances to them is below a threshold defined in the training step, the test sample is labeled with the label of the closest class. Otherwise, it is labeled as “unknown”.

calculates the ratio

$$R = \frac{d(t, s)}{d(t, v)} \quad (6.11)$$

and assign the label according to the following condition:

$$\text{label}(t) = \begin{cases} \text{label}(s), & \text{if } R \leq \text{threshold}, \\ \text{unknown}, & \text{if } R > \text{threshold}. \end{cases} \quad (6.12)$$

Figure 6.5 shows the difference between the two open-set approaches when selecting the closest training neighbors.

6.4 Experiments

In this section, we present the research questions addressed in our experiments, the datasets used, and the adopted evaluation protocol adopted for each research question.

6.4.1 Datasets

We select traditional and widely used datasets of the literature to perform our experiments. Each dataset is detailed in Table 6.1.

Table 6.1: Details of the datasets used in these experiments.

Dataset	# graphs	# classes	Protocol
MAO	68	2	Leave-One-Out
PAH	94	2	10-fold
GREC	1100	22	286 training + 286 validation + 586 testing

- **MAO:** Monoamine Oxidase dataset¹ is a dataset that consists of 68 molecules, with 38 molecules that inhibit the monoamine oxidase and 30 that do not. The standard evaluation protocol adopted for this dataset relies on a Leave-one-out cross-validation, where 67 graphs are used for training and the remaining sample is used for testing.
- **PAH:** The Polycyclic Aromatic Hydrocarbons dataset [49] is composed of 94 graphs representing molecules composed only of carbon atoms. All bound in these molecules are aromatics. The typical evaluation protocol used for this dataset relies on a 10-fold cross-validation procedure. In this protocol, we have ≈ 84 graphs per fold.
- **GREC:** The GREC dataset [105] consists of graphs representing architectural and electronic drawings. The nodes are ending points in the drawings, and the graph edges are the lines and arcs. It contains 1100 graphs divided into 22 classes. The default evaluation protocol of this dataset consists of 286 graphs for training, 286 graphs for validation, and 528 graphs for testing.

6.4.2 Research Questions and Experimental Protocol

In this work, we use different experiment protocols for addressing each research question.

Q1 *What is the impact of the training set size and normalization procedures in the effective performance of the evaluated learning methods?*

In the first question, we want to assess the robustness of the different learning methods with regard to different parameter setting. Recall that our Graph Distance Learning framework relies on the computation of multiple pairwise distance vectors, being therefore computationally costly. We decided, then, to perform experiments using only a subset of the available training sets in our parameter setting investigation. In order to assess the effective performance of the methods for different training set sizes, let s be the number of graphs per class. We vary s in the set $\{2, 5, 10, 20\}$. Also, we use only 10% of the available testing set. The graphs used for training and testing were defined randomly. Our reported results refer to the average effective performance, considering 20 runs using the different randomly selected samples. We also want to assess the impact of different normalization strategies on the effectiveness performance of the evaluated methods. We used the min-max normalization, in which the vectors are normalized between 0 and 1

¹<https://brunl01.users.greyc.fr/CHEMISTRY/index.html> (As of Jan. 2019).

according to minimum and maximum values observed; and the zscore normalization, in which we use the mean and standard deviation to normalize distance score values. In the Graph Distance Learning method, the selected parameters for the SVM closed-set approach was the default ones (RBF kernel with $C = 0$). Open-set approaches OSNN1 and OSNN2 do not have any parameters to setup. Experiments related to Q1 considered the MAO and PAH datasets, and effectiveness results refer to the average normalized accuracy in the graph classification problems defined for each dataset.

Q2 *Which learning method leads to better effectiveness performance?*

Our goal here is to compare the open-set formulations with the SVM-based closed-set solution in the weight cost learning problem. Our evaluation regarding Q2 also considers the use of complex network measurements in the characterization of graph local properties. The experimental protocol is similar to the one described in the previous item. The differences are: we only use the variations of the methods with the best performance observed in the previous experiments, an additional dataset (GREC) is used in our comparisons.

Q3 *How effective are the proposed methods when compared to state-of-the-art solutions?*

Our goal here is to demonstrate that the proposed learning methods yield better or comparable results to those observed for state-of-the-art baselines for different datasets. In order to compare our approach with baselines in the MAO dataset, we consider the evaluation protocol usually employed in the assessment of methods using this dataset (see Section 6.4.1). We also perform experiments to compare the performance of the incremental increase in the size of the training set.

6.5 Results and Analysis

6.5.1 Q1: Impact of normalization and the size of training sets

Figure 6.6 presents the results observed for the evaluated weight learning methods, considering different normalization strategies (e.g., min-max, and zscore). In this figure, we also assess the robustness of the method with regard to the size of the training set size. The first and the second lines of Figure 6.6 refer to the MAO and the PAH datasets, respectively. Good results are obtained with just a few graph examples from the training set and as we can observe 10 graphs per class is a good compromise for the open-set methods. Related to the normalization, the min-max normalization obtained the overall best results in our experiments, thus, we will be using this normalization in the next experiments.

6.5.2 Q2: Identification of the best learning methods

Table 6.2 presents the best results observed for the SVM, OSNN1, and OSNN2 learning methods for the MAO, PAH, and GREC datasets, considering only 10 randomly selected graphs for training. As we can observe, the OSNN2 classifier obtained the best accuracy score considering all datasets. As the OSNN2 classifier considers the distance relation

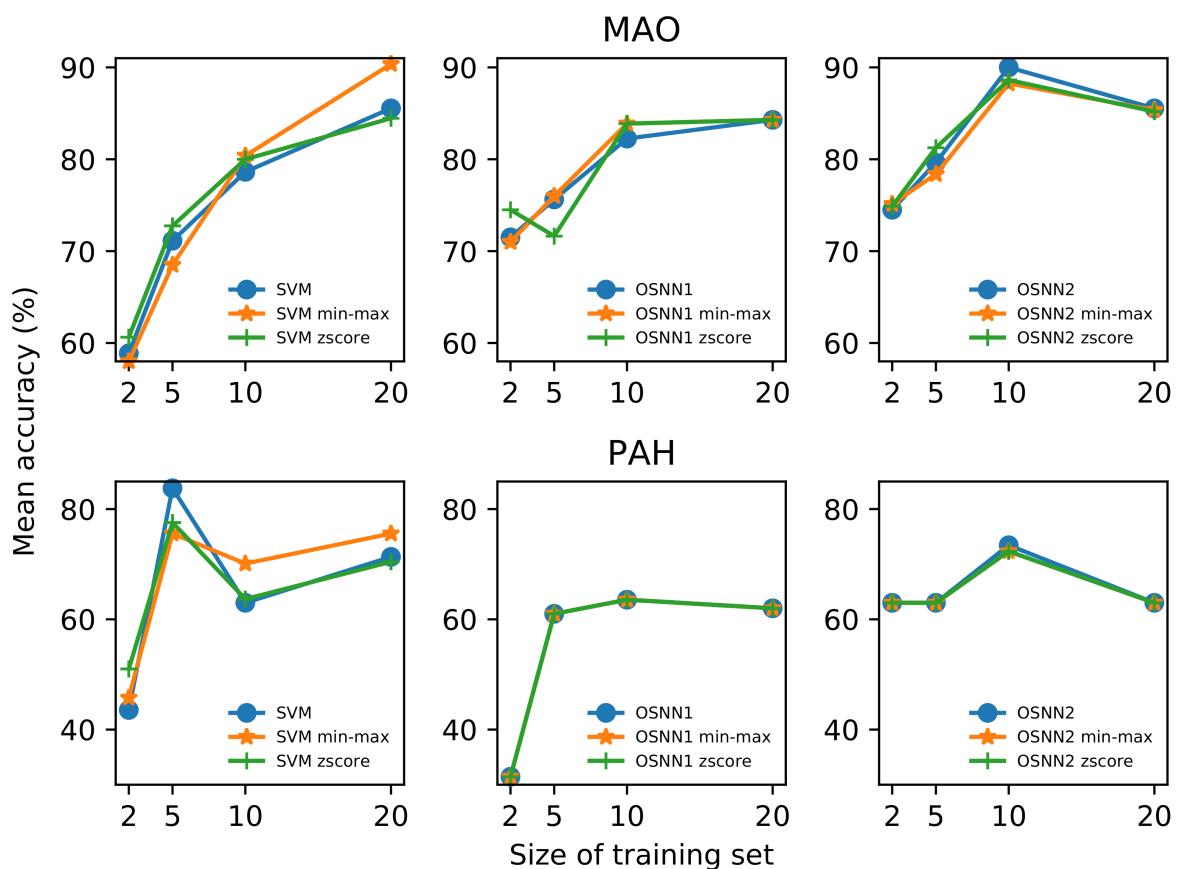


Figure 6.6: Evaluation of the different weight learning strategies with regard to the use of normalization procedures and different training set sizes.

Table 6.2: Best results observed for the different weight learning strategies in terms of normalized accuracy. In all cases, 10 graphs are used for training.

	MAO	PAH	GREC
SVM	80.38	70.11	23.52
OSNN1	83.88	63.56	56.25
OSNN2	88.25	72.33	58.98

Table 6.3: Best results observed for the different weight learning strategies in terms of normalized accuracy, **considering the use of complex network measurements** in the characterization of graph local properties. In all cases, 10 graphs are used for training.

	MAO	PAH	GREC
SVM	79.13	55.33	44.20
OSNN1	90.13	93.67	49.66
OSNN2	95.38	84.11	73.52

between two classes, it can have a better separation of the classes, leading to a high accuracy score.

We also performed some experiments in which we consider the use of complex network measurements in the local properties of the graph. Table 6.3 shows that improving the local representation of the nodes, the overall accuracy increases, especially for the OSNN2 classifier.

6.5.3 Q3: Comparison with state-of-the-art baselines

In this comparison with the state of the art, we perform a few experiments considering the same evaluation protocol used in the literature, and a simple modification using fewer graphs per training. We also just present the results without Table 6.4 presents the obtained results of our solution and state-of-the-art approaches in the MAO dataset. We have slightly modified the leave-one-out protocol to assess the impact of different training set sizes. OSNN2($X-Y$), in the table, refers to the use of the OSNN2 method, training with randomly selected X samples of class 0 and randomly selected Y samples of class 1, performed 10 times. As we can see, our results have not yet beaten the state of the art, but it comes as a close fourth best using only 17 graphs per class in the training set. Our result with all graphs of the training is a little further in the table. This happens mainly because our approach to find the combination of all node signatures results in an overtraining for our classifier, because of the unbalance of the training classes. However, as we can see in Table 6.2 and 6.3, we can achieve close or better results using fewer graphs for training.

6.5.4 Computational complexity and runtimes

Let n be the number of training graphs and v_n the total number of vertices in the training graphs. Similarly, let m be the number of testing graphs and v_m , the total number of vertices in the testing graphs.

Table 6.4: Comparison of our approach with the same evaluation protocol defined in [48] using the MAO dataset.

	MAO
El-Atta et al. [43]	98.5
Mahé et al. [79] [48]	96
Gaüzère et al. Treelet Kernel [48]	94
OSNN2 (17-17)	92.65
OSNN2 (15-15)	91.12
Riesen et al. [107] [48]	91
Neuhaus and Bunke [87] [48]	90
Gaüzère et al. Normalized Graph Laplacian Kernel [48]	90
Gaüzère et al. Normalized Fast Graph Laplacian Kernel [48]	90
OSNN2 (18-18)	89.71
OSNN2 (10-10)	88.24
OSNN2 (38-30)	83.82
Vishwanathan et al. [131] [48]	82
Suard et al. [122] [48]	80
OSNN2 (5-5)	76.47

At the training phase, the computation complexity of the proposed method depends on the (a) computation of the vertex feature vector representation (local descriptor computation); (b) computation of the distance vectors; and (c) the distance learning. The computational costs of each step of the training phase can be defined as:

- (a) Local Descriptor computation: $O(v_n)$;
- (b) Distance Vector computation: $O(v_n^2)$;
- (c) Distance Learning method: $O(v_n^4)$ as pointed out in [100] for the SVM classifier (closed scenario).

The worst case complexity for training is, therefore, $O(v_n^4)$

The test phase comprises (a) the local descriptor computation for the test set; (b) computation of distance vectors considering test and training graphs; (c) population of the Hungarian matrix using the trained classifier; and (d) computation of the Hungarian Algorithm. The computational costs of each step of the test phase can be defined as:

- (a) Local Descriptor computation: $O(v_m)$;
- (b) Distance Vector computation: $O(v_n \times v_m)$;
- (c) Population of the Hungarian matrix: $O(c \times n \times m)$, where c stands for the probability score computation cost defined by the classifier.
- (d) Computation of the Hungarian Algorithm: the Hungarian algorithm computation takes $O(p^3)$ where p is the maximum dimension of the input Hungarian matrix [69]. As this computation is performed $n \times m$, the worst complexity is $O(n \times m \times p^3)$.

Table 6.5: Mean runtimes of each iteration in the MAO dataset with the Leave-One-Out protocol.

Method	Runtime (s)
OSNN2 (17-17)	2680 ± 439
OSNN2 (18-18)	3607 ± 671
OSNN2 (20-20)	5384 ± 458
OSNN2 (38-30)	$74\,391 \pm 2029$

Considering the complexity calculated above, we present a few runtimes of our experiments. Table 6.5 shows the mean runtimes of each iteration in the MAO dataset with Leave-One-Out protocol.

Our proposed approach is somewhat costly because it considers the local descriptions of the graphs to learn the Hungarian matrix cost function. Also, the computation of the Hungarian algorithm itself is quite expensive.

6.6 Conclusions

In this work, we introduced new approaches to learn discriminative costs for a bipartite graph edit distance computation between two graphs. We present a generic framework, and then we describe different methods, based on both closed-set and open-set learning paradigms, used to implement the proposed framework. To the best of our knowledge, this is the first work to model the cost function learning process as an open-set recognition problem. Another novelty of this work relies on the investigation of complex network measurements in the characterization of graph local properties, aiming to obtain more effective cost function matrices. Performed experiments considered widely used datasets and evaluation protocols. Achieved results demonstrate that the proposed framework is effective, leading to comparable and better effectiveness results in different graph classification problems when compared with several baselines. One positive property of our solution relies on its capacity of leading to effective results, even when only a few samples (≈ 10 graphs) are used for training.

In our future work, we intend to deepen the investigations of the use of other complex network measurements in the local characterization of graph properties [33]. We also plan to extend our investigation regarding the use and combination of other open-set recognition approaches [84].

Chapter 7

Conclusions

Many real-world situations can be described using graphs, which have been successfully employed in several applications such as bioinformatics, social network analyses, and databases. The wide spread use of graphs is motivated by the fact that this is a powerful representation, as it allows for encoding relationships not only among objects, but also among their components under a single formalism.

In this thesis, we focused on graph-based approaches to represent and to perform matching between objects. For the representation, we proposed two different approaches, a graph-based image representation and a graph-based multimodal representation. Considering the object matching, we proposed a learning-graph-matching method, presenting a novel framework to learn the cost functions in the graph matching.

In the following sections, we present a summary of the contributions achieved in this thesis, a list of opportunities of investigation in future work, and the papers co-authored during PhD work.

7.1 Summary of Contributions

In this section, we enumerate the achieved contributions of this thesis, referring to the chapter in which we presented it.

Regarding the first question, we proposed an application of the Bag-of-Visual-Graphs method to the scenario of Remote Sensing Images. We combined a color representation with a texture representation, considering the spatial relationship between different interest points. We obtained effective results in two datasets of the literature, as shown in Chapter 3.

The second question regards the combination of different features and/or modalities in the representation of objects. For this question, we proposed two new approaches to combine multiple modalities/representations of multimedia objects using graphs. We first build a graph connecting the representation of different modalities of the object according to our approach and then use the Bag-of-Graphs method to generate a statistical representation of the object. In Chapter 4, we presented these original approaches and validate them in the flooding detection problem.

Our third question refers to the graph matching problem using a learning approach.

We presented in Chapter 5 an original approach in which we learn how to calculate a cost function of the edit distance to match two graphs. We use the combination of two node signatures in the learning process. This learning replaces a specialist in the definition of the costs.

The last question is associated with improvements in the graph matching learning process. We addressed this question by presenting a generic framework to learn discriminative costs for the computation of the edit distance between two graphs. We also shown two different implementations, one of them based on the open-set paradigm. We demonstrated the use of an open set formulation led to improved classification results. Another contribution refers to the use of complex network measurements in the characterization of local properties of graphs in the process of cost function learning. These contributions are presented in Chapter 6.

7.2 Future Work

Beyond the contributions of this work, we still can think some new opportunities of investigation about our work, for example:

- Application of our approaches in the cross-modal scenario. In several situation objects may not be complete, i.e., a modality may be missing. It would be a promising research venue to investigate how our approaches could be used/extended for such cases.
- Several other remote sensing image classification applications, ranging from crop recognition in multiclass classification problems [111] to vegetation type classification in near-surface images [5], may benefit from our bag-of-visual-graph formulation. We proposed this investigation as a promising research venue.
- Another promising research venue is the combination of our proposed bag approach with a hash-based embedding which can help speed up our processing time.
- Investigate the development of new representations by using spectral bands of hyperspectral and multi-spectral remote sensing images as multiple modalities, using our proposed graph-based approach.
- Perform a parametric evaluation of the parameters of both the Bag of KNN Graphs and Bag of Cluster Graphs;
- Regarding our method for learning cost function, one promising research venue would be the investigation of the characterization of graph properties with other complex network measurements [33].
- The fusion of description approaches in open-set recognition problems [84] could be investigated with the goal of improving the effectiveness of the proposed methods for learning cost functions.

7.3 Research Outcomes

In this section, we present the papers, accepted or submitted, elaborated during this thesis.

1. Fernanda B. Silva, **Rafael de Oliveira Werneck**, Siome Goldenstein, Salvatore Tabbone, Ricardo da Silva Torres: **Graph-based bag-of-words for classification**. *Pattern Recognition* 74: 266-285 (2018)
2. **Rafael de Oliveira Werneck**, Ícaro C. Dourado, Samuel G. Fadel, Salvatore Tabbone, Ricardo da Silva Torres: **Graph-Based Early-Fusion for Flood Detection**. *IEEE International Conference on Image Processing* 2018: 1048-1052
3. **Rafael de Oliveira Werneck**, Romain Raveaux, Salvatore Tabbone, Ricardo da Silva Torres. **Learning Cost Functions for Graph Matching**. In Xiao Bai, Edwin R. Hancock, Tin Kam Ho, Richard C. Wilson, Battista Biggio, and Antonio Robles-Kelly, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 345–354, Cham, 2018. Springer International Publishing.
4. **Rafael de Oliveira Werneck**, Romain Raveaux, Salvatore Tabbone, Ricardo da Silva Torres: **Learning Cost Function for Graph Classification with Open-Set Methods**. Submitted to *Pattern Recognition Letters*
5. Otávio Augusto Bizetto Penatti, **Rafael de Oliveira Werneck**, Waldir R. de Almeida, Bernardo V. Stein, Daniel V. Pazinato, Pedro Ribeiro Mendes-Junior, Ricardo da Silva Torres, Anderson Rocha: **Mid-level image representations for real-time heart view plane classification of echocardiograms**. *Computers in Biology and Medicine* 66: 66-81 (2015)
6. Daniel V. Pazinato, Bernardo V. Stein, Waldir R. de Almeida, **Rafael de Oliveira Werneck**, Pedro Ribeiro Mendes-Junior, Otávio Augusto Bizetto Penatti, Ricardo da Silva Torres, Fabio H. Menezes, Anderson Rocha: **Pixel-Level Tissue Classification for Ultrasound Images**. *IEEE Journal of Biomedical and Health Informatics* 20(1): 256-267 (2016)
7. Pedro Ribeiro Mendes-Junior, Roberto Medeiros de Souza, **Rafael de Oliveira Werneck**, Bernardo V. Stein, Daniel V. Pazinato, Waldir R. de Almeida, Otávio A. B. Penatti, Ricardo da Silva Torres, Anderson Rocha: **Nearest neighbors distance ratio open-set classifier**. *Machine Learning* 106(3): 359-386 (2017)
8. Thierry Pinheiro Moreira, Mauricio Lisboa Perez, **Rafael de Oliveira Werneck**, Eduardo Valle: **Where is my puppy? Retrieving lost dogs by facial features**. *Multimedia Tools and Applications* 76(14): 15325-15340 (2017)
9. Keiller Nogueira, Samuel G. Fadel, Ícaro C. Dourado, **Rafael de Oliveira Werneck**, Javier A. V. Muñoz, Otávio A. B. Penatti, Rodrigo Tripodi Calumby, Lin Li, Jefersson Alex dos Santos, Ricardo da Silva Torres: **Data-Driven Flood Detection using Neural Networks**. *MediaEval* 2017

10. Keiller Nogueira, Samuel G. Fadel, Ícaro C. Dourado, **Rafael de Oliveira Werneck**, Javier A. V. Muñoz, Otávio A. B. Penatti, Rodrigo Tripodi Calumby, Lin Tzy Li, Jefersson A. dos Santos, Ricardo da Silva Torres: **Exploiting ConvNet Diversity for Flooding Identification.** *IEEE Geoscience and Remote Sensing Letters* 15(9): 1446-1450 (2018)

Bibliography

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.
- [2] K. Ahmad, P. Konstantin, M. Riegler, N. Conci, and P. Holversen. CNN and GAN Based Satellite and Social Media Data Fusion for Disaster Detection. In *Working Notes Proc. MediaEval Workshop*, 2017.
- [3] S. Alghowinem. A Safer YouTube Kids: An Extra Layer of Content Filtering Using Automated Multimodal Analysis. In *Intelligent Systems and Applications*, pages 294–308, Cham, 2019. Springer International Publishing.
- [4] R. P. Allan and B. J. Soden. Atmospheric Warming and the Amplification of Precipitation Extremes. *Science*, 321(5895):1481–1484, 2008.
- [5] J. Almeida, J. A. dos Santos, B. Alberton, R. da S. Torres, and L. P. C. Morellato. Applying machine learning based on multiscale classifiers to detect remote phenology patterns in Cerrado savanna trees. *Ecological Informatics*, 23:49 – 61, 2014. Special Issue on Multimedia in Ecology and Environment.
- [6] K. Avgerinakis, A. Mountzidou, S. Andreadis, E. Michail, I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris. Visual and textual analysis of social media and satellite images for flood detection @ multimedia satellite task MediaEval 2017. In *Working Notes Proc. MediaEval Workshop*, 2017.
- [7] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453 – 465, 2013.
- [8] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [9] Lu Bai, Lixin Cui, Xiao Bai, and Edwin R. Hancock. Deep depth-based representations of graphs through deep learning networks. *Neurocomputing*, 336:3 – 12, 2019. Advances in Graph Algorithm and Applications.
- [10] Lu Bai, Luca Rossi, Lixin Cui, Zhihong Zhang, Peng Ren, Xiao Bai, and Edwin Hancock. Quantum kernels for unattributed graphs using discrete-time quantum

- walks. *Pattern Recognition Letters*, 87:96 – 103, 2017. Advances in Graph-based Pattern Recognition.
- [11] B. Basnyat, A. Anam, N. Singh, A. Gangopadhyay, and N. Roy. Analyzing Social Media Texts and Images to Assess the Impact of Flash Floods in Cities. In *International Conference on Smart Computing*, pages 1–6. IEEE, 2017.
 - [12] C. Berger, M. Voltersen, S. Hese, I. Walde, and C. Schmullius. Robust Extraction of Urban Land Cover Information From HSR Multi-Spectral and LiDAR Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(5):2196–2211, Oct 2013.
 - [13] B. Bischke, P. Bhardwaj, A. Gautam, P. Helber, D. Borth, and A. Dengel. Detection of Flooding Events in Social Multimedia and Satellite Imagery using Deep Neural Networks. In *Working Notes Proc. MediaEval Workshop*, 2017.
 - [14] B. Bischke, P. Helber, C. Schulze, S. Venkat, A. Dengel, and D. Borth. The Multi-media Satellite Task at MediaEval 2017: Emergence Response for Flooding Events. In *Proceedings of the MediaEval 2017 Workshop*, Ireland, 2017.
 - [15] J. A. Bondy and U. S. R. Murty. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2008.
 - [16] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data Fusion through Cross-modality Metric Learning using Similarity-Sensitive Hashing. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3594–3601, 2010.
 - [17] L. Brun, B. Gaüzère, and S. Fourey. Relationships between Graph Edit Distance and Maximal Common Unlabeled Subgraph. Technical report, GREYC - Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen, July 2012.
 - [18] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689 – 694, 1997.
 - [19] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245 – 253, 1983.
 - [20] H. Bunke, S. Günter, and X. Jiang. Towards Bridging the Gap between Statistical and Structural Pattern Recognition: Two New Concepts in Graph Matching. In *Proceedings of the Second International Conference on Advances in Pattern Recognition*, ICAPR '01, pages 1–11, London, UK, 2001. Springer-Verlag.
 - [21] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1048–1058, 2009.

- [22] A. Çarkacıoğlu and F. Yarman-Vural. SASI: a new texture descriptor for content based image retrieval. In *International Conference on Image Processing*, volume 2, pages 137–140, 2001.
- [23] A. Çarkacıoğlu and F. Yarman-Vural. SASI: a generic texture descriptor for image retrieval. *Pattern Recognition*, 36(11):2615–2633, 2003.
- [24] S. Chatzichristofis and Y. Boutalis. CEDD: Color and Edge Directivity Descriptor: A Compact Descriptor for Image Indexing and Retrieval. In *Computer Vision Systems*, pages 312–322. Springer, 2008.
- [25] S. Chatzichristofis and Y. Boutalis. FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval. In *Workshop on Image Analysis for Multimedia Interactive Services*, pages 191–196, May 2008.
- [26] S. Chatzichristofis, Y. Boutalis, and M. Lux. Selection of the Proper Compact Composite Descriptor for Improving Content based Image Retrieval. In *International Association of Science and Technology for Development*, volume 134643, page 064, 2009.
- [27] L. Chen, W. Yang, K. Xu, and T. Xu. Evaluation of Local Features for Scene Classification Using VHR Satellite Images. In *Joint Urban Remote Sensing Event*, pages 385–388, April 2011.
- [28] M. Chica-Olmo and F. Abarca-Hernández. Computing geostatistical image texture for remotely sensed data classification. *Computers & Geosciences*, 26(4):373 – 383, 2000.
- [29] M. Cho, K. Alahari, and J. Ponce. Learning Graphs to Match. In *IEEE International Conference on Computer Vision*, pages 25–32, 2013.
- [30] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.
- [31] X. Cortés and F. Serratosa. Learning graph-matching edit-costs based on the optimality of the oracle’s node correspondences. *Pattern Recognition Letters*, 56:22–29, 2015.
- [32] X. Cortés and F. Serratosa. Learning Graph Matching Substitution Weights Based on the Ground Truth Node Correspondence. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(2):1650005–1–1650005–22, 2016.
- [33] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56(1):167–242, 2007.
- [34] O. Csillik. Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels. *Remote Sensing*, 9(3):1–19, 2017.

- [35] M. Dalla Mura, S. Prasad, F. Pacifici, P. Gamba, J. Chanussot, and J. A. Benediktsson. Challenges and Opportunities of Multimodality and Data Fusion in Remote Sensing. *Proceedings of the IEEE*, 103(9):1585–1601, Sep. 2015.
- [36] J. G. Daugman. Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.
- [37] T. De Groot. Flood monitoring and mapping using passive microwave remote sensing in Namibia. *Geomatics, Natural Hazards and Risk*, 1(1):19–35, 2010.
- [38] J. P. M. de Sa. *Pattern Recognition: Concepts, Methods, and Applications*. Springer Science & Business Media, 2001.
- [39] J. A. dos Santos, F. A. Faria, R. da S. Torres, A. Rocha, P. H. Gosselin, S. Philipp-Foliguet, and A. Falcão. Descriptor Correlation Analysis for Remote Sensing Image Multi-Scale Classification. In *International Conference on Pattern Recognition*, pages 3078–3081, Nov 2012.
- [40] J. A. dos Santos, O. A. B. Penatti, R. da S. Torres, P. H. Gosselin, S. Philipp-Foliguet, and A. Falcão. Improving Texture Description in Remote Sensing Image Multi-Scale Classification Tasks By Using Visual Words. In *International Conference on Pattern Recognition*, pages 3090–3093, Nov 2012.
- [41] J. A. dos Santos, O. A. B. Penatti, P. H. Gosselin, A. X. Falcaão, S. Philipp-Foliguet, and R. da S. Torres. Efficient and Effective Hierarchical Feature Propagation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4632–4643, Dec 2014.
- [42] J. A. dos Santos, O. A. B. Penatti, and R. da S. Torres. Evaluating the Potential of Texture and Color Descriptors for Remote Sensing Image Retrieval and Classification. In *Conference on Computer Vision Theory and Applications*, pages 203–208, 2010.
- [43] A. H. A. El-Atta and A. E. Hassanien. Two-class support vector machine with new kernel function based on paths of features for predicting chemical activity. *Information Sciences*, 403-404:42 – 54, 2017.
- [44] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71:056103–1–056103–9, May 2005.
- [45] M. Fauvel, J. Chanussot, and J. A. Benediktsson. A spatial-spectral kernel-based approach for the classification of remote-sensing images. *Pattern Recognition*, 45(1):381 – 392, 2012.
- [46] F. Feng, X. Wang, and R. Li. Cross-modal Retrieval with Correspondence Autoencoder. In *ACM International Conference on Multimedia*, MM ’14, pages 7–16, New York, NY, USA, 2014. ACM.

- [47] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [48] B. Gaüzère, L. Brun, and D. Villemin. Two new graphs kernels in chemoinformatics. *Pattern Recognition Letters*, 33(15):2038 – 2047, 2012. Graph-Based Representations in Pattern Recognition.
- [49] B. Gaüzère, L. Brun, and D. Villemin. Graph kernel encoding substituents’ relative positioning. In *International Conference on Pattern Recognition*, pages 637–642, Stockholm, Sweden, August 2014.
- [50] H. Ghassemian. A review of remote sensing image fusion methods. *Information Fusion*, 32:75 – 89, 2016.
- [51] V. Gol’dshtein, G. A. Koganov, and G. I. Surdutovich. Vulnerability and Hierarchy of Complex Networks. *arXiv preprint cond-mat/0409298*, 2004.
- [52] L. Han, R. C. Wilson, and E. R. Hancock. Generative graph prototypes from information theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2013–2027, Oct 2015.
- [53] M. Hashimoto and R. M. Cesar. *Graph-Based Representations in Pattern Recognition*, chapter Object Detection by Keygraph Classification, pages 223–232. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [54] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, June 2016.
- [55] C.-B. Huang and Q. Liu. An Orientation Independent Texture Descriptor for Image Retrieval. In *International Conference on Communications, Circuits and Systems*, pages 772–776, July 2007.
- [56] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–768, Jun 1997.
- [57] Y. Jia, M. Salzmann, and T. Darrell. Learning Cross-modality Similarity for Multinomial Data. In *IEEE International Conference on Computer Vision*, pages 2407–2414, Nov 2011.
- [58] L. Jin, K. Li, H. Hu, G. Qi, and J. Tang. Semantic Neighbor Graph Hashing for Multimodal Retrieval. *IEEE Transactions on Image Processing*, 27(3):1405–1417, March 2018.
- [59] N. Jin, C. Young, and W. Wang. GAIA: Graph Classification Using Evolutionary Computation. In *ACM SIGMOD International Conference on Management of Data*, SIGMOD ’10, pages 879–890. ACM, 2010.

- [60] N. C. Jones and P. A. Pevzner. *An Introduction to Bioinformatics Algorithms*. MIT press, 2004.
- [61] S. Jouili, I. Mili, and S. Tabbone. *Advanced Concepts for Intelligent Vision Systems*, chapter Attributed Graph Matching Using Local Descriptions, pages 89–99. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [62] S. Jouili and S. Tabbone. Graph Matching Based on Node Signatures. In *Graph-Based Representations in Pattern Recognition*, pages 154–163, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [63] F. Jurie and B. Triggs. Creating Efficient Codebooks for Visual Recognition. In *IEEE International Conference on Computer Vision*, volume 1, pages 604–610 Vol. 1, Oct 2005.
- [64] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, June 2010.
- [65] U. Kang, L. Akoglu, and D. H. Chau. Big Graph Mining for the Web and Social Media: Algorithms, Anomaly Detection, and Applications. In *ACM International Conference on Web Search and Data Mining*, WSDM ’14, pages 677–678. ACM, 2014.
- [66] H.-J. Kim and J. M. Kim. Cyclic topology in complex networks. *Physical Review E*, 72:036109–1–036109–4, Sep 2005.
- [67] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97, 1955.
- [68] L. I. Kuncheva. A Theoretical Study on Six Classifier Fusion Strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–286, Feb 2002.
- [69] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.
- [70] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised Learning for Graph Matching. *International Journal of Computer Vision*, 96(1):28–45, 2012.
- [71] K. Li, G. Qi, J. Ye, and K. A. Hua. Linear Subspace Ranking Hashing for Cross-Modal Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1825–1838, Sep. 2017.
- [72] W. Li, J. Joo, H. Qi, and S. Zhu. Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph. *IEEE Transactions on Multimedia*, 19(2):367–381, Feb 2017.

- [73] Y. Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73 – 82, 2004.
- [74] G. Liu, Q. Yang, H. Wang, S. Wu, and M. P. Wittie. Uncovering the Mystery of Trust in An Online Social Network. In *IEEE Conference on Communications and Network Security*, pages 488–496, 2015.
- [75] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.
- [76] X. Lu, X. Zheng, and Y. Yuan. Remote sensing scene classification by unsupervised representation learning. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9):5148–5157, Sep. 2017.
- [77] T. Ma, W. Shao, Y. Hao, and J. Cao. Graph classification based on graph set reconstruction and graph kernel feature reduction. *Neurocomputing*, 296:33 – 45, 2018.
- [78] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):645–657, Feb 2017.
- [79] P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1):3–35, 2009.
- [80] S. Martinis, A. Twele, and S. Voigt. Towards operational near real-time flood detection using a split-based automatic thresholding procedure on high resolution TerraSAR-X data. *Natural Hazards and Earth System Sciences*, 9(2):303–314, 2009.
- [81] P. R. Mendes Júnior, R. M. de Souza, R. de O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. B. Penatti, R. da S. Torres, and A. Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, Mar 2017.
- [82] J. J. Miller. Graph Database Applications and Concepts with Neo4j. In *Southern Association for Information Systems Conference*, number S 36 in SAIS, pages 141–147, 2013.
- [83] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. Multimodal data fusion for sensitive scene localization. *Information Fusion*, 45:307 – 323, 2019.
- [84] M. A. C. Neira, P. R. Mendes Júnior, A. Rocha, and R. da S. Torres. Data-Fusion Techniques for Open-Set Recognition Problems. *IEEE Access*, 6:21242–21265, 2018.
- [85] M. Neuhaus and H. Bunke. Self-Organizing Maps for Learning the Edit Costs in Graph Matching. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Part B*, 35(3):503–514, 2005.

- [86] M. Neuhaus and H. Bunke. Automatic learning of cost functions for graph edit distance. *Information Sciences*, 177(1):239 – 247, 2007.
- [87] M. Neuhaus and H. Bunke. *Bridging the Gap Between Graph Edit Distance and Kernel Machines*. World Scientific Publishing Co., Inc., 2007.
- [88] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal Deep Learning. In *International Conference on Machine Learning*, ICML ’11, pages 689–696. ACM, June 2011.
- [89] K. Nogueira, S. G. Fadel, Í. C. Dourado, R. de O. Werneck, J. A.V. Muñoz, O. A.B. Penatti, R. T. Calumby, L. T. Li, J. A. dos Santos, and R. da S. Torres. Data-Driven Flood Detection using Neural Networks. In *Working Notes Proc. MediaEval Workshop*, 2017.
- [90] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61:539 – 556, 2017.
- [91] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, Jul 2002.
- [92] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
- [93] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic Multimedia Cross-modal Correlation Discovery. In *International Conference on Knowledge Discovery and Data Mining*, KDD ’04, pages 653–658. ACM, 2004.
- [94] R. Pastor-Satorras, A. Vázquez, and A. Vespignani. Dynamical and Correlation Properties of the Internet. *Physical Review Letters*, 87:258701–1–258701–4, Nov 2001.
- [95] O. A. B. Penatti, F. B. Silva, E. Valle, V. Gouet-Brunet, and R. da S. Torres. Visual word spatial arrangement for image retrieval and classification. *Pattern Recognition*, 47(2):705 – 720, 2014.
- [96] O. A. B. Penatti, E. Valle, and R. da S. Torres. Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359–380, February 2012.
- [97] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230:279 – 293, 2017.

- [98] F. Perronnin and C. Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [99] G. Petkos, S. Papadopoulos, E. Schinas, and Y. Kompatsiaris. *Graph-Based Multi-modal Clustering for Social Event Detection in Large Collections of Images*, pages 146–158. Springer International Publishing, 2014.
- [100] J. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Advances in Kernel Methods-Support Vector Learning*, 208:185–208, 07 1998.
- [101] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni. Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, 113, Supplement 1:S110 – S122, 2009. Imaging Spectroscopy Special Issue.
- [102] J. Pokorný, M. Valenta, and J. Kovačič. Integrity constraints in graph databases. *Procedia Computer Science*, 109:975 – 981, 2017.
- [103] S. Qian, T. Zhang, C. Xu, and J. Shao. Multi-Modal Event Topic Model for Social Event Analysis. *IEEE Transactions on Multimedia*, 18(2):233–246, Feb 2016.
- [104] K. Riesen. *Structural Pattern Recognition with Graph Edit Distance - Approximation Algorithms and Applications*. Advances in Computer Vision and Pattern Recognition. Springer, 2015.
- [105] K. Riesen and H. Bunke. IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 287–297, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [106] K. Riesen and M. Ferrer. Predicting the correctness of node assignments in bipartite graph matching. *Pattern Recognition Letters*, 69:8–14, 2016.
- [107] K. Riesen, M. Neuhaus, and H. Bunke. Graph Embedding in Vector Spaces by Means of Prototype Selection. In *Graph-Based Representations in Pattern Recognition*, pages 383–393, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [108] A. A. Ross and R. Govindarajan. Feature Level Fusion Using Hand and Face Biometrics. In *Biometric Technology for Human Identification II*, pages 196–204, 2005.
- [109] J. F. Rosser, D. G. Leibovici, and M. J. Jackson. Rapid flood inundation mapping using social media, remote sensing and topographic data. *Natural Hazards*, 87(1):103–120, 2017.

- [110] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems 30*, pages 4967–4976. Curran Associates, Inc., 2017.
- [111] C. Santos, R. Lamparelli, G. Figueiredo, S. Dupuy, J. Boury, A. C. dos S. Luciano, R. da S. Torres, and G. le Maire. Classification of Crops, Pastures, and Tree Plantations along the Season with Multi-Sensor Image Time Series in a Subtropical Agricultural Region. *Remote Sensing*, 11(3):1–26, 2019.
- [112] W. J. Scheirer, A. de R. Rocha, A. Sapkota, and T. E. Boult. Toward Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, July 2013.
- [113] A. Shahroudy, T. Ng, Y. Gong, and G. Wang. Deep Multimodal Feature Analysis for Action Recognition in RGB+D Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1045–1058, May 2018.
- [114] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1):21 – 74, 2002. 16th ACM Symposium on Computational Geometry.
- [115] F. B. Silva, R. de O. Werneck, S. Goldenstein, S. Tabbone, and R. da S. Torres. Graph-based bag-of-words for classification. *Pattern Recognition*, 74(Supplement C):266 – 285, February 2018.
- [116] F. B. Silva, S. Goldenstein, S. Tabbone, and R. da S. Torres. Image classification based on bag of visual graphs. In *International Conference on Image Processing*, pages 4312–4316, 2013.
- [117] F. B. Silva, S. Tabbone, and R. da S. Torres. BoG: A New Approach for Graph Matching. In *International Conference on Pattern Recognition*, pages 82–87, 2014.
- [118] G. Simone, A. Farina, F. C. Morabito, S. B. Serpico, and L. Bruzzone. Image fusion techniques for remote sensing applications. *Information Fusion*, 3(1):3 – 15, 2002.
- [119] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *IEEE International Conference on Computer Vision*, volume 1, pages 370–377 Vol. 1, Oct 2005.
- [120] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media Hashing for Large-scale Retrieval from Heterogeneous Data Sources. In *International Conference on Management of Data*, SIGMOD ’13, pages 785–796. ACM, 2013.
- [121] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *ACM International Conference on Information and Knowledge Management*, CIKM ’02, pages 102–109. ACM, 2002.

- [122] F. Suard, A. Rakotomamonjy, and A. Bensrhair. Kernel on Bag of Paths For Measuring Similarity of Shapes. In *European Symposium on Artificial Neural Networks*, pages 355–360, 2007.
- [123] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li. Automatic Target Detection in High-Resolution Remote Sensing Images Using Spatial Sparse Coding Bag-of-Words Model. *IEEE Geoscience and Remote Sensing Letters*, 9(1):109–113, Jan 2012.
- [124] M. J. Swain and D. H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.
- [125] H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, June 1978.
- [126] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The New Data in Multimedia Research. *Communications of the ACM*, 59(2):64–73, 2016.
- [127] N. Tkachenko, S. Jarvis, and R. Procter. Predicting floods with Flickr tags. *PloS one*, 12(2):1–13, 2017.
- [128] H. Tong, J. He, M. Li, C. Zhang, and W.-Y. Ma. Graph Based Multi-Modality Learning. In *ACM International Conference on Multimedia*, MULTIMEDIA ’05, pages 862–871. ACM, 2005.
- [129] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual Place Recognition with Repetitive Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, Nov 2015.
- [130] M. Unser. Sum and Difference Histograms for Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):118–125, Jan 1986.
- [131] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph Kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.
- [132] J. Wang, M. Korayem, S. Blanco, and D. J. Crandall. Tracking Natural Events through Social Media and Computer Vision. In *ACM on Multimedia Conference*, pages 1097–1101. ACM, 2016.
- [133] M. Wang, X. S. Hua, R. Hong, J. Tang, G. J. Qi, and Y. Song. Unified Video Annotation via Multigraph Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(5):733–746, May 2009.
- [134] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu. Multimodal Graph-Based Reranking for Web Image Search. *IEEE Transactions on Image Processing*, 21(11):4649–4661, Nov 2012.

- [135] D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442, Jun 1998.
- [136] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan. Cross-Modal Retrieval With CNN Visual Features: A New Baseline. *IEEE Transactions on Cybernetics*, PP(99):1–12, 2016.
- [137] R. de O. Werneck, I. C. Dourado, S. G. Fadel, S. Tabbone, and R. da S. Torres. Graph-Based Early-Fusion for Flood Detection. In *IEEE International Conference on Image Processing*, pages 1048–1052, 2018.
- [138] R. de O. Werneck, R. Raveaux, S. Tabbone, and R. da S. Torres. Learning Cost Functions for Graph Matching. In Xiao Bai, Edwin R. Hancock, Tin Kam Ho, Richard C. Wilson, Battista Biggio, and Antonio Robles-Kelly, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 345–354, Cham, 2018. Springer International Publishing.
- [139] D. R. Wilson and T. R. Martinez. Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, 6(1):1–34, January 1997.
- [140] P. Wu, S. C. H. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao. Online Multimodal Deep Similarity Learning with Application to Image Retrieval. In *ACM International Conference on Multimedia*, MM ’13, pages 153–162. ACM, 2013.
- [141] Bai Xiao, Edwin R. Hancock, and Richard C. Wilson. A generative model for graph matching and embedding. *Computer Vision and Image Understanding*, 113(7):777 – 789, 2009.
- [142] Bai Xiao, Edwin R. Hancock, and Richard C. Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589 – 2606, 2009.
- [143] L. Xie, P. Pan, and Y. Lu. A Semantic Model for Cross-modal and Multi-modal Retrieval. In *International Conference on Multimedia Retrieval*, ICMR ’13, pages 175–182. ACM, 2013.
- [144] L. Xie, L. Zhu, and G. Chen. Unsupervised multi-graph cross-modal hashing for large-scale multimedia retrieval. *Multimedia Tools and Applications*, pages 1–20, 2016.
- [145] C. Yang, J. H. Everitt, Q. Du, B. Luo, and J. Chanussot. Using High-Resolution Airborne and Satellite Imagery to Assess Crop Growth and Yield Variability for Precision Agriculture. *Proceedings of the IEEE*, 101(3):582–592, March 2013.
- [146] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo. Deep Multimodal Representation Learning From Temporal Data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5455, July 2017.

- [147] Q. You, J. Luo, H. Jin, and J. Yang. Cross-modality Consistent Regression for Joint Visual-Textual Sentiment Analysis of Social Multimedia. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 13–22, New York, NY, USA, 2016. ACM.
- [148] A. Zadeh, P. P. Liang, J. Vanbriesen, S. Poria, E. Tong, E. Cambria, M. Chen, and L.-P. Morency. Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph. In *Annual Meeting of the Association for Computational Linguistics (Long Papers)*, volume 1, pages 2236–2246, 2018.
- [149] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. Comparing Stars: On Approximating Graph Edit Distance. *Proceedings of the VLDB Endowment*, 2(1):25–36, 2009.
- [150] X. Zhai, Y. Peng, and J. Xiao. Cross-modality correlation propagation for cross-media retrieval. In *International Conference on Acoustics, Speech and Signal Processing*, pages 2337–2340, March 2012.
- [151] L. Zhang, L. Zhang, D. Tao, and X. Huang. On Combining Multiple Features for Hyperspectral Remote Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):879–893, March 2012.
- [152] H. Zheng, X. Bai, and H. Zhao. A novel approach for satellite image classification using local self-similarity. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 2888–2891, July 2011.
- [153] Z. Zhou, X. Hong, G. Zhao, and M. Pietikäinen. A Compact Representation of Visual Speech Data Using Latent Variables. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):181–187, Jan 2014.