UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Conquering knowledge from images: improving image mining with region-based analysis and associated information

Mirela Teixeira Cazzolato

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Mirela Teixeira Cazzolato

Conquering knowledge from images: improving image mining with region-based analysis and associated information

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Agma Juci Machado Traina

USP – São Carlos
September 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C386c

Cazzolato, Mirela Teixeira
Conquering knowledge from images: improving image mining with region-based analysis and associated information / Mirela Teixeira Cazzolato; orientadora Agma Juci Machado Traina. -- São Carlos, 2019.
138 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2019.

1. Image Mining. 2. Region-Based Analysis. 3. Content-Based Retrieval. 4. Object Tracking. I. Traina, Agma Juci Machado, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Mirela Teixeira Cazzolato

Conquistando conhecimento a partir de imagens: aprimorando a mineração de imagens com análise baseada em regiões e informações associadas

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutora em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Agma Juci Machado Traina

USP – São Carlos
Setembro de 2019

Aos meus pais, irmão, avós e tios.

ACKNOWLEDGEMENTS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, da CAPES-PDSE (Processo nº 88881.134068/2016-01), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), do Projeto RESCUER, financiado pela da Comissão Européia (Processo nº 614154) e pelo CNPq/MCTI (Processo nº 490084-2013-3), e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (Processos nº 2018/24414-2 e nº 2016/17078-0).

A Deus.

Aos meus pais, Nilza e Marcos, pelo apoio incondicional, pela fé que depositam em tudo o que me proponho a fazer, pelo incentivo, exemplo de honestidade, persistência, coragem e alegria. Ao meu irmão, Marcelo, por estar sempre ao meu lado, me dando suporte. Aos meus tios e tias, por todo o apoio. A toda minha família. Aos meus avós, Élia, Marcolino, Maria e José Francisco, vocês me deram base familiar para ir além, e são exemplos de pessoas dignas, trabalhadoras e honestas que usarei como referência para toda minha vida.

À minha orientadora, Profa. Dra. Agma Juci Machado Traina, pela confiança ao me orientar, paciência nas discussões, constantes lições, incentivos a buscar cada vez mais, e pelo exemplo de pessoa. Ao Prof. Dr. Caetano Traina Jr., pelo aprendizado, discussões, dicas e suporte durante meu Doutorado.

Ao Prof. Klemens Böhm, pela orientação, paciência e aprendizado durante meu estágio no Instituto de Tecnologia de Karlsruhe (KIT), na Alemanha. Aos colegas do KIT.

Ao Grupo de Bases de Dados e Imagens (GBdI), por todas as discussões, amizade, suporte e aprendizado que me proporcionaram durante esse período.

Aos funcionários do Instituto de Ciências Matemáticas e de Computação, por fornecerem serviços e insfraestrutura que possibilitam o desenvolvimento de nossos trabalhos.

A todos meus professores, e aos amigos que me deram suporte durante todo esse período. A todos que, de alguma forma, contribuíram com a realização deste trabalho.

Muito obrigada!

“Olho vagarosa ao redor
Que raro, brecha pra enxergar
Mesmo sem conseguir brecar
O que não se vê ainda está lá

Fardo pesado de carregar
Essa coisa do aparentar
Nesse corre-corre, nem dá tempo de olhar
Mas a vida interna abre pra outro lugar

Tão claro o tempo-espaço de ver com outros sentidos
Sentir o cheiro que o dia traz
Pude ser e estar sem precisar mostrar
Sem se preocupar se alguém vai aprovar

Testemunha da devida história ser
Legitimar o próprio olhar
O viés alheio não é o que valida você
Quero absorver sem ninguém mais precisar ver

Puxa o ar do fundo, longo, profundo
Solta com barulho, joga fora pro mundo
Silêncio, silêncio, eu quero escutar
O tudo e o nada interno que há

Para pra pensar se esse é o teu lugar
Aquele bom em que deveria estar
Presta atenção só no som do mar
Que te conecta com Jah Jah”

(Te Conecta de Priscilla Novaes Leone, a Pitty)

RESUMO

CAZZOLATO, M. T. **Conquistando conhecimento a partir de imagens: aprimorando a mineração de imagens com análise baseada em regiões e informações associadas**. 2019. 138 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

A popularização de redes sociais e o uso generalizado de *smartphones* e equipamentos avançados em hospitais têm gerado dados complexos e sequências de dados, tais como imagens de alta qualidade, em grande quantidade. Fornecer ferramentas apropriadas para extrair conhecimento útil de tais dados é um grande desafio. Enquanto diversas técnicas em potencial têm sido propostas para analisar imagens, grande parte dessas técnicas consideram a imagem inteira na análise. Assim, regiões que não são de interesse são consideradas na etapa de análise, sem distinção apropriada e consequentemente prejudicando diversas tarefas. Esta pesquisa de Doutorado baseou-se na tese de que, ao tirar vantagem de pequenas representações de imagens, é possível melhorar os resultados de diferentes tarefas de análise. Para provar esta tese, primeiramente segmentou-se imagens em regiões pequenas. Uma das abordagens usadas para isso foi a extração de superpixel. Depois, foram aplicados algoritmos de classificação, agrupamento, e análise temporal nos dados, de acordo com a aplicação. A tese foi avaliada em três cenários de aplicação. No primeiro cenário, foram analisadas regiões de imagens de situações de emergência, obtidas por meio de redes sociais e que apresentavam regiões de fumaça. Os métodos propostos são capazes de segmentar regiões de fumaça e melhorar a classificação global de imagens em até 23% em comparação ao estado da arte. No segundo cenário, foram abordadas imagens do contexto médico, contendo doenças pulmonares intersticiais. As imagens foram classificadas considerando a incerteza de cada região do pulmão em conter diferentes anormalidades, representando os resultados obtidos por meio de uma visualização baseada em mapas de calor. A abordagem proposta foi melhor que os competidores na tarefa de classificação de regiões pulmonares, apresentando melhores resultados em até quatro de cinco anormalidades. No terceiro cenário, foram tratadas sequências de imagens microscópicas, exibindo embriões se desenvolvendo ao longo do tempo. Com o uso de informações das imagens baseadas em regiões, foi possível rastrear e predizer trajetórias de células ao longo do tempo, e também construir o vetor de movimento das mesmas. As abordagens propostas mostraram uma melhora de até 57% em qualidade, e uma melhora de tempo no *pipeline* de rastreamento de até 81.9%. Esta tese de Doutorado contribuiu para o estado da arte introduzindo métodos de análise de imagem baseados em região para os três cenários de aplicação mencionados anteriormente.

Palavras-chave: Mineração de imagens, análise baseada em regiões, recuperação baseada em conteúdo, rastreamento de objetos.

ABSTRACT

CAZZOLATO, M. T. **Conquering knowledge from images: improving image mining with region-based analysis and associated information**. 2019. 138 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

The popularization of social media, combined with the widespread use of smartphones and the use of advanced equipment in hospitals and medical centers has generated sequences of complex data, including images of high quality and in large quantity. Providing appropriate tools to extract meaningful knowledge from such data is a big challenge. While many potential techniques have been proposed to analyze images, most of the processing performed by image mining techniques consider the entire image. Thus, regions that are not of interest are considered in the analysis step, without proper distinction and consequently damaging most tasks. This doctorate research focused on the thesis that by taking advantage of small representations of images we can improve the overall results of different image analysis tasks. Then, we employed classification, clustering, and temporal data analysis algorithms, according to the application. We evaluate this thesis in three application scenarios. In the first scenario, we analyzed regions of images from emergencies, gathered from social media and which depict smoke regions. We were able to segment smoke regions and improve the classification of smoke images by up to 23%, compared to global approaches. In the second scenario, we worked with images from the medical context, containing Interstitial Lung Diseases (*ILD*). We classified the images considering the uncertainty of each lung region to contain different abnormalities, representing the obtained results with a heat map visualization. Our approach outperformed its competitors in the classification of lung regions by up to four of five classes of abnormalities. In the third scenario, we dealt with sequences of microscopic images depicting embryos being developed over time. Using region-based information of images, we were able to track and predict cells over time and build their motion vector. Our approaches showed an improvement of up to 57% in quality, and a speed-up of the tracking pipeline by up to 81.9%. Therefore, this doctorate research contributed to the state-of-the-art by introducing methods of region-based image analysis for the three application scenarios mentioned above.

Keywords: Image mining, region-based analysis, content-based retrieval, object tracking.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

k-NN	k-Nearest Neighbors
AC	Accuracy
BF	Brute Force
CBIR	Content-Based Image Retrieval
DF	Distance Function
DM	Data Mining
DT	Decision Tree
DTW	Dynamic Time Warping
EM	Expectation Maximization
ER	Error Rate
ExecTime	Execution time
FN	False Negative examples
FP	False Positive examples
FEM	Feature Extraction Method
F_1	F-Measure
FnCost	False Negative Cost
FpCost	False Positive Cost
GMM	Gaussian Mixture Model
IBL	Instance-Based Learning
ILD	Interstital Lung Disease
KDD	Knowledge Discovery in Databases
MAM	Metric Access Method
MBR	Minimum Bounding Rectangle
MLE	Maximum Likelihood Estimation
MLP	Multi Layer Perceptron
MND	Multivariate Normal Distribution
NB	Naïve Bayes
ND	Normal Distribution
N	Negative examples
PCA	Principal Component Analysis
PDF	Probability Distribution Function

P	Positive examples
PR	Precision
RF	Random Forests
RE	Recall
RESCUER	Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management
ROI	Region of Interest
CH	Color Histogram
CL	Color Layout
CS	Color Structure
CT	Color Temperature
EH	EdgeHistogram
Hr	Haralick
LBP	Local Binary Pattern
SLIC	Simple Linear Iterative Clustering
NH	Normalized Histogram
OF	Optical Flow
SC	Scalable Color
TS	Texture Spectrum
TB	Texture Browsing
SVM	Support Vector Machine
Zr	Zernike
TDR	Tracking Detection Error
TNR	True Negative Rate
TotCells	Total number of cells
TPR	True Positive Rate
TN	True Negative examples
TP	True Positive examples
UND	Univariate Normal Distribution

LIST OF SYMBOLS

$\mathscr{E}$ — Extractor or Feature Extraction Method

$< \mathbb{S}, \delta >$ — Metric space

δ — Metric distance function

$\mathbb{S}$ — Domain of elements

$\mathbb{R}$ — Domain of real numbers

$\mathbb{V}$ — Domain of feature vectors

S — Set of elements

ξ — Radius value

s — Complex element or object

$< \mathscr{E}, \delta >$ — Descriptor

v — Feature vector

L_p — Distances from the Minkowski family

ℓ — Label value

$\mathscr{L}$ — Set of labels

T — Training set

d — Number of dimensions

P — Number of pixels

B — Number of superpixels

b — Desired number of superpixels

e — Compactness factor of superpixels

B' — Size of a superpixel

$\mathscr{O}$ — Big O notation for computational complexity

K — Cluster

$\mathscr{C}$ — Classifier

k-NNq — *k-NN* operator

Rq — Range query operator

$f(v)$ — Probability density function

X — Random variable

Σ — Covariance matrix

$|\Sigma|$ — Determinant of the covatiance matrix

I — Image

Q — Set of superpixels

q — Superpixel region

t — Point in time (timestamp)

C — Set of cells from an embryo

c — Cell from an embryo

$\mathscr{T}$ — Trajectory of cells

th — Threshold distance

$\hat{S}$ — Set of seeds or seed points

$\hat{s}$ — Seed or seed point of a cell

w — Window size

pw — Portion of the window discarded

$\hat{p}(\hat{s}_1, \hat{s}_2)$ — Pair of two seed points

$Pol(x)$ — Lagrange's polynomial

CONTENTS

CHAPTER

1

INTRODUCTION

Online platforms generate huge amounts of data at a very fast pace. Extracting meaningful knowledge from complex data such as images, sequences of images, videos, and time series has become a huge challenge. Social media employs images since its inception in 2005. Thus, the amount of images on social media is overwhelming, and the approximate number of images on social media can be counted in the billions (ØRNAGER; LUND, 2018; COSTA *et al.*, 2017). This popularization of social media, combined with the increasing access and use of smartphones and other mobile devices, collaborate with the generation and sharing of such data, all the time. Also, specialized environments such as hospitals, clinics, and laboratories now have the potential of generating complex data of high quality, in large quantity and with associated information (GONÇALVES *et al.*, 2017). This scenario brings out the need for processing the available data effectively and efficiently. The appropriate analysis of the available information can help authorities in emergency situations (CHINO *et al.*, 2015; OLIVEIRA *et al.*, 2016), supporting education and medical decision-making (SANTOS *et al.*, 2018; FERREIRA *et al.*, 2018), as well as speeding-up pipelines known to be time-consuming (STEGMAIER *et al.*, 2014; HE *et al.*, 2017; ULMAN *et al.*, 2017).

1.1 Motivation

A wide range of techniques have been proposed to perform analysis of images, and many of them have great potential. Taking advantage of existing approaches to represent and find patterns on images can be meaningful. However, most of the processing performed by image mining techniques consider the entire image. This hurts most tasks since regions that are not of interest are considered in the analysis step, without proper distinction (OLIVEIRA *et al.*, 2016; CHINO *et al.*, 2015). Region-based analysis of images can improve the detection and content-based retrieval results. This is performed by segmenting regions of interest from

images before performing data mining tasks. Meanwhile, when dealing with sequences of images, the processing pipeline should be fast and take advantage of the objects' movement to infer knowledge. Accordingly, the segmentation of such objects can also be useful to improve the analysis.

This Ph.D. research aimed at answering the following question: *"How can we improve the image mining task of different application contexts relying on complex data and associated information?"* One of the main challenges regarding this research question is that image mining can be *complex*, *diverse* and *dependent on the application domain*. It can be complex due to the different forms of data representation. Image mining can be diverse regarding the different tasks that are possible to perform, such as image classification/labeling, content-based retrieval, object identification, noise removal, and image fusing. Also, such tasks can be different among themselves only regarding the different application domains. For instance, object identification may refer to detecting pedestrians in the urban context, or refer to a textured pattern that can indicate cancer or a similar abnormality in the medical context.

1.2 Problem Statement

As discussed in the previous section, the main goal of this Ph.D. research is to improve image mining by taking advantage of visual findings obtained by region-based approaches. Accordingly, we propose methods to extract meaningful information from single and sequences of images, considering problems in which the global mining approach is not accurate. Due to the complexity and diversity of issues in the context of image mining, in this Ph.D. research we propose the following thesis:

> **Thesis.** *The analysis of image regions, combined with additional information, leads to more accurate mining results regarding the entire image and also helps the processing of sequences of images, speeding-up costly pipelines and making it possible to infer knowledge from objects' movement.*

This Ph.D. research focused on image mining methods, assisted by region-based analysis and associated information. Particularly, we consider the aforementioned thesis by working with data obtained from three context applications: (i) images from emergency scenarios, that were gathered from social media and depict smoke regions; (ii) images from the medical context, containing Interstitial Lung Diseases (*ILD*); and (iii) with sequences of microscopic images, depicting embrios being developed over time and the associated information. Properly, we support the previously mentioned thesis by proposing methods to solve the three following research problems. Notice that each contribution concerns one application context.

Research problem 1. *Mining images from emergency scenarios:*

Social media and crowdsourcing generate images at a fast pace. Particularly, in emergency scenarios, public authorities aim to provide fast and accurate responses, while dealing with intense flows of information in little time. The early detection of fire, smoke, and explosions assist the rescue forces in preventing further risks to human life and financial losses. While the global detection of fire in images has shown accurate results, detecting smoke in still images has proven to be a challenging task. Overall, the major difficulties of this task refer to the absence of movement and the fact that smoke does not present well-defined color patterns. For instance, the smoke color can change as the temperature increases, according to the material being burned, due to the inherent transparency, and its heavy dependence on weather conditions. Based on this scenario, we formulated the following questions: (i) *What patterns can be used to classify images regarding the presence of smoke?* (ii) *Can we improve the segmentation of smoke images with a region-based classification approach?* (iii) *Can the classification of smoke images be enhanced using the segmentation outputs?*

Research problem 2. *Mining medical images with uncertainty:*

The advent of capturing and storing technologies have made it possible for hospitals and medical centers to generate large volumes of imaging exams increasingly. Content-Based Image Retrieval (CBIR) applications assist physicians in analyzing such images, leaning on fast, accurate, and reliable methods, mainly based on historical and already analyzed exams. In particular, when detecting abnormalities in lung CT scans, different abnormalities can present very similar visual patterns. Also, the detection approach must consider that a single CT slice/image of the lungs may show normal and abnormal regions. Consequently, performing the labeling of the entire images as a whole unity is neither accurate nor adequate. Relying on this scenario, we formulated the following research questions: (i) *How can we model the healthy pulmonary tissue pattern and visualize, for each lung region, its likelihood of being abnormal?* (ii) *How can we model different lung abnormalities, and then provide a proper visualization of the most probable findings of each lung region?*

Research problem 3. *Analyzing moving objects in sequences of microscopic images:*

A sequence of images depicts the development of a given embryo over time. It starts with a single cell that splits over time, as the embryo is developing. The traditional pipeline to deal with such images encompasses image acquisition by microscopes, transferring them to a local computer and processing the images by segmentation algorithms. Then, tracking algorithms construct the trajectory vector of each cell, over time. The problem here is that performing the entire pipeline is time-consuming, as the volume of images is substantial and an embryo can take hours to develop. Based on this scenario, we aimed to answer the following research questions:(i) *How can we efficiently track cells over time, relying on the output of segmentation*

algorithms, and obtain cells' trajectory vectors? (ii) *What are the alternative configurations that allow using previous positions of cells to predict their future movement effectively, and obtain reliable approximate trajectory vectors?* (iii) *Can we speed-up the tracking task with indexing structures, and consequently improve the overall accuracy?*

1.3 Contributions

The main contributions of this Ph.D. research are three-fold, each of which addressing one of the research problems mentioned earlier.

Contribution 1. *The SmokeBlock method:*

We studied various configurations regarding the representation of images and the best classifiers to use in the context of fire and smoke. As a result, we propose the *SmokeBlock* method. *SmokeBlock* performs the segmentation of smoke regions from the images by dividing them into groups of pixels containing similar visual patterns (called superpixels). Then, the method classifies each region as smoke or not, composing as its first output a segmented image with only smoke regions. In the sequence, *SmokeBlock* uses the segmented image to perform a global classification of the images. The experimental analysis was carried using real data from Flickr, and the obtained results were compared against state-of-the-art methods for feature extraction. *SmokeBlock* achieved performance superior than its competitors, for the task of smoke detection.

The works (CAZZOLATO *et al.*, 2016) and (CAZZOLATO *et al.*, 2017) report all results related to this contribution.

Contribution 2. *The BREATH and dp-BREATH methods:*

Considering that different parts of a single lung image may present both normal and abnormal characteristics, we first propose the *BREATH* method. *BREATH* starts by segmenting lung tissues using a superpixel-based approach. Following, it trains a statistical model to represent normal tissues and, finally, *BREATH* generates a heat map showing abnormal regions that require attention from the physicians.

Further, we proposed the *dp-BREATH* method to extend *BREATH*. *dp-BREATH* models and classifies the highlighted lung regions, according to their probability of containing each of the studied abnormalities. The method relies on a Gaussian Mixture Model to represent each lung abnormality, assuming that the lung regions follow a normal distribution when grouped separately. Consequently, *dp-BREATH* has shown to provide a high recognition of radiological patterns, with the likelihood of a selected lung region to contain abnormalities.

The works (CAZZOLATO *et al.*, 2017) and (CAZZOLATO *et al.*, 2019) report *BREATH*, *dp-BREATH*, and all related results regarding this contribution.

Contribution 3. *The CM-Predictor method:*

CM-Predictor is a method which takes advantage of previous positions of cells to estimate their motion. It extends the traditional pipeline of cell tracking. When estimation takes place, the costly acquisition, data transfer, and image segmentation steps are omitted. *CM-Predictor* speeds-up the tracking pipeline, monitors the prediction error and adapts the model whenever needed. We also proposed baseline tracking algorithms (*Direct-Tracker* and *Clever-Tracker*) to serve as a basis for comparison and validation. Experimental results show that *CM-Predictor* can accurately estimate the motion vectors, maintaining the prediction quality of other algorithms, and performing faster than them. Further, we evaluate how the methods can benefit from indexing structures to speed-up the tracking task.

This contribution was reported in the work (CAZZOLATO; TRAINA; BÖHM, 2018).

1.4 Outline

This chapter summarized the motivation, the addressed problems, objectives, and main contributions of this doctorate Ph.D. research. The remaining parts of this monograph are organized as follows: Chapter 2 presents the relevant background, needed to follow the work. Chapter 3 shows how we can take advantage of region-based classification of images to improve content-based image retrieval. Chapter 4 presents how the study of regions of images can improve the characterization of different patterns presented within. Chapter 5 presents how we can improve the tracking of moving objects by studying their intrinsic temporal behavior. Additionally, we explore scenarios where different configurations are allowed, and to which extent they are positive and valid. Chapter 6 concludes this thesis, stating its main contributions, the publications generated during the work, and points out different opportunities for future work.

CHAPTER

2

BACKGROUND

In this chapter, we present the relevant background related to complex data, knowledge discovery, and moving objects.

2.1 Complex Data and Knowledge Discovery

Large amounts of multimedia information are generated every instant, due to the lowering cost of acquisition devices, the easiness of data communication, to the advent of social media, as well as the dissemination of mobile devices. The increasing quantity of data considered complex, such as images, sequences of images, video, audio and time series, corroborates with the necessity of methods and technologies that support the automatic manipulation and knowledge extraction of these data. This manipulation must be efficient, with the lowest computational cost possible, and be effective as it should present the meaningful semantics embedded in the data. Unlike traditional data (such as numbers, small characters arrays, and dates), complex data do not possess the order property. This makes impossible to use comparison operators, such as "$<$", "$>$", "$\leq$" e "$\geq$" (BUENO *et al.*, 2009; CHINO *et al.*, 2015). In the same way, equality operators ("$=$" and "$\neq$") generally are not used, since we rarely want to verify if two complex elements are equal.

Complex elements are represented by feature vectors, that are numeric signatures of elements' data content, generated by the so-called Feature Extraction Methods (*FEMs*). A similarity measure compares a pair of complex elements and is often referred to as a Distance Function (*DF*). Consequently, the following definitions of Metric Space (Definition 2.1) and Metric distance function (Definition 2.2) become necessary as they provide the basis for complex data comparison.

Definition 2.1 (Metric Space)**.** A **metric space** is an ordered pair $< \mathbb{S}, \delta >$, where $\mathbb{S}$ is the elements' domain, and $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}^+$ is a metric (SAMET, 2006).

Definition 2.2 (Metric)**.** A **metric** δ computes the distance between pairs of elements $< s_i, s_j >$ $|s_i, s_j \in \mathbb{S}$, and follows the properties of symmetry, non-negativity, triangular inequality, and the identity of indiscernibles.

For all elements $s_i, s_j, s_k \in \mathbb{S}$, δ must hold the following properties to be considered metric:

- **Symmetry.** $\delta(s_i, s_j) = \delta(s_j, s_i)$: the distance is a symmetric function.
- **Non-negativity.** $\delta(s_i, s_j) \geq 0$: the distance is a non-negative number.
- **Triangular inequality.** $\delta(s_i, s_j) \leq \delta(s_i, s_k) + \delta(s_k, s_j)$: the distance between two objects s_i e s_j in space cannot be higher than the distance between them passing through a third object s_k.
- **Identity of indiscernibles.** $\delta(s_i, s_i) = 0$: the distance of an object to itself is 0.

The definitions of metric space and metric distance functions are important to enable the use of indexing structures for complex data, that are called Metric Access Methods (*MAMs*). We detail *MAMs* in Section 2.1.4, when we refer to the content-based image retrieval.

Another important concept regarding complex data is the **descriptor**, described following in Definition 2.3.

Definition 2.3 (Descriptor)**.** A **descriptor** is composed of a pair $< \mathscr{E}, \delta >$, where $\mathscr{E}$ is a *FEM* and δ is a distance function.

Following we present the main concepts related to *FEMs* and distance functions, with examples of existing methods grabbed from the literature.

2.1.1 Feature Extraction Methods

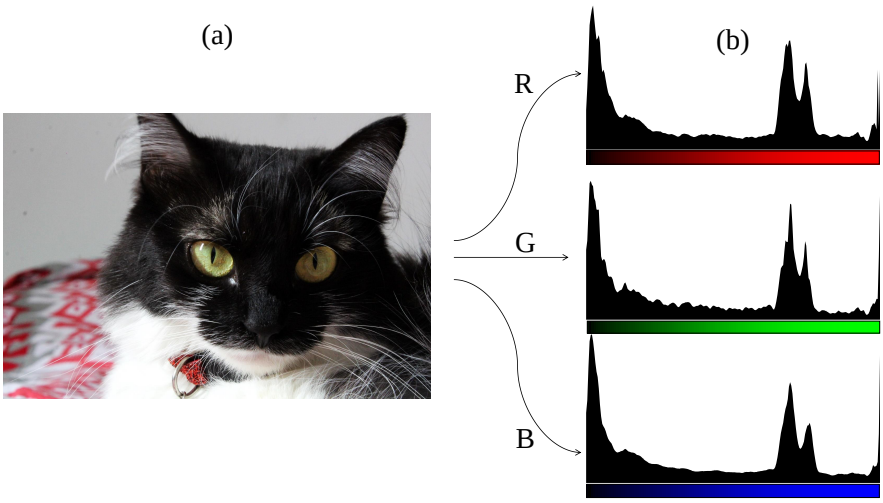
Feature Extraction Methods (*FEMs*) generate feature vectors to represent the content of complex data elements. Definition 2.4 formally describes *FEMs*.

Definition 2.4 (Feature Extraction Method - *FEM*)**.** Let $\mathbb{S}$ be the domain of complex data, and $\mathbb{V}$ be the domain of feature vectors. A **Feature Extraction Method** is a non-bijective function $\mathscr{E} : \mathbb{S} \rightarrow \mathbb{V}$ capable of representing a complex object $s_q \in \mathbb{S}$ in $\mathbb{V}$ as v_q.

Given a complex object $s_q \in \mathbb{S}$ and a *FEM* $\mathscr{E}$, a **feature vector** is an element generated by $\mathscr{E}(s_q) = v_q$, with $v_q \in \mathbb{V}$, such that v_q represents a complex object s_q in $\mathbb{V}$.

Regarding the different types of complex data, **in this thesis we focus on the analysis of images and sequences of images**, that are aligned in time. Most existing *FEMs* generate feature vectors as numeric results extracted from images. *FEMs* can generate different representations such as shape, texture distribution, color distribution or intensities, and movement. Among the most widespread *FEMs* in the literature, we have the Color Histogram (*CH*). *CH* serves as an effective and efficient representation of the image content, is simple and fast to compute, and enables the characterization of the color distribution in the image. Its representation can be local or global (PARKER, 2010; GONZALEZ; WOODS, 2008; SHAPIRO; STOCKMAN, 2001). Another important characteristic of *CH* is the robustness to the translation and rotation of the visualization axis. It subtly changes with relation to scale, occlusion, and change in the viewing angle. Figure 1 shows an example of an image and its corresponding *RGB* features. *CH* is built by a quantization of the space, consisting of grouping similar colors in bins. The size (or range of values) of these bins can be defined according to the application need. After defining the bins, *CH* counts the frequency of each color intensity in the image. *CH* is constructed using a color space, such as *RGB*, *YCbCr*, *HSL* and *HSV* (PARKER, 2010; ZHANG; LI; LI, 2009; HUANG *et al.*, 2018).

Figure 1 – An (a) input image and (b) its corresponding color histogram, represented in the *RGB* color space and with the color channels separated.



Source: Elaborated by the author.

The *MPEG*-7 standard was proposed by *ISO/IEC JTC*1 (MULTIMEDIA, 2002). It defines expected low-level representations of images using color, texture, and shape features. Such representations became widely used in the literature by implementations of *FEMs* (OJALA; AITTOLA; MATINMIKKO, 2002; HYUN; KIM; OH, 2015; MUNADI *et al.*, 2015), such as:

- **Color:** *Color Layout* (*CL*), *Color Structure* (*CS*), *Scalable Color* (*SC*), *Color Temperature* (*CT*), *Dominant Color* (*DC*), *Color Correlogram* (*CC*) e *Group-of-Frames* (*GF*);
- **Shape:** *Contour Shape* (*ContS*), *Shape Spectrum* (*SS*) e *Region Shape* (*RS*);

- **Texture:** *Edge Histogram* (*EH*), *Texture Browsing* (*TB*) e *Homogeneous Texture* (*HT*).

For instance, the *Texture Browsing* (*TB*) *FEM* extracts texture patterns by applying Gabor filters in the image (LEE; CHEN, 2005), as illustrated in Figure 2(b). *TB*'s resulting feature vector is composed of 12 features: two of texture regularity; six of directionality, defining the most dominant texture orientation; and four of roughness, representing two dominant texture scales.

Another example is the *Color Layout* (*CL*) *FEM*, which describes the color distribution of the image considering the spatial location of pixels. The image is divided into squared sub-regions, defined by a parameter, and then *CL* labels each region with its correspondent average color (KASUTANI; YAMADA, 2001). Figure 2(c) illustrates the regions created by *CL*. After this step, the average colors are transformed to the *YCbCr* color space, and then *CL* applies the Cosine Discrete Transform in each channel of *YCbCr*. *CL* extracts the low-frequency coefficients by performing a zig-zag image reading. Aiming to reduce the dimensionality, *CL* selects only the most prominent frequencies of the resulting feature vector (KASUTANI; YAMADA, 2001).

Figure 2 – Representing an image using different *FEMs*: (a) The original image. (b) *TB*'s Gabor filter, with the following setting: filter size $fs = (21, 21)$, standard deviation of the Gaussian filter $\sigma = 8.0$, orientation of the normal to the parallel stripes $\theta = \pi/4$, wavelength of the sunusoidal factor $\lambda = 10.0$, spatial aspect ratio $\gamma = 0.5$, and phase offset $\psi = 0$. And (c) *CL* color quantization, representing an image of size 5184×3456 pixels with a matrix of size 30×30 pixels.

(a) (b) (c)

Source: Elaborated by the author.

A digital camera typically captures the objects depicted in the images in an arbitrary orientation. This results in a representation that is distorted by certain geometric transformations (LIN; FANG, 2009). This is a recurrent problem regarding shape *FEMs*. In fact, Chaudhuri (2013) points out that shape is a powerful tool to describe objects, differentiating them from each other. The shape representation of an image usually looks for effective and perceptually relevant features, based on both boundaries information as well as the inside information contents. Among the many developed features in recent years are the shape signature, histograms of signatures, invariant shapes, moments, curvatures, shape content, and spectral characteristic (CHAUDHURI, 2013).

Haralick (*Hr*) (HARALICK; SHANMUGAM; DINSTEIN, 1973) and *Local Binary Patterns* (*LBP*) (OJALA; PIETIKAINEN; MAENPAA, 2002) are two examples of widely used

FEMs for texture patterns. *Hr* assumes that the textural information from an image is in the overall or "average" spatial relationship that gray tones in images have to one another. Accordingly, *Hr* considers that a set of gray-tone spatial-dependence matrices adequately specifies the texture information. Such matrices are computed for various angular relationships and distances between neighboring resolution cell pairs on the image (HARALICK; SHANMUGAM; DINSTEIN, 1973). *LBP* is a theoretically and computationally simple approach, that is robust to grayscale variations and rotation invariant. The algorithm recognizes uniform local binary patterns, in the sense that uniform patterns provide a vast majority, sometimes over 90%, of the 3×3 texture patterns in examined surface textures (OJALA; PIETIKAINEN; MAENPAA, 2002; OJALA; AITTOLA; MATINMIKKO, 2002; FENG *et al.*, 2017).

Besides color, texture, and shape visual features, a piece of very important information related to sequences of images and videos is the **movement**. With sequential data, applications can take advantage of the movement information by considering the spacial relation of the images in time. We discuss further details about the analysis of sequences of images in Section 2.4, when we talk about Moving Objects. Regarding the feature extraction task, one example of *FEM* for sequential data is the *Optical Flow* (*OF*) algorithm (NEUMANN, 1984). *OF* represents the movements' state in the entire image/frame, regarding a specific instant in time (CHAUDHRY *et al.*, 2009). This representation enables the identification of the movement type of an object with relation to the static background of an image/frame. According to Neumann (1984), *OF* is an intermediary variation of the image difference in time. It assigns a velocity vector to each pixel, to describe the pixels' temporal displacement in the view plane. Although widely used, *OF* presents a high computational complexity.

FEMs can be employed globally, considering the entire image, or locally, considering for instance only specific regions of the image, objects or control points. In practice, different *FEMs* can be combined to describe various characteristics and improve the content representation being analyzed. Examples of works in this line of research are (CHAUDHRY *et al.*, 2009; CHINO *et al.*, 2018; BEDO *et al.*, 2015; NESSO-JR *et al.*, 2018). After extracting the relevant feature vectors with *FEMs*, it is necessary to establish a metric that allows the comparison of pairs of such vectors. We discuss such metrics provided by proper distance functions, in the following subsection.

2.1.2 Distance Functions

Distance Functions (*DF*) are (dis)similarity measures, used to measure the similarity (or difference) between a pair of elements $< s_i, s_j >$, that are represented in the same *FEM*, i.e. s_i and s_j are from the same domain $\mathbb{V}$. Definition 2.5 formalizes this concept.

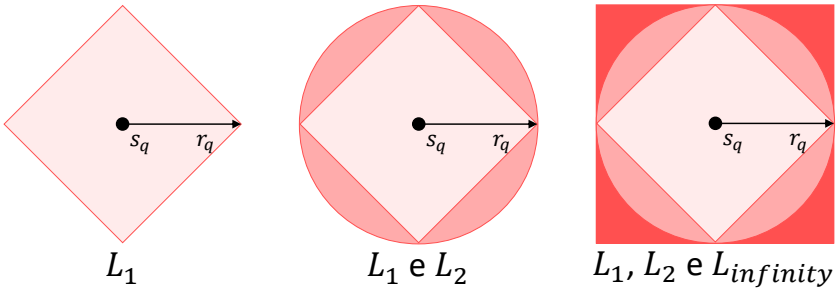
Definition 2.5 (Distance Function - *DF*)**.** Let $\mathscr{E}$ be a *FEM* and s_i and s_j be two complex objects. A **Distance Function** measures the distance between s_i and s_j, and returns a real value to represent how much (dis)similar both objects are from each other: $\delta(s_i, s_j) \rightarrow [0, 1]$.

As mentioned before, for a *DF* $\delta : \mathbb{S} \times \mathbb{S}$ to be metric it must hold to the metric space restrictions (symmetry, non-negativity, triangular inequality and identity of indiscernibles). Different *DFs* can measure the distance between a pair of objects in distinct manners, depending on the data domain and the application scenario. Well-known examples of *DFs* used in the literature are the ones from the *Minkowski* family, which contemplates *Manhattan* (*City-Block*), *Euclidean*, and *Chebychev*, among others (HAN; KAMBER; PEI, 2011; ZEZULA *et al.*, 2006). This family of distances is also known as L_p. Let v_a and v_b be the feature vectors of dimension d, corresponding to the complex objects s_a and s_b, respectively. The *DFs* from the *Minkowski* family follow Equation 2.1 in individual cases, using the corresponding p value.

$$L_p = \sqrt{\sum_{i=1}^{d} |v_{ai} - v_{bi}|^p} \tag{2.1}$$

Figure 3 shows the search space for different values of p, in the bi-dimensional space. *Manhattan* (L_1) represents a lozenge and consists of the sum of the absolute differences between the elements. *Euclidean* (L_2) is widely used and is represented by a circle since all points have the same distance to the query center. Finally, *Chebychev* (L_{infinity} or L_∞), also known as the maximum distance, is represented by a square. The respective formulas are shown in Table 1.

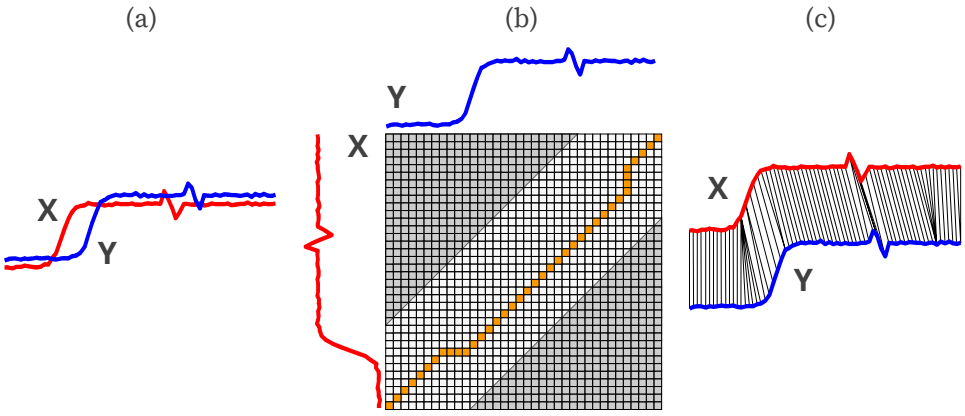
Figure 3 – Search space of the *Minkowski* family, in the bi-dimensional space: *Manhattan* (*City-Block* or L_1), *Euclidean* (L_2) e *Chebychev* (L_{infinity} or L_∞).



Source: Elaborated by the author.

The *Dynamic Time Warping* (*DTW*) distance is another example of a well-known and widely used distance. *DTW* searches for the optimal alignment between two sequences, that are dependent in time, under certain restrictions, as illustrated by Figure 4. Intuitively, *DTW* distorts the sequences in a non-linear way, in order to match each other (MÜLLER, 2007). *DTW* compares two time-dependent sequences X and Y, where $X = (x_1; x_2; ...; x_n)$ is of size $n \in \mathbb{N}$ and $Y = (y_1; y_2; ...; y_m)$ is of size $m \in \mathbb{N}$. X and Y can be time series or sequences of features, sampled in equidistant points in time. *DTW* compares both sequences by constructing an $N \times M$ matrix of synchronization. *DTW* employs a local *DF* to compare the pair of sequences X and Y (*Euclidean*, for instance). Thus, the (i^{th}, j^{th}) element of the matrix corresponds to the value obtained by local *DF*. According to Müller (2007), *DTW* is not metric, as it is not normally

Figure 4 – *DTW* searches for the optimal alignment between two sequences X and Y: (a) the comparison of X and Y, that are misaligned in time; (b) matrix of synchronization with the optimal path between the sequences; (c) the resulting alignment of the sequences.



Source: Adapted from Keogh and Ratanamahatana (2004).

defined as positive, and it does not satisfy the triangular inequality, even in the case that the local distance employed is metric.

Table 1 summarizes examples of *DFs*, the corresponding formula, and the information about if the *DF* is metric. The Kullback-Leibler Divergence (*KLD*) and Jeffrey Divergence (*JD*) are examples of non-metric *DFs*, useful for image classification and retrieval. *KLD* does not hold the symmetry and triangular inequality properties. *JD* is a symmetric variation of *KLD*, but it does not follow the triangular inequality property (BEDO *et al.*, 2016).

Table 1 – Example of *DFs*, the respective formulas and the information about each distance being a metric. Here, v_a and v_b are two feature vectors of d dimensions.

Distance Function	***Formula***	***Is Metric***
Manhattan (*City-Block*)	$\sum_{i=1}^{d} \lvert v_{ai} - v_{bi} \rvert$	Yes
Euclidean	$\sqrt{\sum_{i=1}^{d} (v_{ai} - v_{bi})^2}$	Yes
Chebychev	$\lim_{p \to \infty} (\sum_{i=1}^{d} \lvert v_{ai} - v_{bi} \rvert^p)^{\frac{1}{p}}$	Yes
Canberra	$\sum_{i=1}^{d} \frac{\lvert v_{ai} - v_{bi} \rvert}{\lvert v_{ai} \rvert + \lvert v_{bi} \rvert}$	Yes
Kullback-Leibler Divergence	$\sum_{i=1}^{d} v_{ai} \ln(\frac{v_{ai}}{v_{bi}})$	No
Jeffrey Divergence	$\sum_{i=1}^{d} (v_{ai} - v_{bi}) \ln(\frac{v_{ai}}{v_{bi}})$	No

Source: Elaborated by the author.

2.1.3 Data Mining

The *Knowledge Discovery in Databases* (*KDD*) was proposed in (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) and is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Additionally to this definition, in (MAIMON; ROKACH, 2010) the authors describe *KDD* as an automatic exploratory analysis and modeling of large data repositories. *KDD* is an iterative process, and can be divided into seven main steps: data cleaning and integration (performed during pre-processing), data selection, transformation, mining, patterns evaluation, and knowledge presentation (HAN; KAMBER; PEI, 2011).

Data Mining (*DM*) is the primary step of the *KDD* process. *DM* consists of the effective application of algorithms in the data, aiming at discovering meaningful hidden patterns. In (ZAKI; MEIRA JR., 2014) the authors state that *DM* comprises the core algorithms that enable one to gain fundamental insights and knowledge from massive data. Additionally, it consists of an interdisciplinary field, including concepts from areas such as database systems, statistics, machine learning, and pattern recognition. *DM* algorithms perform based on diverse heuristics and can be divided into tasks. These tasks differ from each other regarding the way they search for and recognize existing patterns. *DM* can be divided into three tasks, namely classification, clustering, and frequent patterns discovery (HAN; KAMBER; PEI, 2011). In this work, we focus on classification and clustering algorithms, thus we discuss these two tasks with more details next. For the sake of simplicity, we use the terms data objects, examples, and tuples as synonyms in this work.

Classification Algorithms

Classification consists of a supervised learning process that aims to find a model (or function) to describe and distinguish between classes of data (HAN; KAMBER; PEI, 2011). This model is trained from a set of n labeled (or pre-classified) data objects. Classification can also be seen as a function (or simply classifier) $\mathscr{C}$, that maps an object s into a label ℓ, such that $\mathscr{C}(s) = \ell$ (MELLO; PONTI, 2018). The constructed model can be further used to predict the class of new objects, which the class label is unknown.

The choice of the appropriate classifier must consider the context (application) and the available data (data type, distribution, if it has noisy and missing values, and others). The classification task has many applications, such as meteorological predictions, shopping pattern recognition, finances, and image classification. We are interested in the latter. Many classifiers have shown to be fit for the image classification task in the literature. Example of recent works are (BEDO *et al.*, 2015; KALE *et al.*, 2016; KARARGYRIS *et al.*, 2016; CHINO *et al.*, 2018). Usually, they rely on color, texture, and other available features to perform classification. We present well-known examples of classifiers next.

Support Vector Machines (*SVMs*) were proposed in (VAPNIK, 1995) to classify both linear and nonlinear data. According to Han, Kamber and Pei (2011), *SVM* transforms the training data into a higher dimension using a nonlinear mapping. Once in the new dimension, *SVM* searches for the linear optimal separating hyperplane found using support vectors and margins. This hyperplane is also known as the "decision boundary", which separates the classes. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. *SVMs* are known to be extremely slow but highly accurate. It has the ability to model complex nonlinear decision boundaries while being much less willing to overfitting than other classifiers. *SVMs* can also be used for numeric prediction.

Decision Trees (*DTs*) are popularly known due to their construction and representation simplicity. *DTs* are self-explanatory and do not require a data mining expertise to understand the resulting model, as it is intuitive and easy to take in. They are capable of representing multidimensional data in a simple and fast way, generally presenting high accuracy. However, a high accurate *DT* model depends on the training data available, and it can be difficult to find the optimal tree model. A *DT* structure is composed of internal nodes (representing a test in an attribute), branches (representing the outcomes of the test), and the leaves (holding the class labels). *DTs* algorithms employ a node division heuristic to decide when and how to split the internal nodes of the tree. The most known and employed division heuristics are the Gini Index and Information Gain. Also, a *DT* relies on a stopping criteria to finish the construction of the model (HAN; KAMBER; PEI, 2011; AGARWAL; SHARMA, 2011). Examples of classic *DT* algorithms are C4.5 (QUINLAN, 1993) and CART (BREIMAN *et al.*, 1984).

The classifiers as mentioned earlier are called *eager learners* since they have a training phase to construct the classification model. The classifiers follow predetermined criteria, and (sometimes) make suppositions regarding the data distribution. Subsequently, this model is applied to a testing set, aimed at evaluating its effectiveness. Then, the model is used to classify new instances of data. On the other hand, *IBL* classifiers (***Instance-Based Learning***) (AHA; KIBLER; ALBERT, 1991) generalize the classification predictions through the available data, since the training data is completely stored. That is the reason this category of classifiers is named *lazy learners*. Most *IBL* classifiers use *Euclidean* distance, but other measures can be used as well and may be more appropriated, according to the data distribution and context (HAN; KAMBER; PEI, 2011).

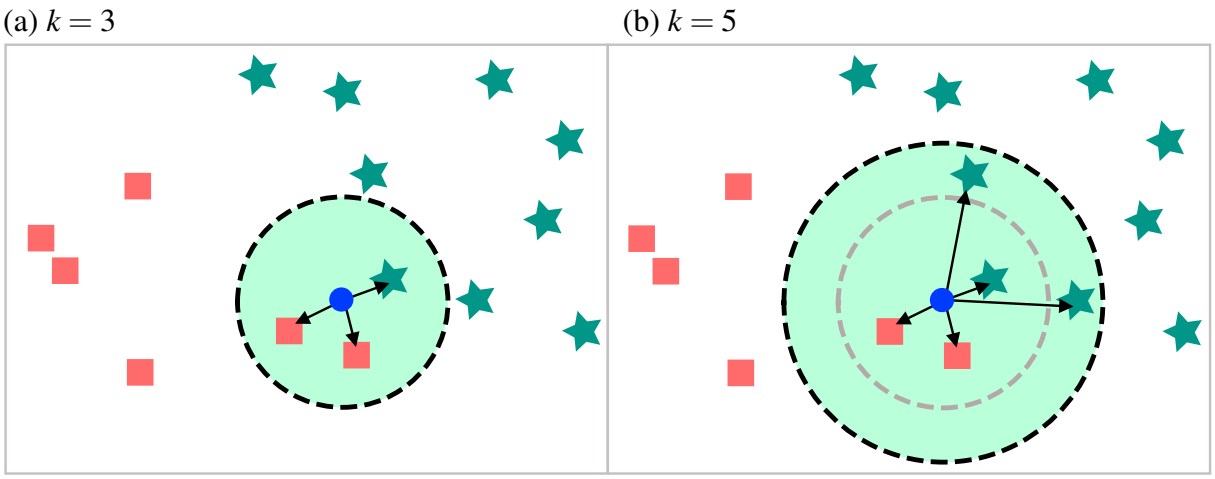
A well-known *IBL* classifier is the *k-NN* (***k-Nearest Neighbors***). *k-NN* is a non-parametric approach that makes no assumption regarding the data. The algorithm uses a distance function to determine which of the elements of a dataset is the closest instance to a query element. Once the closest element to the query is located (for $k = 1$), the corresponding label is used to classify the query element. If $k > 1$, the majority class of the k-nearest elements is used to classify the query element.

Figure 5 shows two examples of *k-NN* queries, (a) one with $k = 3$ and (b) other with

$k = 5$. Notice that, for both queries, the same dataset is used, but different values of k. The dashed circle has its center as the query center, and the squares and stars represent data elements from both available classes. With $k = 3$, (a) the majority class is the square since two of the three nearest elements are squares. With $k = 5$, (b), the majority class is the star. When a tie happens (for example, with $k = 4$ both classes would have two nearest elements to the query), *k-NN* chooses the class whose summation of distances to the nearest elements is smaller. If the tie persists, an arbitrary class is chosen.

The classic *k-NN* approach assumes that all nearest elements of a given query have the same importance (all of them have the same weigh). Modifications in this classifier include weighting the neighbor elements according to their proximity to the query center.

Figure 5 – Example of *k-NN* queries. In (a) the majority class is the square, and in (b) the majority class is the star.



Source: Elaborated by the author.

When referring to ***Ensemble*** approaches, several classifiers are trained together, and a voting process combines their results. A widely known example of an algorithm from this category is the ***Random Decision Forests*** (*RF*) (HO, 1995). *RF* combines tree predictors, in a way that each tree depends on the values of a random array, built independently and with the same distribution of the other trees in the forest (BREIMAN, 2001).

The ***Multi-Layer Perceptron*** (*MLP*) is an ***Artificial Neural Network*** classifier. *MLP* maps the input data into a set of appropriate outputs. Its structure is composed of one input layer, one output layer, and one of more layers between them.

Naïve Bayes (*NB*) is a ***Probabilistic*** classifier based on the Bayes theorem. Bayesian classifiers can predict the probability of a given object to belong to a specific class (HAN; KAMBER; PEI, 2011). Particularly, *NB* assumes that the effect of an attribute value in a given class is independent of the other attributes. This property is the conditional independence of a class, considered "naïve" because *NB* simplifies the corresponding calculations. Although very

simple, *NB* has shown to be accurate, and even comparable to other classifiers such as decision trees and a few neural networks (HAN; KAMBER; PEI, 2011). One of the advantages of *NB* is that it does not require a training set to obtain high classification results (SAMI; EL-BENDARY; HASSANIEN, 2012).

In this thesis we employed many of the aforementioned off-the-shelf classifiers. The classifiers used in our work that were not mentioned in this section will be introduced together with the proper results.

Clustering Algorithms

In the clustering task, algorithms perform unsupervised learning, and the class labels are previously known. Clustering is the process of organizing data into similar groups. Objects belonging to the same group have more similarity between them, and less similarity between the objects belonging to the other groups (HAN; KAMBER; PEI, 2011; ZAKI; MEIRA JR., 2014). Clustering algorithms can also be used as helping tools. For instance, they can be used to obtain knowledge regarding the data distribution, to observe the characteristics of each group for further analysis, and as a preprocessing tool to support other algorithms, such as classifiers. Distance functions compute the (dis)similarity among the objects being clustered.

Clustering algorithms can be divided into categories, namely representative-based (or partitioning-based) clustering, hierarchical, density-based, grid-based and model-based (ZAKI; MEIRA JR., 2014; FAHAD *et al.*, 2014). We briefly discuss each of these categories in the following.

Partitioning-based methods find mutual-exclusive groups based on distance. They normally represent the center of each group using an element determined by the mean or medoid of the elements within the group. The algorithms divide the data into partitions, each one representing a group. Each group is required to contain at least one element and to belong to exactly one group. Examples of algorithms from this category are ***k-means***, ***k-medoids***, ***PAM*** (KAUFMAN; ROUSSEEUW, 2005) and ***CLARANS*** (NG; HAN, 2002).

Hierarchy-based methods group the available data in a dendrogram, which consists of a hierarchical representation with multiple levels. The intermediary nodes determine the proximity among the elements, and leaf nodes store the data elements. The initial group of elements is gradually divided into other groups as the hierarchy continues. Algorithms from this category can employ a bottom-up or top-down approach. Bottom-up approaches initialize the groups with one object and then join two or more groups recursively. Top-down approaches start with all elements in one group and recursively divides them until a stopping criterion is satisfied (normally, until k groups are found). Examples of hierarchical algorithms are ***BIRCH*** (ZHANG; RAMAKRISHNAN; LIVNY, 1996) and ***CURE*** (GUHA; RASTOGI; SHIM, 1998).

Density-based methods find groups with arbitrary shapes and split the objects according

to their density regions, connectivity, and boundary. Groups are dense regions (with many data objects), separated by regions of low density, and present natural protection against the presence of outliers. ***DBScan*** (SANDER *et al.*, 1998) is an example of a density-based method.

Grid-based methods use a grid structure to divide the data space. The main advantage of these methods is the fast processing, which is normally independent of the number of objects. This occurs because the algorithm scans the dataset only once to compute the statistical values for the grids. Then, the grouping is performed based on the collected information. An example of grid-based algorithm is ***STING*** (WANG; YANG; MUNTZ, 1997).

Model-based methods optimize the adjust between the data and predefined mathematical models. The algorithms of this category are based on the premise that the data is generated by a mixture of underlying probability distributions. The number of groups is automatically determined based on the statistics and considering the outliers. ***MCLUST*** (FRALEY; RAFTERY, 1999) is an example of a model-based algorithm.

In this thesis we employed clustering algorithms to look for regions of images depicting similar visual patterns. Details about this are given in Chapter 4.

Evaluation measures

A mining algorithm needs to be evaluated to check how accurate and generalized it is for the classification/clustering of unseen examples. A trained model is applied to a test set of data (also labeled), and the results are evaluated. Many evaluation measures can be used to check how accurate is the model. Table 2 summarizes some of the most known measures, employed in this thesis. Here we refer as **positive examples** (P) the ones coming from the class of interest, and as **negative examples** (N) otherwise. Many of these measures rely on four terms: (i) **true positives** (TP) are positive examples correctly labeled as positive; (ii) **true negatives** (TN) are negative examples correctly labeled as negative; (iii) **false positives** (FP) are negative examples incorrectly labeled as positive; and **false negatives** (FN) are positive examples incorrectly labeled as negative.

Accuracy (AC) is the set of examples correctly classified. Error rate (ER) refers to the misclassification rate, and is corresponding to $1 - AC$. The true positive rate (TPR) is also known as sensitivity and recall, and refers to the proportion of positive examples correctly classified. The true negative rate (TNR) or specificity is the proportion of negative examples correctly classified. F_1 is the harmonic mean of precision and recall.

All evaluation measures used in this thesis and not introduced in this section are related to the domain, and will be explained along the specific results, in Chapters 3, 4 and 5.

Table 2 – Example of evaluation measures used to evaluate data mining algorithms.

Measure	***Acronym***	***Formula***
Accuracy	*AC*	$\frac{TP+TN}{P+N}$
Error rate	*ER*	$\frac{FP+FN}{P+N}$
Precision	*PR*	$\frac{TP}{TP+FP}$
True positive rate, sensitivity or recall	*TPR*	$\frac{TP}{P}$
True negative rate or specificity	*TNR*	$\frac{TN}{N}$
F-Measure	F_1	$2 \times PR \times TPR$

Source: Adapted from Han, Kamber and Pei (2011).

2.1.4 Content-Based Image Retrieval

Content-Based Image Retrieval (*CBIR*) techniques allow users to query for images that are similar to a given query center. *CBIR* techniques combine *FEMs*– that represent images as multidimensional objects, with *DFs*– that compare such objects. In this context, similarity queries are usually performed by two basic operators, ***k-NN* queries** and ***Range queries*** (TRAINA *et al.*, 2011). Figure 6 gives examples of (a) a *k-NN* query ($k = 6$) and a Range query (with radius ξ), and we formalize them as follows:

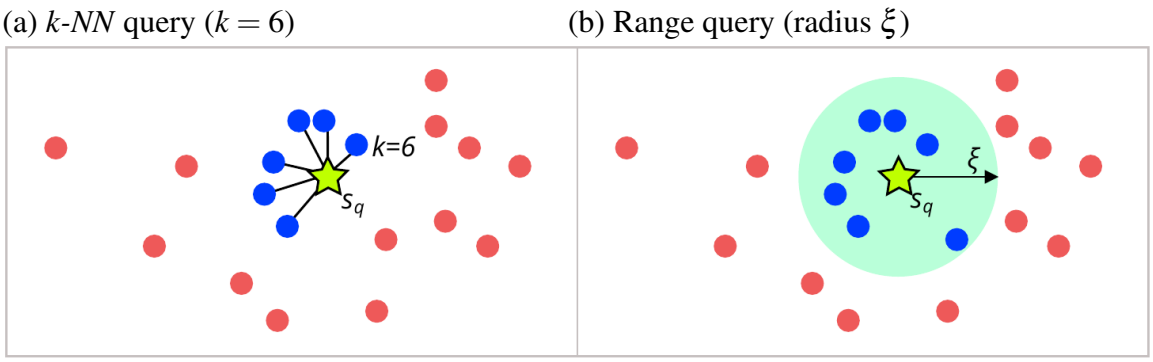
- ***k-NN query:*** Let $k\text{-}NNq(v_q, k)$ be the operator to represent a *k-NN* query, S be a set of elements, v_q be the feature vector of a query element $s_q \in S$, and k be the number of elements to be retrieved. The *k-NN* query retrieves the k most similar elements to s_q, sorted from the nearest (1st element) to the farthest (k-th element).

- ***Range query:*** Let $Rq(v_q, \xi)$ be the operator to represent a Range query, S be a set of elements, v_q be the feature vector of a query element $s_q \in S$, and ξ be the radius value. The range query retrieves all elements within the distance ξ, sorted from the nearest to the farthest element.

Classic similarity queries are usually performed by sequentially scanning all elements of the dataset. This includes the comparison (by a *DF*) of all elements to the query element. However, this approach is not indicated to big sets of data since it is time-consuming. Indexing approaches optimize these queries, using appropriate structures to speed-up the search.

Metric Access Methods (*MAMs*) divide the search space and support content-based queries. Examples of existing *MAMs* are *M-Tree* (CIACCIA; PATELLA; ZEZULA, 1997), *Slim-Tree* (TRAINA-JR. *et al.*, 2000) and *Onion-Tree* (CARÉLO *et al.*, 2009). For instance, *Slim-Tree* is a dynamic and balanced tree structure. It is constructed based on a bottom-up

Figure 6 – Examples of (a) *k-NN* and (b) Range query operators.



Source: Elaborated by the author.

approach, starting at the leaves and going towards the root. Like *M-Tree* and similar structures, *Slim-Tree* gather elements of a dataset in fixed-sized pages, which each page corresponds to a node of the tree (TRAINA-JR. *et al.*, 2000; SOUZA; CAZZOLATO; TRAINA, 2016). All *Slim-Tree* elements are kept in the leaves and arranged hierarchically in the tree. The structure uses representative elements as the centers of regions in the data space. Such regions have a coverage radius to determine which elements are inside the given region. These elements are associated with the node containing the representative element. Also, existing works focus on the query optimization of *MAMs* (SOUZA; RAZENTE; BARIONI, 2014; FILHO *et al.*, 2001; TRAINA-JR. *et al.*, 2007).

Speeding-up k-NN queries.

Brute-Force (*BF*) search algorithms are efficient for small datasets, but as the number of samples increases, *BF* becomes very costly and unfeasible. Let n be the number of elements in a dataset and d be the number of dimensions (data attributes). Only for datasets with $n < 30$ elements, $\log(n)$ is comparable to n, and the brute-force approach can overcome tree structures.

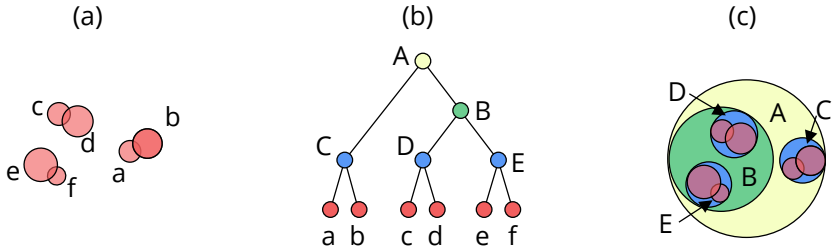
Diverse specialized structures have been proposed to speed-up the *k-NN* queries. Two examples of such structures are the *kd-Tree* and *Ball-Tree* algorithms. In this work, we used both algorithms to match points of moving objects along time, and we detail this in Chapter 5. Following, we briefly describe these structures.

kd-Tree stands for the k-dimensional tree, which is a generalization of the two-dimensional *Quad-Trees* and three-dimensional *Oct-Trees* (BENTLEY, 1975). It recursively partitions the data space along the data axes. The ***kd-Tree*** structure construction is known to be very fast, since no d-dimensional distance calculations are performed because the partitioning is made only along the data axes. *kd-Trees* perform *k-NN* queries on $\mathcal{O}(\log(n))$ distance computations. Also, *kd-Trees* have shown to perform very fast with $d < 20$, and it becomes inefficient as d grows. Recent works include the adaptation of the *kd-Tree* construction method using a point-based approach, in order to reduce the computational construction time of the structure (HUANG *et*

al., 2018). *kd-Trees* also have been used to estimate real-time earthquake parameter estimation (YIN; ANDREWS; HEATON, 2018).

Ball-Trees overcome the problem of with high dimensions of *kd-Tree* by partitioning the data space in a series of nesting hyper-spheres (OMOHUNDRO, 1989). The structure consists of a complete binary tree. A ball is a region bounded by a hyper-sphere in the n-dimensional *Euclidean* space $\mathbb{R}^n$. The construction of *Ball-Trees* is more costly than *kd-Trees*' structures. However, *Ball-Tree* has the advantage of being very efficient on highly structured data, even with high values of d. Recent work proposes an approach to determine the communication radius of an automatic light trap based on the balltree structure (PHUONG *et al.*, 2016).

Figure 7 – Example a *Ball-Tree* structure. From (a) a set of balls in the plane, (b) the corresponding binary tree is constructed, and (c) depicts the resulted balls in *Ball-Tree*. The colors indicate each level of the tree.



Source: Adapted from Omohundro (1989).

Dimensionality reduction

The high dimensionality of feature vectors obtained from sets of images is a recurrent problem. High dimensional data can bring difficulties to algorithms for data analysis and mining, except those who perform nonlinear classification (ZAKI; MEIRA JR., 2014). Thus, it is important to check if the dimensionality can be reduced, while the essential data properties are preserved. By selecting a set of attributes, the dimensionality reduction techniques remove all attributes that are irrelevant, weakly relevant or redundant (HAN; KAMBER; PEI, 2011).

Codification techniques try to obtain a "compressed" representation of the original data. Examples of such techniques are data compression (*Wavelets transform* and *Principal Component Analysis* (*PCA*)), attribute selection (by removing irrelevant attributes) and attribute construction (when a set of attributes is derived from the original dataset) (HAN; KAMBER; PEI, 2011). For instance, *PCA* transforms or projects the original data into a smaller space. It searches the r-dimensional base that best captures the data variance. The direction with more variance is projected, and is referred to as the first principal component (ZAKI; MEIRA JR., 2014). *PCA*'s orthogonal direction captures the second higher variance as the second principal component, and so on. The direction that maximizes the variance is also the one that minimizes the quadratic error.

The Theory of Fractals can also be used to reduce data dimensionality. In (TRAINA-JR. *et al.*, 2010) the authors proposed the use of the theory of fractals to approximate the intrinsic dimension of a dataset. Each dimension of a d-dimensional feature vector is represented as an attribute, and the proposed technique selects the most important attributes (dimensions) to describe the dataset. All attributes that do not change the fractal dimension are removed.

In this thesis we employed *PCA* to represent the information from images (their feature vectors) in lower dimensionality. In this context, *PCA* was used as a support to a visualizing tool, as well as a pre-processing technique for image segmentation and analysis, which we detail next.

2.2 Image Segmentation and Analysis

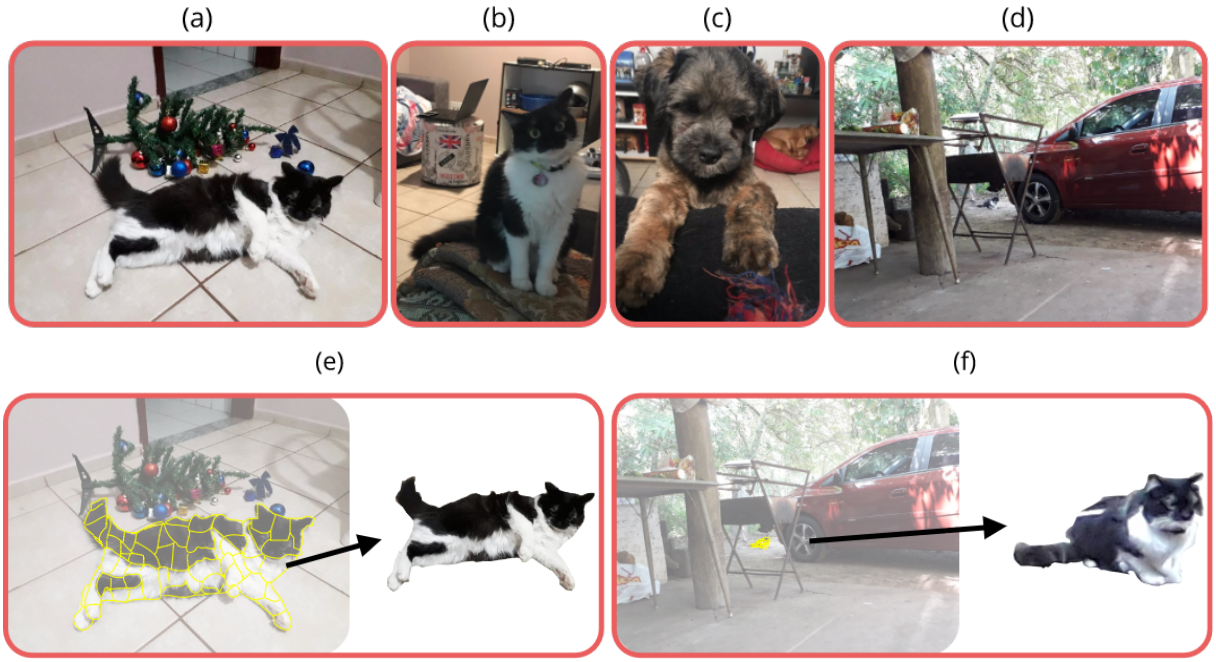
When dealing with entire images, many visual concepts can be depicted in the image beside the object of interest. For instance, consider a case where we want to classify images as <*cat*> (if there is a cat depicted in the image) or as <*not cat*>, otherwise. Figure 8 shows examples of images. While obvious options such as (a) and (b) show a cat, and thus belong to class <cat>, examples such as the (c) and (d) could easily be classified as <*not cat*>. However, notice that there *is* a cat in image (d), but not in evidence (we can find it in front of the car). In this case, the detection/classification is not trivial, even for human eyes. Accordingly, color-based global classifiers could use the color information of the car, for instance, and discard the possibility of the image to depict a cat. On the other hand, shape-based descriptors could mislead the classifier to label image (c) as <*cat*>. Options (e) and (f) show the object of interest segmented, what can help the classifier to label the image of interest.

In this thesis we are particularly interested in showing that, by observing specific regions of images or sequences of images, we can improve the overall analysis of visual features. This process includes the analysis of image regions, the composition of the image represented only by relevant regions (which is a segmented image), and then the analysis of the resulting image. Segmentation subdivides an image into its constituent regions or objects (PARKER, 2010). Segmenting nontrivial images is a difficult task in image processing. The segmentation can be crucial to image analysis, as it can determine the eventual success or failure of computerized analysis procedures.

Segmentation algorithms can work based on discontinuity when the image is partitioned based on abrupt changes and intensity, or based on similarity when the image is divided into regions that are similar to a given criterion (GONZALEZ; WOODS, 2008). In this work, we focused on the latter approach, ***segmentation based on similarity***. When segmenting according to the similarity, algorithms partition images into regions that are similar, according to a set of predefined criteria. Examples of such an approach are thresholding, region growing, and region splitting and merging.

Image segmentation can be carried in a vast range of techniques. In this work, we focus

Figure 8 – When global information is not sufficient: at first sight, only images (a) and (b) would be labeled as <*cat*>. However, image (d) also depicts a cat (in front of the car). Segmenting objects of interest (e and f) can improve the overall detection of visual patterns.



Source: Elaborated by the author.

on the segmentation of images based on visually-similar regions of pixels, called superpixels, that can be further analyzed considering the specific semantics of interest. We detail superpixels next.

Superpixels

A superpixel algorithm groups pixels into perceptually meaningful atomic regions. Such regions can be used to replace the image structure, based on the pixel grid. Superpixels capture image redundancy and can be used to extract image features while reducing the complexity of subsequent tasks regarding the image processing (ACHANTA *et al.*, 2012). Additionally, superpixel segmentation can be considered low or mid-level representations, that show to be useful to tasks like semantic segmentation, or generation of objects in detection systems (NEUBERT; PRÖTZEL, 2015). Existing superpixel algorithms can be categorized into graph-based and gradient-based approaches (or methods) (ACHANTA *et al.*, 2012; WANG *et al.*, 2017).

Table 3 shows examples of existing superpixels, with the original work, category, and computational complexity. In the corresponding computational complexities presented, B is the number of superpixels, P is the number of pixels, and a is a small constant. Figure 9 shows examples of segmented images using the mentioned approaches.

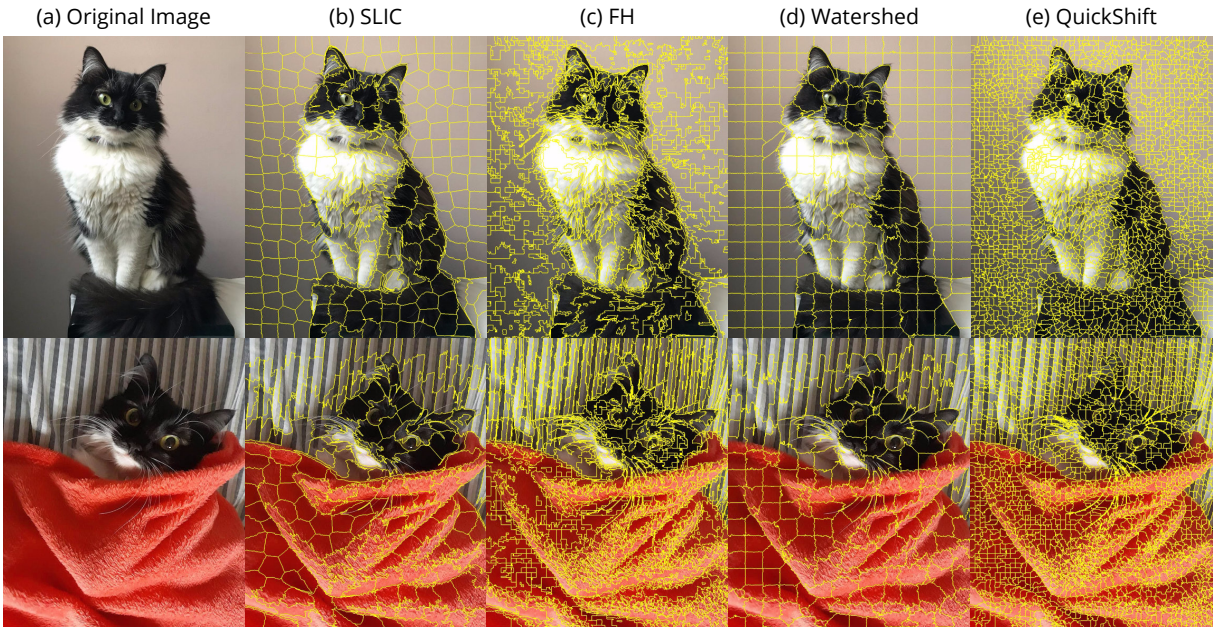
The ***Simple Linear Iterative Clustering*** (***SLIC***) was proposed in (ACHANTA *et al.*,

Table 3 – Comparison of superpixel algorithms from the literature.

Method	*Authors*	*Category*	*Complexity*
SLIC	(ACHANTA *et al.*, 2012)	Gradient-based	$\mathcal{O}(B)$
FH	(FELZENSZWALB; HUTTENLOCHER, 2004)	Graph-based	$\mathcal{O}(P \log P)$
Watershed	(VINCENT; SOILLE, 1991)	Gradient-based	$\mathcal{O}(P \log P)$
QuickShift	(VEDALDI; SOATTO, 2008)	Gradient-based	$\mathcal{O}(aP^2)$

Source: Adapted from Wang *et al.* (2017).

Figure 9 – Example of images depicting superpixel boundaries, generated with (b) *SLIC*, (c) FH, (d) Watershed and (e) QuickShift.



Source: Elaborated by the author.

2012). It takes as parameters b and e, which are respectively the *desired* number of superpixels and the compactness of the superpixels, and represents color images in the *CIELAB* color space. Let B' be the size of the superpixel, determined by $nsuperpixels' = \sqrt{P/b}$, where P is the number of pixels. *SLIC* consists of an adaptation of *k-means* for the superpixel regions generation, limiting the search space for pixels within a $2B' \times 2B'$ of distance to the center point. This makes *SLIC* very fast while presenting accurate results regarding the segmentation. The algorithm starts by sampling b initial cluster centers on a regular grid. Each pixel is assigned to a cluster center, and an update step adjusts the cluster centers, according to the lowest gradient position in a 3×3 neighborhood. After associating each pixel to the corresponding cluster, *SLIC* adjusts the cluster centers to be the mean vector of all the pixels belonging to the cluster. *SLIC* uses an error to repeat update steps until convergence is achieved. Also, a post-processing step enforces connectivity by re-assigning disjoint pixels to nearby superpixels.

FH was proposed in (FELZENSZWALB; HUTTENLOCHER, 2004). It uses a graph-

based representation to cluster pixels as nodes on a graph, in a way that each superpixel is the minimum spanning tree of the constituent pixels. *FH* has shown to adhere well to image boundaries, but produces superpixels with irregular sizes and shapes (ACHANTA *et al.*, 2012). However, it does not offer control regarding the number of superpixels and their compactness.

The ***Watershed*** technique was proposed in (VINCENT; SOILLE, 1991). It performs a gradient ascent from local minima to produce watersheds, which are lines that separate catchment basins. In (HU; ZOU; LI, 2015) the authors propose an efficient watershed-based superpixel approach, called *SCoW* (Spatial-Constrained Watershed). The authors use an edge-preserving scheme to align superpixel boundaries to image edges. Thus, they obtained a balance between the homogeneity and the compactness. *SCoW* provides control regarding the number of superpixels or their compactness, and the approach produced high-quality superpixels as traditional superpixel algorithms.

Finally, ***QuickShift*** was proposed in (VEDALDI; SOATTO, 2008). It consists of a mode-seeking segmentation method, that moves each point in the feature space to the nearest neighbor that increases the Parzen density estimation (WANG *et al.*, 2017). *QuickShift* requires a set of non-intuitive parameters, that need to be tuned (ACHANTA *et al.*, 2012). Also, it does not provide the option to control the size or the number of superpixels and is relatively slow.

Other examples of superpixel algorithms are *SEEDS*, *N-cut* and *MeanShift* (WANG *et al.*, 2017). In this thesis we used *SLIC* superpixel algorithm as a tool to obtain color-coherent regions of images, as we show in Chapters 3 and 4.

2.3 Modeling Patterns with Uncertainty

All concepts covered up to this section refer to the analysis and assignment of objects according to specific labels. However, many real-world problems rely on uncertainty, and only one label is not always sufficient to describe the content of an object. Accordingly, in this section we discuss how a specific object can participate in multiple groups (clusters) in a probabilistic manner. The **probability theory** is a key concept in the field of pattern recognition (BISHOP, 2007). It provides concepts to deal with uncertainty, and we describe some of the important points in this section.

One can say that the goal of cluster analysis is to find hidden categories within the data. Such categories can be inferred from the data, without the necessity of actual labels for the objects. Then, we can assume that the inferred categories are a distribution over the data space. Specifically, this distribution can be mathematically represented using a ***Probability Distribution Function*** (*PDF*), also referred to as a distribution function (HAN; KAMBER; PEI, 2011).

In this context, the ***Normal Distribution*** (*ND*) – also known as *Gaussian Distribution* – is one of the most important *PDFs*. This is especially because many physically observed variables

follow an approximately *ND*. Accordingly, a sampling distribution of an arbitrary probability distribution of the means follows a *ND*. The *ND* also plays an important role as the parametric distribution of choice in clustering, density estimation, and classification (ZAKI; MEIRA JR., 2014).

Following we review the basic concepts of *NDs*, univariate, and multivariate analysis. Then, we briefly discuss how such concepts can be employed in the analysis of image regions, which is within the scope of this thesis.

2.3.1 Normal Distribution: Basic Concepts

In this section we introduce basic statistical concepts regarding the *ND*. The following concepts were described based on (ZAKI; MEIRA JR., 2014), (HAN; KAMBER; PEI, 2011), (RENCHER; CHRISTENSEN, 2012) and (BISHOP, 2007). Given the following $n \times d$ matrix:

$$D = \begin{pmatrix} X_1 & X_2 & \cdots & X_d \\ \hline x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix},$$

each row represents a data object x_i in the d-dimensional space, such that $x_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T \in \mathbb{R}^d$. This is called the *row view* of the matrix. Similarly, each column represents a data object X_j in the n-dimensional space, such that $X_j = (x_{1j}, x_{2j}, \cdots, x_{nj})^T \in \mathbb{R}^n$. This is called the *column view of the matrix*. Finally, in the *probabilistic view* the d attributes are modeled as a vector random variable, $X = (X_1, X_2, \cdots, X_d)^T$, and the points x_i are independent and identically distributed as X.

We denote the *expectation* value of a function f as $E[f]$. For instance, the average value of a given function $f(x)$ under a probability distribution $p(x)$ is denoted by $E[f]$, and is given by (for a discrete distribution):

$$E[f] = \sum_x p(x) f(x)$$

Following, the *sample mean* is given by Equation 2.2, the *sample variance* is given by Equation 2.3, and the *sample standard deviation* is given by Equation 2.4.

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \quad (2.2) \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2 \quad (2.3) \qquad \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2} \quad (2.4)$$

Given two random variables X_1 and X_2, *the covariance* Σ is given by Equation 2.5. Since $\sigma_{12} = \sigma_{21}$, Σ is a symmetric matrix.

$$\Sigma = E[(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T] = E\left[\begin{pmatrix} X_1-\mu_1 \\ X_2-\mu_2 \end{pmatrix} \quad \begin{pmatrix} X_1-\mu_1 & X_2-\mu_2 \end{pmatrix}\right] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix} \tag{2.5}$$

The *multivariate mean* vector is the mean of each attribute, given as Equation 2.6.

$$\mu = E[\mathbf{X}] = \begin{pmatrix} E[X_1] \\ E[X_2] \\ \vdots \\ E[X_d] \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{pmatrix} \tag{2.6}$$

Accordingly, the *multivariate covariance* is given by the $d \times d$ symmetric covariance matrix, which gives the covariance value for each pair of attributes as Equation 2.7. Here, the diagonal element σ_i^2 specifies the attribute variance for X_i. Also, the off-diagonal elements $\sigma_{ij} = \sigma_{ji}$ represent the covariance between the pair of objects X_i and X_j.

$$\Sigma = E[(\mathbf{X}-\mu)(\mathbf{X}-\mu)^{\mathbf{T}}] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{pmatrix} \tag{2.7}$$

Next, we describe the main concepts related to the univariate normal distribution.

Univariate Normal Distribution

Given the mean μ and the variance σ^2, a random variable X follows a ***Univariate Normal Distribution*** (*UND*) if its *PDF* of X satisfies Equation 2.8 as follows:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}. \tag{2.8}$$

Here, $(x-\mu)^2$ is the distance value of x from the distribution mean μ. Consequently, the probability density decreases exponentially as a function of the distance from the mean. The maximum value of the density occurs when $x = \mu$, since $f(\mu)$ is inversely proportional to the σ of the distribution. Following we define the multivariate normal distribution.

Multivariate Normal Distribution

Let $X = (X_1, X_2, \cdots, X_d)^T$ be a d-dimensional vector random variable. X has a ***Multivariate Normal Distribution*** (*MND*) if its joint multivariate *PDF* is given as Equation 2.9, where μ

is the mean and Σ is the covariance matrix.

$$f(x|\mu,\Sigma) = \frac{1}{(\sqrt{2\pi}^2)\sqrt{|\Sigma|}} \exp\left\{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}\right\} \tag{2.9}$$

Here, $|\Sigma|$ is the determinant of the covariance matrix. Similarly to the univariate case, $(x-\mu)^T\Sigma^{-1}(x-\mu)$ measures the ***Mahalanobis*** distance of x to the mean μ of the distribution, considering all the variance-covariance information between the attributes. The *Mahalanobis* distance is a generalization of the *Euclidean* distance, since *Euclidean* ignores the covariance information between attributes, whereas the *Mahalanobis* distance explicitly considers it.

2.3.2 Gaussian Mixture Models and the EM Algorithm

A ***Mixture Model*** (*MM*) assumes that a set of observed objects is a mixture of instances that come from multiple probabilistic groups (clusters). In a *MM*, each observed object is independently generated by two steps. The first chooses a probabilistic cluster according to the probabilities of the clusters. The second step selects a sample according to the *PDF* of the chosen cluster (HAN; KAMBER; PEI, 2011).

The *k-means* algorithm works in a greedy manner, minimizing the squared error of points from their respective cluster means. Accordingly, it is said that *k-means* performs *hard clustering* since each point is assigned to only one cluster. However, as mentioned at the beginning of Section 2.3, the uncertainty is present in many real-world problems, and sometimes only one label is not sufficient to describe the data content.

Let $\{K_1,\cdots,K_k\}$ specify the partitioning of n points into k clusters, in which each cluster $K_i|1 \leq i \leq k$ is characterized by a *MND* $f_i(x)$, as given by Equation 2.9. In this case, the cluster mean $\mu_i \in \mathbb{R}^d$ and the covariance matrix $\Sigma_i \in \mathbb{R}^{d\times d}$ are unknown parameters. Also, it is assumed that the *PDF* of X is given as a ***Gaussian Mixture Model*** (*GMM*) over k cluster normals, as Equation 2.10. The prior probabilities $P(K_i)$ are the mixture parameters, and they must satisfy the condition of Equation 2.11. Finally, *GMM* is characterized by the mean μ_i, the covariance matrix Σ_i and the mixture probability $P(K_i)$ for each of the k *NDs*. Equation 2.12 shows the model parameters compactly.

$$f(x) = \sum_{i=1}^{k} f_i(x)P(K_i) = \sum_{i=1}^{k} f(x|\mu_i,\Sigma_i)P(K_i) \tag{2.10}$$

$$\sum_{i=1}^{k} P(K_i) = 1 \tag{2.11}$$

$$\theta = \{\mu_1,\Sigma_1,P(K_1),\cdots,\mu_k,\Sigma_k,P(K_k)\} \tag{2.12}$$

Let $P(D|\theta)$ denote the *likelihood* of θ as the conditional probability of a given dataset D and the model parameters θ. Since all n points x_j are considered random samples of X, the likelihood of θ is given by Equation 2.13. Here, the goal of the ***Maximum Likelihood Estimation*** (*MLE*) is to choose the parameters θ that maximize the likelihood, as given by Equation 2.14.

$$P(D|\theta) = \prod_{j=1}^{n} f(x_j) \qquad (2.13) \qquad\qquad \theta^* = \arg\max_{\theta}\{P(D|\theta)\} \qquad (2.14)$$

MLE maximizes the likelihood function since it turns the product over the points into a summation of the maximum value of the likelihood and log-likelihood coincide (ZAKI; MEIRA JR., 2014). Since this maximizing task is hard, the *EM* algorithm is used.

The Expectation Maximization Algorithm

The ***Expectation Maximization*** (*EM*) algorithm generalizes *k-means* by modeling the data as a mixture of normal distributions (ZAKI; MEIRA JR., 2014). *EM* finds the mean and covariance matrix cluster parameters by maximizing the likelihood of the data. Unlike *k-means*, which makes hard assignments, *EM* is a *soft clustering* approach that returns the probability that a point belongs to each existing cluster.

Regarding fuzzy and probabilistic model-based clustering, *EM* starts with an initial set of parameters and iterates until the clustering converges (HAN; KAMBER; PEI, 2011). This means that the resulting clusters cannot be improved, or the changes are smaller than a threshold. Each *EM* iteration consists of two steps, namely the expectation and the maximization. During the expectation, *EM* assigns objects to clusters according to the current parameters of probabilistic clusters. Then, during the maximization, *EM* finds the new clustering or parameters that maximize the expected likelihood in probabilistic model-based clustering (HAN; KAMBER; PEI, 2011).

Modeling Complex Objects with GMM

The concepts regarding Normal Distributions presented in this section can be applied to model complex data. As mentioned before, many real-world applications show a behavior similar to the *ND PDF*. Accordingly, in this thesis we show that similar behavior can be observed when modeling regions of images. When dealing with uncertainty, performing a **hard clustering** or single-label assignment (for instance, by classification) is neither sufficient nor accurate since a specific region of an image can depict more than one visual finding. Thus, a **soft analysis** is recommended to consider the likelihood of image regions to contain each of the available labels. For this purpose, feature vectors extracted from each image region are modeled using Gaussian Mixture Models. We represent each image region as an object and compute its likelihood to belong to each of the modeled groups. As a result, we have multiple labels that can be assigned to a single object, with different probabilities.

Chapter 4 discuss the use of Gaussian Mixture Models for this context, with the application of the concepts described in this section in images from the medical context.

2.4 Moving Objects

Moving objects are basically geometries that change over time (GÜTING; SCHNEIDER, 2005). In this context, time is generally perceived as a one-dimensional space, considered from the past to the future. As time can be modeled as a bounded or infinite information, we are interested in bounded time models, since they assume some origin and an end of time.

Many applications can be represented as moving objects, such as hurricanes depicted by satellites, pedestrian and vehicle trajectories prediction, and tracking of biological objects, such as cells and particles. As stated in (LI, 2017), moving objects can be viewed in two forms. The first form is as *moving points*, that describe objects for which only the time-dependent position is of interest. The second form is as *moving regions*, which have both time-dependent position and spatial extent taken into account.

The movement of objects and its concise representation, combined with uncertainty, have been of interest to the database community in the past decade. In (TAO *et al.*, 2004), the authors focused on the problem of predicting the motion pattern. They observed that individual trajectories may vary significantly, but most motion types show self-similar behavior. Accordingly, they argue that one can often predict the current location of individual trajectories by looking at the ones in the recent past. They propose a framework that indexes object locations and processes queries based on the estimated position of objects. In (ALMUHISEN; DURAND; QUAFAFOU, 2018) the authors characterize the behavior of moving objects obtained by GPS locations of smartphones, cars, etc. They employed frequent pattern mining to mine correspondences between hidden patterns and trajectories. For this purpose, they tagged city maps to visualize the behavior of different spatio-temporal values.

Differently, in (SALTENIS *et al.*, 2000) the authors modeled positions of moving objects as functions of time. They argue that modeling the movement of objects both facilitates predictions and solves the problem of frequent updates, which are required to approximate continuous movement in a traditional setting.

In this thesis we borrowed these ideas to target microscopic images. We show in Chapter 5 that developing cells can be represented as a specific kind of moving objects, and we take advantage of this useful representation to track and predict cells over time.

2.5 Final Considerations

In this chapter, we presented the main concepts related to this thesis. All concepts and related works not presented herein will be introduced in the corresponding chapters, when the application context is introduced. The existing techniques to represent and extract knowledge from complex data give support to a wide range of applications.

This work focuses on the analysis of images and sequences of images. **Our thesis** is that

by **taking advantage of small representations of images** we can **improve the overall results of different image analysis tasks**. To prove this, we start by segmenting images into small regions. One of the approaches used for this task was the superpixel extraction. Once we have the representation of different regions of the images, in the following we performed classification, clustering and retrieval tasks on them.

Regarding images, the classification and clustering tasks were performed using both hard and soft analysis. For the hard analysis, Chapter 3 shows how we employed off-the-shelf classifiers, such as *Naïve Bayes*, *SVM*, and *RF*, to segment images and consequently improve the global classification of images using the segmentation outputs. For the soft analysis, Chapter 4 describes how we used Gaussian Mixture Models to classify and highlight regions of images presenting uncertainty, related to their corresponding labels. Regarding sequences of images, Chapter 5 shows how the concepts of moving objects, combined with segmentation and complex data representation, can improve the task of tracking objects over time.

The literature covered in this chapter is quite broad regarding the subjects presented. It was not our intent to exhaustively discuss all of these subjects. Rather, the goal was to cover the required background and knowledge to understand the contributions of this Ph.D. research, presented in the next chapters.

CHAPTER

3

REGION-BASED IMAGE CLASSIFICATION

In this chapter, we approach the *first research problem* of this Ph.D. research, which regards image mining from emergency scenarios, taking advantage of region-based approaches. The corresponding contributions were published, and are available in the works (CAZZOLATO *et al.*, 2016) and (CAZZOLATO *et al.*, 2017).

An intense flow of information is gathered in a short period in large-scale events such as the Brazilian street carnival, the FIFA Football World Cup and the Olympic Games. When emergency situations occur in such contexts, public authorities must be able to provide fast and accurate responses to emergencies. Dealing with such an amount of information in real-time is a challenging task. Computer systems have great potential to support the decision making process, providing timely information and data analysis (HUANG; CHENG; CHIU, 2013). In this context, the early detection of fire, smoke, and explosions can assist rescue forces in preventing further risks to human life, thus reducing financial and patrimonial losses.

Generally, the available technology for fire and smoke detection are based on hardwired solutions. They employ infrared or ultraviolet local sensors to determine the chemical presence of fire and/or of smoke particles, also relying on temperature measures. However, such solutions can be unsuitable to crowded urban areas or large open spaces, not only because of environment difficulties, but also due to the high cost of installation, communication, and maintenance of local sensors (CELIK; ÖZKARAMANH; DEMIREL, 2007).

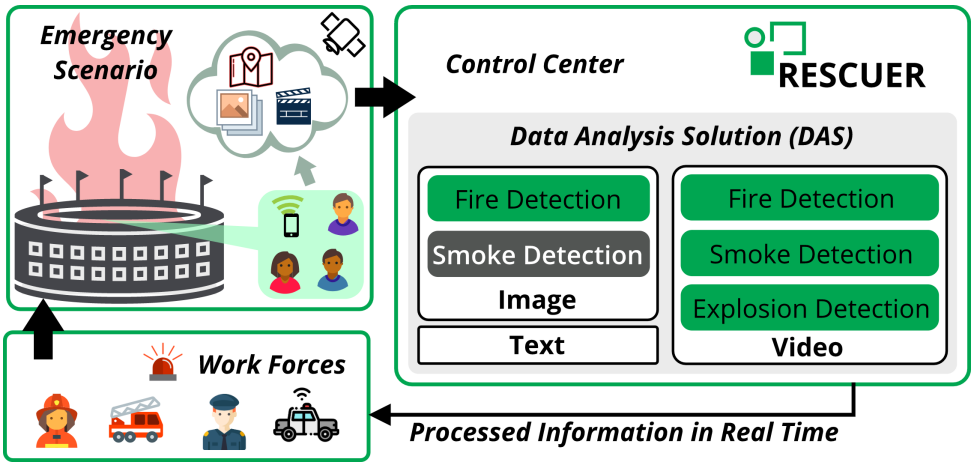
3.1 Context Application

Cameras embedded in mobile devices can provide digital visual information that can be an alternative to local fire/smoke sensors (CHEN *et al.*, 2006). For this, social networks can be used to gather images, videos, and textual information regarding the emergency event, shared by people involved in the incident. This is the core of the ***RESCUER*** (Reliable and

Smart Crowdsourcing Solution for Emergency and Crisis Management) project[1], which aimed at developing a system that uses crowdsourcing images, videos, and text sent by mobile devices to support decision making during emergencies. Figure 10 gives an overview of *RESCUER*.

This chapter focuses on the problem of detecting smoke in still images, obtained from emergency situations. The results presented herein were incorporated to the *RESCUER* system, in the data analysis solution (*Rescuer-DAS*), aimed at identifying the pieces of information presenting useful data regarding fire, smoke and/or explosion. Accordingly, we propose *SmokeBlock* to detect smoke, relying on region-based image analysis.

Figure 10 – Overview of the *RESCUER* Project: users send images, videos, text, and location information gathered from the emergency scenario. The *RESCUER* control center processes the available data, feeding the workforces with processed information in real-time.



Source: Elaborated by the author.

3.2 Motivation and Problem Definition

One of the main challenges regarding smoke detection in still images is the absence of movement. Existing techniques depend on a set of sequential images (video) to identify fire and/or smoke (CELIK; ÖZKARAMANH; DEMIREL, 2007). Unlike fire detection, which can be solved by global analysis using rule-based color models (CALDERARA; PICCININI; CUCCHIARA, 2011), the smoke identification problem does not present well-defined color patterns, since it changes as its temperature increases. Besides, the smoke color is heavily dependent on weather conditions and the material being burnt. Video-based strategies present good results for fire and smoke detection, but results of smoke identification in images are rather modest (CELIK; ÖZKARAMANH; DEMIREL, 2007).

To overcome the challenges of smoke detection on still images, we employ a region-based analysis through superpixel segmentation. As mentioned in Section 2.2, a superpixel region

[1] The *RESCUER* Project: <www.rescuer-project.org>

of an image corresponds to a group of pixels presenting coherent patterns according to some visual features. Superpixels capture redundancy and reduce the complexity of subsequent image processing, having proved to be useful for depth estimation, image segmentation, skeletonization, body model estimation, and object localization (ACHANTA *et al.*, 2010). In this work, we use color and texture visual features to represent the images and their regions.

We are interested in solving the following problem: *given a set of images obtained from social media or crowdsourcing, find the subset of images that depict smoke while minimizing the rate of false positives.* We refer to the images that depict smoke as *positive examples* while images without visual traces of smoke are *negative examples*. The objective here is to find positive examples so that humans or resource-intensive computer vision methods can focus on the analysis of such images. Accordingly, our secondary goal is to *minimize the number of false positives*, i.e., images that do not depict smoke but are classified as smoke.

3.3 Related Work

Previous approaches for smoke detection were designed for video data relying on motion analysis, which is not suitable for accurate smoke detection in still images (CELIK; DEMIREL, 2009; TIAN *et al.*, 2014). *SmokeBlock* improves the state-of-the-art by using a combination of color and texture to reduce false positives. Also, the method relies on a small set of parameters that do not depend on the user to tune.

To the best of our knowledge, the majority of reported techniques for smoke detection is related to video analysis (BOHUSH; BROUKA, 2013; CALDERARA; PICCININI; CUCCHIARA, 2011; TIAN *et al.*, 2014). Video analysis relies on a pipeline processing composed of two stages, namely static and motion analysis. In the static analysis stage, each frame is processed as a still image. Then, the techniques conduct the motion analysis stage over the temporal features of the video (BOHUSH; BROUKA, 2013).

In (CELIK; ÖZKARAMANH; DEMIREL, 2007) the authors employed a two-rule color model to classify the pixels as smoke, relying on the RGB color space. The proposal (CHEN *et al.*, 2006) follows the same rationale, as it employs a three-rule color model to classify each pixel represented in the HSI color space. Both works (CELIK; ÖZKARAMANH; DEMIREL, 2007) and (CHEN *et al.*, 2006) construct models that are limited by the coarse correspondence between the rule-based color models and the smoke characterization. This results in a large number of false-positive frames. Therefore, in such approaches, Motion Analysis is required to produce acceptable results (VARADARAJAN; MILLER; ZHOU, 2015), reducing the number of false positives by employing frame difference and other motion strategies (TIAN *et al.*, 2014).

To avoid the aforementioned limitations of rule-based color approaches – which are adequate only for video analysis, we employ a superpixel local segmentation along with color and texture features simultaneously. We detail our approach next.

3.4 The *SmokeBlock* Algorithm

SmokeBlock detects smoke on still images without any information about time/motion. The method's pipeline encompasses the preprocessing step, with the superpixel and feature extraction tasks, and the classification step, where the region-based smoke detection is performed, along with the global smoke classification. Algorithm 1 presents *SmokeBlock*, and we describe each of its steps next.

Algorithm 1: *SmokeBlock*

Input : Unlabeled image I
Superpixel classifier $\mathscr{C}_Q$
Image classifier $\mathscr{C}_I$

Output : Image label $\ell \in \{smoke, not\text{-}smoke\}$

1 $I_{seg} \leftarrow I$; ▷ *Initialize the segmented image*
2 $Q \leftarrow SLIC(I, b, e)$; ▷ *$b$ is the desired number of superpixels, e is the compactness*
3 **for** $q_i \in Q$ **do**
4 **if** $\mathscr{C}_Q(q_i) \neq smoke$ **then**
5 $I_{seg}[q_i] \leftarrow \emptyset$; ▷ *Remove pixels from not-smoke regions*
6 $v_{\text{color}} \leftarrow \mathscr{E}_{\text{color}}(I_{\text{seg}})$; ▷ *Extract color from segmented image*
7 $v_{\text{texture}} \leftarrow \mathscr{E}_{\text{texture}}(I_{\text{seg}})$; ▷ *Extract texture from segmented image*
8 $v \leftarrow v_{\text{color}} \oplus v_{\text{texture}}$; ▷ *Concatenate color and texture*
9 $\ell \leftarrow \mathscr{C}_I(v)$; ▷ *Classify segmented image with color and texture*
10 **return** I_{seg}, ℓ; ▷ *Return segmented image and predicted label*

SmokeBlock: Preprocessing

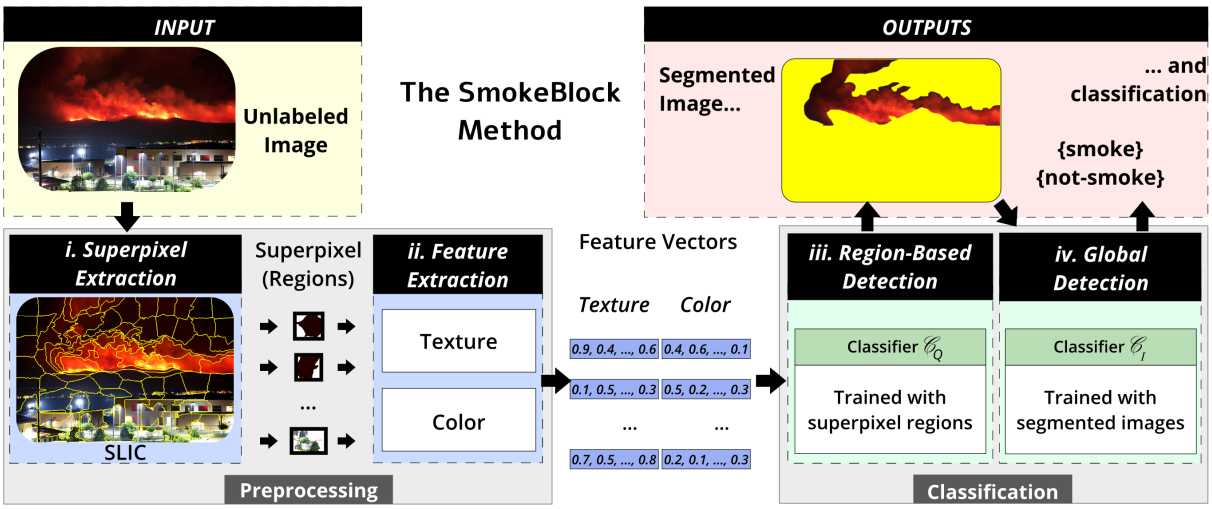
Figure 11 depicts the pipeline of *SmokeBlock*. Given an input image I that is unlabeled, (i) *SmokeBlock* starts by extracting the superpixels using *SLIC* algorithm. This results in a set Q of B superpixels $q_i \in Q$ that potentially contain traces of smoke. Then, (ii) the method extracts color and texture features from each superpixel region, resulting in a set of feature vectors.

By extracting features from superpixels instead of individual pixels, we reduce the redundancy, since pixels with similar visual properties that are spatially close will be processed together (as a single superpixel), thus improving performance. Also, the use of superpixels allows us to group pixels and extract texture features, which is not possible for individual pixels. *SmokeBlock* concatenates the color and texture feature vectors and forwards them to the classification step, described next.

SmokeBlock: Classification

As depicted in Figure 11-iii, *SmokeBlock* performs a **Region-Based Classification** over each superpixel region. For this, *SmokeBlock* employs a classifier that was previously trained over

Figure 11 – The *SmokeBlock* method: given an input image, (i) *SmokeBlock* employs *SLIC* to extract superpixels regions. For each superpixel region, (ii) *SmokeBlock* extracts color and texture features, outputting the corresponding feature vectors. Then, (iii) each region is classified as smoke or not smoke, and segmented image is outputed with only smoke regions. Finally, (iv) *SmokeBlock* performs the global classification relying on the segmented image.



Source: Adapted from Cazzolato *et al.* (2016).

a set T of manually annotated superpixels, considering the set of labels $\mathscr{L} = \{smoke, not\text{-}smoke\}$. We report the image annotation task in Section 3.5.1. We trained a classifier $\mathscr{C}_Q : T \rightarrow \ell$ that decides whether a superpixel region depicts smoke or not, based on its feature vector. Accordingly, ***SmokeBlock*'s first output is a segmented image**, composed of only regions classified as smoke.

In the (iv) **Global Smoke Detection** step, *SmokeBlock* classifies the image as smoke or not smoke. This can be useful in a scenario with a large set of images gathered from a social media service, and we want to find images that are likely to contain smoke. A naive approach would be to use *SmokeBlock*'s superpixel regions and classify an image as positive (suspicious) if at least one superpixel was classified as depicting smoke. The drawback is that if a single superpixel is wrongly classified, the entire image will also be wrongly classified as containing smoke.

Accordingly, *SmokeBlock* extracts a new set of feature vectors, now regarding only the regions of I that were segmented by the Region-Based Classification step. Then, the method builds a final representation of the segmented image using the Principal Component Analysis (*PCA*) dimensionality reduction method, over the feature vectors. The Classification Model (iv) relies on a supervised classifier $\mathscr{C}_I$ to label the segmented image. This approach is superior when compared to global feature extraction (see experimental results in the next section. This occurs because *SmokeBlock* only extracts features from regions of the image that already is a candidate for having smoke.

SmokeBlock's modularized scheme allows the addition of new feature extractors and

classifiers. Moreover, the method can be integrated with existing fire detection methods, which enables the development of a *framework* for fire and smoke detection. Notice that *SmokeBlock* uses *Haralick* (*Hr*) and *Color Layout* (*CL*) *FEMs*, and a supervised classifier. These settings were obtained experimentally, as we detail next.

3.5 Experimental Analysis

In this section, we detail the experiments performed to define the best settings of *SmokeBlock*. The evaluation comprises the impact of color, texture, and shape extractors for smoke segmentation. We compare the results to the rule-based color models of *Celik* and *Chen*. The global classification is also evaluated, by comparing *SmokeBlock*'s precision and the number of false positives against other global approaches. With this experimental analysis, we aim to determine:

1. *The most suitable low-level features for SmokeBlock:* we evaluate the impact of color, shape, and texture *FEMs* to represent the candidate regions for smoke detection. The best settings are used in the remaining experiments;

2. *SmokeBlock's accuracy for smoke segmentation:* we compare *SmokeBlock* against state-of-the-art competitors, discussing improvements and limitations of our method for smoke segmentation;

3. *SmokeBlock's performance for smoke detection:* we evaluate the superpixel-based classification of our method in comparison to global classification, regarding *F-Measure* (F_1), *Precision* (*PR*) and *Recall* (*RE* or *TPR*).

Environment Setup. All experiments were performed in an Intel Core i7-4770(3.40GHz), 16GB RAM, Ubuntu 14.04 (64-bit) OS machine. *SLIC* parameter b – the desired number of superpixels – was set to 100, according to empirical tests.

3.5.1 Material

We simulate *RESCUER*'s environment using the Flickr API[2] to download images from Flickr, under the Creative Commons license. All images were retrieved using textual queries such as "smoke fire", or "smoke forest". Figure 12 shows examples of images from the created dataset, which we named *Flickr-Smoke*.

Notice that even with queries related to smoke, some of the images did not contain visual traces of smoke. Each image was manually annotated to build the ground-truth for evaluation. The annotation was performed by seven subjects aging between 20 and 30 years, familiar with

[2] The Flickr API: <www.flickr.com/services/api/>.

the issue, and non-color-blinded. In the first annotation round, two subjects labeled each image. Next, images with divergent annotations were labeled by a third subject to untie the classification. The average disagreement was 7.2%. The resulting dataset contains 832 images labeled as *smoke* and 834 as *not smoke*.

Flickr-Smoke was built in the context of the *RESCUER* project, in collaboration with colleges from the Databases and Images Group (GBdI). Now *Flickr-Smoke* is a subset of a bigger dataset, named *FiSmo*. *FiSmo* contains images and videos gathered from emergency situations and is available online. Additional information can be found in the work (CAZZOLATO *et al.*, 2017). *FiSmo* is available at <github.com/mtcazzolato/dsw2017>. *SmokeBlock* was coded in C++, using OpenCV and Artemis libraries. The complete source code is available at <github.com/mtcazzolato/smokeblock.git>.

Figure 12 – Examples of images from the *Flickr-Smoke* dataset.

Source: Adapted from Cazzolato *et al.* (2017).

3.5.2 Finding the Best Configuration for SmokeBlock

The first aspect we analyze is the content representation of the superpixels, which depends on the employed *FEM*. Accordingly, we address the first question:

Q1. Which are the most suitable low-level visual features to detect smoke in still images?

We experimented with a set of ten widely employed *FEMs* regarding color, texture and shape. Table 4 summarizes the *FEMs* and corresponding acronyms, that are available in the Artemis image processing library[3].

We evaluate the representativeness of each extractor with a subset of 1,202 manually labeled superpixels from *Flickr-Smoke*. Half of them are from class *smoke* and the half are from *not smoke*, and Figure 13 shows a few examples. We caried the evaluation of *FEMs* using the *Naïve-Bayes* (*NB*) classifier. We computed the *F-Measure* obtained by each feature extractor using ten-fold cross validation. Figure 14 shows the results regarding the *F-Measure* values

[3] Artemis library: <gbdi.icmc.usp.br>

Table 4 – *FEMs* tested for *SmokeBlock*.

Visual Feature	***FEM***	***Acronym***	***Details***
Color	*Color Layout*	*CL*	16 dimensions, 8 from channel Y (luminance), 4 from channel Cr (red), and 4 from channel Cb (blue)
	Color Structure	*CS*	128 dimensions of quantized values
	Color Temperature	*CT*	3 dimensions, from XYZ color space
	Scalable Color	*SC*	256 dimensions, from HSV color space
	Normalized Histogram	*NH*	256 dimensions, in grayscale
Texture	*Edge Histogram*	*EH*	150 dimensions, with 30 clusters
	Haralick	*Hr*	24 dimensions, with 4 angles of each feature: variance, entropy, uniformity, homogeinity, moment2thMatrix and reverse variance
	Local Binary Patterns	*LBP*	177 dimensions, 8-neighbors pixel for each channel, with 58 uniform texture codes and 1 non-uniform
	Texture Spectrum	*TS*	8 dimensions
Shape	*Zernike*	*Zr*	38 dimensions, Zernike moments

Source: Elaborated by the author.

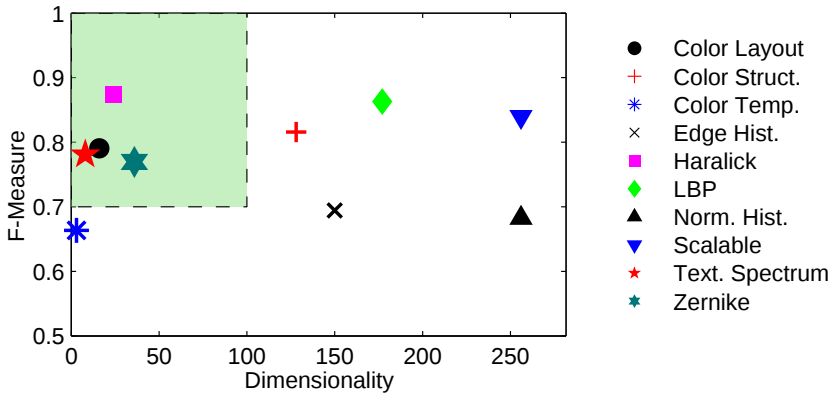
Figure 13 – Examples of superpixels manually labeled, used to train the machine learning classifiers employed in our methodology.

Source: Adapted from Cazzolato *et al.* (2017).

and the dimensionality of each *FEM*. The best extractors (yellow region) are those with high performance and low dimensionality (high compactness).

Low-level features with compact representation are those more suitable to deal with the "dimensionality curse". The results illustrated by Figure 14 provide the answer for question *Q1*. Hence, based on this plot, we selected *CL* and *Hr FEMs*, which represent respectively color and texture, and have a ballance between low dimensionality and high *F-Measure*.

Figure 14 – Comparison of *FEMs* regarding performance (*F-Measure*) and dimensionality. The best *FEMs* have high performance and low dimensionality, and are placed inside the green area.



Source: Adapted from Cazzolato *et al.* (2016).

3.5.3 Smoke Segmentation

In this section, we describe a qualitative analysis performed to answer to the second question:

Q2. Is SmokeBlock more suitable than existing approaches for smoke segmentation?

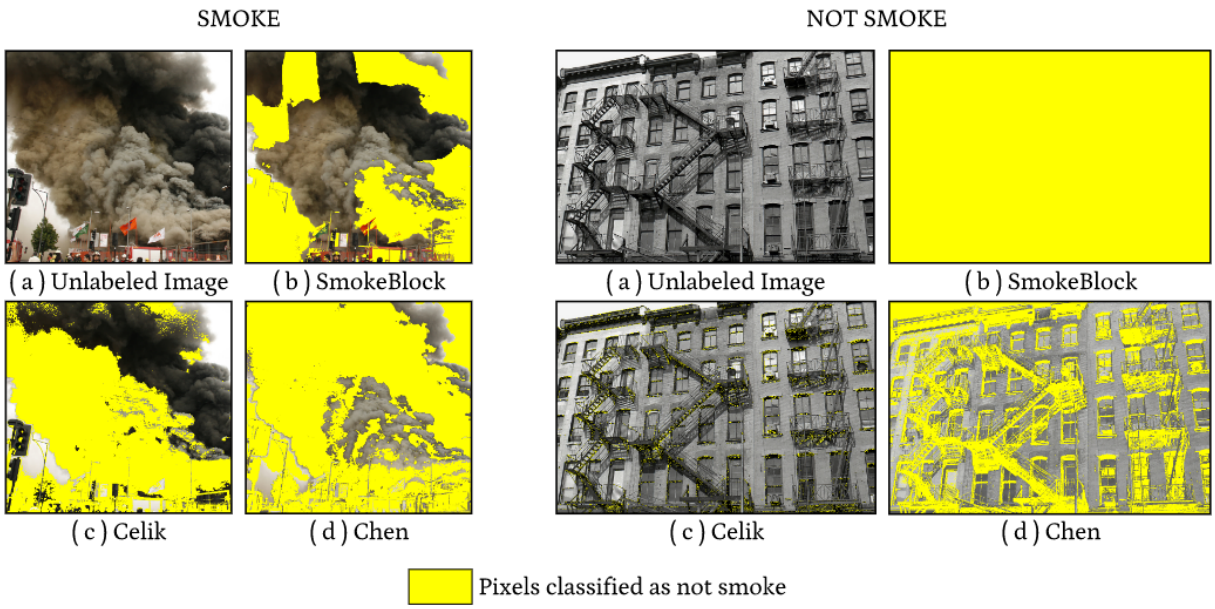
We compare our method with the related works of Çelik *et al.* (CELIK; DEMIREL, 2009) and Chen *et al.* (CHEN *et al.*, 2006).

We employed *SmokeBlock* with *CL* and *Hr FEMs* (selected in the previous step), and with the Naïve Bayes classifier. *NB* was set with different skewed cost-ratio for FN:FP (referred to as *FpCost* and *FnCost*) parameters. *FpCost* is the false positive cost and reduces the number of false positives. Similarly, *FnCost* is the false negative cost and reduces the number of false negatives. For a fair comparison, we also tuned the parameters of *Celik* and *Chen* methods. *Celik*'s single global threshold parameter was set to 25. *Chen*'s alpha value was set to 18, and its remaining parameters were set to their default values.

Figure 15 presents the differences between the strategies for two representative situations, one depicting smoke and another one without traces of smoke. *SmokeBlock* presented the highest accuracy for the detection of candidate smoke regions. *Celik* and *Chen* failed to identify most of the smoke pixels, detecting smoke pixels where there weren't any (false positives). *SmokeBlock* correctly discarded those, reinforcing our hypothesis that the combination of local color and texture improves smoke detection.

Figure 16 gives more evidence on the better results of *SmokeBlock*. The experiment was performed for 7 random images from *Flickr-Smoke*. We show the number of false-positive classifications of pixels, which the fewer the better, against the *F-Measure*, which higher values are desired. Accordingly, the best results are those inside the yellow area. The best configuration

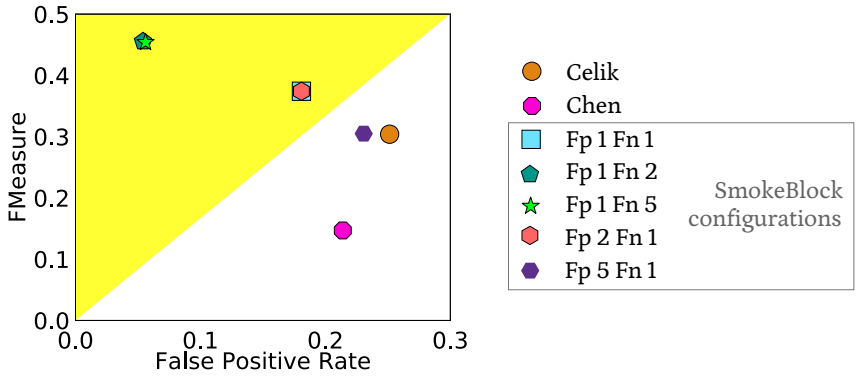
Figure 15 – Yellow pixels were classified as *not smoke*. In the first example, (a) *SmokeBlock* provided the most accurate smoke segmentation. In the second example, (a) *SmokeBlock* correctly discarded regions without smoke. (c) *Celik* and (d) *Chen* misclassified a large number of pixels.



Source: Adapted from Cazzolato *et al.* (2016).

of *SmokeBlock* was the one with $FpCost = 1$ and $FnCost = 2$. This configuration was used in the smoke detection experiment, which we detail next.

Figure 16 – Smoke pixel classification with different configurations of *SmokeBlock*, *Chen* and *Celik*.



Source: Elaborated by the author.

3.5.4 Smoke Detection

This section provides an analysis to answer the third and last question:

Q3. Can SmokeBlock overpass global-feature extractor methods in the task of spotting smoke on images?

To answer this question we perform a ten-fold cross-validation over *Flickr-Smoke*. We compare *SmokeBlock* with features extracted from the entire images. We refer to these configurations as Global *CL*, Global *Hr*, Global *TS*, and Global *Zr*, according to employed *FEM*. *SmokeBlock* is different from these configurations because it only extracts visual features from regions that have a high probability of having smoke, which refers to the segmented regions outputted by *SmokeBlock*.

SmokeBlock's Global Smoke Detection module (iv) was performed using three different classifiers, namely Naive-Bayes (*NB*), *k-NN*, and Random Forests (*RF*). Table 5 shows the comparison of *SmokeBlock*'s *F-Measure* to the the global approaches. We observed that the characteristics of the classifier strongly influence performance. *SmokeBlock* beats all global configurations, with higher *F-Measure* values than global extractions regardless of the classification strategy. However, neither the color-only nor the texture-only methods were capable of detecting smoke for a *F-Measure* higher than 0.64.

Table 5 – *F-Measure* comparison, considering different classifiers and *FEMs*. The highest values of each row are highlighted in gray.

	F-Measure				
Classifier	*Global CL*	*Global TS*	*Global Zr*	*Global Hr*	*SmokeBlock*
NB	0.59	0.46	0.51	0.50	0.64
k-NN	0.61	0.61	0.51	0.61	0.68
RF	0.60	0.61	0.51	0.62	0.71

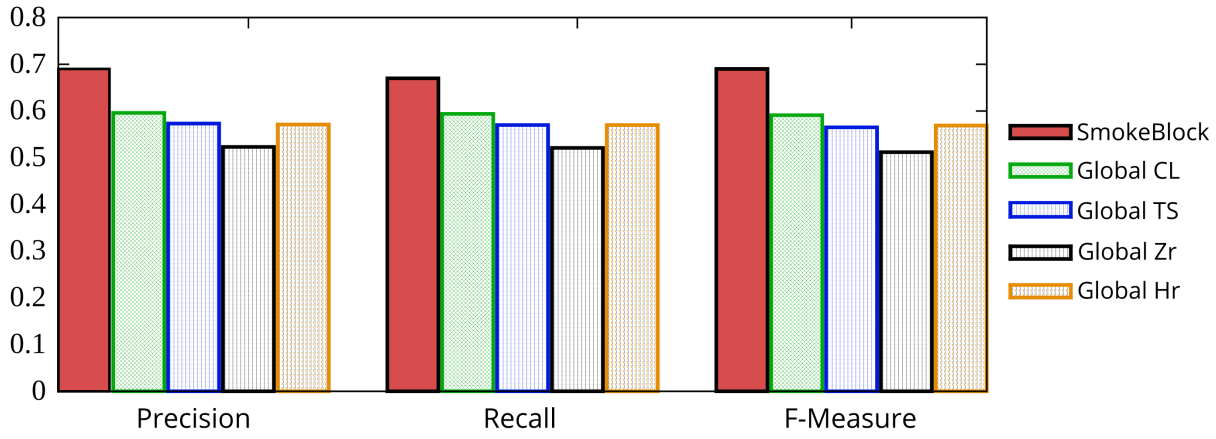
Source: Adapted from Cazzolato *et al.* (2016).

Figure 17 compares the precision, recall, and F-measure using the *RF* classifier for the entire *Flickr-Smoke* dataset. In absolute numbers, *SmokeBlock* was better than global strategies of *CL* by 14%, *TS* by 12%, *Zr* by 23% and *Hr* by 12% with respect to *F-Measure*.

Figure 18(a) presents the precision, recall, and F-Measure obtained when classifying just the smoke images of *Flickr-Smoke*. *SmokeBlock* presented the best overall performance by a margin of 9% for precision. *CL* presented the second-highest value for precision, but its recall was the second-lowest.

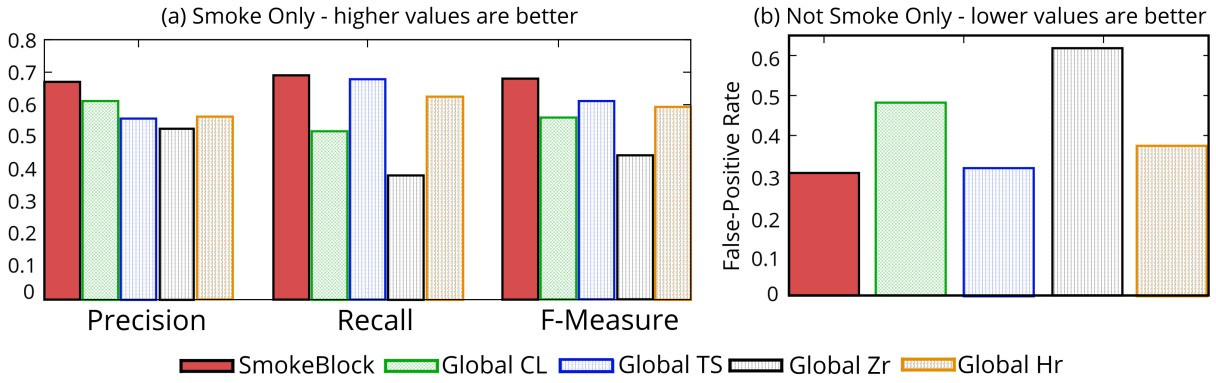
Figure 18(b) shows the false positive rate of the approaches, regarding the images from *Flickr-Smoke* without smoke. *SmokeBlock* obtained the lowest value for the false-positive rate, which is a desirable result. The precision achieved by the *CL* method (as shown in Figure 17) can be justified by means of its highest false-positive rate for not smoke images, which is not a desirable behavior. The same rationale can be employed for the *Hr* strategy. Therefore, both strategies – *CL* and *Hr*– are unsuitable for smoke detection in the perspective of avoiding false positives. Concluding, *SmokeBlock* reduced the false positive rate for not smoke images up to 7% by using previous segmented regions represented by color and texture.

Figure 17 – Smoke or not smoke using the *RF* classifier: *SmokeBlock* achieved the highest precision, recall, and F-Measure.



Source: Adapted from Cazzolato *et al.* (2016).

Figure 18 – Comparison of the techniques for smoke only and not smoke only subsets of *Flickr-Smoke*.



Source: Adapted from Cazzolato *et al.* (2016).

3.6 Lessons Learned

We have learned that color-based models are not feasible for the identification of different smoke patterns. This lack of color pattern is specialy applicable when working with images from urban scenarios. Changes on temperature, ilumination and the material being burned interfere on the smoke color. Also, as smoke present similar visual patterns than other objects, it is crucial to train the classification model with examples from the relevant similar objects that can appear in the evaluated scenario. For instance, in the urban scenario, where *SmokeBlock* was employed, patterns of mist, water and clouds, among others, should be used in the training phase of the classification model, for proper distinction.

3.7 Final Considerations

In this chapter, we dealt with the problem of smoke detection in images gathered from social media. Our goal was to use region-based analysis of images to improve the global classification. Regarding the application scenario, we have the following contributions: *(i) SmokeBlock*: a flexible, scalable, and accurate method for smoke detection to be used as a model for future developments in the field, *(ii)* Evaluation: we compared *SmokeBlock* with its competitors, providing relevant discussion to support further works on smoke detection, and *(iii) Flickr-Smoke* dataset: we built a human-annotated dataset of real images suitable as ground-truth for similar problems concerning image classification. The *SmokeBlock* approach and correspondent results were published, and are available in the works (CAZZOLATO *et al.*, 2016) and (CAZZOLATO *et al.*, 2017).

Now regarding the objectives of this Ph.D. research, in this chapter *we approached the first research problem of mining images from emergency scenarios.* The reported results indicate that a region-based classification can, indeed, improve the overall results of classification, regarding difficult image classification tasks. We focused on a hard classification approach, assigning a single label to each image or region. However, in many real-world problems, we have to deal with uncertainty. In the next chapter, we show how to address this problem by performing a region-based analysis of images using soft classification/clustering approaches.

CHAPTER

4

PROBABILISTIC REGION-BASED IMAGE CLASSIFICATION AND VISUALIZATION

In this chapter, we approach the *second research problem* of this Ph.D. research, aimed at performing image mining of medical information considering the embedded uncertainty. The corresponding contributions were published, and are available in the works (CAZZOLATO *et al.*, 2017) and (CAZZOLATO *et al.*, 2019).

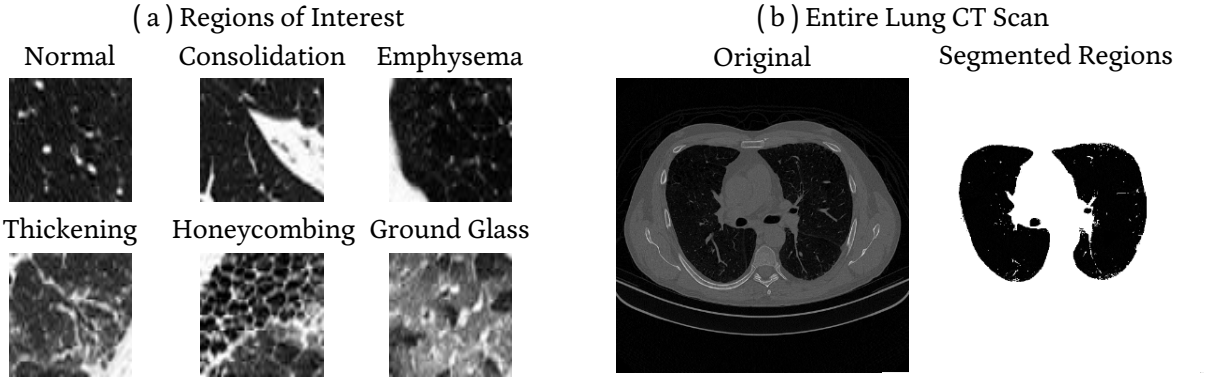
Hospitals and medical centers generate large volumes of imaging exams on a daily basis. This brings a pressing demand for automated and precise methods and systems to perform image classification and retrieval. Such tasks can be critical to support decision-making processes, as reported in the works (AKGUL *et al.*, 2011; LE *et al.*, 2018). Content-Based Image Retrieval (CBIR) applications assist physicians in the analysis of large amounts of images by providing fast, accurate, and reliable diagnosis. *CBIR* systems often rely on historical and already analyzed exams (LI *et al.*, 2018; ZHANG; METAXAS, 2016). Particularly, when dealing with high-resolution tomographic images, one can accurately identify morphological alterations such as emphysema, airway dilatation, and wall thickening, bronchiectasis, interstitial opacities consistent with inflammation and/or fibrosis and others (KOENIGKAM-SANTOS; WEINHEIMER, 2017). The main goal of the research reported in this chapter is to assess the detection of lung abnormalities in chest Computed Tomography (CT) scans, considering the uncertainty of the problem.

4.1 Context Application

Lung diseases can be diagnosed through chest CT scans, usually based on the visual clues present in the images. Such visual clues include changes in normal attenuation and the presence of focal or diffuse opacities. Here, we analyze the set of patterns *consolidation, emphysema, thickening, honeycombing, and ground glass*, additionally to the normal pattern. These patterns

appear on the lung regions of CT scans, thus the images ideally should be segmented before the analysis takes place. Figure 19(a) shows examples of these visual patterns and (b) the segmented lung regions.

Figure 19 – Examples of relevant patterns from chest CT scans, and segmented lung regions from the original CT scan.



Source: Adapted from Cazzolato *et al.* (2017), Cazzolato *et al.* (2019).

The similarity between the visual patterns of each abnormality is one of the major challenges related to the analysis of chest CT scans. For instance, interstitial thickening and honeycombing abnormalities may present a similar texture. This occurs because interstitial thickening may result in pulmonary opacities of varying morphology, such as reticulations with traction bronchiectasis (lucent/cystic areas), that may resemble honeycombing (ELICKER *et al.*, 2008; HANSELL *et al.*, 2008). On the other hand, consolidation, emphysema, and ground glass generally present distinct texture patterns. Moreover, the detection approach must consider that a single CT slice of the lungs may present normal and abnormal regions.

4.2 Motivation and Problem Definition

The tasks of pulmonary regions' segmentation and the detection of abnormalities reported in the literature suffer from many issues (NAKAGOMI *et al.*, 2013; CANDEMIR *et al.*, 2014; ZHANG *et al.*, 2013; KARARGYRIS *et al.*, 2016; SANTOSH; ANTANI, 2018; WANG *et al.*, 2018). There is no consensus, however, regarding which are the most suitable features to be used for lung tissue representation, or which is the best classifier to use. Additionally, only the lung regions should be considered in the pattern recognition process. This requires a preprocessing step to delimit and separate the pulmonary tissue regions. Radiologists and students can take more advantage of applications that highlight relevant regions of the image. Such regions present patterns that demand a more cautious analysis instead of fully automated diagnosing methods. Accordingly, in this part of our research, we aimed at solving two problems:

1. *How to **model the healthy pulmonary tissue pattern** and compose a **heat map visualization** of lung regions, according to their **likelihood of being abnormal**?*

2. *How to **model each lung abnormality** and provide a visualization of the **most probable** one contained in a certain lung region, according to its **likelihood of containing each lung abnormality**?*

To address the first problem, we proposed the *BREATH* method (CAZZOLATO *et al.*, 2017). Then, we extended *BREATH* by proposing the *dp-BREATH* method (CAZZOLATO *et al.*, 2019) to also attend to the second problem. Accordingly, from now on, we refer to our method as *dp-BREATH*, since it also includes the functionalities of *BREATH*. *dp-BREATH* segments the lung regions of chest CT scans by dividing them into superpixels. All superpixel regions identified as pulmonary tissue are classified by a supervised model, aimed at detecting the presence or absence of abnormalities based on the analysis of previously known characteristics. Since a lung may present regions with distinct patterns, such as healthy and honeycombing, our approach performs a continuous classification of lung regions. This classification considers a score value, that indicates the deviation of the region from an average normal tissue. Also, *dp-BREATH* provides a visualization that helps the radiologist to focus on abnormal regions. The same principle is applied to determine the most probable abnormality for a given lung region. The main contributions of *dp-BREATH* are four-fold:

1. Automatically segment regions containing pulmonary tissue using superpixel;
2. Model the pattern of healthy pulmonary tissue, and determine the likelihood score of a superpixel to contain abnormalities;
3. Provide a heat map representation of the abnormalities found in the image, highlighting the score of each region;
4. Model the behavior of the five considered lung abnormalities, determining the probability of a superpixel to contain each abnormality.

4.3 Related Work

The literature reports works approaching the issues of classifying pulmonary patterns based on visual features for several years now. In (MALONE *et al.*, 2004), the authors aimed to recognize pulmonary patterns by applying a *SVM* classifier on a set of 18 textural features. These features were extracted from image blocks, and the authors obtained high accuracy by combining different block sizes through a Bayesian approach. Although the idea of classifying lung regions separately is interesting, they used different-sized and squared block regions (16×16, 8×8, 4×4). Thus, the segmentation of lungs using these blocks did not adhere adequately to the textural and color visual patterns of the lungs. Besides, their proposal makes use of a training sample of blocks that are already labeled. They used a sample of pre-classified images selected by a specialist, which are highly representative of 6 patterns (normal, lung, fibrosis, emphysema, fat,

muscle and bone), making it unfeasible to work with CT images without a subset of pre-classified labels.

In (HUA *et al.*, 2011), the authors proposed an alternative method to segment lung regions from CT scans. They created a technique based on graph search, leveraged by a cost function combining intensity with gradient, boundary smoothness, and rib information, being capable of automatically segment lung CT images to detect Interstitial Lung Disease (ILD). Unfortunately, they do not provide information regarding specific regions of the lung, since their work is focused on the improvement of the lung segmentation regarding the remaining parts of the CT images.

In (SORENSEN; SHAKER; BRUIJNE, 2010) the authors used Local Binary Patterns (*LBP*) (OJALA; PIETIKAINEN; MAENPAA, 2002) features to perform instance-based classification to detect different levels of emphysema. The authors argue that they used a texture *FEM* to detect the interrelation between pixels, which provides richer information about patterns in lung CT scans. They performed an experimental analysis with the *k-NN* classifier and a dataset of 168 manually annotated Regions of Interest (ROIs), and their approach presented high accuracy. As output, their method provided a grayscale visualization of the posterior probability of each pixel of the lungs to contain emphysema. This probability was computed according to four classes (normal tissue, centrilobular emphysema, paraseptal emphysema, and panlobular emphysema). The reported results are very promising, but the work focuses on only the severity of emphysema, for three classes of patients (nonsmokers, healthy smokers and COPD smokers, where COPD stands for Chronic Obstructive Pulmonary Disease).

Still regarding the ILD classification problem, the work (KALE *et al.*, 2016) analyzes the discriminatory performance of artificial neural networks and *SVM* classifiers. Such approaches are based on wavelet features, which were extracted from high-resolution CT images in two datasets, MedGIFT, and a private one. They studied how the classifier performance varies when the training and testing stages are performed over each dataset. Then, they compared the results by performing a test on a subset that merges samples from the two original datasets. It was shown that the latter approach gives the best results for both classifiers. This finding can nourish the development of other approaches since they are easily generalized for different datasets. One drawback of their work is that the proposed approach focuses on classifying full images only. Thus, images presenting different abnormalities and healthy tissue will only be classified as one of the ILD classes considered (consolidation, emphysema, ground glass, and nodular), or as normal.

In (ALMEIDA; RANGAYYAN; AZEVEDO-MARQUES, 2015) the authors carried a statistical analysis relying on fractal and texture *FEMs*. They extracted features from squared pre-labeled ROIs within six classes (normal, consolidation, interstitial thickening, honeycombing, and ground glass). A Gaussian Mixture Model (GMM) was used to identify each class, and achieved up to 65% correct classifications, for at least one class. The idea of fitting the lung

abnormalities using GMM has shown to be interesting, but their work only focused on the analysis of full images.

The works mentioned above focus on the automatic detection of diseases in lung scans. They either employ segmentation techniques too inflexible when compared to a superpixel-based approach or focus on automatic classification of whole images. This does not allow further inquiry on specific image regions, what physicians may find relevant.

Finally, two commercial solutions analyze pulmonary tissues, namely CALIPER (RAGHUNATH *et al.*, 2016) and IMBIO (IMBIO, 2019). CALIPER provides a visualization of pulmonary regions, focusing on the detection of histologically proven emphysema. IMBIO is a tool focused on the visualization of pulmonary abnormalities. The tool provides different functionalities regarding lung density and texture analysis. IMBIO has a texture analysis to map pulmonary regions as normal, ground glass, reticular, honeycomb or hyperlucen. As it is a proprietary solution, we do not have details about its algorithms and implementation. This prevents us from comparing IMBIO's results with our approach.

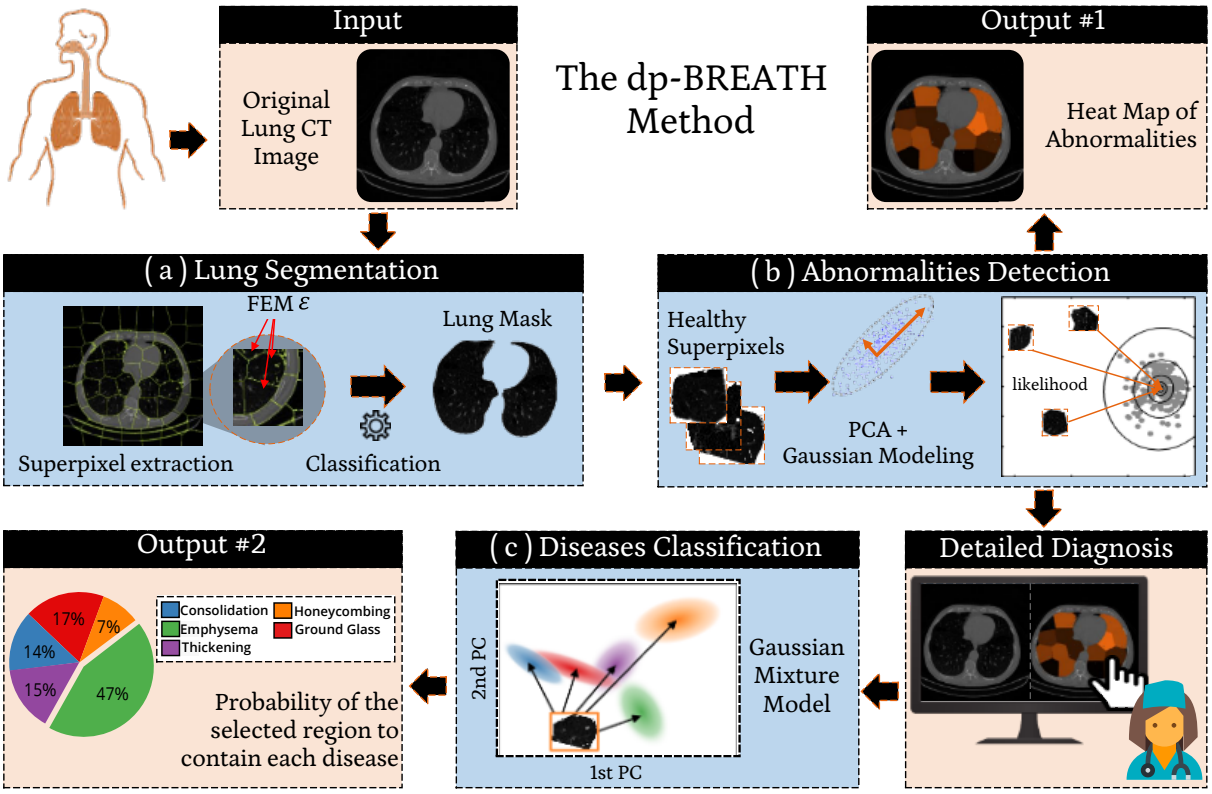
In this chapter we propose *dp-BREATH*, which is composed of a superpixel-based segmentation to discriminate pulmonary from non-pulmonary regions, together with a statistical model that characterizes normal and abnormal regions based on texture and gray-level features for ILD detection, also quantifying the region's deviation from normality. We consider that discriminating among specific disease patterns can be uncertain in many cases, and thus we perform a continuous classification to assist physicians in clinical environments. We detail our approach next.

4.4 The *dp-BREATH* Algorithm

dp-BREATH (Diseases Probabilistic BREATH) is an automatic detector of abnormalities in pulmonary regions of chest CT scans. Figure 20 shows its pipeline. *dp-BREATH* receives as input a lung CT scan, which is (a) segmented using a superpixel classification step to separate only lung tissue regions. The method (b) characterizes healthy pulmonary tissues using the *PCA* and a statistical modeling. The resulting model determines the likelihood of a superpixel to contain an abnormality. Accordingly, the first output of *dp-BREATH* is a heat map representation of the lung: abnormal regions are shown in lighter colors, and healthy regions are shown in darker colors. A detailed diagnosis is given when selecting a superpixel region in the heat map. The, (c) *dp-BREATH* uses Gaussian Mixture Model to compute the likelihood of the selected region to be one of each of the following radiological patterns considered: consolidation, emphysema, interstitial thickening, honeycombing or ground glass. Thus, the second output of *dp-BREATH* is a graphic representation of the classification. This representation shows the probability of the selected region to contain each radiological pattern.

dp-BREATH includes three main steps, which we detail in the next subsections:

Figure 20 – The *dp-BREATH* method pipeline, from the input image, (a) segmentation, (b) region-based classification on normal and abnormal, and (c) then the detailed analysis using *GMM*.



Source: Adapted from Cazzolato *et al.* (2019).

a) Segmentation of lung regions using superpixels;

b) Abnormality detection by modeling the visual properties of healthy pulmonary tissues. This step generates a heat map representation of the lung regions;

c) Characterization of pulmonary abnormalities using *GMM* and a continuous classification. This provides the probability of each lung region to contain each disease.

4.4.1 Segmentation of Lung Regions from CT Images

dp-BREATH receives as input a CT scan taken from a patient. The image depicts both pulmonary tissue and other structures. Figure 20(a) shows the segmentation of the lungs using the superpixel algorithm. *dp-BREATH* employs SLIC to divide the image into such regions, that are represented by a feature vector, obtained with the application of a *FEM*. The *FEM* used is the Local Binary Pattern (*LBP*), since it allows comparing pulmonary tissue regions from CT scans even with different settings for image contrast enhancement. The input for *LBP* is the Minimum Boundary Rectangle (*MBR*) of the superpixel. We ignore the pixels from the *MBR* that are not part of the superpixel. This is done by a binary mask that assigns the value 1 for pixels belonging to the superpixel and 0 otherwise. After extracting the features, *dp-BREATH* classifies

each superpixel using the set of labels $\mathscr{L} = \{lung, body, other\}$. These labels represent the three main patterns depicted in a lung CT scan. Accordingly, the first output of *dp-BREATH* is an image composed of only superpixels classified as *lung*.

4.4.2 Characterization of Healthy and Abnormal Tissue

In this subsection, we aim to check the following assumption: "CT scans' visual patterns can be used to discriminate between healthy and abnormal pulmonary tissue". To do so, we show how *dp-BREATH*:

1. Characterizes the visual properties of healthy pulmonary tissues using a statistical model;
2. Spots the normal and abnormal regions delimited by superpixels;
3. Represents the likelihood of each pulmonary region to contain an abnormality using heat maps.

Figure 20(b) shows that *dp-BREATH* starts characterizing the healthy pulmonary tissue by projecting the superpixels' features into a lower-dimensional feature space using the two first components of *PCA*. The visual features used here are the same extracted in step (i), Lung Segmentation.

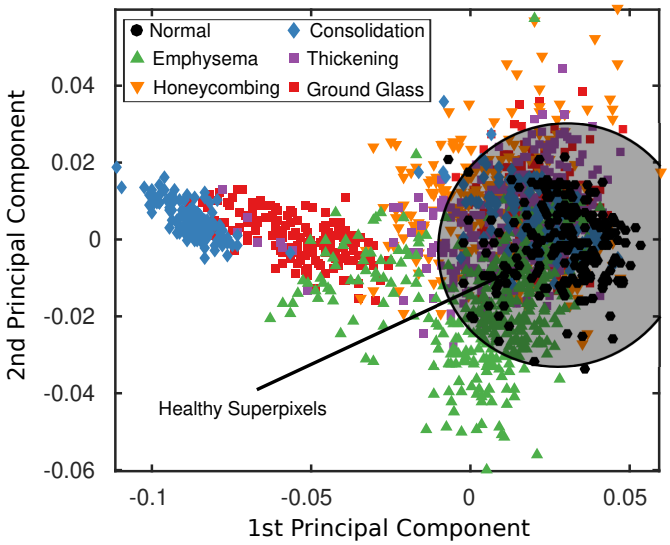
Figure 21 shows a two-dimensional *PCA* projection of all superpixels. Each dot represents a projected superpixel feature vector. Since we know that superpixels from images labeled as normal present healthy visual features, we model a Multivariate Normal Distribution (*MND*) on them (the black dots represents normal regions). Accordingly, *dp-BREATH* considers all points inside the dark ellipse as healthy pulmonary tissue.

One can observe that the majority of the healthy superpixels form a single elliptical-shaped cluster of points. Consequently, there is a concentration of superpixels from other classes around the healthy cluster, since lungs presenting abnormalities also have regions with healthy tissue. To train this model, we estimate the parameters of a Multivariate Normal Distribution (*MND*) to describe the *PCA*-projected visual features only from healthy pulmonary tissue. Let v be a *PCA*-projected feature vector, and $f(v)$ be the Probability Distribution Function (*PDF*). The *PDF* of the *MND* returns the likelihood that $\mathbf{v}$ describes a healthy tissue as defined in (BISHOP, 2007):

$$f(v) = \frac{\exp\left(-\frac{1}{2}(v-\mu)^T\Sigma^{-1}(v-\mu)\right)}{2\pi|\Sigma|^{1/2}} \tag{4.1}$$

where μ is the mean and Σ the covariance matrix from the feature vectors, and $|\Sigma|$ is the determinant of Σ. To estimate μ and Σ we employ the Maximum Likelihood Estimation (*MLE*) using only healthy superpixels.

Figure 21 – *PCA* projection of all superpixel feature vectors, along with the representation of the statistical model of the healthy tissue (black dots) as a dark ellipse.



Source: Cazzolato *et al.* (2019).

The fitted *MND* has shown to accurately describe the feature vectors' distribution of superpixels representing healthy regions of the images. The model is simple and accurate. It also leads to an important cornerstone of *dp-BREATH*, as it provides a representation of how likely a lung region is abnormal in comparison with the normal region model.

Representing lung abnormalities with heat maps

The *PCA*-projected *MND* provides an accurate description of the distribution of healthy feature vectors. The *PDF* value $f(v)$ provides an estimation of the likelihood that v was extracted from a CT region depicting a healthy pulmonary tissue. Figure 20(b) depicts this step. Small values of $f(v)$ indicate that v was likely extracted from an abnormal region.

dp-BREATH computes $f(v)$ for each superpixel, and then generates a heat map representation of the lungs. The highest *PDF* value is painted with the darkest color (black), thus representing most likely healthy tissues. The smallest *PDF* value is painted with a lighter color (we use orange), and represents the superpixel extracted from CT regions most likely depicting abnormality. *dp-BREATH*'s heat map can be presented alongside the original CT image, thus assisting physicians in the diagnosis process. They highlight regions of the lung that should be more carefully observed.

From the heat map representation, the physician can have a detailed description of the superpixel regions regarding a radiological pattern, which concerns the next step of *dp-BREATH*, described next.

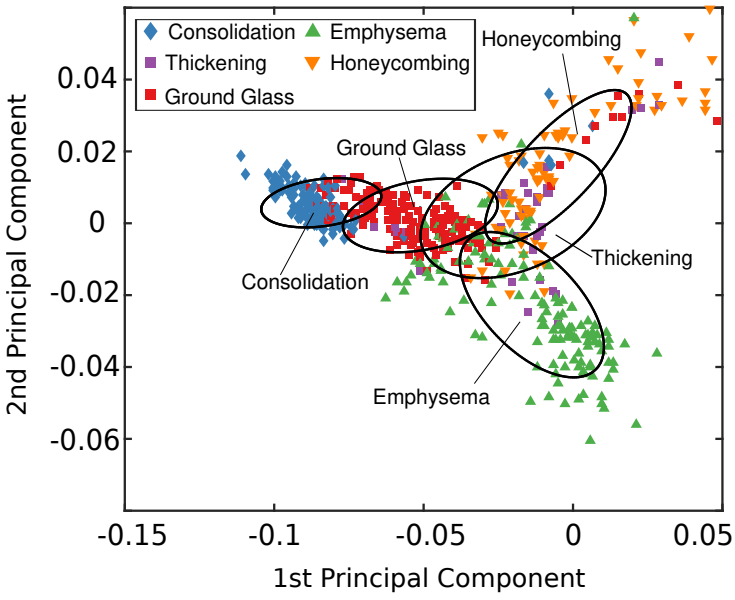
4.4.3 Detection of Radiological Patterns in the Lungs

Figure 20(c) shows a further assessment performed by *dp-BREATH*, which is the discrimination of the five radiological patterns a given tissue region is more likely to indicate. *dp-BREATH* approaches this question using the Gaussian Mixture Model (GMM) clustering, which we detail herein.

The characterization of each disease starts with a preparation step, that removes the superpixels classified as healthy from the dataset. The gray region of Figure 21 shows the removed superpixels. This step removes the normal superpixels (black dots) and those with a likelihood value $f(v)$ higher than a specified threshold, that is, those with a high probability of being healthy. The threshold value we employed covers 99% of the superpixels from normal images (dark dots), and was chosen based on the perceived separability between the healthy superpixels and the others, indicated by the ellipse boundary depicted in Figure 21.

Now, only the five classes of radiological patterns must be considered. *GMM* initialized by considering the remaining superpixels and their corresponding classes, aiming at choosing a suitable set of initial cluster centers. For each of the five classes of radiological patterns, *dp-BREATH* computes a point based on the mean of the coordinates of the first and second principal components, and the covariance matrix of the superpixels in this class. Figure 22 shows the initial groups, obtained before applying GMM. Each ellipse corresponds to a cluster that is a class representative. Here, each group consists of a single fitted Gaussian per class of abnormality. Observe that, before further processing, all groups present major overlapping among them.

Figure 22 – The traditional approach: with a single Gaussian per class, the initial groups show major regions of overlapping between radiological patterns.

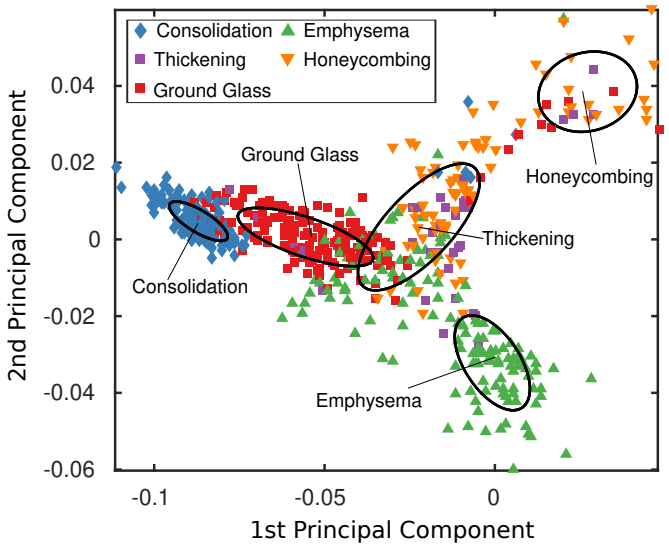


Source: Cazzolato *et al.* (2019).

GMM iteratively fits a Gaussian Distribution for the five clusters, separately. As *GMM*

is a soft clustering method, it fits the data using the iterative Expectation-Maximization (*EM*) algorithm (MOON, 1996). By iterating over the groups, *EM* might converge to local optimum, reducing the overall overlapping between the diseases. Figure 23 shows the *PCA* projection of the dataset when the clustering converged, after 71 iterations. It shows a reasonable degree of separability between the clusters for the classes consolidation, emphysema, and ground glass. However, honeycombing and interstitial thickening remain mixed due to their intrinsic visual and structural patterns. Thickening of the interstitial connective tissue network leads to reticular opacities. These opacities can be divided into interlobular septal thickening, reticulation associated with traction bronchiectasis (which can be accompanied by signs of pulmonary fibrosis such as honeycombing), and honeycombing (ELICKER *et al.*, 2008). Honeycombing is a final stage for a set of radiological patterns, and it can also present interstitial thickening patterns in CT scans.

Figure 23 – *GMM* statistical model visualization: *PCA* projection of superpixel feature vectors, where classes consolidation, emphysema and ground glass allow a good separation, while honeycombing and interstitial thickening are mixed.



Source: Cazzolato *et al.* (2019).

The clustering step finishes by employing the result set of Gaussian Distributions to compute the scores for each superpixel. This score indicates how strong is the assignment of each superpixel to each cluster of abnormalities. The score is computed as the likelihood $f(v)$, using μ and Σ of the corresponding cluster. The five scores are normalized and have a sum equal to 1, resulting in the probability of each superpixel to belong to each class. This is the second output of *dp-BREATH*.

4.5 Experimental Analysis

In this section, we detail the experiments performed to validate the *dp-BREATH* method.

Environment Setup. The experiments were performed in an Intel Core i7-4770(3.40GHz), 16GB RAM machine, with Fedora 28 (64-bit) OS. *SLIC* parameter b – the desired number of superpixels – was set to 100, according to empirical tests.

4.5.1 Material

We validate *dp-BREATH* with a dataset of chest CT scans, obtained from the Clinical Hospital of the Ribeirão Preto Medical School, Brazil. The dataset is composed of 246 image slices obtained from 108 exams, labeled with the following classes: 40 of normal pulmonary tissue; 40 with consolidation radiological pattern; 44 with emphysema radiological pattern; 42 with interstitial thickening radiological pattern; 39 with honeycombing; and 41 with ground glass radiological pattern. Each scan has 512×512 pixels, with a slice thickness of 1 millimeter.

The extracted features employed in this work are available for download in a git repository[1]. We provide *LBP* and *CH* features extracted from the images and all ROIs, the *LBP* and two-dimensional *PCA* feature vectors used in the projection of the superpixels (ref. to Figure 22). We also provide the corresponding *PDF* probabilities of each superpixel for each abnormality class. From now on, we will refer to this dataset as *dpB-Dataset*.

4.5.2 Segmentation of Lung Regions

We start by manually separating three samples of superpixels extracted from CT scans: *pulmonary tissue* (1,742 superpixels), *body* (2,031 superpixels) and *other* (2,356 superpixels). *Body* refers to the region surrounding the lungs (gray regions of the CTs). Then, we extracted the feature vectors of each set using the *CH* and *LBP FEMs*. Table 6 shows the comparison of the classification results using *k-NN*, Random Forest (*RF*), Multilayer Perceptron (*MLP*) and Support Vector Machine (*SVM*), performing a ten-fold cross-validation. We observed that even off-the-shelf classifiers were able to accurately differentiate the chest CT scans' regions, with an F-Measure of up to 0.99 using *CH* and up to 0.96 using *LBP*.

4.5.3 Detecting Abnormalities in Lung Tissues

dp-BREATH takes advantage of superpixels depicting healthy pulmonary tissue to check how well it detects abnormalities. It is important to highlight that, although we have the class of every image (e.g., normal or emphysema), we do not have the corresponding label for individual superpixels. This occurs because superpixels from the same image can depict visual patterns from different classes.

[1] Material available at <github.com/mtcazzolato/dp-breath>

Table 6 – Lung segmentation using *dp-BREATH*: the classifiers accurately differentiate different regions of CT scans.

		F-Measure			
FEM	***Dimensions***	1***NN***	***RF***	***MLP***	***SVM***
CH	32	0.989	0.993	0.988	0.948
	64	0.992	0.994	0.986	0.954
	128	0.994	0.994	0.984	0.956
	256	0.992	0.992	0.973	0.950
LBP	16	0.908	0.939	0.331	0.859
	256	0.894	0.965	0.345	0.660

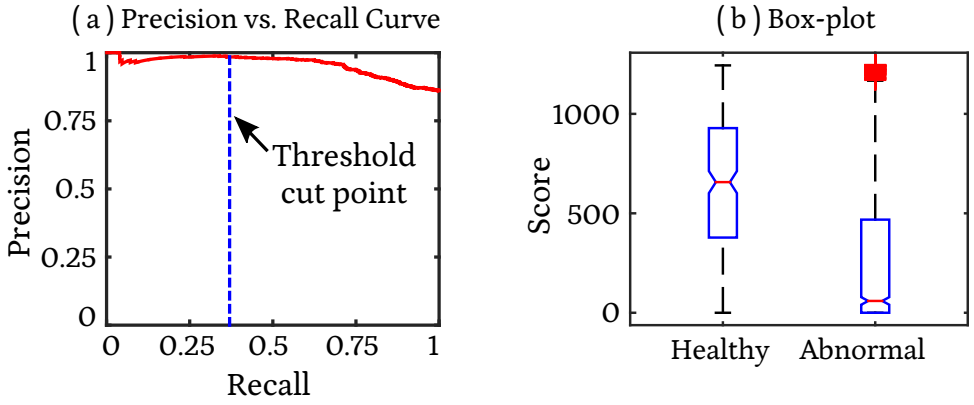
Source: Adapted from Cazzolato *et al.* (2019).

Accordingly, we used a representative dataset of regions of interest (ROIs) of pulmonary tissue (COSTA; HUMPIRE-MAMANI; TRAINA, 2012) to validate our method. This dataset has $3,258$ ROIs of size 64×64 pixels, labeled by an expert using the five classes of radiological patterns. We extracted *LBP* features from each ROI, and trained a *k-NN* classifier with $k = 3$ using the *City-Block* distance. The classifier was shown to be highly accurate, with an accuracy of $97.8\% \pm 0.7\%$ after performing ten-fold cross-validation on the ROI data. Accordingly, *dp-BREATH* uses this classifier to label the superpixels as either positive (presenting abnormalities) or negative (healthy tissue). The resulting labels were used as the ground truth for the experiments described next.

We first evaluate how well *dp-BREATH* detects abnormal pulmonary tissues. Figure 24(a) shows the precision vs. recall (sensitivity) curve when detecting superpixels with abnormalities. The best performance corresponds to a curve close to the top. To classify a superpixel as healthy or not, we used the likelihood score. The curve was built varying the decision threshold with respect to the likelihood score. *dp-BREATH* obtained a precision higher than 86% for every value of recall, presenting an average accuracy of 95%. The dashed vertical line represents the cut point, and the points below the score value are considered abnormal. This cut point is represented by the black ellipse of Figure 21.

We also analyze how well the first statistical model is able to describe healthy visual patterns. Figure 24(b) shows a box-plot comparing the likelihood score returned by the *dp-BREATH*'s statistical model for superpixels classified either as healthy (normal) or abnormal. The assigned scores clearly separate the majority of superpixels. More specifically, the interquartile range (indicated by the boxes) of healthy superpixels has upper scores with almost no overlap with the interquartile range of the superpixels presenting abnormalities.

Figure 24 – (a) The Precision vs. Recall curve, showing that *dp-BREATH* detects superpixels depicting abnormalities with precision close to 1 for all values of recall. (b) The box-plot comparing the distribution of likelihood scores returned by the statistical model, which shows that *dp-BREATH* separates healthy and abnormal pulmonary tissues.



Source: Adapted from Cazzolato *et al.* (2019).

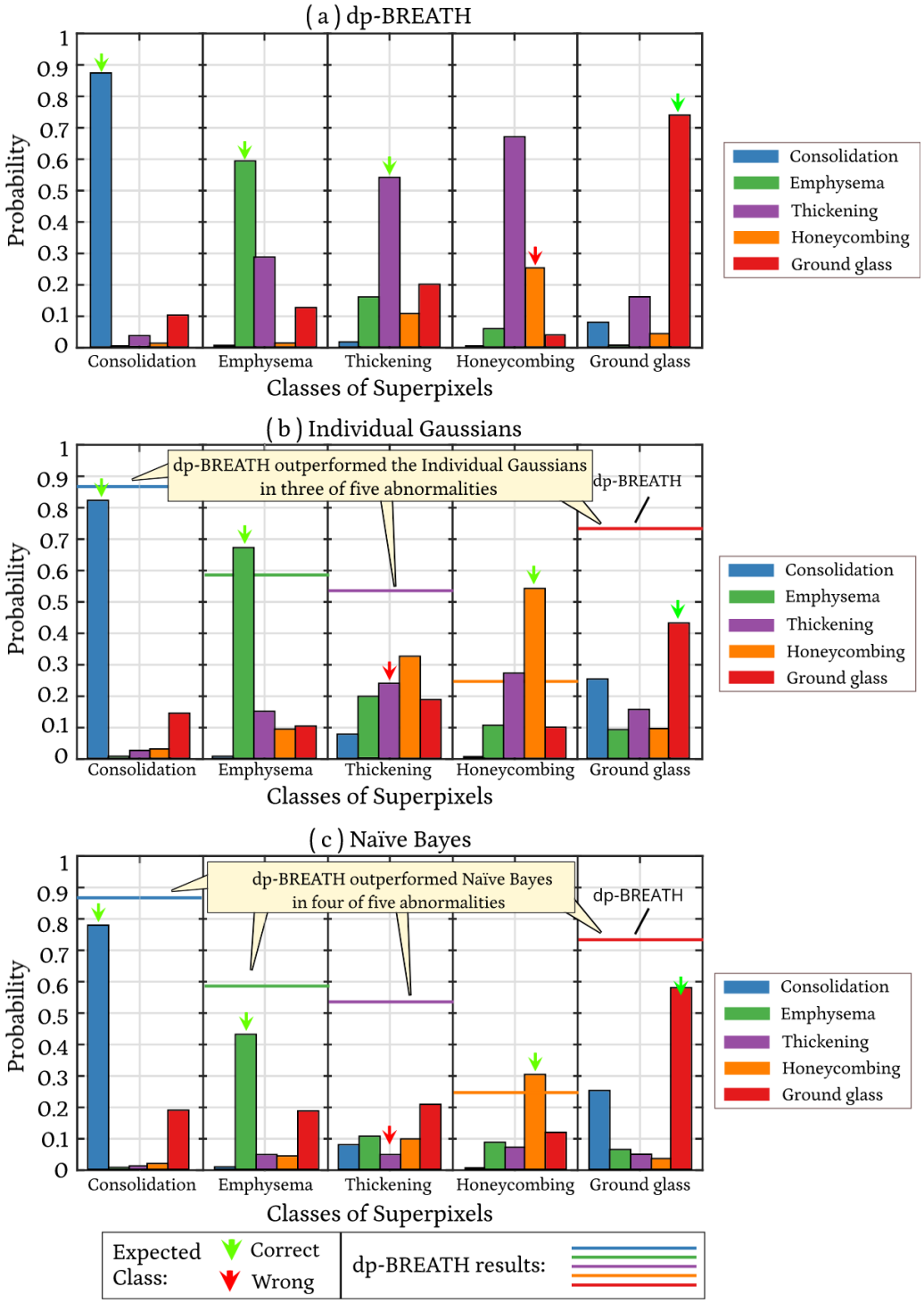
4.5.4 Detecting Lung Radiological Patterns in Specific Regions

dp-BREATH computes the probability of each superpixel to contain each radiological pattern by grouping regions through their labels. The method computes the average probability of each superpixel from a class to spot each radiological pattern. Figure 25 presents the comparison of *dp-BREATH* with the individual fitted (pure) Gaussian for each abnormality (see Figure 21) and the *Naïve Bayes* (*NB*) classifier.

Figure 25(a) shows that *dp-BREATH* obtained the best results on classifying each region as the class that it belongs to, except for honeycombing. Consolidation, emphysema, and ground glass were the classes that presented the best separation in the clustering step, with respectively 87%, 59% and 73% of probability. On the other hand, interstitial thickening and honeycombing presented a mixture behavior, reflected in the honeycombing results, where 66% of the superpixels were classified as are most probable to be from class thickening. We have learned from the literature and the feedback of specialists that honeycombing and interstitial thickening may present similar texture patterns, which can explain why several superpixels from honeycombing were misclassified as interstitial thickening.

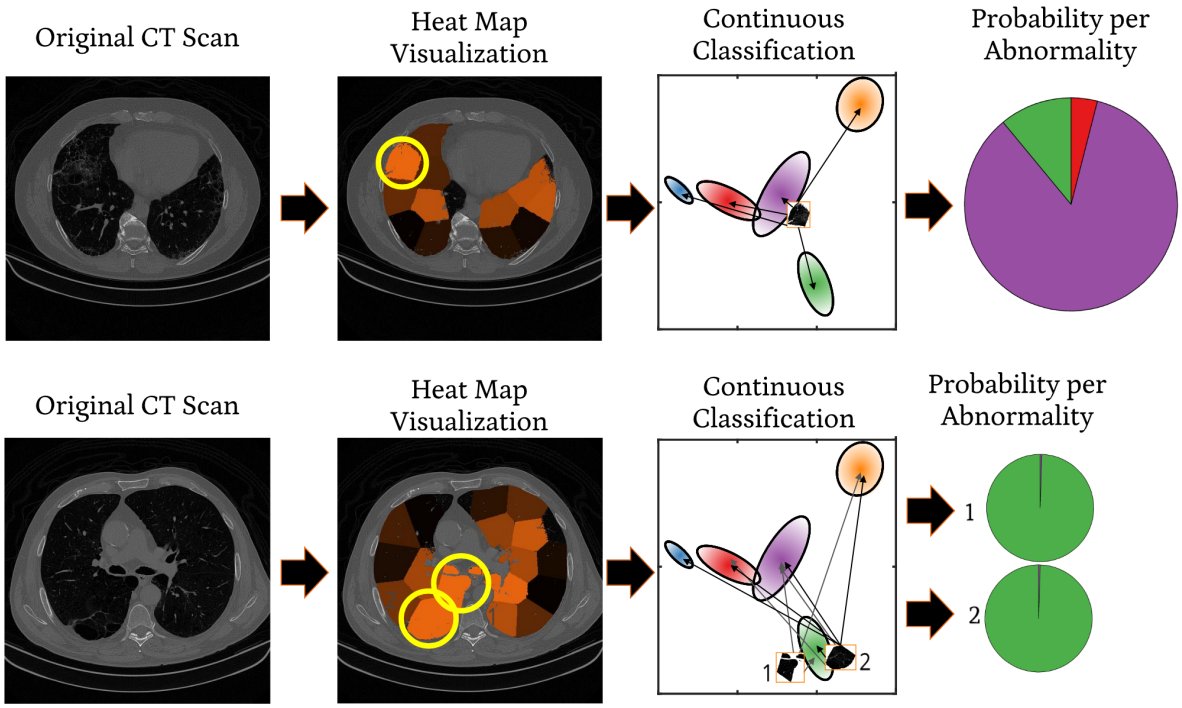
Regarding the individual Gaussian groups (Figure 21(b)), the classification of consolidation superpixels showed 82% of probability, emphysema 67%, interstitial thickening 23%, honeycombing 54% and ground glass 73%. *NB* (Figure 21(c)) classified consolidation with 78% of probability, emphysema with 69%, interstitial thickening with 11%, honeycombing with 53% and ground glass with 58%. The three approaches misclassified the superpixels of either thickening or honeycombing. *dp-BREATH* outperformed the individual Gaussian groups in three of five abnormalities, and *NB* in four of five abnormalities.

Figure 25 – Probability classification results: (a) *dp-BREATH* correctly classified the majority of lung patterns from abnormal superpixels. The dashed horizontal lines in (b) and (c) represent the results of *dp-BREATH* (a), easing comparison. We can observe that *dp-BREATH* was better than its competitors.



Source: Adapted from Cazzolato *et al.* (2019).

Figure 26 – Visualizing lung abnormalities: *dp-BREATH* outputs a heat map representing the probability of each superpixel to contain an abnormality or being normal. Orange represent lung regions with the highest probability of being abnormal, and the darker colors represent probably healthy regions. Also, *dp-BREATH* computes the likelihood of a selected superpixel to contain each radiological pattern.



Source: Adapted from Cazzolato *et al.* (2019).

4.5.5 Visualizing the Probability of Abnormalities with dp-BREATH

dp-BREATH composes a heat map representation of the image from the characterization of healthy and abnormal tissues. Figure 26 shows two examples of outputs. The statistical model used to classify lung regions computes the likelihood of each superpixel to be healthy tissue. *dp-BREATH* uses lighter colors (towards orange) to represent regions likely to be abnormal, and darker colors (towards black) to represent healthy regions.

The output heat map highlights regions presenting different texture patterns, potentially guiding the physician to more carefully analyze the regions most likely to have abnormalities. When the physician selects a lung region, *dp-BREATH* computes its distance to each class of abnormalities. A pie chart shows the probability of the chosen superpixel for each abnormality. This result can be used as a continuous classification, with the goal of interactively lead the physician to a more detailed analysis.

4.6 Lession Learned

Dividing a lung CT scan into grid-based regions (as squares or rectangles) do not capture the visual coherence of lung regions as superpixels do. Using an approach that considers the

similarities of a group of pixels are most likely to generate regions with distinct patterns among other regions. Also, the global classification of single CT lung scans is not reliable, once a single image (usually) presents more than one pattern. Finally, the hard classification of a single region may not be appropriate. The set of lung abnormalities considered in this work have classes that are early stages of other abnormalities, such as “consolidation”.

4.7 Final Considerations

In this chapter, we presented the problem of classifying medical images considering uncertainty. Our goal was to employ a region-based analysis of lung scans to improve the global analysis of the different abnormalities depicted together. Regarding the application scenario, we have the following contributions: (i) *dp-BREATH*: an accurate method that models the healthy pulmonary tissue pattern and also each of the considered abnormalities. *dp-BREATH* provides a visualization of the most probable visual patterns contained in a particular lung region, according to its likelihood of containing each lung abnormality. (ii) Evaluation: we compared *dp-BREATH* with existing classifiers, showing its high accuracy, and providing an extensive discussion regarding the pathological lung patterns considered concerning the obtained results. (iii) *dpB-Dataset*: we made the features used in our validation available online, aimed at supporting the reproducibility of our findings. *BREATH* and *dp-BREATH* approaches and corresponding results were published, and are available respectively in the works (CAZZOLATO *et al.*, 2017) and (CAZZOLATO *et al.*, 2019).

Concerning the goals of this Ph.D. research, in this chapter *we approached the second research problem of mining images with uncertainty*. The reported results show that region-based approaches can deal with different visual patterns together, providing a complete analysis to the specialist. On the contrary of Chapter 3, we show that performing soft classification/clustering can benefit the analysis results, improving the overall accuracy of the methods.

BREATH and *dp-BREATH* focus on the analysis of regions from still images and the related information, such as labels. However, when dealing with sequences of images, region-based approaches can benefit from the temporal factor to provide a fast and approximated result, also considering the associated additional information. We show this scenario in the next chapter.

CHAPTER

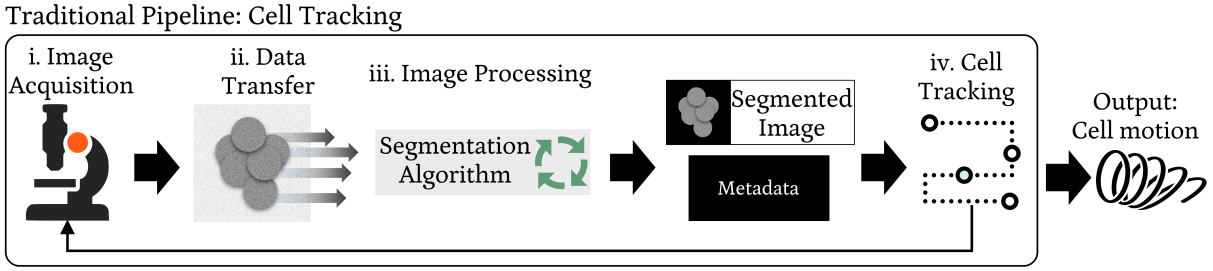
5

TRACKING AND PREDICTING MOVING OBJECTS IN SEQUENCES OF IMAGES

In this chapter, we approach the *third research problem* of this Ph.D. research, with the goal of analyzing moving objects detected in regions of sequences of images. The corresponding contributions were published, and are available in the work (CAZZOLATO; TRAINA; BÖHM, 2018).

The problem of tracking cells from sequences of images depicting developing embryos is important regarding the study of the dynamics of biological processes (BALOMENOS *et al.*, 2017; ELFWING *et al.*, 2004). This task must take place automatically, to enable scalability. Figure 27 depicts the traditional pipeline for cell tracking. It encompasses steps of (i) image acquisition, (ii) data transfer, (iii) image processing and then (iv) the actual tracking. The pipeline is iterative (the thin arrow represents this) since it performs the four steps for each new image of the sequence. All steps must be performed ideally in real-time. For the sake of simplicity, we assume that all images from a sequence are equidistant in time.

Figure 27 – The traditional pipeline of cell tracking.

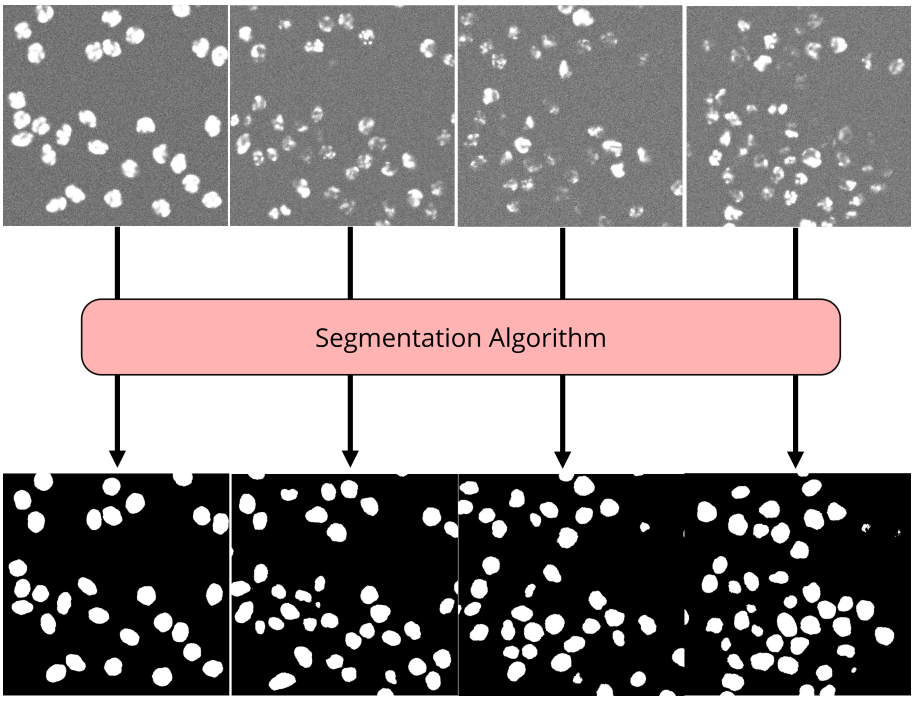


Source: Adapted from Cazzolato, Traina and Böhm (2018).

5.1 Context Application

Conventionally, segmentation algorithms process sequences of microscopic images to separate cells from the background. Figure 28 depicts the segmentation task, with the original images and the output of segmented cells, obtained by a segmentation algorithm. Examples of such approaches were presented in (SPINA *et al.*, 2018) and (STEGMAIER *et al.*, 2018). They output cells' positions, which consist of spatial coordinates. Limitations of image capturing and segmentation may introduce artifacts in cell curvatures, making the outputted information not very accurate, and consequently, the tracking task can be a difficult problem (BALOMENOS; TSAKANIKAS; MANOLAKOS, 2015). Conventional tracking algorithms establish cell-to-cell (or seed-to-cell) correspondences along the sequence of images, by matching cells according to their locations. Figure 29 illustrates the construction of the trajectory vector to describe cell motion, with a new seed point being matched. They usually combine the segmentation task with the tracking task. When tracking cells, conventional trackers also build vectors of cell trajectories. However, this takes place late in the data-processing chain, so the vectors are not used for tracking.

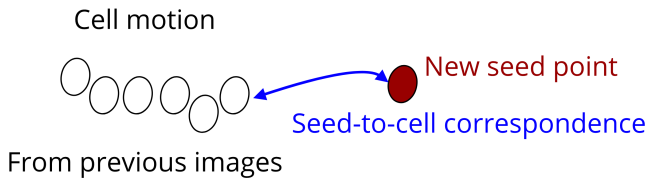
Figure 28 – The segmentation algorithm separates cells from the background of images.



Source: Elaborated by the author.

We study whether and to which extent establishing an explicit representation of the images early in the chain gives way to better cell tracking. By providing such an explicit representation, one can use it to accurately and efficiently predict/estimate cell positions along time. This solution has the advantage of avoiding the transmission of the full images to a local computer and storing them before any analysis takes place. The study described in this chapter

Figure 29 – Example of cell trajectory, and a seed-to-cell correspondence of an existing cell to a newly detected seed point.



Source: Elaborated by the author.

was carried during an internship at the Karlsruhe Institute of Technology (KIT), Germany. Accordingly, the aforementioned problem is, currently, the bottleneck of the entire embryo-breeding workflow at KIT, and it takes hours for one embryo.

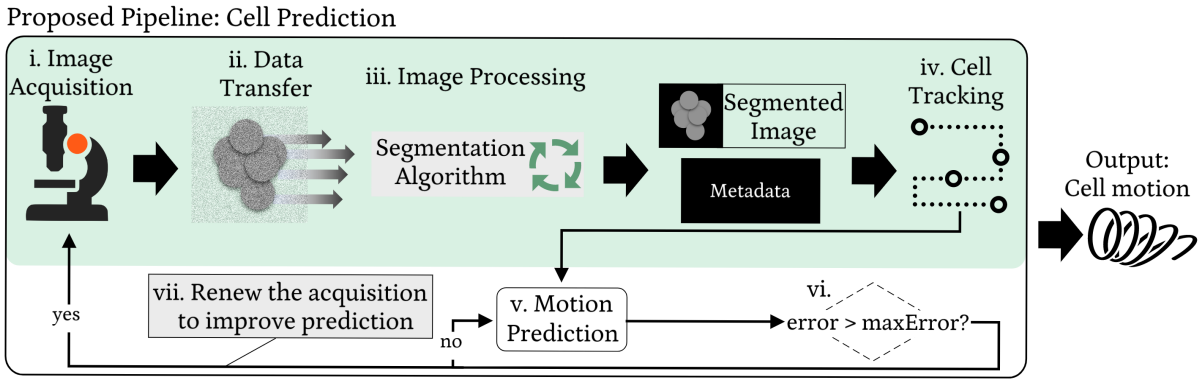
Moving objects and moving objects databases fields have been received attention from the database research community over the years (ALMUHISEN; DURAND; QUAFAFOU, 2018; SALTENIS *et al.*, 2000; TAO *et al.*, 2004). Traditionally, works from the literature have used moving objects to represent traffic participants, such as cars, pedestrians, among others. In this work, we aim at representing cells as moving objects, since we judge worthwhile to investigate whether such ideas can help with the tracking of cells from embryos.

5.2 Motivation and Problem Definition

We propose a modification on the traditional pipeline of cell tracking, intercalating the traditional steps with cell prediction. Figure 30 depicts the proposed pipeline. The boxes inside the green area are from the traditional pipeline, which we omit in iterations when estimation takes place. With this modification, different kinds of improvement should be possible, be it in detection quality, be it in tracking speed. However, different metrics to assess the improvements are necessary, and we discuss such measures as well. Also, there are alternative models of cell motion, such as the number of points used for interpolation, and the distance threshold used for matches. In this chapter, we establish the design space of our methods and evaluate the plausible options systematically.

The proposed solutions must update the representation of the motions on the fly, as new data arrives. When taking advantage of previous cell positions for prediction, the solution must monitor the prediction error at each iteration. This allows the model to adapt itself when necessary. In this context, extreme solutions like throwing away the entire previous model or adapting the model only minimally do not appear to be promising. Thus, our proposal requires reorganizing the interaction between the various components, such as image segmentation or tracking, at the same time maintaining the solution light-weight. Very importantly, the proposed solution must work orthogonally to existing tracking tools, meaning that they can be combinable

Figure 30 – The proposed pipeline of cell tracking: with the cell motion prediction, steps from the traditional pipeline (inside the green area) can be omitted in many iterations, speeding-up the entire process.



Source: Adapted from Cazzolato, Traina and Böhm (2018).

with existing implementations of the other components of the pipeline. With this in mind, we state that our goal here is not to "replace" existing techniques for image segmentation and tracking. We want to use available implementations and focus on the establishment of vectors describing cell motions. Lastly, modularity is a related, yet different requirement. Dealing with the cells' segmentation and tracking separately makes the pipeline more flexible by being able to replace individual components.

In this part of our research, we aimed at answering the following research problems:

1. How can we efficiently track cells over time and establish their trajectory vectors, relying on the output of segmentation algorithms?

2. What are the alternative configurations that allow the use of previous cells' positions to predict their future movement in an effective and reliable manner?

We address the aforementioned problems by proposing *CM-Predictor*, which takes advantage of previous locations of cells to estimate their motion along time. The estimation of cell motion can speed up the generation of the output. This estimation is performed using Lagrange's polynomials (ABDEL-AKHER; SELIM; ALY, 2015), after the acquisition of enough cell points. We compare the estimated results with baseline tracking methods (*Direct-Tracker* and *Clever-Tracker*), that we propose to serve as a basis for comparison.

5.3 Basic Concepts and Naming

In this section, we introduce the basic concepts and naming used in the context application, regarding cell tracking and prediction.

Sequences of microscopic images depict the development of an embryo, starting from a single **cell** that splits over time. The images are equidistant, which means that they were obtained after fixed intervals of time, depending on the quality and configuration of the acquisition equipment. Each cell in the image has its spatial location referred to as a **seed** (or seed point) at time t. A match between a new seed point and an already existing cell (detected from previous images) is a **seed-to-cell correspondence**.

Accurately monitoring topological changes, e.g., splitting objects over time, relies on the accuracy of cell detection, by a segmentation algorithm, at each time frame (LI *et al.*, 2016). Segmentation algorithms separate objects from the background, thus detecting and segmenting cells. One example of such an algorithm is *TWANG* (STEGMAIER *et al.*, 2014), which we employed in this work as it is fast and accurate.

TWANG starts by detecting seed points in the images. The authors use a LoG (Laplacian of Gaussian) blob detector with different scales to find spherical objects in the images. Then, *TWANG* homogeneously distributes the seed points among different threads, independently performing further calculations in parallel. For this task, the method crops a region around each seed point, aiming to process as few pixels as possible. *TWANG* computes the Gaussian smoothed left-right derivative image and a weighted dot product of the normalized gradient for each region. The properties of the segmented regions are immediately extracted from the cropped images. *TWANG* outputs segmented images, with the detected cells and metadata information, containing the position of each cell in each image.

Given that a cell changes its position at each image, and considering a sequence of images, a **trajectory** consists of a list of seed points. For instance, given a Cell c, detected in images from time interval $[2,n]$, its trajectory is given by $\mathscr{T} = (\hat{s}_2, \ldots, \hat{s}_n)$, where $\hat{s}_i$ is the seed of a cell c at time t_i with $2 \leq i \leq n$. We review the related work on cell tracking next.

5.4 Related Work

In this section, we tackle the problem of tracking microscopic objects.

The outputs generated by cell tracking approaches can be used by biologists to analyze changes induced by the use of substances, such as contrast agents, observing the evolution in cell motion and morphology (HE *et al.*, 2017). The literature reports many works regarding the tracking of cells. In (CHAKRABORTY; ROY-CHOWDHURY, 2015), the authors propose representing cells of an embryo using a graph structure. Their method focuses on cells that are in close contact with each other, i.e., share an edge or boundary. They build a graph structure for the cells every time a new image is processed. Its overall accuracy is good, with up to 87% and 97% for spatial tracking, considering two different image datasets. In the work (JIUQING; XU; XIANHANG, 2017) the authors define six local events to describe linking patterns between pairs of consecutive images, which are move, divide, appear, disappear, split, merge. Unlike our

proposed pipeline, their proposal consists of a joint detection and tracking method. They predict cell trajectories by solving a linear programming problem. The model parameters are learned with a structured SVM.

The work (HILSENBECK *et al.*, 2016) presents a review of existing tools. Among the mentioned tools is the *TLM-Tracker* (KLEIN *et al.*, 2012), which we use as a reference to compare with our methods since it attempts to perform a fully automated cell tracking and is available online. *TLM-Tracker* has several steps. It starts with the pre-processing of input images, such as image filters. The segmentation algorithm also searches for seed-to-cell correspondences in two consecutive images, using two different approaches: (i) based on the areas of cells, (ii) based on the center points of cells. The first approach (i) gets the minimum bounding rectangle (MBR) of each cell, from two consecutive images. Then it gets the distance between the cells in time by computing the relative overlap of their MBRs. If there is no overlap, the second approach (ii) computes the Euclidean distance between the center points of cells. *TLM-Tracker* has a joint detection and tracking step to predict cell positions. This prediction is computed using a fitted polynomial: the approaches restrain the search space for the segmentation algorithm, to look for the probable match point of a certain cell in the next image. Although the tool was originally proposed to work with elongated cells, the authors state that *TLM-Tracker* works well with different kinds of cells (KLEIN *et al.*, 2012). *TLM-Tracker* finishes its pipeline with the visualization of the motion vector of the embryo.

The aforementioned tracking approaches work with cell points detected from sequences of images, which consists of the traditional pipeline to obtain the motion vectors and can be time-consuming and computationally expensive (due to image acquisition, transfer, and segmentation). To overcome this issue, in this chapter, we propose *CM-Predictor*, a method that establishes the motion vectors of cells, based on cell points from previous images. We target at useful representations of cells in microscopic images, which we consider a specific kind of moving object from now on.

5.5 The Traditional Pipeline: Tracking Cells

The segmentation algorithm reads input images and outputs the segmented image and metadata, which contains the detected cells' position represented as seed points. Tracking algorithms match such seed points to their corresponding existing cells. We propose two algorithms for the tracking task, namely *Direct-Tracker* and *Clever-Tracker*. Any segmentation algorithm that outputs the cell positions in a given image can be used here. We employed the *TWANG* algorithm, proposed in the paper of *TWANG* (STEGMAIER *et al.*, 2014), since it consists of a simple, fast and still accurate alternative.

5.5.1 Data Structures for Trackers

The proposed tracking approaches use a distance-threshold th to decide when a match between a pair of seed points occur. The pair of seed points in question are coordinates from the cell detected from time $t-1$, that have already been added to the description of cells of the embryo, and the coordinates of a new seed point, detected at the current time point t. With this threshold, the algorithms do not need to test all possible combinations of matching seeds, speeding up this task. Table 7 shows the **input** parameters and **output** of the trackers.

Table 7 – Input parameters and output of the trackers.

Input Parameter	**Description**
$\hat{S}_{t-1}$	A list with all seed points detected at time $t-1$. The seeds are sorted by the order they were detected by the segmentation algorithm, and they have already been inserted to the description of cells from an embryo.
$\hat{S}_t$	A list with all seed points detected at time t, sorted in the same manner as the previous item.
th	The distance threshold value.
Output	**Description**
embryo	An embryo, composed of moving cells, each one represented as a sequence of seed points.

Source: Elaborated by the author.

5.5.2 Adding a Cell Match

Let us consider C as the set of cells seen so far in the sequence of images. Each seed point in $\hat{S}_{t-1}$ stands for a cell $c \in C$. The proposed trackers use Function *AddMatch* to add a new match of a cell, and we explain in the following. The tracker algorithms add the new seed point $\hat{s}_2 \in \hat{S}_2$ to a cell C, that contains the matching seed point $\hat{s}_1 \in \hat{S}_1$ (Lines 1–3). Both seeds from times $t-1$ and t are removed from the lists $\hat{S}_{t-1}$ and $\hat{S}_t$, respectively (Lines 4–5). If an existing cell has more than one new seed point as a match at time t, the trackers assume that the cell has been split, and create a new cell for each match with a new seed. *Direct-Tracker* and *Clever-Tracker* use the same data structures and the same method for adding another cell to the embryo.

5.5.3 Cell Tracker Approaches

Here we introduce two cell tracker algorithms, *Direct-Tracker* and *Clever-Tracker*. The approaches select the pair of seeds $\hat{p}(\hat{s}_{t-1}, \hat{s}_t)$, that is matched based on the corresponding criterion:

Function AddMatch(*embryo*, $\hat{s}_1$, $\hat{s}_2$, $\hat{S}_1$, $\hat{S}_2$)

1 Let C be the set of cells of *embryo* seen so far;
2 $c \leftarrow$ cell $\in C$ containing seed s_1; ▷ *Get the cell with the seed being matched*
3 c.addSeed($\hat{s}_2$); ▷ *Add new seed to the existing cell*
4 Remove $\hat{s}_1$ from $\hat{S}_1$;
5 Remove $\hat{s}_2$ from $\hat{S}_2$;
6 **return** *embryo*; ▷ *Return embryo data with the new seed point*

- *Direct-Tracker* works in a greedy fashion, matching the last seed point of the existing cell to the first seed point within the distance threshold th;
- *Clever-Tracker* maps each new seed point to its nearest cell.

Direct-Tracker

Algorithm 2 presents *Direct-Tracker*. For each seed ($\hat{s}_t$) detected at the current time t, *Direct-Tracker* searches for a matching seed at time $t-1$ ($\hat{s}_{t-1}$) (lines 3–5). Notice that $\hat{s}_{t-1}$ belongs to an existing cell. If the distance between $\hat{s}_t$ and $\hat{s}_{t-1}$ is less than th, they are considered a match (line 7). *Direct-Tracker* deems the first existing seed point ($\hat{s}_{t-1}$) within a th of distance from ($\hat{s}_t$) a match. However, this does not ensure that $\hat{s}_{t-1}$ is the nearest seed point to $\hat{s}_t$. *Clever-Tracker* in turn has this characteristic, as we detail next.

Algorithm 2: *Direct-Tracker*

Input : $\hat{S}_{t-1}$: a list with all seeds from time $t-1$
$\hat{S}_t$: a list with all seeds from time t
th: the distance threshold

Output : The *embryo*, with existing and new seed points.

1 **begin**
2 initialization;
3 **foreach** *seed* $\hat{s}_t \in \hat{S}_t$ **do**
4 $matched\_t \leftarrow -1$; ▷ *Set flag as not matched yet*
5 **foreach** *seed* $\hat{s}_{t-1} \in \hat{S}_{t-1}$ **do**
6 $\delta \leftarrow ComputeDistance(\hat{s}_t, \hat{s}_{t-1})$; ▷ *Compute distance between seed points*
7 **if** *(*$\delta < th$ *AND* $matched\_t = -1$*)* **then**
8 *AddMatch*(*embryo*, $\hat{s}_t$, $\hat{s}_{t-1}$, $\hat{S}_t$, $\hat{S}_{t-1}$);
9 $matched\_t \leftarrow 1$; ▷ *The current seed has been matched*
10 Create new cells for remaining seeds and add them to *embryo* ▷ *For new seeds with no matching cells*
11 **return** *embryo*

Direct-Tracker's results depend on the order of seeds. But as the segmentation algorithm (*TWANG*) usually searches for cells following the order of pixels, the detected cells tend to

maintain the same order when listed by the segmentation algorithm, with minor changes due to new cells, resulted from cell splits.

Clever-Tracker

Clever-Tracker uses Function *CheckNearestSeed* to search the matching existing cell to each new seed point, as shown in Algorithm 3. Let $\hat{s}_q$ be a seed point from time t. Function *CheckNearestSeed* checks if there is a seed $\hat{s}_{t-1} \in \hat{S}_{t-1}$, which $\delta(\hat{s}_q, \hat{s}_{t-1}) < mindist$, i.e., if there is a seed close to $\hat{s}_q$ (lines 10–14). If the distance between two seed points is less than th, it is considered a match.

Algorithm 3 presents *Clever-Tracker*. For each new seed at t (line 3), *Clever-Tracker* searches for its closest seed (belonging to an existing cell) from time $t-1$ (line 5). Function *CheckNearestSeed* (line 5) is responsible to search for the nearest seed, and the matches are added by Function *AddMatch* (line 7).

Algorithm 3: *Clever-Tracker*

Input : $\hat{S}_{t-1}$: a list with all seeds from time $t-1$
$\hat{S}_t$: a list with all seeds from time t
th: the distance threshold

Output : The *embryo*, with existing and new seed points.

```
1  begin
2      initialization;
3      foreach seed ŝ_t ∈ Ŝ_t do
4          mindist ← +∞;                          ▷ Initialize minimum distance seen so far
5          (matched_t, mindist) ← CheckNearestSeed(ŝ_t, Ŝ_{t-1}, th, mindist, −1)
6          if (matched_t > −1) then
7              AddMatch(embryo, s_t, ŝ_{matched_t}, Ŝ_t, Ŝ_{t-1});
8      Create new cells for remaining seeds and add them to embryo   ▷ For new seeds
         with no matching cell
9      return embryo
10 Function CheckNearestSeed(ŝ_q, Ŝ, th, mindist, matched_t)
11     foreach seed ŝ_i ∈ Ŝ do
12         δ ← ComputeDistance(ŝ_q, ŝ_i);
13         if (δ < mindist AND δ < th) then
14             mindist ← δ;                       ▷ Set the minimum distance seen so far
15             matched_t ← i;                     ▷ Set tge index of the closest seed
16     return matched_t, mindist;
```

Direct-Tracker and *Clever-Tracker* consist of simple tracking approaches, suited for the traditional tracking pipeline. They will be used as a basis for our experimental analysis. In the

next section, we describe the estimation of cell positions, based on previously detected seed points.

5.6 The Proposed Pipeline: Establishing Cell Motion

In the last section, we showed tracking algorithms which detect seed points to construct cell trajectories, following the traditional pipeline. We propose changing this pipeline to predict future positions of cells, taking advantage of previously detected ones, to establish cell motion. Accordingly, prediction takes place when we have enough cell points for motion estimation. We propose the *CM-Predictor* (**C**ell **M**otion **Predictor**) algorithm, presented in Algorithm 4 and detailed next.

5.6.1 Motion Estimation

CM-Predictor estimation of new seed points is performed using Lagrange's polynomials (ABDEL-AKHER; SELIM; ALY, 2015), see Equation 5.1. $Pol(x)$ is the polynomial of degree $\leq n-1$, which passes through n data points. $(x_0,y_0),...,(x_j,y_j),...,(x_{n-1},y_{n-1})$ are these n points, where no two x_i are equal, and $0 \leq j \leq n$. Also, $Pol_j(x)$ is the polynomial at j.

$$Pol(x) = \sum_{j=1}^{n} Pol_j(x), \tag{5.1}$$

where

$$Pol_j(x) = y_j \cdot \prod_{k \in \{1,...,n\}-\{j\}} \frac{x - x_k}{x_j - x_k}.$$

Function InterpolatePoints is the pseudo-code for estimating the points of each cell, relying on Equation 5.1.

Function InterpolatePoints(*embryo*)

```
1 foreach cell c in embryo do
2     Ŝ ← c.getSeeds();                    ▷ Get seeds from current cell
3     for (i = 0 to Ŝ.size() − 1) do
4         newSeed.x ← Predict x using Eq. 5.1;
5         newSeed.y ← Predict y using Eq. 5.1;
6         newSeed.z ← Predict z using Eq. 5.1;
7         Add newSeed to the embryo; ▷ newSeed is composed of x, y and z estimated
            coordinates
8 return embryo;
```

If the predicted cell position outputted by *Interpolate-Points* is at least th of distance from the last seed point of the cell, the prediction is deemed correct. Otherwise, this is considered an interpolation error. The algorithms continue to interpolate as long as the percentage of wrongly

predicted cells is less than *maxError*, where $0 \leq maxError \leq 1$. When the error is higher than *maxError*, *CM-Predictor* discards the oldest points used for interpolation, as we discuss later.

5.6.2 Parameters

Table 8 shows the **input** parameters and **output** of *CM-Predictor*. *CM-Predictor* uses the current motion vector until the error exceeds a threshold. When the error is too much, the oldest points of the current motion vector are discarded. Additionally, the window is renewed, as new detected seed points are added to the window used for the interpolation. The window size w is an exogenous parameter in this current study, and we will show its influence experimentally.

Table 8 – Input parameters and output of *CM-Predictor*.

Input Parameter	**Description**
$\hat{S}_{t-1}$	A list with all seeds detected at time $t-1$.
$\hat{S}_t$	A list with all seeds detected at time t.
th	A distance threshold.
w	The window size.
pw	Share of the window discarded after *maxError* is reached.
maxError	Maximum error allowed when interpolating points.
Output	**Description**
embryo	An embryo, composed of moving cells, each one represented as a sequence of seed points.

Source: Elaborated by the author.

5.6.3 The Cell Motion Predictor

Algorithm 4 is the pseudocode of *CM-Predictor*. The tracking takes place until all images from the sequence have been analyzed. While *CM-Predictor* has not processed w images (line 4), a tracker approach matches the cells. Here, the approaches from Section 5.5 can be used by calling Function *PerformTracking* (lines 5 and 6). The window is incremented (line 7), and *CM-Predictor* checks if w images have been processed (lines 8 and 9). If so, the predictor uses the last w seeds of each cell for the interpolation. Here, we consider active cells those that had seed points added at the last iteration $t-1$. Accordingly, in iteration t *CM-Predictor* estimates the next seed points of active cells by calling *Interpolate-Points* (line 11). This means that we use the w last seeds added to the embryo to estimate the next points (lines 16–22).

CM-Predictor estimates new points using Equation 5.1, returning the embryo with actual and predicted seed positions. We apply a product of the distance threshold $c \times th$ to bound the

distance between the existing cell and its predicted new position. This is done to ensure that the predicted point is closer to its original cell than th. If not, the prediction is considered a mistake, and the number of errors is incremented.

After each iteration *CM-Predictor* checks if the number of errors is not acceptable, i.e., greater than $maxError$ (line 12). If so, $pw\%$ of the oldest points in the window are removed and the algorithm processes the next image (from time $t+1$) (line 13). Otherwise, *CM-Predictor* continues to interpolate the seed points, now at time $t+1$. This process repeats until all images from the sequence are processed (line 3).

Algorithm 4: *CM-Predictor*

Input : $\hat{S}_{t-1}$: a list of seeds from time $t-1$
$\hat{S}_t$: a list of seeds from time t
th: a distance threshold
w: the window size
pw: percentage of the window to be discarded
$maxError$: the maximum error allowed in the interpolation of points

Output : The *embryo*, with existing and new seed points

```
1  begin
2      initialization;
3      while (t < sizeOfSequence) do
4          if (window < w) then
5              foreach seed ŝ_timestamp ∈ Ŝ_t do
6                  embryo ← PerformTracking(); ▷ Track using detected seed points
7              window++; ▷ Increment window
8              if (window = w) then
9                  Initialize polynomial; ▷ Use the last seed points of each active cell
10         else
11             embryo ← Interpolate-Points(embryo); ▷ Add estimated points to the
                embryo
12             if (error > maxError) then
13                 Remove pw% of the oldest points in the window and process next
                    image
14         t++;
```

We study the effects of *CM-Predictor*'s parameters experimentally, as described in Section 5.7.4.

5.7 Experimental Analysis

In this section, we detail the experiments performed to validate the *CM-Predictor* approach and the various methods.

Environment Setup. The experiments were performed in an Intel Core i7-4770(3.40GHz), 16GB RAM machine, with Fedora 28 (64-bit) OS.

5.7.1 Evaluation Measures

We employ different measures to validate our methods. The total number of cells created by the trackers, the tracking detection error, and the trajectory of cells are quality measures. The average execution time assesses the performance of the algorithms. Table 9 summarizes the measures, and we explain them as follows. As mentioned before, tracking methods matches new seed points with existing ones and creates cells for seed points that were not matched. We use **total number of cells** created by the tracker and added to the embryo as a measure, and it must ideally be close to the number of cells reported by the ground truth (*GT*) and *TWANG*. Both numbers are considered for evaluation because the *GT* reports the *actual* number of cells present in the image, while *TWANG* outputs the cells which the trackers must work with.

Table 9 – Evaluation measures employed to the trackers and predictors.

Measure	***Description***	***Better Values***
TDR	Tracking detection error	High (close to 1)
TotCells	Total number of cells	Close to the ground truth
Cell trajectory	Trajectory of cells, according to their distance to the image centroid	Visually similar to the ground truth
ExecTime	Average execution time to perform tracking or prediction tasks	Low values (faster)

Source: Elaborated by the author.

A **tracking error** occurs when the algorithm misses a seed-to-cell correspondence or adds a non-existing one. There are two types of this error. The first one occurs when the algorithm erroneously created a new cell, consequently adding a tracking error. The second one occurs when the algorithm adds the new seed point to an existing cell when it should have created a new cell instead. With the number of tracking errors we are able to compute the **Tracker Detection Rate** (***TDR***) (BALOMENOS; TSAKANIKAS; MANOLAKOS, 2015). *TDR* is defined by $TDR = TP/n$, where TP is the True Positive, and in this application refers to the number of images with no tracking errors, and n is the number of images (or time stamps) in the sequence. *TDR* allows to evaluate seed-to-cell correspondences, and values close to 1 represent optimal and desired results. The cell motion estimated by *CM-Predictor* is evaluated by computing the **distance of cell points to the image centroid**. This allows us to observe how cells were matched along time, and to compare the position of cells detected in the image to the ones predicted by *CM-Predictor*. We also evaluate the performance of trackers by computing the **average execution times** (*ExecTime*). This time is computed during the tracking and prediction

steps only, as we do not consider the segmentation time of the approaches. More specifically, *ExecTime* is measured by running each algorithm 1,000 times, and then we take the average.

5.7.2 Material

We perform our evaluation using benchmark datasets provided by the Cell Tracking Challenge[1] (ULMAN *et al.*, 2017). Each of its datasets has its original images, ground truth annotations (manually done) and segmented images. The annotations have the following format: *"id of cell"*, *"initial time"* (which is the first time the cell has been detected in the sequence), *"final time"* (which is the last time the cell has been detected), and *"id of parent cell"* (available only if the cell resulted from the split of another one). Table 10 summarizes the datasets used in each experiment we report in this section. We show the original name and the number of images available, together with the images' type, which can be $2D+t$ or $3D+t$. Figure 31 contains examples of such images, obtained from different timestamps, to exemplify the development of the embryos.

Table 10 – Datasets used in each experiment.

Experiment	*Dataset*	*Set*	*Type*	*# images*
Exp1.1	Fluo-N3DH-SIM+	01	3D+*t*	150
Exp1.2	Fluo-N3DH-SIM+	02	3D+*t*	80
Exp2.1	Fluo-N2DH-GOWT1	01	2D+*t*	92
Exp2.2	Fluo-N2DH-GOWT1	02	2D+*t*	92

Source: Cazzolato, Traina and Böhm (2018).

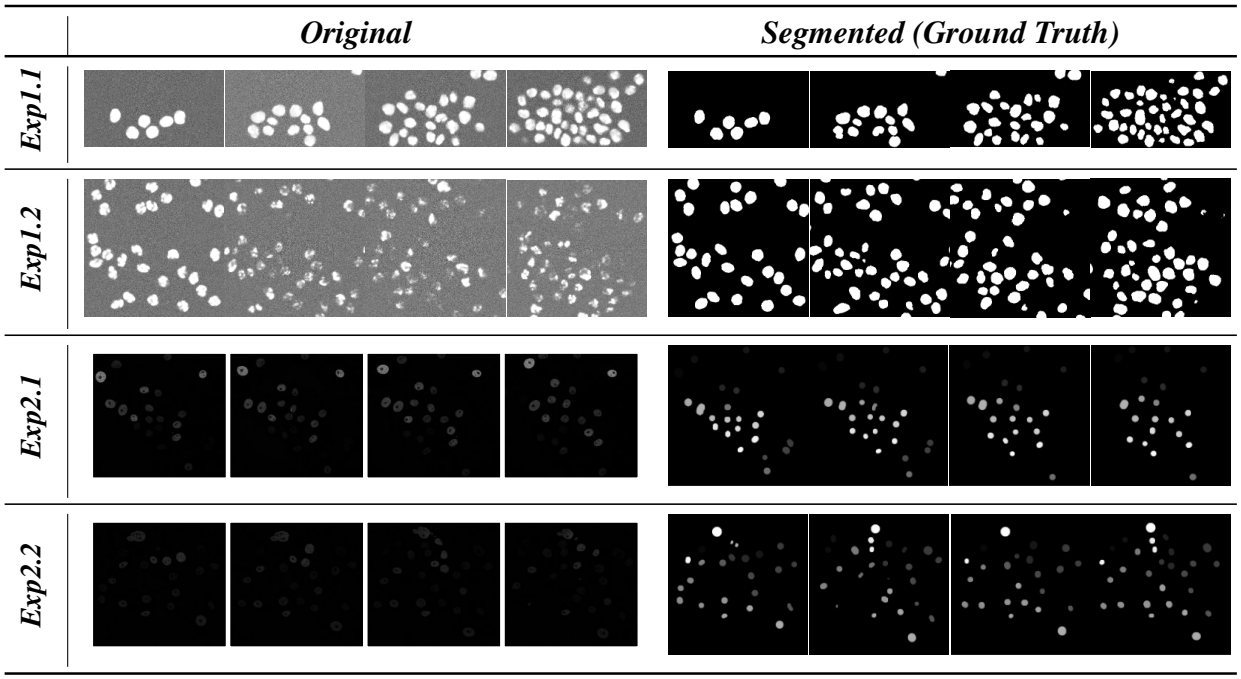
5.7.3 Parameter Setup and Preprocessing Step to Detect Cells

The parameters used for the preprocessing step of the tracking/prediction pipeline are as follows. Regarding the segmentation step, *TWANG* has the parameter values $SpacingX = 0.3$, $SpacingY = 0.3$ and $SpacingZ = 2$ for all experiments, obtained experimentally. The other parameters have their default values, as reported in (STEGMAIER *et al.*, 2014). *TLM-Tracker* provides a set of segmentation algorithms to detect cells from images. We used the Chan-Vese algorithm for Exp2.1 and Watershed for Exp2.2, as they had yielded the best segmentation results, obtained experimentally. *TLM-Tracker* does not support 3D+*t* images, so we have not employed it in Experiments Exp1.1 and Exp1.2. We employed the Euclidean distance to match the points with the *Direct-Tracker*, *Clever-Tracker*, and *CM-Predictor* approaches.

Following we report the results of the algorithms on detecting the cells from the used datasets. We have applied *TWANG* and *TLM-Tracker* to the images (in the segmentation step)

[1] The Cell Tracking Challenge: <codesolorzano.com/Challenges/CTC/Welcome.html>

Figure 31 – Examples of original images (from different timestamps) and the corresponding segmented images.



Source: Elaborated by the author.

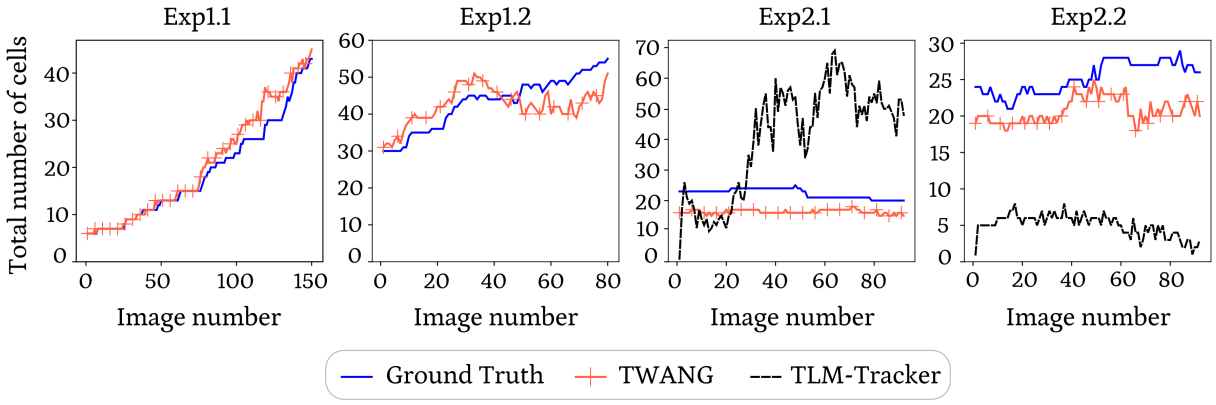
to obtain the seed points that the trackers can then use subsequently. Detecting seed points can be seen as a preprocessing step of our proposal, which extracts the cell information from images at each timestamp. Although this is not the primary goal of our proposal, the detected number of cells influences the tracking and prediction tasks, as it can deviate from the ground truth information. Figure 32 compares the total number of cells detected with *TWANG* and *TLM-Tracker*, in comparison to the ground truth. We observe that *TLM-Tracker* features more divergence than *TWANG* compared to the ground truth. *TLM-Tracker* did not detect all cells in the images from Exp2.1 and Exp2.2, mainly because the cells are faded, i.e., they are not easy to recognize.

Once we have the positions of the cells and associated information, next we proceed to the tracking and prediction steps.

5.7.4 Tracking and Predicting Trajectories of Cells

Firstly we define the best values for the **distance threshold** th parameter. Since this is the only parameter required by *Direct-Tracker* and *Clever-Tracker*, we employ them to define the best value to be used in each experiment. Figure 33 depicts the *TDR* results using *Direct-Tracker* and *Clever-Tracker*. Each point of the plots represents the *TDR* result obtained after processing all images of the sequence, using a specific distance-threshold value. Accordingly, the size of the dots reports the number of cells created by the algorithms. Ideally, the size of the dots should be the same (or similar) to the dark dots, which represent the ground truth (*GT*) number of

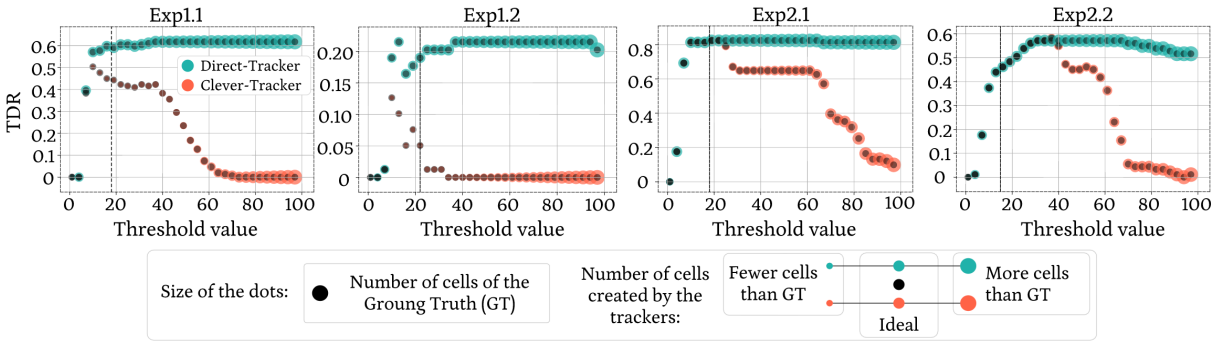
Figure 32 – Total number of cells reported by the ground truth and detected by *TWANG* and *TLM-Tracker*, during the preprocessing step. Notice that *TLM-Tracker* was only employed in Exp2.1 and Exp2.2, since it does not support 3D+*t* images.



Source: Adapted from Cazzolato, Traina and Böhm (2018).

cells. In summary, we look for high *TDR* values and dot sizes close to the *GT*. We observe that the number of cells is higher as *th* increases, thus diverging even more from the ground truth. For small *th* values ($\approx th < 10$), *TDR* results also tend to be low, and the best values are those between approximately 10 and 30. The *th* values used in the experiments are those represented by the dashed vertical lines: $th = 18$ for *Exp1.1*, $th = 22$ for *Exp1.2*, $th = 18$ for *Exp2.1* and $th = 15$ for *Exp2.2*.

Figure 33 – Finding the best threshold *th* value to be used by the trackers.

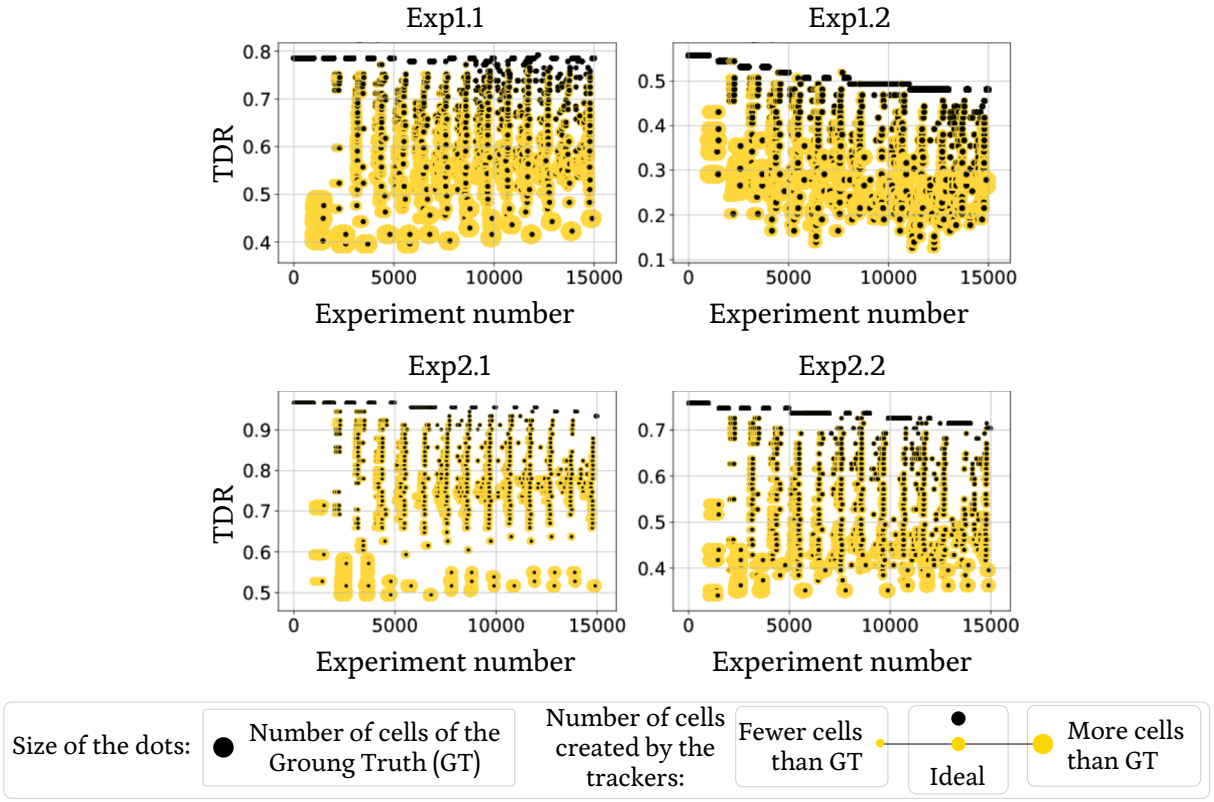


Source: Adapted from Cazzolato, Traina and Böhm (2018).

Evaluating the impact of CM-Predictor's parameters

Table 11 shows the combination of parameters generated for *CM-Predictor*. We show that it resulted in $15 \times 10 \times 10 \times 10 = 15{,}000$ combinations, from now on called configurations. Figure 34 presents the *TDR* results of all configurations. Again, the size of the dots corresponds to the number of cells created, and the dark dots inform us of the number of cells reported by the ground truth. The best results are at the top of the charts. Each point represents the result obtained after the entire image sequence, using the specific configuration. *CM-Predictor* has

Figure 34 – Estimating the best parameters of *CM-Predictor*, by generating different configurations of parameters.



Source: Adapted from Cazzolato, Traina and Böhm (2018).

obtained low *TDR* results when creating many more cells than expected, and high *TDR* results when the number of cells created is smaller than the one of the ground truth. As our goal here is to obtain high *TDR* values and a number of cells close to the ground truth, we only use a subset of the configurations to evaluate the other parameters. Accordingly, we first select all combinations with $TDR > 0.7$ for Exp1.1, $TDR > 0.4$ for Exp1.2, $TDR > 0.85$ for Exp2.1, and $TDR > 0.65$ for Exp2.2.

Table 11 – Combination of parameters tested for *CM-Predictor*.

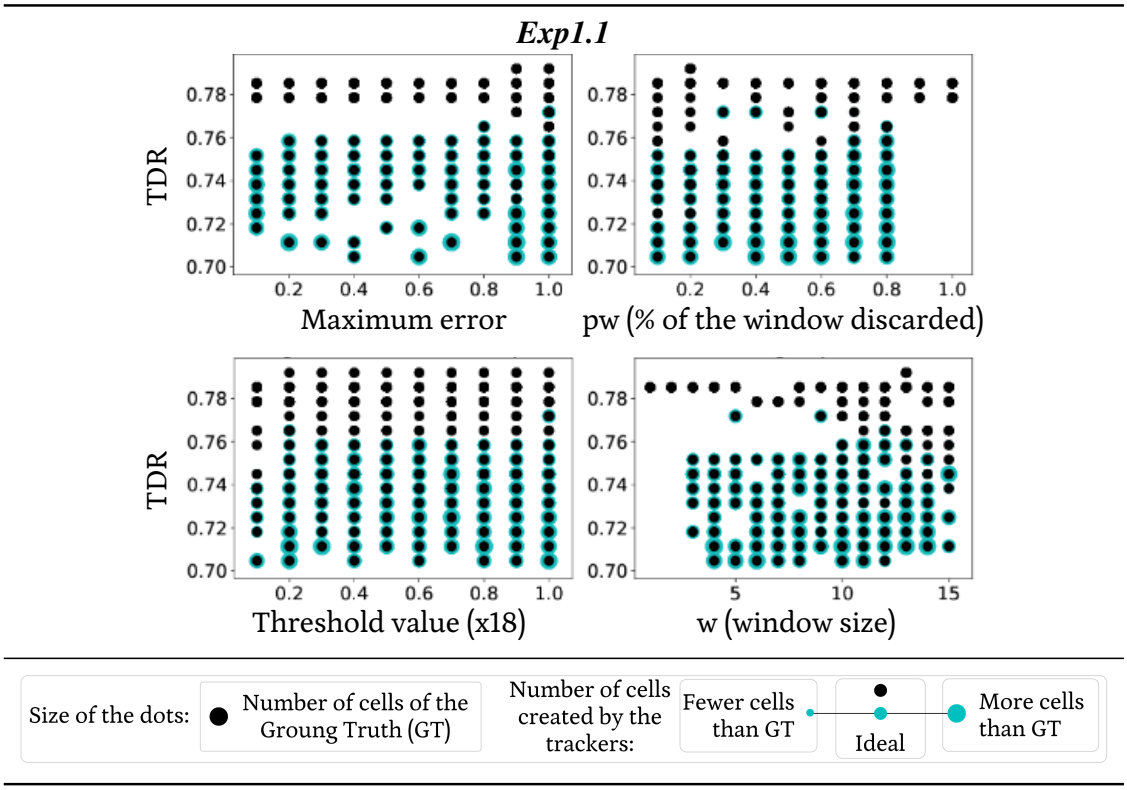
Parameter	***Range of values (min to max)***	***Increment step***
w	1 to 15	1.0
pw	0.1 to 1.0	0.1
$maxError$	$0.1 \times th$ to $1.0 \times th$	0.1
th	$0.1 \times th$ to $1.0 \times th$	0.1

Source: Elaborated by the author.

Figures 35, 36, 37 and 38 show the representative charts, with the outcomes of the configurations selected in the previous step. The best results obtained in Exp1.1 used parameters

$w < 4$, $pw < 0.4$, $maxError$ between 0.4 and 0.8, and the distance threshold of $0.5 \times th$. In Exp1.2, the highest number of cells has been created with $w = 2$, the distance threshold $0.9 \times th$, $pw < 0.6$, and independently of the value of $maxError$. Notice that, in this configuration, the whole window is discarded once *CM-Predictor* removes $\lceil w \times pw \rceil$ points from the window when the error becomes too large. This occurred because, with high th values, points that are not very close from each other in the space do not match, yielding more errors and forcing the window to be updated. On the other hand, with only two points in the window *CM-Predictor* generates poor estimates, resulting on a large distance between points and more cells being created. The best results were achieved with $w = 3$, with high *TDR* results using $4 \leq w \leq 7$. In both Exp2.1 and Exp2.2, $w < 4$ yield high *TDR* results with the number of cells close to the *GT*. The distance threshold of $th > 0.8$ presented the highest concentration of points, meaning that when *CM-Predictor* allows for a relatively higher distance between points to be a match, the results are the best. In general, for all datasets, *CM-Predictor* presented its best behavior with parameter values $w = 3$, $pw = 0.3$ and $maxError = 0.9$.

Figure 35 – Best configurations of parameters for *CM-Predictor* in *Exp1.1*.

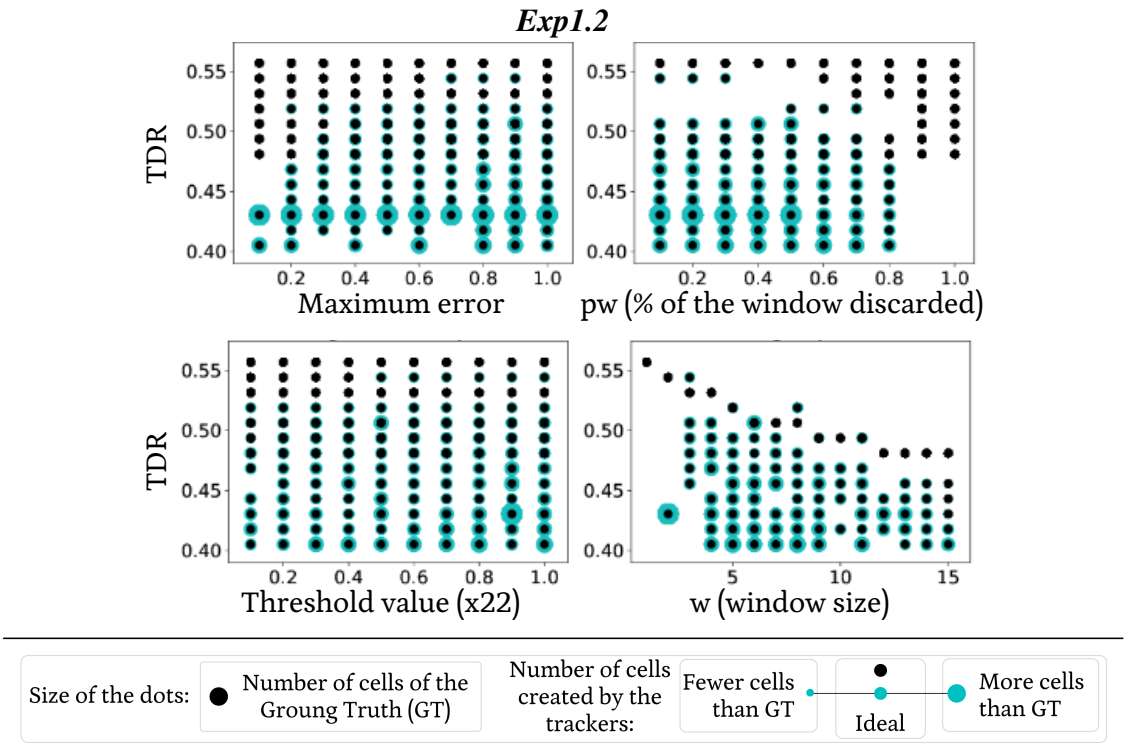


Source: Adapted from Cazzolato, Traina and Böhm (2018).

Comparing the Different Approaches

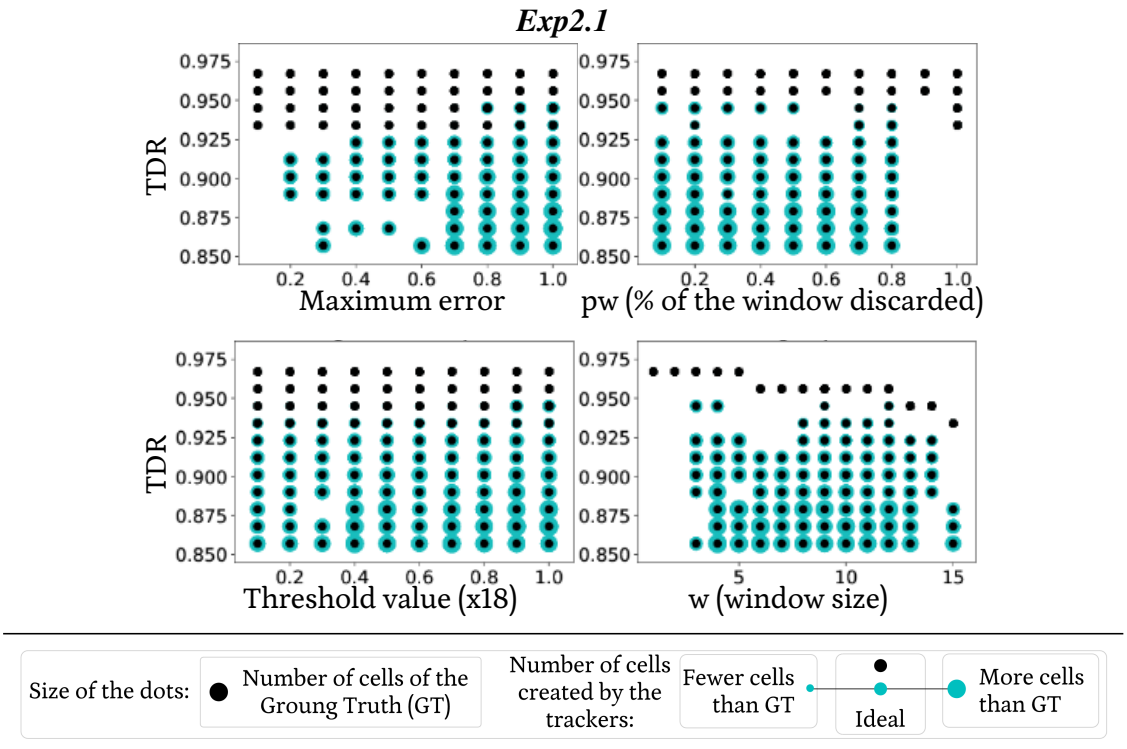
Here we compare the results of *CM-Predictor* with the ones of *Direct-Tracker*, *Clever-Tracker*, and *TLM-Tracker*. *CM-Predictor* parameters employed in these experiments are: $maxError = 0.3$, $th \times 0.4$, $pw = 0.3$ and $w = 4$, with the same th values reported in Figure

Figure 36 – Best configurations of parameters for *CM-Predictor* in *Exp1.2*.



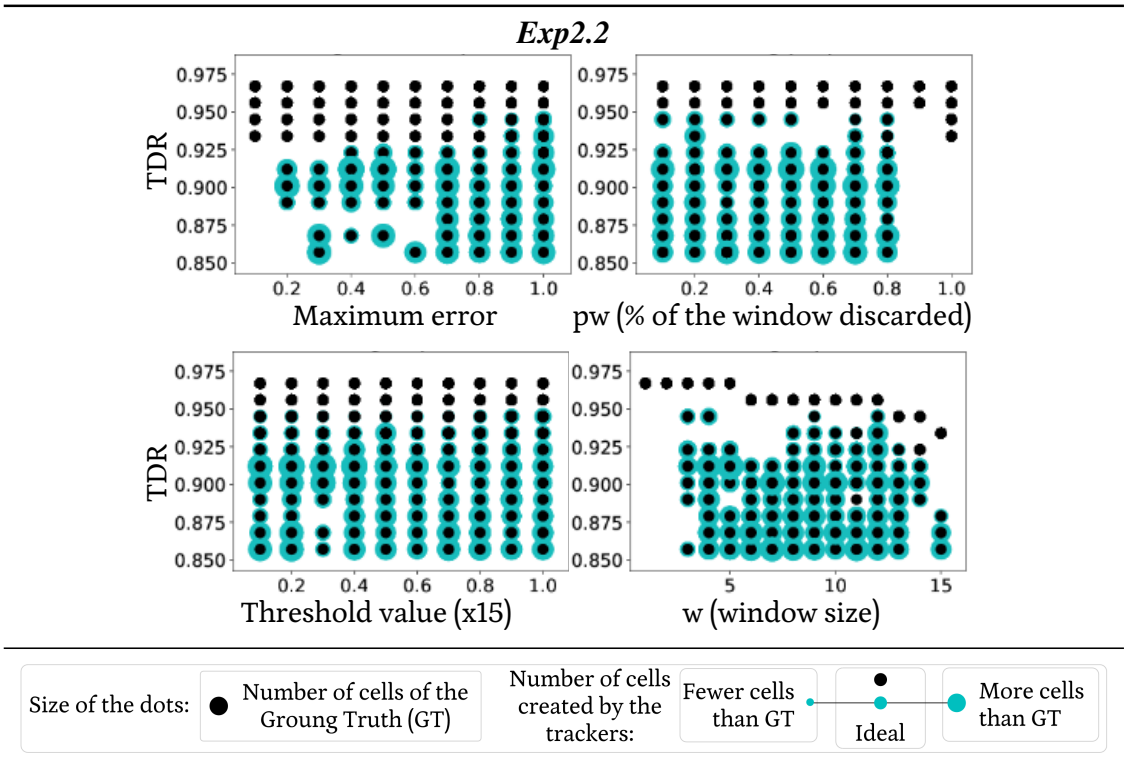
Source: Adapted from Cazzolato, Traina and Böhm (2018).

Figure 37 – Best configurations of parameters for *CM-Predictor* in *Exp2.1*.



Source: Adapted from Cazzolato, Traina and Böhm (2018).

Figure 38 – Best configurations of parameters for *CM-Predictor* in *Exp2.2*.



Source: Adapted from Cazzolato, Traina and Böhm (2018).

33. Table 12 lists the *TDR* results the total numbers of cells created. Our newly proposed methods yield the best results for all experiments, compared to *TLM-Tracker*. As expected, *Direct-Tracker* has created the highest number of cells in Exp1.1 and Exp1.2. This is because its matching has not been correct, as it matches the first match candidate within *th*, and has led to low *TDR* values. Additionally, the high number of false positives and false negatives in the cells detected by *TWANG* (in the preprocessing step) has bogged down the matching precision of *Direct-Tracker*. *Clever-Tracker* has yielded similar results.

During prediction iterations, *CM-Predictor* does not create new cells or delete existing ones, and this is one reason why it has created fewer cells than the other ones. Accordingly, when the number of cells from one image to the next one does not change much, *TDR* tends to perform better, since it does not create false positives. This justifies why *CM-Predictor*'s overall *TDR* results are better than the other algorithms.

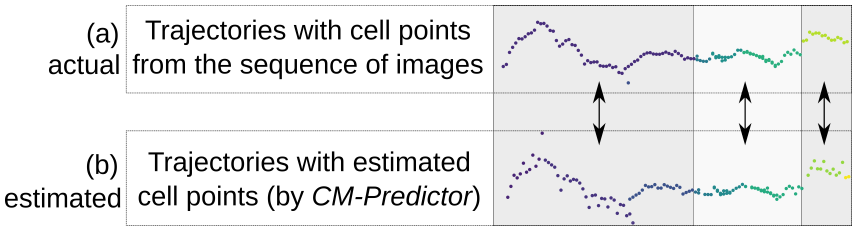
Following we show the trajectories of cells outputted by each algorithm. Figure 39 explains how to compare cells trajectories visually. Notice that we are interested in showing the differences between the actual points (detected by the segmentation algorithms in the images) and the estimated ones. Figures 40, 41 and 42 show the trajectories of cells for Exp1.1, Exp2.1 and Exp2.2, respectively. *Direct-Tracker* and *Clever-Tracker* yield very similar results. This is because they use the same cell points, provided by the segmentation algorithm *TWANG*. *CM-Predictor* was able to predict cell points similar to the actual ones. Even though *CM-*

Table 12 – Comparison of the tracking and prediction approaches.

Algorithm		***Exp1.1***	***Exp1.2***	***Exp2.1***	***Exp2.2***
Direct-Tracker	*TDR*	0.45	0.05	0.82	0.46
	TotCells	440	584	31	81
Clever-Tracker	*TDR*	0.73	0.24	0.62	0.68
	TotCells	144	259	31	81
CM-Predictor	*TDR*	0.58	0.19	0.82	0.46
	TotCells	154	538	185	100
TLM-Tracker	*TDR*	–	–	0.0	0.29
	TotCells	–	–	956	142

Source: Cazzolato, Traina and Böhm (2018).

Figure 39 – Cell trajectories with (a) actual and (b) estimated points.



Source: Cazzolato, Traina and Böhm (2018).

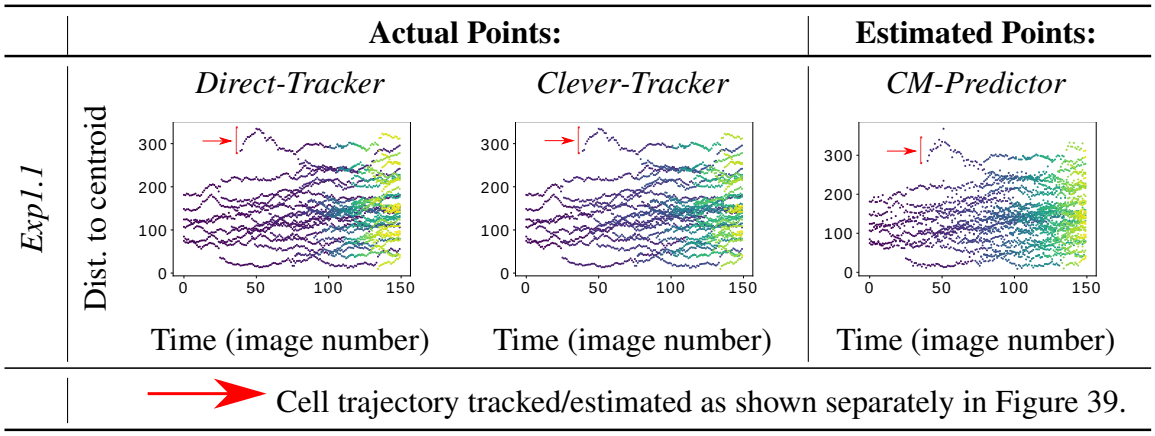
Predictor has predicted most of the points, the estimated motion vectors are similar to the ones detected by *TWANG*. This shows that predicting cells positions based on recently observed cell points is a good approach. Our competitor, *TLM-Tracker*, did not detect many cells in the image segmentation, and consequently, its trajectories differ from the ones given by the other algorithms. Observing the images, we can see that *TLM-Tracker* has created many more cells in Exp2.1 and did not detect all cells in Exp2.2.

Overall, the configurations of *CM-Predictor* show results that are comparable with the ones obtained by tracking algorithms. This shows the advantages of our proposal:

- The tracking detection rate (*TDR*) of *CM-Predictor* is close to the ones reported by the tracking algorithms;
- The predicted trajectories provided by *CM-Predictor* are similar to the ones composed of actual cell points.

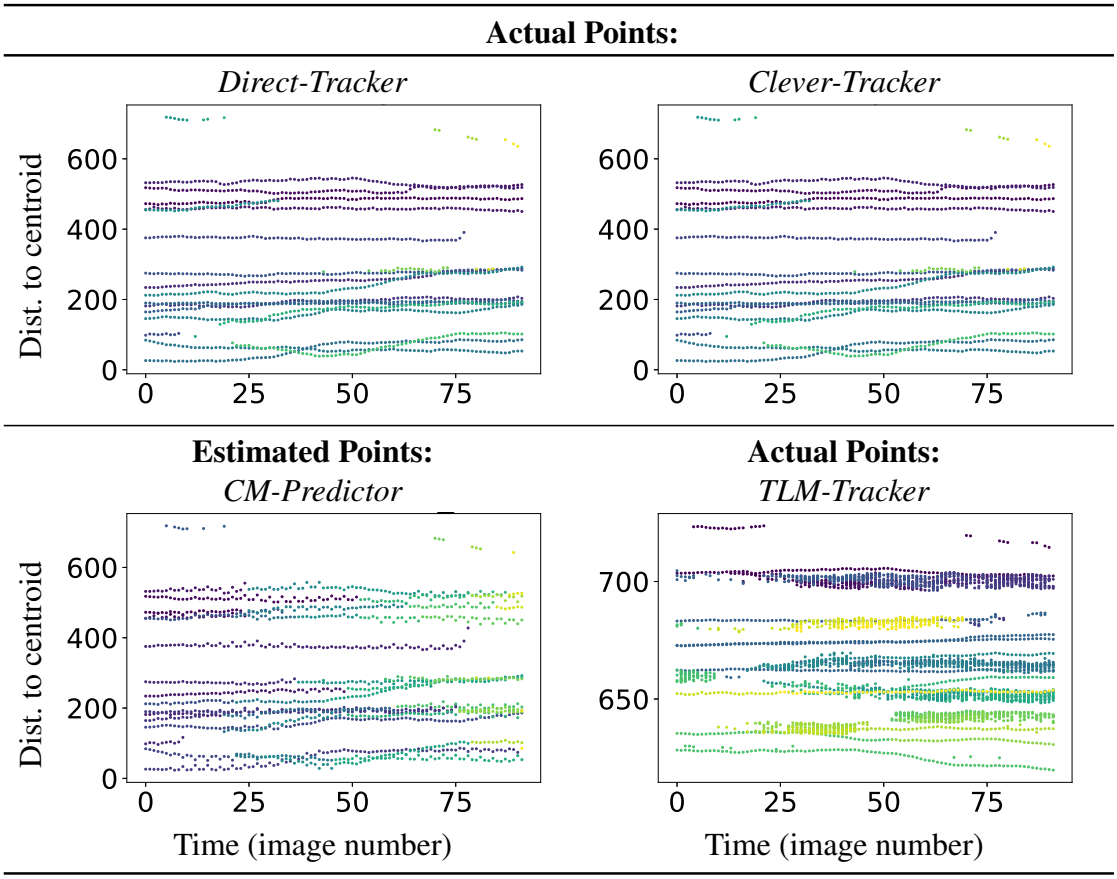
All this means that we are able to improve the tracking pipeline by including predictions of cell motions. We show the performance of *CM-Predictor* and the tracking approaches next.

Figure 40 – Cell trajectories for experiments Exp1.1.



Source: Cazzolato, Traina and Böhm (2018).

Figure 41 – Cell trajectories for experiment Exp2.1.

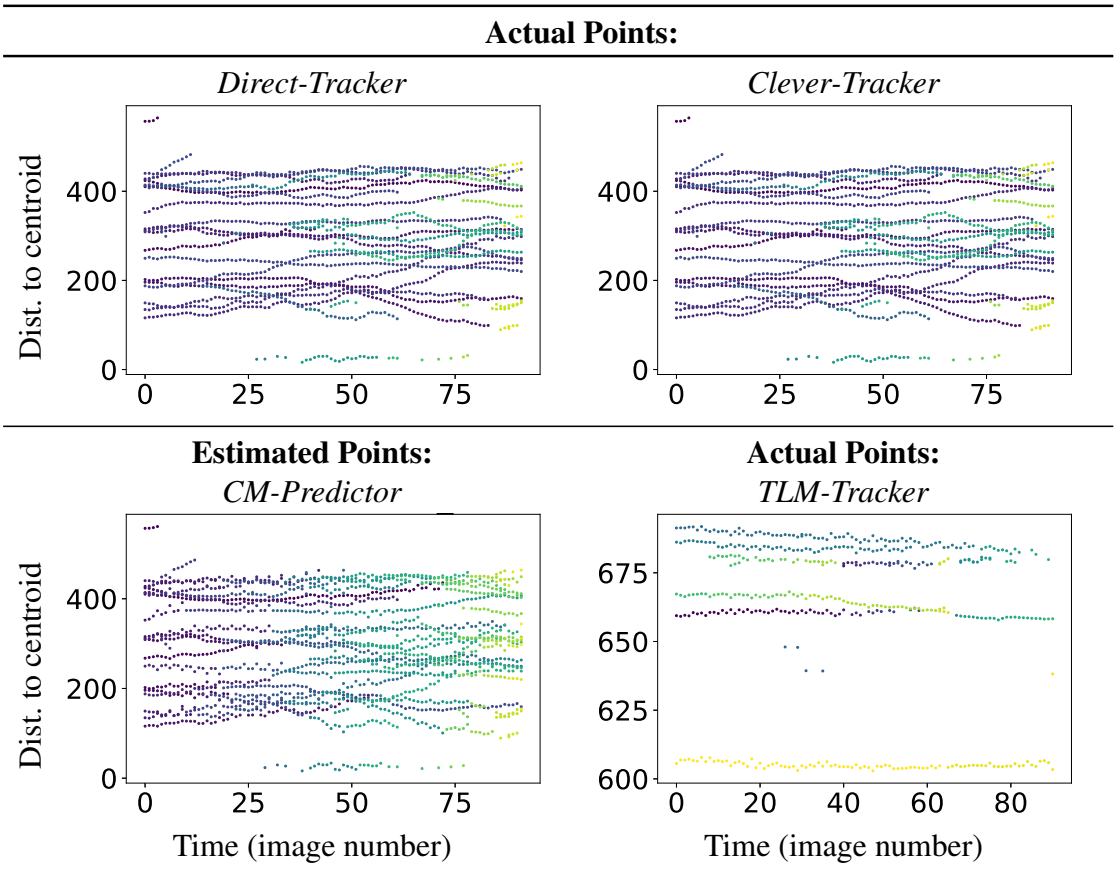


Source: Cazzolato, Traina and Böhm (2018).

5.7.5 Performance Analysis

Table 13 presents the average execution times (in seconds) of the proposed approaches. For a fair comparison, we do not present the numbers for *TLM-Tracker* in the same table, since it has been implemented in *MATLAB*, while the other algorithms are available in *C++*. *TLM-*

Figure 42 – Cell trajectories for experiment Exp2.2.



Source: Cazzolato, Traina and Böhm (2018).

Tracker has performed in 0.348 seconds (Exp2.1) and in 0.416 seconds (Exp2.2). *Direct-Tracker* has shown to be the fastest tracking approach since it works greedily and matches the first cell by simply checking if it is within the distance threshold. *Clever-Tracker* performs more comparisons between new seeds and existing cells. *CM-Predictor* performs no comparisons between points in iterations which prediction takes place, making it much faster than the two tracking approaches. Finally, *TLM-Tracker* works according to the same principle as *Direct-Tracker*, but with the additional cost of predicting the overlapping area of cells, if this option is selected to improve the results of the tracking step.

5.7.6 Discussion

In the last subsections, we have systematically showed how to check desirable values for the parameters required by *CM-Predictor*. The values of w, pw, $maxError$ and th heavily depend on the images being analyzed. Accordingly, we only need to choose the best parameter values once, for a specific type of embryo. This is because the shape and spatial movement of cells are similar between sequences of images. Consequently, once the optimal values are known, one can also use them in subsequent analyses.

Table 13 – Average execution time (in seconds) obtained by the algorithms during the tracking and prediction tasks.

	Average Execution Time (s)			
Algorithm	***Exp1.1***	***Exp1.2***	***Exp2.1***	***Exp2.2***
Direct-Tracker	0.12	0.16	0.42	0.06
Clever-Tracker	0.18	0.27	0.06	0.09
CM-Predictor	0.06	0.20	0.06	0.04

Source: Adapted from Cazzolato, Traina and Böhm (2018).

5.8 Track and Prediction Using Indexing Data Structures

In this section we aim at answering the following question:

- Can we speed-up the tracking task with indexing structures, consequently improving the accuracy?

We address this problem by employing indexing structures over the seed points being matched. *Direct-Tracker*, *Clever-Tracker* and *CM-Predictor* rely on a Brute-Force (*BF*) search strategy to match seed points to existing cells. Although *BF* can be efficient for small sets of data, as the number of samples increases *BF* becomes very costly and unfeaseble. Two examples of existing structures to speed-up *k-NN* queries are *kd-Tree* and *Ball-Tree*, previously described in Section 2.1.4.

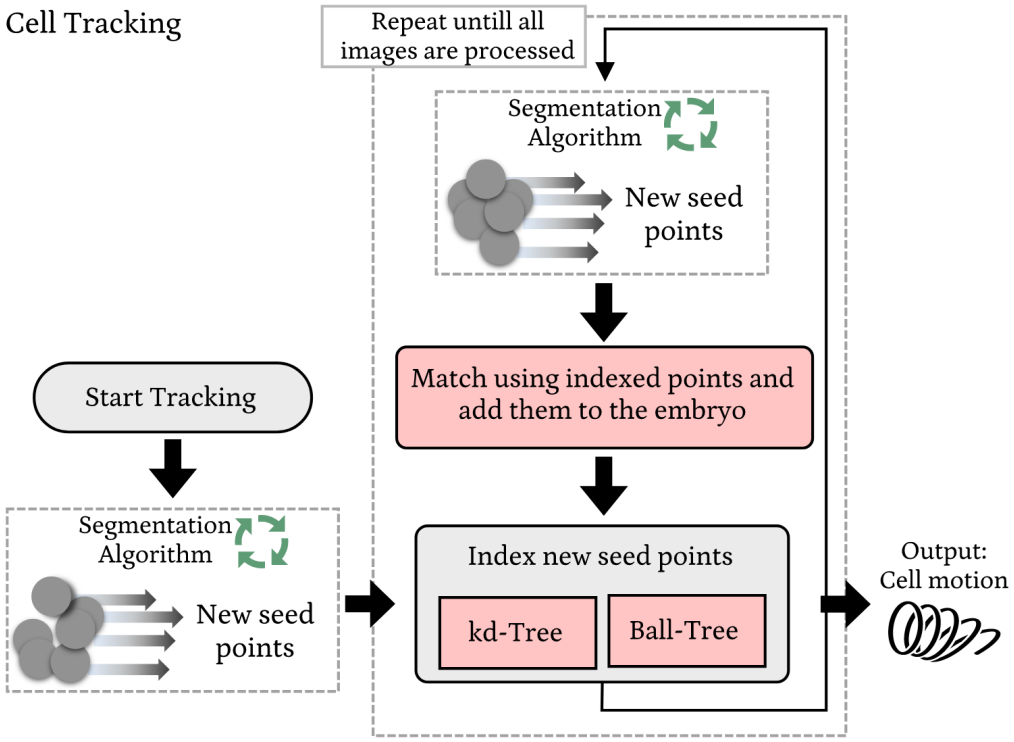
We used *kd-Tree* and *Ball-Tree* structures to match new seed points to the already existing cells. Figure 43 illustrates the tracking pipeline using the indexing structures. After reading the cell information at each iteration, the algorithm uses the previously indexed seed points to match the new seeds. Similarly, when employing the predictor approach, the algorithm uses the indexing structures in iterations whenever the cell tracking takes place. Accordingly, we will refer to the new versions of the tracking and prediction algorithms as *kdt-Tracker* and *kdtCM-Predictor* (when using *kd-Tree*), and *blt-Tracker* and *bltCM-Predictor* (when using *Ball-Tree*).

Evaluating the Execution Time

Environment Setup. The algorithms tested in this section were implemented in Python 3.7.2, and the experiments were performed on a Intel Core i7-4770(3.40GHz) 16GB RAM machine, with Fedora 29 (64-bit) OS.

Figure 44 presents the execution time of the tracking approaches. Although there is the extra cost of building the structure at each iteration of the pipeline (as new cells arrive), the lower cost of cell matching compensates the execution time. For *Exp1.1*, the execution times of the

Figure 43 – Working flow of tracking and prediction approaches using indexing structures.



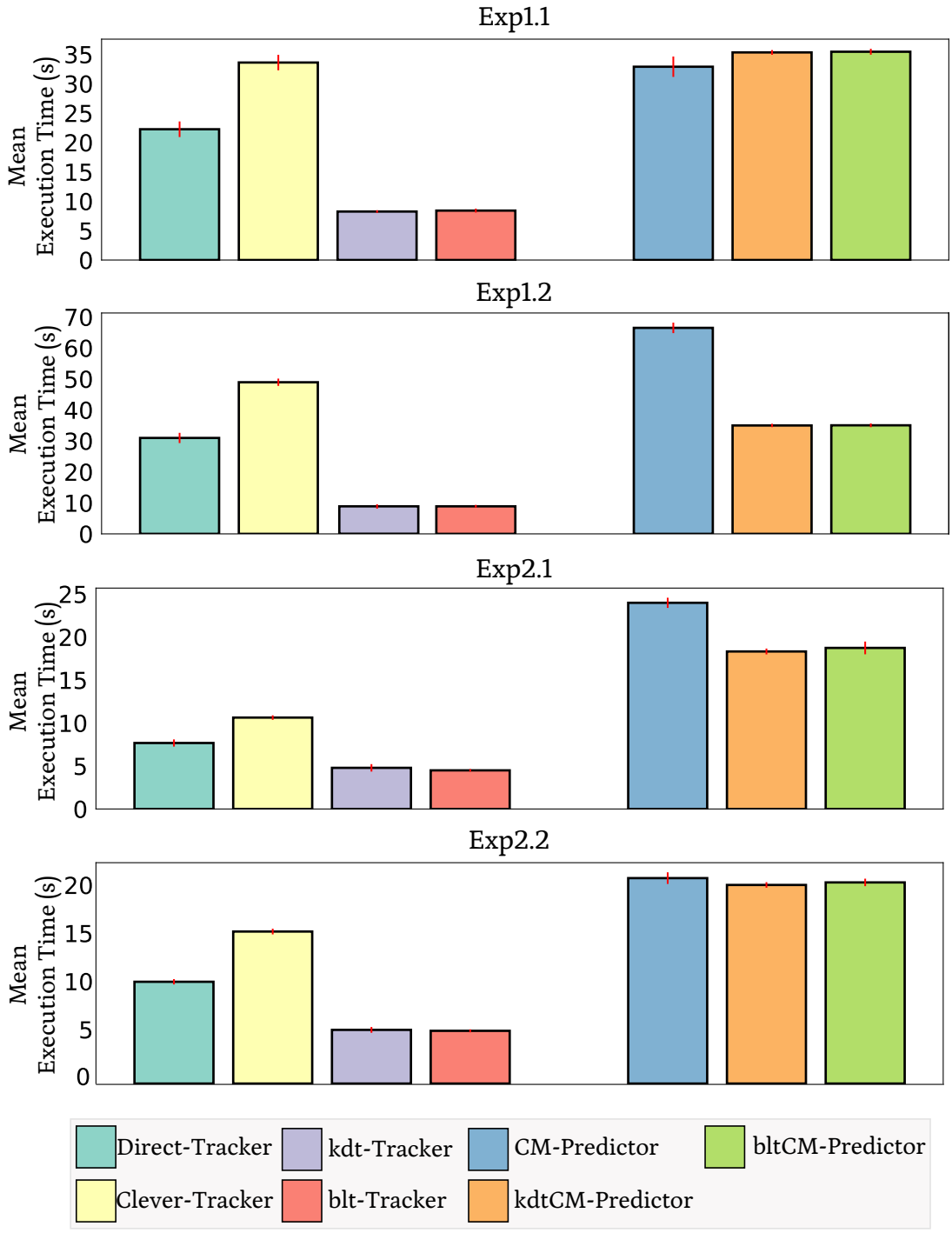
Source: Elaborated by the author.

indexed versions of *CM-Predictor* were higher because, in this particular case, the algorithm created more cells. With the indexing structures, **we removed the threshold** th **parameter**, since we do not rely on it to consider a seed point a match or not. Accordingly, we were able to successfully speed-up the tracking pipeline, obtaining by up to 81.9% of improvement regarding the mean execution time of tracking approaches, and up to 47.3% of improvement regarding the mean execution time of prediction approaches.

5.9 Lessons Learned

We have learned that *Direct-Tracker* directly depends on the order of seed points detected by the cell segmentation algorithm. Accordingly, if the employed algorithm randomly outputs the position of cells, *Direct-Tracker* will perform many wrong seed-to-cell correspondences. This problem occurs because *Direct-Tracker* is greedy, as it matches any seed point within the given threshold distance. Also, the chosen segmentation algorithm does not need to provide the perfect shape of cells. However, the tracking and prediction approaches must have as input the approximated location of as many cells as possible, detected by the segmentation algorithm. For instance, if a seed point appears in one image at time t, disappears in the next point of time $t+1$, and appears again at $t+2$, it will be considered two different cells: one from t, and one from $t+2$. Such miss detections harm the overall tracking and prediction results. A workaround employed in the literature is the removal of orphan cells, which consist of cell trajectories with a

Figure 44 – Execution time of trackers using indexing structures.



Source: Elaborated by the author.

size of ≤ 3.

5.10 Final Considerations

In this chapter we have studied and implemented the tracking and prediction of cell trajectories. This information is available from sequences of microscopic images depicting developing embryos. Regarding the application scenario and after having proposed two tracking algorithms (*Direct-Tracker* and *Clever-Tracker*), our core contribution is a new predictor algorithm (*CM-Predictor*). We have focused on the prediction of cell positions based on previous ones, and we were able to speed up the tracking, while maintaining high accuracy. *CM-Predictor* was up to three times faster than *Clever-Tracker* and six times faster than *Direct-Tracker* when tracking/predicting cells. Our experiments show that *CM-Predictor* performs well with both 2D+*t* and 3D+*t*. It has been able to accurately estimate the motion of cells of a developing embryo over time. Regarding TDR results, *CM-Predictor* was up to 49% better than *Direct-Tracker* (in Exp1.2), up 31% better than *Clever-Tracker* (in Exp2.1) and up to 66% better than *TLM-Tracker* (in Exp2.2), using parameters $w = 3$, $pw = 0.3$ and $maxError = 0.9$. We conclude that *CM-Predictor* is an enhancement for cell tracking, serving its purpose of improving and speeding up the detection cells' motion.

CM-Predictor, *Direct-Tracker* and *Clever-Tracker* approaches, together with all corresponding results, were published in the work (CAZZOLATO; TRAINA; BÖHM, 2018). They are direct results obtained during an internship period at the Karlsruhe Institute of Technology (KIT), in Germany.

Concerning the objectives of this PhD research, in this chapter *approached the third research problem of analyzing moving objects in sequences of images*. The reported results show the advantages of using region-based approaches to track objects over time. We have shown that we designed a solution that allow us to use previous positions of objects to effectively estimate their future movement, consequently obtaining reliable approximate trajectory vectors. *CM-Predictor*, *Direct-Tracker* and *Clever-Tracker* are able to effectively estimate and track cells over time. Also, we showed that indexing structures are able to speed-up the tracking and prediction tasks by up to 81.9% and 47.3%, respectively.

CHAPTER

6

CONCLUSIONS

In this Ph.D. research, we focused on methods that take advantage of region-based approaches to improve the mining and analysis of images and their additional information. Specifically, we presented three main contributions, that we discuss next, together with the research question and thesis of the Ph.D. research.

This chapter is organized as follows. Section 6.1 states the main contributions of this thesis. Section 6.2 lists the future work we envision. Section 6.3 summarizes the list of publications resulted from this Ph.D. research and related works.

6.1 Contributions

As we stated in the Introduction chapter of this thesis, the main goal of this Ph.D. research was to answer the following question: *"How can we improve the image mining task of different application contexts relying on complex data and associated information?"*. This is a challenging question since it involves three characteristics of the image mining task: it can be complex, diverse, and dependent on the application domain. From this scenario, we obtained three main contributions:

1. The *SmokeBlock* method, capable of detecting and segmenting image regions depicting smoke. We show that considering a hard problem such as the classification of images as containing smoke or not, by employing region-based approaches, we can obtain more accurate results than global approaches.

2. *BREATH* and *dp-BREATH* methods, that also take advantage of region-based approaches, but perform soft classification and clustering analysis, labeling and highlighting image regions according to their probability of containing abnormalities.

3. *CM-Predictor*, *Direct-Tracker*, and *Clever-Tracker* methods, that consider the region-based analysis of images combined with time information. The methods can predict and track cells over time, obtained from evolving embryos. We accurately improved the tracking pipeline with alternative approaches for motion estimation. Also, we showed how indexing structures can be used to improve matching tasks by speeding-up the process and to reduce the number of required parameters from the trackers. Finally, we presented alternative models for the prediction of the cells' movement.

A **general contribution** of this Ph.D. research consists of a methodology to handle visual patterns of images. Such methodology consists of dividing the image (or sequences of images) into visually-similar small regions, and process each region (or sets of regions) according to the semantics of the problem. For *SmokeBlock*, *dp-BREATH*, and the trackers and predictor, this approach brought better results than the global approaches.

Regarding the thesis of this doctorate research:

> **Thesis.** *The analysis of image regions, combined with additional information, leads to more accurate mining results regarding the entire image and can also help the processing of sequences of images, speeding-up costly pipelines and making it possible to infer knowledge from objects' movement.*

We are able to say that, in fact, region-based analysis of images can lead to improvements regarding data mining and analysis tasks, considering the applications evaluated. The use of available labels and time information also helps to infer knowledge from moving objects, obtained from regions of images, making it possible to provide an efficient and reliable representation of such objects.

Next, we point out future developments that can be done, based on the results of the research achieved in this doctorate work.

6.2 Future Work

As future work, we envision the following developments:

- Extending *SmokeBlock* using Deep Learning approaches: Recent works from the literature have reported highly accurate results regarding the classification of images, from different contexts. For instance, the fire detection task for still images has been improved in the work (SHARMA *et al.*, 2017). Accordingly, different Deep Learning models can be trained to also enhance the smoke detection task in still images.

- Extending *BREATH* and *dp-BREATH* models to other medical application scenarios: The probabilistic approaches employed to map lung abnormalities according to their

distribution can be extended to work with different types of images, such as skin ulcer photos, MRIs and brain scans.

- Extending *BREATH* and *dp-BREATH* to consider the location of abnormalities: We have learned from specialists that the location of lung visual patterns can also determine the possible abnormalities presented in the area. Accordingly, a broader analysis, considering the region of the lung that each superpixel belongs, can bring promising results to the analysis.

- Combining segmentation algorithms to cell prediction and tracking approaches, such as *Direct-Tracker*, *Clever-Tracker*, and *CM-Predictor*: By combining the segmentation and prediction/tracking tasks, improvements can be achieved, not only in execution time but also in the accuracy of the analysis.

- Provide an interactive visualization of trajectories of cells: Biologists can take advantage of intuitive representations of the obtained/estimated trajectory vectors.

- Use different configurations of the tracking pipeline: By interpolating points of intermediary windows, the exactitude of the predicted points can be better than by extrapolating future points. In this case, the algorithm would read two windows of points, with an interval between them, which will be filled up by the interpolated points.

6.3 List of Publications

We describe as the *main publications* the ones directly resulted from this Ph.D. research. *Complementary publications* are those resulted from contributions with the research group GBdI, that also helped to achieve the results reported in this thesis and contributed to the formation of the Ph.D. candidate. We list all publications next, together with the *Qualis-CC Capes* category from when the work was published (when available), and the year of publication.

Main publications:

- ***Journal CMPB 2019 - Qualis-CC Capes A2 (CAZZOLATO et al., 2019):*** CAZZOLATO, M. T.; SCABORA, L. C.; NESSO-JR., M. R.; MILANO-OLIVEIRA, L. F.; COSTA, A. F.; KASTER, D. S.; KOENIGKAN-SANTOS, M.; AZEVEDO-MARQUES, P. M.; TRAINA-Jr, C.; TRAINA, A. J. M.. ***dp-BREATH*: Heat maps and probabilistic classification assisting the analysis of abnormal lung regions.** Journal of Computer Methods and Programs in Biomedicine (CMPB), May, 2019. p. 27–34.

- ***CIKM 2018 - Qualis-CC Capes A1 (CAZZOLATO; TRAINA; BÖHM, 2018):*** CAZZOLATO, M. T.; TRAINA, A. J. M.; BOEHM, K.. **Efficient and Reliable Estimation of**

Cell Positions. In: International Conference on Information and Knowledge Management, Turin, Italy. October 22-26, 2018. p. 1043–1052.

- ***CBMS 2017 - Qualis-CC Capes B1 (CAZZOLATO et al., 2017):*** CAZZOLATO, M. T.; SCABORA, L. C.; COSTA, A. F.; NESSO-JR., M. R.; MILANO-OLIVEIRA, L. F.; KASTER, D. S.; TRAINA-Jr, C.; TRAINA, A. J. M.. **BREATH: Heat Maps Assisting the Detection of Abnormal Lung Regions in CT Scans.** In: 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS). Thessaloniki, Greece. June 22-24, 2017. p. 248–253.

- ***SBBD-DSW 2017 (CAZZOLATO et al., 2017):*** CAZZOLATO, M. T.; AVALHAIS, L. P. S.; CHINO, D. Y. T.; RAMOS, J. S.; SOUZA, J. A.; RODRIGUES-Jr, J. F.; TRAINA, A. J. M.. **FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis.** In: SBBD2017 - SBBD Proceedings of Satellite Events of the 32nd Brazilian Symposium on Databases - DSW (Dataset Showcase Workshop). Uberlândia, Brazil. October 2-5, 2017. p. 213–223.

- ***SAC 2016 - Qualis-CC Capes A1 (CAZZOLATO et al., 2016):*** CAZZOLATO, M. T.; BEDO, M. V. N.; COSTA, A. F.; SOUZA, J. A.; TRAINA-Jr, C.; RODRIGUES-Jr, J. F.; TRAINA, A. J. M.. **Unveiling smoke in social images with the SmokeBlock approach.** In: 31st Annual ACM Symposium on Applied Computing. Pisa, Italy. April 4-8, 2016. p. 49–54.

Complementary contributions:

- ***SBBD 2019 (to appear):*** CAZZOLATO, M. T.; RODRIGUES, L. S.; SCABORA, L. C.; ZABOT, G.; VASCONCELOS, G. Q.; CHINO, D. Y. T.; JORGE, A. E. S.; CORDEIRO, R. L. F.; TRAINA-Jr, C.; TRAINA, J. M.. **A DBMS-Based System for Content-Based Retrieval and Analysis of Skin Ulcer Images in Medical Practice** In: 34th Brazilian Symposium on Databases (SBBD - to appear). Fortaleza, Brazil. 2019. p 1–12.

- ***SBBD-DSW 2019 (to appear):*** CAZZOLATO, M. T.; GIUNTINI, F. T.; RUIZ, L. P.; KIRCHNER, L. F.; PASSARELLI, D. A.; REIS, M. J. D.; TRAINA-Jr., C.; UEYAMA, J.; TRAINA, A. J. M.. **Beyond Tears and Smiles with *ReactSet*: Records of Users' Emotions in Facebook Posts.** In: SBBD2019 - SBBD Proceedings of Satellite Events of the 34th Brazilian Symposium on Databases - DSW (Dataset Showcase Workshop). Fortaleza, Brazil. October 7-10, 2019. p. 1–11.

- ***SBBD-DSW 2019 (to appear):*** SCABORA, L.; SPADON, G.; RODRIGUES, L. S.; CAZZOLATO, M. T.; ARAUJO, M. V. S.; SOUSA, E. P. M.; TRAINA, A. J. M.; RODRIGUES-Jr., J. F.; TRAINA-Jr., C.. G-FranC: **A dataset of Criminal Activities mapped as a**

Complex Network in a Relational DBMS. In: SBBD2019 - SBBD Proceedings of Satellite Events of the 34th Brazilian Symposium on Databases - DSW (Dataset Showcase Workshop). Fortaleza, Brazil. October 7-10, 2019. p. 1–11.

- ***SIBGRAPI 2019 - Qualis-CC Capes B1 (to appear):*** RAMOS, J. S.; CAZZOLATO, M. T.; FAICAL, B.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M.. **Fast and smart segmentation of paraspinal muscles in magnetic resonance imaging with CleverSeg.** In: 32nd Conference on Graphics, Patterns and Images (SIBGRAP - to appear), 2019. p. 1-6.

- ***CBMS 2019 - Qualis-CC Capes B1 (ZABOT et al., 2019):*** ZABOT, G.; FAICAL, B.; CAZZOLATO, M. T.; SCABORA, L. C.; TRAINA, A. J. M.; TRAINA-Jr, C.. **UCORM: Indexing Uncorrelated Metric Spaces for Concise Content-Based Retrieval of Medical Images.** In: 32nd IEEE Internatinal Symposium on Computer-Based Medical Systems (CBMS), 2019. p. 306-311.

- ***CBMS 2019 - Qualis-CC Capes B1 (RAMOS et al., 2019):*** RAMOS, J. S.; CAZZOLATO, M. T.; FAICAL, B.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M.. **3DBGrowth: volumetric vertebrae segmentation and reconstruction in magnetic resonance imaging.** In: 32nd IEEE Internatinal Symposium on Computer-Based Medical Systems (CBMS), 2019. p. 435-440.

- ***Journal JBHI 2018 - Qualis-CC Capes A1 (OLIVEIRA et al., 2018):*** OLIVEIRA, P. H.; SCABORA, L. C. ; CAZZOLATO, M. T.; OLIVEIRA, W. D.; PAIXAO, R. S.; TRAINA, A. J. M.; TRAINA-Jr, C.. **Employing Domain Indexes to Efficiently Query Medical Data from Multiple Repositories.** IEEE Journal of Biomedical and Health Informatics. December, 2018. p. 1–9.

- ***CBMS 2018 - Qualis-CC Capes B1 (NESSO-JR et al., 2018):*** NESSO-JR, M. R.; CAZZOLATO, M. T.; SCABORA, L. C.; OLIVEIRA, P. H.; SPADON, G.; SOUZA, J. A.; OLIVEIRA, WILLIAN; RODRIGUES-Jr., J. F.; TRAINA, A. J. M.; TRAINA-Jr, C.. **RAFIKI: Retrieval-Based Application for Imaging and Knowledge Investigation.** In: 31st IEEE Internatinal Symposium on Computer-Based Medical Systems (CBMS). Karlstad, Sweeden. June 18-21, 2018. p. 71–76.

- ***CBMS 2018 - Qualis-CC Capes B1 (CHINO et al., 2018):*** CHINO, D. Y. T.; SCABORA, L. C.; CAZZOLATO, M. T.; JORGE, A. E. S.; TRAINA-Jr., C.; TRAINA, A. J. M.. **ICARUS: Retrieving Skin Ulcer Images Through Bag-of-Signatures.** In: 31st IEEE Internatinal Symposium on Computer-Based Medical Systems (CBMS). Karlstad, Sweden. June 18-21, 2018. p. 82–87.

- ***CBMS 2017 - Qualis-CC Capes B1 (OLIVEIRA et al., 2017a):*** OLIVEIRA, P. H.; SCABORA, L. C. ; CAZZOLATO, M. T.; OLIVEIRA, W. D.; TRAINA, A. J. M.; TRAINA-Jr,

C.. **Efficiently Indexing Multiple Repositories of Medical Image Databases.** In: 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS). Thessaloniki, Greece. June 22-24 2017. p. 286–291.

- ***SBBD-DSW 2017 (OLIVEIRA et al., 2017b):*** OLIVEIRA, P. H.; SCABORA, L. C.; CAZZOLATO, M. T.; BEDO, M. V. N.; TRAINA, A. J. M. ; TRAINA-Jr., C.. **MAMMOSET: An Enhanced Dataset of Mammograms.** In: SBBD2017 - SBBD Proceedings of Satellite Events of the 32nd Brazilian Symposium on Databases - DSW (Dataset Showcase Workshop). Uberlândia, Brazil. October 2-5, 2017. p. 256–266.

- ***SISAP 2017 - Qualis-CC Capes B2 (LAVERDE et al., 2017):*** LAVERDE, N. A.; CAZZOLATO, M. T.; TRAINA, Agma J. M.; TRAINA-Jr., C.. **Semantic Similarity Group By Operators for Metric Data.** In: 10th International Conference on Similarity Search and Applications (SISAP). Munich, Germany. October 4-6, 2017. p. 247–261.

- ***ISM 2016 - Qualis-CC Capes B1 (Blanco et al., 2016):*** BLANCO, G.; BEDO, M. V. N.; CAZZOLATO, M. T.; SANTOS, L. F. D.; JORGE, A. E. S.; TRAINA, A. J. M.; AZEVEDO-MARQUES, P. M.; TRAINA JR, C.. **A Label-Scaled Similarity Measure for Content-Based Image Retrieval.** In: IEEE International Symposium on Multimedia. San Jose, USA. December, 11-13, 2016. p. 20–25.

- ***SAC 2016 - Qualis-CC Capes A1 (SOUZA; CAZZOLATO; TRAINA, 2016):*** SOUZA, J. A.; CAZZOLATO, M. T. ; TRAINA, A. J. M.. **ClusMAM: fast and effective unsupervised clustering of large complex datasets using metric access methods.** In: 31st Annual ACM Symposium on Applied Computing. Pisa, Italy. April 4-8, 2016. p. 986–991.

- ***LNBIP 2015 (BEDO et al., 2016):*** BEDO, M. V. N.; OLIVEIRA, W. D.; CAZZOLATO, M. T.; COSTA, A. F.; BLANCO, G.; RODRIGUES-Jr, J. F.; TRAINA, A. J. M.; TRAINA, C.. **Fire Detection from Social Media Images by Means of Instance-Based Learning.** In: Slimane Hammoudi; Leszek Maciaszek; Ernest Teniente; Oliver Camp; Jose Cordeiro. (Org.). Lecture Notes in Business Information Processing. 1ed.: Springer International Publishing. 2015. p. 23–44.

- ***ICEIS 2015 - Qualis-CC Capes B1 (BEDO et al., 2015):*** BEDO, M. V. N.; BLANCO, G.; OLIVEIRA, W. D.; CAZZOLATO, M. T.; COSTA, A. F.; RODRIGUES-Jr, J. F.; TRAINA, A; J. M.; TRAINA-Jr, C.. **Techniques for Effective and Efficient Fire Detection from Social Media Images.** In: 17th International Conference on Enterprise Information Systems (ICEIS). Barcelona, Spain. April 27-30, 2015. p. 34-45.

BIBLIOGRAPHY

ABDEL-AKHER, M.; SELIM, A.; ALY, M. M. Initialised load-flow analysis based on lagrange polynomial approximation for efficient quasi-static time-series simulation. **IET Generation, Transmission Distribution**, v. 9, n. 16, p. 2768–2774, 2015. ISSN 1751-8687. Available: <http://dx.doi.org/10.1049/iet-gtd.2015.0866>. Citations on pages 94 and 100.

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SUESSTRUNK, S. **SLIC Superpixels**. EPFL, 2010. Available: <http://infoscience.epfl.ch/record/149300/files/SLIC_Superpixels_TR_2.pdf>. Citation on page 63.

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SÜSSTRUNK, S. SLIC superpixels compared to state-of-the-art superpixel methods. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 11, p. 2274–2282, Nov 2012. ISSN 0162-8828. Available: <http://dx.doi.org/10.1109/TPAMI.2012.120>. Citations on pages 51, 52, and 53.

AGARWAL, C.; SHARMA, A. Image understanding using decision tree based machine learning. In: **Information Technology and Multimedia (ICIM), 2011 International Conference on**. Kuala Lumpur, Malaysia: IEEE, 2011. p. 1–8. Available: <http://dx.doi.org/10.1109/ICIMU.2011.6122757>. Citation on page 43.

AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 6, n. 1, p. 37–66, jan 1991. ISSN 0885-6125. Available: <http://dx.doi.org/10.1007/BF00153759>. Citation on page 43.

AKGUL, C. B.; RUBIN, D. L.; NAPEL, S.; BEAULIEU, C. F.; GREENSPAN, H.; ACAR, B. Content-based image retrieval in radiology: Current status and future directions. **Journal of Digital Imaging**, v. 24, n. 2, p. 208–222, 2011. ISSN 08971889. Available: <http://dx.doi.org/10.1007/s10278-010-9290-9>. Citation on page 75.

ALMEIDA, E. S.; RANGAYYAN, R. M.; AZEVEDO-MARQUES, P. M. Gaussian mixture modeling for statistical analysis of features of high-resolution CT images of diffuse pulmonary diseases. In: **IEEE International Symposium on Medical Measurements and Applications, MeMeA, Torino, Italy, May 7-9**. Turin, Italy: IEEE, 2015. p. 1–5. Available: <http://dx.doi.org/10.1109/MeMeA.2015.7145162>. Citation on page 78.

ALMUHISEN, F.; DURAND, N.; QUAFAFOU, M. Detecting behavior types of moving object trajectories. **International Journal of Data Science and Analytics**, v. 5, n. 2, p. 169–187, Mar 2018. ISSN 2364-4168. Available: <http://dx.doi.org/10.1007/s41060-017-0076-8>. Citations on pages 58 and 93.

BALOMENOS, A. D.; TSAKANIKAS, P.; ASPRIDOU, Z.; TAMPAKAKI, A. P.; KOUTSOUMANIS, K. P.; MANOLAKOS, E. S. Image analysis driven single-cell analytics for systems microbiology. **BMC Systems Biology**, v. 11, n. 1, p. 43:1–43:21, 2017. Available: <http://dx.doi.org/10.1186/s12918-017-0399-z>. Citation on page 91.

BALOMENOS, A. D.; TSAKANIKAS, P.; MANOLAKOS, E. S. Tracking single-cells in overcrowded bacterial colonies. In: **2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. Milan, Italy: IEEE, 2015. p. 6473–6476. ISSN 1094-687X. Available: <http://dx.doi.org/10.1109/EMBC.2015.7319875>. Citations on pages 92 and 103.

BEDO, M. V. N.; BLANCO, G.; OLIVEIRA, W. D.; CAZZOLATO, M. T.; COSTA, A. F.; JR., J. F. R.; TRAINA, A. J. M.; JR., C. T. Techniques for effective and efficient fire detection from social media images. In: **ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April, 2015**. Barcelona, Spain: Scitepress, 2015. p. 34–45. Available: <http://dx.doi.org/10.5220/0005341500340045>. Citations on pages 39, 42, and 124.

BEDO, M. V. N.; OLIVEIRA, W. D. de; CAZZOLATO, M. T.; COSTA, A.; BLANCO, G.; RODRIGUES-JR, J. F.; TRAINA, A.; TRAINA-JR, C. Fire detection from social media images by means of instance-based learning. In: SPRINGER (Ed.). **Lecture Notes in Business Information Processing**. Cham: Springer International Publishing, 2016. v. 241, p. 23–44. ISBN 978-3-319-29132-1. Available: <http://dx.doi.org/10.1007/978-3-319-29133-8_2>. Citations on pages 41 and 124.

BENTLEY, J. L. Multidimensional binary search trees used for associative searching. **Commun. ACM**, ACM, New York, NY, USA, v. 18, n. 9, p. 509–517, 1975. Available: <http://dx.doi.org/10.1145/361002.361007>. Citation on page 48.

BISHOP, C. M. **Pattern recognition and machine learning, 5th Edition**. Berlin, Heidelberg: Springer, 2007. (Information science and statistics). ISBN 9780387310732. Available: <https://www.springer.com/us/book/9780387310732>. Citations on pages 53, 54, and 81.

Blanco, G.; Bedo, M. V. N.; Cazzolato, M. T.; Santos, L. F. D.; Jorge, A. E. S.; Traina-Jr., C.; Azevedo-Marques, P. M.; Traina, A. J. M. A label-scaled similarity measure for content-based image retrieval. In: **2016 IEEE International Symposium on Multimedia (ISM), San Jose, CA, USA. December, 11-13.** San Jose, CA, USA: IEEE, 2016. p. 20–25. Available: <http://dx.doi.org/10.1109/ISM.2016.0014>. Citation on page 124.

BOHUSH, R.; BROUKA, N. Smoke and flame detection in video sequences based on static and dynamic features. In: **Signal Processing**. Poznan, Poland: IEEE, 2013. p. 20–25. ISSN 2326-0262. Available: <https://ieeexplore.ieee.org/document/6710590>. Citation on page 63.

BREIMAN, L. Random forests. **Machine Learning**, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Available: <http://dx.doi.org/10.1023/A:1010933404324>. Citation on page 44.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and Regression Trees**. Monterey: Wadsworth, 1984. ISBN 0-534-98053-8. Available: <https://dblp.org/rec/bib/books/wa/BreimanFOS84>. Citation on page 43.

BUENO, R.; KASTER, D. S.; TRAINA, A. J. M.; TRAINA JR., C. Time-aware similarity search: A metric-temporal representation for complex data. In: MAMOULIS, N.; SEIDL, T.; PEDERSEN, T.; TORP, K.; ASSENT, I. (Ed.). **Advances in Spatial and Temporal Databases**. Berlin, Heidelberg: Springer, 2009, (Lecture Notes in Computer Science, v. 5644). p. 302–319. ISBN 978-3-642-02981-3. Available: <http://dx.doi.org/10.1007/978-3-642-02982-0_20>. Citation on page 35.

CALDERARA, S.; PICCININI, P.; CUCCHIARA, R. Vision based smoke detection system using image energy and color information. **Mach. Vis. and App.**, Springer-Verlag, v. 22, n. 4, p. 705–719, 2011. ISSN 0932-8092. Available: <http://dx.doi.org/10.1007/s00138-010-0272-1>. Citations on pages 62 and 63.

CANDEMIR, S.; JAEGER, S.; PALANIAPPAN, K.; MUSCO, J. P.; SINGH, R. K.; XUE, Z.; KARARGYRIS, A.; ANTANI, S. K.; THOMA, G. R.; MCDONALD, C. J. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. **IEEE Trans. Med. Imaging**, v. 33, n. 2, p. 577–590, 2014. Available: <http://dx.doi.org/10.1109/TMI.2013.2290491>. Citation on page 76.

CARÉLO, C. C.; POLA, I. R.; CIFERRI, R. R.; TRAINA, A. J.; TRAINA-JR., C.; CIFERRI, C. D. A. The onion-tree: Quick indexing of complex data in the main memory. In: **Proceedings of the 13th East European Conference on Advances in Databases and Information Systems**. Berlin, Heidelberg: Springer-Verlag, 2009. (ADBIS '09), p. 235–252. ISBN 978-3-642-03972-0. Available: <http://dx.doi.org/10.1007/978-3-642-03973-7_18>. Citation on page 47.

CAZZOLATO, M. T.; AVALHAIS, L. P. S.; CHINO, D. Y. T.; RAMOS, J. S.; SOUZA, J. A.; RODRIGUES-JR, J. F.; TRAINA, A. J. M. Fismo: A compilation of datasets from emergency situations for fire and smoke analysis. In: **SBBD2017 - SBBD Proceedings of Satellite Events of the 32nd Brazilian Symposium on Databases - DSW (Dataset Showcase Workshop)**. SBC, 2017. p. 213–223. ISBN 978-85-7669-399-4. Available: <http://sbbd.org.br/2017/wp-content/uploads/sites/3/2017/10/proceedings-satellite-events-sbbd-2017.pdf>. Citations on pages 32, 61, 67, 68, 73, and 122.

CAZZOLATO, M. T.; BEDO, M. V. N.; COSTA, A. F.; SOUZA, J. A. de; JR., C. T.; JR., J. F. R.; TRAINA, A. J. M. Unveiling smoke in social images with the smokeblock approach. In: **Proceedings of the 31st Annual ACM Symposium on Applied Computing, April 4-8, 2016**. Pisa, Italy: ACM, 2016. p. 49–54. Available: <http://dx.doi.org/10.1145/2851613.2851634>. Citations on pages 32, 61, 65, 69, 70, 71, 72, 73, and 122.

CAZZOLATO, M. T.; SCABORA, L. C.; NESSO-JR, M. R.; MILANO-OLIVEIRA, L. F.; COSTA, A. F.; KASTER, D. d. S.; KOENIGKAM-SANTOS, M.; AZEVEDO-MARQUES, P. M.; TRAINA-JR, C.; TRAINA, A. J. M. dp-breath: Heat maps and probabilistic classification assisting the analysis of abnormal lung regions. **Computer Methods and Programs in Biomedicine**, v. 173, p. 27–34, May 2019. ISSN 0169-2607. Available: <http://dx.doi.org/10.1016/j.cmpb.2019.01.014>. Citations on pages 32, 75, 76, 77, 80, 82, 83, 84, 86, 87, 88, 89, 90, and 121.

CAZZOLATO, M. T.; SCABORA, L. d. C.; COSTA, A. F.; NESSO-JR, M. R.; MILANO-OLIVEIRA, L. F.; KASTER, D. d. S.; TRAINA-JR, C.; TRAINA, A. J. M. Breath: Heat maps assisting the detection of abnormal lung regions in ct scans. In: **2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)**. Thessaloniki, Greece: IEEE, 2017. p. 248–253. Available: <http://dx.doi.org/10.1109/CBMS.2017.82>. Citations on pages 32, 75, 76, 77, 90, and 122.

CAZZOLATO, M. T.; TRAINA, A. J. M.; BÖHM, K. Efficient and reliable estimation of cell positions. In: **Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, October 22-26, 2018**. Torino, Italy: ACM, 2018. p. 1043–1052. Available: <http://dx.doi.org/10.1145/3269206.3271734>. Citations on pages 33, 91, 94, 104, 106, 107, 108, 109, 110, 111, 112, 113, 114, 117, and 121.

CELIK, T.; DEMIREL, H. Fire detection in video sequences using a generic color model. **Fire Safety J.**, v. 44, n. 2, p. 147–158, 2009. ISSN 0379-7112. Available: <http://dx.doi.org/10.1016/j.firesaf.2008.05.005>. Citations on pages 63 and 69.

CELIK, T.; ÖZKARAMANH, H.; DEMIREL, H. Fire and smoke detection without sensors: Image processing based approach. In: **EUSIPCO**. Poznan, Poland: IEEE, 2007. p. 1794–1798. ISSN 1051-8215. Available: <https://ieeexplore.ieee.org/document/7099116>. Citations on pages 61, 62, and 63.

CHAKRABORTY, A.; ROY-CHOWDHURY, A. K. Context aware spatio-temporal cell tracking in densely packed multilayer tissues. **Medical Image Analysis**, v. 19, n. 1, p. 149 – 163, 2015. ISSN 1361-8415. Available: <https://doi.org/10.1016/j.media.2014.09.008>. Citation on page 95.

CHAUDHRY, R.; RAVICHANDRAN, A.; HAGER, G.; VIDAL, R. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: **Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on**. Miami, FL, USA: IEEE, 2009. p. 1932–1939. ISSN 1063-6919. Available: <http://dx.doi.org/10.1109/CVPR.2009.5206821>. Citation on page 39.

CHAUDHURI, D. Global contour and region based shape analysis and similarity measures. **Defense Sciente Journal**, v. 63, n. 1, p. 74–88, 2013. Available: <http://dx.doi.org/10.14429/dsj.63.3767>. Citation on page 38.

CHEN, T.; YIN, Y.-H.; HUANG, S.; YE, Y. The smoke detection for early fire-alarming system base on video processing. In: **IIH-MSP**. Pasadena, CA, USA: IEEE, 2006. p. 427–430. Available: <http://dx.doi.org/10.1109/IIH-MSP.2006.265033>. Citations on pages 61, 63, and 69.

CHINO, D.; AVALHAIS, L.; RODRIGUES, J.; TRAINA, A. Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. In: **Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on**. Salvador, Brazil: IEEE, 2015. p. 95–102. Available: <http://dx.doi.org/10.1109/SIBGRAPI.2015.19>. Citations on pages 29 and 35.

CHINO, D. Y. T.; SCABORA, L. de C.; CAZZOLATO, M. T.; JORGE, A. E. S.; TRAINA-JR., C.; TRAINA, A. J. M. ICARUS: retrieving skin ulcer images through bag-of-signatures. In: **31st IEEE International Symposium on Computer-Based Medical Systems, CBMS 2018, June 18-21, 2018**. Karlstad, Sweden: IEEE, 2018. p. 82–87. Available: <http://dx.doi.org/10.1109/CBMS.2018.00022>. Citations on pages 39, 42, and 123.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. M-tree: An efficient access method for similarity search in metric spaces. In: **Proceedings of the 23rd International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. (VLDB '97), p. 426–435. ISBN 1-55860-470-7. Available: <http://www.vldb.org/conf/1997/P426.PDF>. Citation on page 47.

COSTA, A. F.; HUMPIRE-MAMANI, G.; TRAINA, A. J. M. An efficient algorithm for fractal analysis of textures. In: **25th SIBGRAPI**. Ouro Preto, Brazil: IEEE Computer Society, 2012. (SIBGRAPI '12), p. 39–46. ISBN 978-0-7695-4829-6. Available: <http://dx.doi.org/10.1109/SIBGRAPI.2012.15>. Citation on page 86.

COSTA, A. F.; YAMAGUCHI, Y.; TRAINA, A. J. M.; JR., C. T.; FALOUTSOS, C. Modeling temporal activity to detect anomalous behavior in social media. **TKDD**, ACM, v. 11, n. 4, p. 49:1–49:23, 2017. Available: <http://dx.doi.org/10.1145/3064884>. Citation on page 29.

ELFWING, A.; LEMARC, Y.; BARANYI, J.; BALLAGI, A. Observing growth and division of large numbers of individual bacteria by image analysis. **Applied and Environmental Microbiology**, v. 70, n. 2, p. 675–678, 2004. Available: <http://dx.doi.org/10.1128/AEM.70.2.675-678.2004>. Citation on page 91.

ELICKER, B.; PEREIRA, C. A. d. C.; WEBB, R.; LESLIE, K. O. High-resolution computed tomography patterns of diffuse interstitial lung disease with clinical and pathological correlation. **The Brazilian Journal of Pulmonology**, v. 34, p. 715 – 744, 09 2008. ISSN 1806-3713. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18982210>. Citations on pages 76 and 84.

FAHAD, A.; ALSHATRI, N.; TARI, Z.; ALAMRI, A.; KHALIL, I.; ZOMAYA, A.; FOUFOU, S.; BOURAS, A. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. **Emerging Topics in Computing, IEEE Transactions on**, v. 2, n. 3, p. 267–279, Sept 2014. ISSN 2168-6750. Available: <http://dx.doi.org/10.1109/TETC.2014.2330519>. Citation on page 45.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Advances in knowledge discovery and data mining. In: FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. (Ed.). Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996. chap. From Data Mining to Knowledge Discovery: An Overview, p. 1–34. ISBN 0-262-56097-6. Available: <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1230/1131>. Citation on page 42.

FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient graph-based image segmentation. **International Journal of Computer Vision**, v. 59, n. 2, p. 167–181, 2004. Available: <http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77>. Citation on page 52.

FENG, J.; LIU, X.; DONG, Y.; LIANG, L.; PU, J. Structural difference histogram representation for texture image classification. **IET Image Processing**, v. 11, n. 2, p. 118–125, 2017. ISSN 1751-9659. Available: <http://dx.doi.org/10.1049/iet-ipr.2016.0495>. Citation on page 39.

FERREIRA, J. R.; KOENIGKAM-SANTOS, M.; CIPRIANO, F. E. G.; FABRO, A. T.; MARQUES, P. M. de A. Radiomics-based features for pattern recognition of lung cancer histopathology and metastases. **Computer Methods and Programs in Biomedicine**, v. 159, p. 23–30, 2018. Available: <http://dx.doi.org/10.1016/j.cmpb.2018.02.015>. Citation on page 29.

FILHO, R. F. S.; TRAINA, A. J. M.; JR., C. T.; FALOUTSOS, C. Similarity search without tears: The OMNI family of all-purpose access methods. In: **Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001**. Heidelberg, Germany: Elsevier, 2001. p. 623–630. Available: <http://dx.doi.org/10.1109/ICDE.2001.914877>. Citation on page 48.

FRALEY, C.; RAFTERY, E. A. Mclust: Software for model-based cluster analysis. **Journal of Classification**, v. 16, n. 2, p. 297–306, 1999. ISSN 1432-1343. Available: <http://dx.doi.org/10.1007/s003579900058>. Citation on page 46.

GONÇALVES, R. R. do V.; JR., J. Z.; ROMANI, L. A. S.; AMARAL, B. F. do; SOUSA, E. P. M. de. Agricultural monitoring using clustering techniques on satellite image time series of low spatial resolution. In: **9th International Workshop on the Analysis of Multitemporal Remote Sensing Images, MultiTemp 2017, Brugge, Belgium, June 27-29, 2017**. Brugge, Belgium: IEEE, 2017. p. 1–4. Available: <http://dx.doi.org/10.1109/Multi-Temp.2017.8035234>. Citation on page 29.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing, 3rd Edition**. Upper Saddle River, NJ, USA: Pearson Education, 2008. ISBN 9780135052679. Available: <https://dl.acm.org/citation.cfm?id=1076432>. Citations on pages 37 and 50.

GUHA, S.; RASTOGI, R.; SHIM, K. Cure: An efficient clustering algorithm for large databases. In: **Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 1998. (SIGMOD '98), p. 73–84. ISBN 0-89791-995-5. Available: <http://doi.acm.org/10.1145/276304.276312>. Citation on page 45.

GÜTING, R. H.; SCHNEIDER, M. **Moving Objects Databases**. Morgan Kaufmann, 2005. ISBN 0-12-088799-1. Available: <https://doi.org/10.1016/B978-0-12-088799-6.X5000-2>. Citation on page 58.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790, 9780123814791. Available: <https://doi.org/10.1016/C2009-0-61819-5>. Citations on pages 40, 42, 43, 44, 45, 47, 49, 53, 54, 56, and 57.

HANSELL, D. M.; BANKIER, A. A.; MACMAHON, H.; MCLOUD, T. C.; MüLLER, N. L.; REMY, J. Fleischner society: Glossary of terms for thoracic imaging. **Radiology**, v. 246, n. 3, p. 697–722, 2008. Available: <http://dx.doi.org/10.1148/radiol.2462070712>. Citation on page 76.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. **Systems, Man and Cybernetics, IEEE Transactions on**, SMC-3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472. Available: <http://dx.doi.org/10.1109/TSMC.1973.4309314>. Citations on pages 38 and 39.

HE, T.; MAO, H.; GUO, J.; YI, Z. Cell tracking using deep neural networks with multi-task learning. **Image and Vision Computing**, Elsevier, v. 60, p. 142–153, 2017. ISSN 0262-8856. Available: <http://dx.doi.org/10.1016/j.imavis.2016.11.010>. Citations on pages 29 and 95.

HILSENBECK, O.; SCHWARZFISCHER, M.; SKYLAKI, S.; SCHAUBERGER, B.; HOPPE, P. S.; LOEFFLER, D.; KOKKALIARIS, K. D.; HASTREITER, S.; SKYLAKI, E.; FILIPCZYK, A.; STRASSER, M.; BUGGENTHIN, F.; FEIGELMAN, J. S.; KRUMSIEK, J.; BERG, A. J. J. van den; ENDELE, M.; ETZRODT, M.; MARR, C.; THEIS, F. J.; SCHROEDER, T. Software tools for single-cell tracking and quantification of cellular and molecular properties. **Nature Biotechnology**, v. 34, p. 703–706, Jul 2016. Available: <https://doi.org/10.1038/nbt.3626>. Citation on page 96.

HO, T. K. Random decision forests. In: **Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on**. Montreal, Quebec, Canada: IEEE, 1995. v. 1, p. 278–282. Available: <http://dx.doi.org/10.1109/ICDAR.1995.598994>. Citation on page 44.

HU, Z.; ZOU, Q.; LI, Q. Watershed superpixel. In: **2015 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2015. p. 349–353. Citation on page 53.

HUA, P.; SONG, Q.; SONKA, M.; HOFFMAN, E. A.; REINHARDT, J. M. Segmentation of pathological and diseased lung tissue in CT images using a graph-search algorithm. In: **8th ISBI**. Chicago, IL, USA: IEEE, 2011. p. 2072–2075. ISSN 1945-7928. Available: <http://dx.doi.org/10.1109/ISBI.2011.5872820>. Citation on page 78.

HUANG, S.; CHENG, F.; CHIU, Y. Efficient contrast enhancement using adaptive gamma correction with weighting distribution. **Trans. on Image Process.**, v. 22, n. 3, p. 1032–1041, 2013. Available: <http://dx.doi.org/10.1109/TIP.2012.2226047>. Citation on page 61.

HUANG, Y.; ZHAO, Z.; QI, C.; NIE, Z.; LIU, Q. H. Fast point-based kd-tree construction method for hybrid high frequency method in electromagnetic scattering. **IEEE Access**, v. 6, p. 38348–38355, 2018. Available: <http://dx.doi.org/10.1109/ACCESS.2018.2853659>. Citations on pages 37 and 49.

HYUN, J. I.; KIM, H. K.; OH, W. G. Study on performance of mpeg-7 visual descriptors for deformable object retrieval. In: **Frontiers of Computer Vision (FCV), 2015 21st Korea-Japan Joint Workshop on**. Mokpo, South Korea: IEEE, 2015. p. 1–5. Available: <http://dx.doi.org/10.1109/FCV.2015.7103701>. Citation on page 37.

IMBIO. **Lung Texture Analysis (LTA) – CT Imaging for ILD's and Other Fibrotic Conditions**. 2019. Accessed on 02/11/2019. Available: <https://imbio.com/lung-texture>. Citation on page 79.

JIUQING, W.; XU, C.; XIANHANG, Z. Cell tracking via structured prediction and learning. **Machine Vision and Applications**, Springer, Berlin, Heidelberg, v. 28, n. 8, p. 859–874, Nov 2017. ISSN 1432-1769. Available: <http://dx.doi.org/10.1007/s00138-017-0872-0>. Citation on page 95.

KALE, M.; MUKHOPADHYAY, S.; DASH, J. K.; GARG, N. K. M. Differentiation of several interstitial lung disease patterns in hrct images using support vector machine: role of databases on performance. In: . San Diego, CA, US: SPIE, 2016. v. 9785, p. 9785 – 9785 – 6. Available: <http://dx.doi.org/10.1117/12.2216743>. Citations on pages 42 and 78.

KARARGYRIS, A.; SIEGELMAN, J.; TZORTZIS, D.; JAEGER, S.; CANDEMIR, S.; XUE, Z.; SANTOSH, K. C.; VAJDA, S.; ANTANI, S.; FOLIO, L.; THOMA, G. R. Combination of texture and shape features to detect pulmonary abnormalities in digital chest x-rays. **International Journal of Computer Assisted Radiology and Surgery**, v. 11, n. 1, p. 99–106, Jan 2016. ISSN 1861-6429. Available: <http://dx.doi.org/10.1007/s11548-015-1242-x>. Citations on pages 42 and 76.

KASUTANI, E.; YAMADA, A. The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In: **Image Processing, 2001. Proceedings. 2001 International Conference on**. Thessaloniki, Greece: IEEE, 2001. v. 1, p. 674–677 vol.1. Available: <http://dx.doi.org/10.1109/ICIP.2001.959135>. Citation on page 38.

KAUFMAN, L.; ROUSSEEUW, P. J. **Finding Groups in Data: An Introduction to Cluster Analysis**. John Wiley, 2005. ISBN 9780470316801. Available: <http://dx.doi.org/10.1002/9780470316801>. Citation on page 45.

KEOGH, E.; RATANAMAHATANA, A. Everything you know about dynamic time warping is wrong. **3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA**, 2004. Available: <http://www.cs.ucr.edu/~eamonn/DTW_myths.pdf>. Citation on page 41.

KLEIN, J.; LEUPOLD, S.; BIEGLER, I.; BIEDENDIECK, R.; MüNCH, R.; JAHN, D. Tlm-tracker: software for cell segmentation, tracking and lineage analysis in time-lapse microscopy

movies. **Bioinformatics**, v. 28, n. 17, p. 2276–2277, 2012. Available: <http://dx.doi.org/10.1093/bioinformatics/bts424>. Citation on page 96.

KOENIGKAM-SANTOS, M.; WEINHEIMER, O. Diffuse Lung Diseases (Emphysema, Airway and Interstitial Lung Diseases). In: **Medical Image Analysis and Informatics: Computer-Aided Diagnosis and Therapy**. 1. ed. Boca Raton, FL, US: Taylor & Francis, 2017. p. 203–218. ISBN 9781498753197. Available: <https://www.crcpress.com/Medical-Image-Analysis-and-Informatics-Computer-Aided-Diagnosis-and-Therapy/Azevedo-Marques-Mencattini-Salmeri-Rangayyan/p/book/9781498753197>. Citation on page 75.

LAVERDE, N. A.; CAZZOLATO, M. T.; TRAINA, A. J. M.; TRAINA-JR., C. Semantic similarity group by operators for metric data. In: **Similarity Search and Applications - 10th International Conference, SISAP 2017, October 4-6, 2017, Proceedings**. Munich, Germany: Springer, Cham, 2017. p. 247–261. Available: <http://dx.doi.org/10.1007/978-3-319-68474-1_17>. Citation on page 124.

LE, V.; YANG, D.; ZHU, Y.; ZHENG, B.; BAI, C.; SHI, H.; HU, J.; ZHAI, C.; LU, S. Quantitative CT analysis of pulmonary nodules for lung adenocarcinoma risk classification based on an exponential weighted grey scale angular density distribution feature. **Computer Methods and Programs in Biomedicine**, Elsevier, v. 160, p. 141–151, 2018. Available: <http://dx.doi.org/10.1016/j.cmpb.2018.04.001>. Citation on page 75.

LEE, K.-L.; CHEN, L.-H. An efficient computation method for the texture browsing descriptor of mpeg-7. **Image Vision Comput.**, Butterworth-Heinemann, Newton, MA, USA, v. 23, n. 5, p. 479–489, May 2005. ISSN 0262-8856. Available: <http://dx.doi.org/10.1016/j.imavis.2004.12.002>. Citation on page 38.

LI, L. Constraint databases and moving objects. In: **Encyclopedia of GIS.** Boston, MA: Springer, 2017. p. 329–334. Available: <http://dx.doi.org/10.1007/978-0-387-35973-1_189>. Citation on page 58.

LI, Y.; ROSE, F.; PIETRO, F. di; MORIN, X.; GENOVESIO, A. Detection and tracking of overlapping cell nuclei for large scale mitosis analyses. **BMC Bioinformatics**, BMC Bioinformatics, v. 17, p. 183, 2016. Available: <http://dx.doi.org/10.1186/s12859-016-1030-9>. Citation on page 95.

LI, Z.; ZHANG, X.; MÜLLER, H.; ZHANG, S. Large-scale retrieval for medical image analytics: A comprehensive review. **Medical Image Analysis**, v. 43, p. 66 – 84, 2018. ISSN 1361-8415. Citation on page 75.

LIN, W.-S.; FANG, C.-H. Lossless parameterisation of image contour for shape recognition. **Computer Vision, IET**, v. 3, n. 1, p. 36–46, March 2009. ISSN 1751-9632. Available: <http://dx.doi.org/10.1049/iet-cvi:20070042>. Citation on page 38.

MAIMON, O.; ROKACH, L. **Data Mining and Knowledge Discovery Handbook**. Springer US, 2010. (Series in Solid-State Sciences). ISBN 9780387098234. Available: <http://dx.doi.org/10.1007/978-0-387-09823-4>. Citation on page 42.

MALONE, J.; ROSSITER, J. M.; PRABHU, S.; GODDARD, P. Identification of disease in CT of the lung using texture-based image analysis. In: **38th ASILOMAR**. Pacific Grove, CA, USA: IEEE, 2004. v. 2, p. 1620–1624. Available: <http://dx.doi.org/10.1109/ACSSC.2004.1399431>. Citation on page 77.

MELLO, R. F. de; PONTI, M. A. **Machine Learning - A Practical Approach on the Statistical Learning Theory**. Springer, 2018. ISBN 978-3-319-94988-8. Available: <http://dx.doi.org/10.1007/978-3-319-94989-5>. Citation on page 42.

MOON, T. K. The expectation-maximization algorithm. **IEEE Signal Processing Magazine**, v. 13, n. 6, p. 47–60, Nov 1996. ISSN 1053-5888. Available: <http://dx.doi.org/10.1109/79.543975>. Citation on page 84.

MÜLLER, M. **Information Retrieval for Music and Motion**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN 3540740473. Available: <http://dx.doi.org/10.1007/978-3-540-74048-3>. Citation on page 40.

MULTIMEDIA, I. Mpeg-7: The generic multimedia content description standard, part 1. **IEEE MultiMedia**, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 9, n. 2, p. 78–87, apr 2002. ISSN 1070-986X. Available: <http://dx.doi.org/10.1109/93.998074>. Citation on page 37.

MUNADI, K.; ARNIA, F.; SYARYADHI, M.; KIYA, H. A content-based image retrieval system for visually protected image databases. In: **Multimedia and Broadcasting (APMediaCast), 2015 Asia Pacific Conference on**. Kuta, Indonesia: IEEE, 2015. p. 1–6. Available: <http://dx.doi.org/10.1109/APMediaCast.2015.7210285>. Citation on page 37.

NAKAGOMI, K.; SHIMIZU, A.; KOBATAKE, H.; YAKAMI, M.; FUJIMOTO, K.; TOGASHI, K. Multi-shape graph cuts with neighbor prior constraints and its application to lung segmentation from a chest ct volume. **Medical Image Analysis**, Elsevier, v. 17, n. 1, p. 62 – 77, 2013. ISSN 1361-8415. Available: <http://dx.doi.org/10.1016/j.media.2012.08.002>. Citation on page 76.

NESSO-JR, M. R.; CAZZOLATO, M. T.; SCABORA, L. C.; OLIVEIRA, P. H.; SPADON, G.; SOUZA, J. A. de; OLIVEIRA, W. D.; CHINO, D. Y. T.; JR., J. F. R.; TRAINA, A. J. M.; TRAINA-JR., C. RAFIKI: retrieval-based application for imaging and knowledge investigation. In: **31st IEEE International Symposium on Computer-Based Medical Systems, CBMS 2018, Karlstad, Sweden, June 18-21, 2018**. Karlstad, Sweden: IEEE, 2018. p. 71–76. Available: <http://dx.doi.org/10.1109/CBMS.2018.00020>. Citations on pages 39 and 123.

NEUBERT, P.; PRÖTZEL, P. Benchmarking superpixel descriptors. In: **2015 23rd European Signal Processing Conference (EUSIPCO)**. Nice, France: IEEE, 2015. p. 614–618. ISSN 2076-1465. Available: <http://dx.doi.org/10.1109/EUSIPCO.2015.7362456>. Citation on page 51.

NEUMANN, B. Optical flow. **SIGGRAPH Comput. Graph.**, ACM, New York, NY, USA, v. 18, n. 1, p. 17–19, Jan. 1984. ISSN 0097-8930. Available: <http://doi.acm.org/10.1145/988525.988528>. Citation on page 39.

NG, R.; HAN, J. Clarans: a method for clustering objects for spatial data mining. **Knowledge and Data Engineering, IEEE Transactions on**, v. 14, n. 5, p. 1003–1016, Sep 2002. ISSN 1041-4347. Available: <http://dx.doi.org/10.1109/TKDE.2002.1033770>. Citation on page 45.

OJALA, T.; AITTOLA, M.; MATINMIKKO, E. Empirical evaluation of mpeg-7 xm color descriptors in content-based retrieval of semantic image categories. In: **Pattern Recognition, 2002. Proceedings. 16th International Conference on**. Quebec City, Quebec, Canada: IEEE, 2002. v. 2, p. 1021–1024 vol.2. ISSN 1051-4651. Available: <http://dx.doi.org/10.1109/ICPR.2002.1048479>. Citations on pages 37 and 39.

OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 24, n. 7, p. 971–987, Jul 2002. ISSN 0162-8828. Available: <http://dx.doi.org/10.1109/TPAMI.2002.1017623>. Citations on pages 38, 39, and 78.

OLIVEIRA, P. H.; FRAIDEINBERZE, A. C.; LAVERDE, N. A.; GUALDRON, H.; GONZAGA, A. S.; FERREIRA, L. D.; OLIVEIRA, W. D.; JR., J. F. R.; CORDEIRO, R. L. F.; JR., C. T.; TRAINA, A. J. M.; SOUSA, E. P. M. de. On the support of a similarity-enabled relational database management system in civilian crisis situations. In: **ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems, Volume 1, April 25-28, 2016**. Rome, Italy: Scitepress, 2016. p. 119–126. Available: <http://dx.doi.org/10.5220/0005816701190126>. Citation on page 29.

OLIVEIRA, P. H.; SCABORA, L. C.; CAZZOLATO, M. T.; OLIVEIRA, W. D.; TRAINA, A. J. M.; TRAINA-JR, C. Efficiently indexing multiple repositories of medical image databases. In: **2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)**. Thessaloniki, Greece: IEEE, 2017. p. 286–291. Available: <http://dx.doi.org/10.1109/CBMS.2017.81>. Citation on page 123.

OLIVEIRA, P. H.; SCABORA, L. C.; CAZZOLATO, M. T.; BEDO, M. V. N.; TRAINA, A. J. M.; TRAINA-JR., C. MAMMOSET: An Enhanced Dataset of Mammograms. In: **SBBD2017 - SBBD Proceedings of Satellite Events of the 32nd Brazilian Symposium on Databases - DSW (Dataset Showcase Workshop)**. SBC, 2017. p. 256–266. ISBN 978-85-7669-399-4. Available: <sbbd.org.br/2017/wp-content/uploads/sites/3/2017/10/proceedings-satellite-events-sbbd-2017.pdf>. Citation on page 124.

OLIVEIRA, P. H.; SCABORA, L. C.; CAZZOLATO, M. T.; OLIVEIRA, W. D.; PAIXãO, R. S.; TRAINA, A. J. M.; TRAINA-JR, C. Employing domain indexes to efficiently query medical data from multiple repositories. **IEEE Journal of Biomedical and Health Informatics**, IEEE, p. 1–9, 2018. ISSN 2168-2194. Available: <http://dx.doi.org/10.1109/JBHI.2018.2881381>. Citation on page 123.

OMOHUNDRO, S. M. **Five Balltree Construction Algorithms**. International Computer Science Institute, 1989. (Technical report). Available: <http://www.icsi.berkeley.edu/ftp/global/pub/techreports/1989/tr-89-063.pdf>. Citation on page 49.

ØRNAGER, S.; LUND, H. **Images in Social Media: Categorization and Organization of Images and Their Collections**. Morgan & Claypool Publishers, 2018. (Synthesis Lectures on Information Concepts, Retrieval, and Services). Available: <http://dx.doi.org/10.2200/S00821ED1V01Y201712ICR062>. Citation on page 29.

PARKER, J. R. **Algorithms for Image Processing and Computer Vision**. 2nd. ed. Wiley Publishing, 2010. ISBN 0470643854, 9780470643853. Available: <https://dl.acm.org/citation.cfm?id=1965423>. Citations on pages 37 and 50.

PHUONG, G. N. T.; LUONG, H. H.; PHAM, T. H.; HUYNH, H. X. A parallel algorithm for determining the communication radius of an automatic light trap based on balltree structure. In: **2016 Eighth International Conference on Knowledge and Systems Engineering, KSE 2016, Hanoi, Vietnam, October 6-8, 2016**. Hanoi, Vietnam: IEEE, 2016. p. 139–143. Available: <http://dx.doi.org/10.1109/KSE.2016.7758043>. Citation on page 49.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Available: <https://dl.acm.org/citation.cfm?id=152181>. Citation on page 43.

RAGHUNATH, S.; RAJAGOPALAN, S.; KARWOSKI, R.; MALDONADO, F.; BARTHOLMAI, B.; PEIKERT, T. Computer-aided lung informatics for pathology evaluation and rating (caliper) analysis of chest ct to detect histologically proven emphysema. In: **D28. Novel Imaging Methods to Study Lung Disease**. AST Journals, 2016. p. A6613–A6613. Available: <https://www.atsjournals.org/doi/abs/10.1164/ajrccm-conference.2016.193.1_MeetingAbstracts.A6613>. Citation on page 79.

RAMOS, J. S.; CAZZOLATO, M. T.; FAIçAL, B. S.; NOGUEIRA-BARBOSA, M. H.; TRAINA-JR., C.; TRAINA, A. J. M. 3dbgrowth: Volumetric vertebrae segmentation and reconstruction in magnetic resonance imaging. In: **2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)**. Cordoba, Spain: IEEE, 2019. p. 435–440. ISSN 2372-9198. Available: <http://dx.doi.org/10.1109/CBMS.2019.00091>. Citation on page 123.

RENCHER, A. C.; CHRISTENSEN, W. F. **Methods of Multivariate Analysis**. 3rd. ed. Wiley Publishing, 2012. ISBN 9780470178966, 9781118391686. Available: <http://dx.doi.org/10.1002/9781118391686>. Citation on page 54.

SALTENIS, S.; JENSEN, C. S.; LEUTENEGGER, S. T.; LÓPEZ, M. A. Indexing the positions of continuously moving objects. In: **Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.** New York, NY, USA: ACM, 2000. p. 331–342. Available: <http://dx.doi.org/10.1145/342009.335427>. Citations on pages 58 and 93.

SAMET, H. **Foundations of Multidimensional and Metric Data Structures**. Elsevier, Morgan Kaufmann, 2006. (Morgan Kaufmann). ISSN 10:0-12-369-446-9. ISBN 9780123694461. Available: <https://www.elsevier.com/books/foundations-of-multidimensional-and-metric-data-structures/samet/978-0-12-369446-1>. Citation on page 36.

SAMI, M.; EL-BENDARY, N.; HASSANIEN, A. Automatic image annotation via incorporating naive bayes with particle swarm optimization. In: **Information and Communication Technologies (WICT), 2012 World Congress on**. Trivandrum, India: IEEE, 2012. p. 790–794. Available: <http://dx.doi.org/10.1109/WICT.2012.6409182>. Citation on page 45.

SANDER, J.; ESTER, M.; KRIEGEL, H.-P.; XU, X. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. **Data Min. Knowl. Discov.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, n. 2, p. 169–194, jun 1998. ISSN 1384-5810. Available: <http://dx.doi.org/10.1023/A:1009745219419>. Citation on page 46.

SANTOS, L. F. D.; BLANCO, G.; OLIVEIRA, D. de; TRAINA, A. J. M.; JR., C. T.; BEDO, M. V. N. Exploring diversified similarity with kundaha. In: **Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, October 22-26, 2018**. Torino, Italy: ACM, 2018. p. 1903–1906. Available: <http://dx.doi.org/10.1145/3269206.3269220>. Citation on page 29.

SANTOSH, K. C.; ANTANI, S. Automated chest x-ray screening: Can lung region symmetry help detect pulmonary abnormalities? **IEEE Transactions on Medical Imaging**, IEEE, v. 37, n. 5, p. 1168–1177, May 2018. ISSN 0278-0062. Available: <http://dx.doi.org/10.1109/TMI.2017.2775636>. Citation on page 76.

SHAPIRO, L.; STOCKMAN, G. **Computer Vision**. Prentice Hall, 2001. ISBN 9780130307965. Available: <https://books.google.com.br/books?id=FftDAQAAIAAJ>. Citation on page 37.

SHARMA, J.; GRANMO, O.-C.; GOODWIN, M.; FIDJE, J. T. Deep convolutional neural networks for fire detection in images. In: BORACCHI, G.; ILIADIS, L.; JAYNE, C.; LIKAS, A. (Ed.). **Engineering Applications of Neural Networks**. Cham: Springer International Publishing, 2017. p. 183–193. Available: <https://doi.org/10.1007/978-3-319-65172-9_16>. Citation on page 120.

SORENSEN, L.; SHAKER, S. B.; BRUIJNE, M. de. Quantitative analysis of pulmonary emphysema using local binary patterns. **IEEE Transactions on Medical Imaging**, IEEE, v. 29, n. 2, p. 559–569, Feb 2010. ISSN 0278-0062. Available: <http://dx.doi.org/10.1109/TMI.2009.2038575>. Citation on page 78.

SOUZA, J. A. de; CAZZOLATO, M. T.; TRAINA, A. J. M. Clusmam: fast and effective unsupervised clustering of large complex datasets using metric access methods. In: **Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016**. Pisa, Italy: ACM, 2016. p. 986–991. Available: <http://dx.doi.org/10.1145/2851613.2851661>. Citations on pages 48 and 124.

SOUZA, J. A. de; RAZENTE, H. L.; BARIONI, M. C. N. Optimizing metric access methods for querying and mining complex data types. **J. Braz. Comp. Soc.**, Springer, v. 20, n. 1, p. 17:1–17:14, 2014. Available: <http://dx.doi.org/10.1186/s13173-014-0017-5>. Citation on page 48.

SPINA, T. V.; STEGMAIER, J.; FALCÃO, A. X.; MEYEROWITZ, E.; CUNHA, A. SEGMENT3D: A web-based application for collaborative segmentation of 3d images used in the shoot apical meristem. In: **15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, April 4-7, 2018**. Washington, DC, USA: IEEE, 2018. p. 391–395. Available: <http://dx.doi.org/10.1109/ISBI.2018.8363600>. Citation on page 92.

STEGMAIER, J.; OTTE, J. C.; KOBITSKI, A.; BARTSCHAT, A.; GARCIA, A.; NIENHAUS, G. U.; STRäHLE, U.; MIKUT, R. Fast segmentation of stained nuclei in terabyte-scale, time resolved 3d microscopy image stacks. **PLOS ONE**, Public Library of Science, v. 9, n. 2, p. 1–11, 02 2014. Available: <http://dx.doi.org/10.1371/journal.pone.0090036>. Citations on pages 29, 95, 96, and 104.

STEGMAIER, J.; SPINA, T. V.; FALCÃO, A. X.; BARTSCHAT, A.; MIKUT, R.; MEYEROWITZ, E.; CUNHA, A. Cell segmentation in 3d confocal images using supervoxel merge-forests with cnn-based hypothesis selection. In: **15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, April 4-7, 2018**. Washington, DC, USA: IEEE, 2018. p. 382–386. Available: <http://dx.doi.org/10.1109/ISBI.2018.8363598>. Citation on page 92.

TAO, Y.; FALOUTSOS, C.; PAPADIAS, D.; LIU, B. Prediction and indexing of moving objects with unknown motion patterns. In: **Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 2004. p. 611–622. ISBN 1-58113-859-8. Available: <http://dx.doi.org/10.1145/1007568.1007637>. Citations on pages 58 and 93.

TIAN, H.; LI, W.; WANG, L.; OGUNBONA, P. Smoke detection in video: An image separation approach. **Int. J. of Comp. Vis.**, Springer US, v. 106, n. 2, p. 192–209, 2014. ISSN 0920-5691. Available: <http://dx.doi.org/10.1007/s11263-013-0656-6>. Citation on page 63.

TRAINA, A. J. M.; TRAINA-JR., C.; BALAN, A. G. R.; RIBEIRO, M. X.; BUGATTI, P. H.; WATANABE, C. Y. V.; AZEVEDO-MARQUES, P. M. Feature extraction and selection for decision making. In: DESERNO, T. M. (Ed.). **Biomedical Image Processing**. Heidelberg: Springer, 2011, (Biological and Medical Physics, Biomedical Engineering). p. 197–223. Available: <https://doi.org/10.1007/978-3-642-15816-2_8>. Citation on page 47.

TRAINA-JR., C.; FILHO, R. F. S.; TRAINA, A. J. M.; VIEIRA, M. R.; FALOUTSOS, C. The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. **VLDB J.**, v. 16, n. 4, p. 483–505, 2007. Available: <http://dx.doi.org/10.1007/s00778-005-0178-0>. Citation on page 48.

TRAINA-JR., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS, C. Slim-trees: High performance metric trees minimizing overlap between nodes. In: ZANIOLO, C.; LOCKEMANN, P.; SCHOLL, M.; GRUST, T. (Ed.). **Advances in Database Technology — EDBT 2000**. Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1777). p. 51–65. ISBN 978-3-540-67227-2. Available: <http://dx.doi.org/10.1007/3-540-46439-5_4>. Citations on pages 47 and 48.

TRAINA-JR., C.; TRAINA, A. J. M.; WU, L.; FALOUTSOS, C. Fast feature selection using fractal dimension. **JIDM**, v. 1, n. 1, p. 3–16, 2010. Available: <http://seer.lcc.ufmg.br/index.php/jidm/article/view/4>. Citation on page 50.

ULMAN, V. *et al.* An objective comparison of cell-tracking algorithms. **Nature Methods**, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved, v. 14, p. 1141–1152, 2017. Available: <http://dx.doi.org/10.1038/nmeth.4473>. Citations on pages 29 and 104.

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8. Available: <http://dx.doi.org/10.1007/978-1-4757-3264-1>. Citation on page 43.

VARADARAJAN, S.; MILLER, P.; ZHOU, H. Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. **Patt. Recogn.**, Elsevier, v. 48, n. 11, p. 3488–3503, 2015. ISSN 0031-3203. Available: <http://dx.doi.org/10.1016/j.patcog.2015.04.016>. Citation on page 63.

VEDALDI, A.; SOATTO, S. Quick shift and kernel methods for mode seeking. In: **Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, October 12-18, 2008, Proceedings, Part IV**. Marseille, France: Springer, 2008. p. 705–718. Available: <http://dx.doi.org/10.1007/978-3-540-88693-8_52>. Citations on pages 52 and 53.

VINCENT, L.; SOILLE, P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE, v. 13, n. 6, p. 583–598, 1991. Available: <http://dx.doi.org/10.1109/34.87344>. Citations on pages 52 and 53.

WANG, M.; LIU, X.; GAO, Y.; MA, X.; SOOMRO, N. Q. Superpixel segmentation: A benchmark. **Sig. Proc.: Image Comm.**, v. 56, p. 28–39, 2017. Available: <http://dx.doi.org/10.1016/j.image.2017.04.007>. Citations on pages 51, 52, and 53.

WANG, W.; YANG, J.; MUNTZ, R. R. Sting: A statistical information grid approach to spatial data mining. In: **Proceedings of the 23rd International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. (VLDB '97), p.

186–195. ISBN 1-55860-470-7. Available: <http://dl.acm.org/citation.cfm?id=645923.758369>. Citation on page 46.

WANG, Z.; XIN, J.; SUN, P.; LIN, Z.; YAO, Y.; GAO, X. Improved lung nodule diagnosis accuracy using lung CT images with uncertain class. **Computer Methods and Programs in Biomedicine**, Elsevier, v. 162, p. 197–209, 2018. Available: <http://dx.doi.org/10.1016/j.cmpb.2018.05.028>. Citation on page 76.

YIN, L.; ANDREWS, J.; HEATON, T. Reducing process delays for real-time earthquake parameter estimation - an application of KD tree to large databases for earthquake early warning. **Computers & Geosciences**, Elsevier, v. 114, p. 22–29, 2018. Available: <http://dx.doi.org/10.1016/j.cageo.2018.01.001>. Citation on page 49.

ZABOT, G. F.; CAZZOLATO, M. T.; SCABORA, L. C.; FAIçAL, B. S.; TRAINA, A. J. M.; TRAINA-JR., C. UCORM: Indexing uncorrelated metric spaces for concise content-based retrieval of medical images. In: **2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)**. Cordoba, Spain: IEEE, 2019. p. 306–311. ISSN 2372-9198. Available: <http://dx.doi.org/10.1109/CBMS.2019.00070>. Citation on page 123.

ZAKI, M. J.; MEIRA JR., W. **Data Mining and Analysis - Fundamental Concepts and Algorithms**. New York, NY, USA: Cambridge University Press, 2014. ISBN 9780521766333. Available: <http://www.dataminingbook.info/pmwiki.php>. Citations on pages 42, 45, 49, 54, and 57.

ZEZULA, P.; AMATO, G.; DOHNAL, V.; BATKO, M. **Similarity Search - The Metric Space Approach**. Berlin, Heidelberg: Springer Publishing Company, 2006. 1-191 p. (Advances in Database Systems, v. 32). ISBN 978-0-387-29146-8. Available: <http://dx.doi.org/10.1007/0-387-29151-2>. Citation on page 40.

ZHANG, F.; SONG, Y.; CAI, W.; ZHOU, Y.; SHAN, S.; FENG, D. Context curves for classification of lung nodule images. In: **DICTA**. Hobart, TAS, Australia: IEEE, 2013. p. 1–7. Available: <http://dx.doi.org/10.1109/DICTA.2013.6691494>. Citation on page 76.

ZHANG, S.; METAXAS, D. Large-scale medical image analytics: Recent methodologies, applications and future directions. **Medical Image Analysis**, Springer, v. 33, p. 98 – 101, 2016. ISSN 1361-8415. Available: <http://dx.doi.org/10.1016/j.media.2016.06.010>. Citation on page 75.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. Birch: An efficient data clustering method for very large databases. In: **Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 1996. (SIGMOD '96), p. 103–114. ISBN 0-89791-794-4. Available: <http://doi.acm.org/10.1145/233269.233324>. Citation on page 45.

ZHANG, Z.; LI, W.; LI, B. An improving technique of color histogram in segmentation-based image retrieval. In: **Information Assurance and Security, 2009. IAS '09. Fifth International Conference on**. Xi'an, China: IEEE, 2009. v. 2, p. 381–384. Available: <http://dx.doi.org/10.1109/IAS.2009.156>. Citation on page 37.


ICMC
USP
SÃO CARLOS