UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Managing feature extraction, mining and retrieval of complex data: applications in emergency situations and medicine

Daniel Yoshinobu Takada Chino

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

ICMC USP
SÃO CARLOS

Daniel Yoshinobu Takada Chino

Managing feature extraction, mining and retrieval of complex data: applications in emergency situations and medicine[1]

Thesis submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Agma Juci Machado Traina

USP – São Carlos
August 2019

[1] This work has been supported by grant #2014/25125-3, São Paulo Research Foundation (FAPESP) and Coordenação de Aperfeiçoamento de Pessoal de NívelSuperior - Brasil (CAPES).

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

```
        Chino, Daniel Yoshinobu Takada
C539m      Managing feature extraction, mining and
        retrieval of complex data: applications in
        emergency situations and medicine / Daniel
        Yoshinobu Takada Chino; orientadora Agma Juci
        Machado Traina. -- São Carlos, 2019.
           142 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. Content-Based Image Retrieval. 2. Image
        Segmentation. 3. Bag-of-Visual-Words. 4. Fire
        Detection. 5. Skin Ulcer. 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

Daniel Yoshinobu Takada Chino

Tratando o problema de extração de características, mineração e recuperação de dados complexos: aplicações em situações de emergência e medicina[2]

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor 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
Agosto de 2019

[2] Este trabalho foi financiado pelo processo nº 2014/25125-3, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e Coordenação de Aperfeiçoamento de Pessoal de NívelSuperior - Brasil (CAPES).

ACKNOWLEDGEMENTS

Gostaria de agradecer primeiramente, à minha noiva Mayumi que compartilhou comigo todos os momentos desse doutorado, me confortando nos momentos de dificuldade. À nossa cachorrinha Mame que sempre nos confortou com seu amor incondicional.

Aos meus pais, que sempre me incentivaram e apoiaram nos momentos de indecisões e escolhas de novos caminhos, permitindo que eu pudesse continuar meus estudos. Ao meu irmão que sempre me aconselhou e à minha irmã que sempre me mostrou o quanto o esforço é recompensador.

À minha orientadora Profa. Agma Juci Machado Traina por toda a paciência, apoio e todo o conhecimento transmitido.

Agradeço também ao Prof. Christos Faloutsos, que me orientou e aconselhou durante minha estadia na CMU.

Aos amigos do GBdI pelas discussões e momentos de alegria. Agradecimentos especiais à Letrícia Avalhais, ao Lucas Scabora e à Mirela Cazzolato pelas colaborações, conselhos e noites perdidas trabalhando.

À Dra. Ana Elisa Serafim Jorge por nos auxiliar na área médica e sempre nos apresentar novos desafios no tratamento de úlceras cutâneas crônicas.

Aos funcionários e professores do ICMC-USP.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, também pelos processos nº 2014/25125-3 e 2015/13346-8, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)[3] e pelo projeto RESCUER finaciado pela Comissão Europeia (Processo 614154) e pelo CNPq/MCTI (Processo 490084/2013-3).

[3] As opiniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade do(s) autor(es) e não necessariamente refletem a visão da FAPESP e da CAPES.

“ Study hard what interests you the most
in the most undisciplined, irreverent
and original manner possible.”
(Feynmann, Richard)

RESUMO

CHINO, D. Y. T. **Tratando o problema de extração de características, mineração e recuperação de dados complexos: aplicações em situações de emergência e medicina**. 2019. 142 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.

O tamanho e complexidade dos dados gerados por mídias sociais e imagens médicas tem crescido rapidamente. Diferentemente de dados tradicionais, não é possível lidar com imagens dentro de seus domínios originais. Aumentando assim os desafios para a descoberta de conhecimento. Técnicas de processamento de imagens podem auxiliar em diversas tarefas de tomada de decisão. Imagens provenientes de crowdsourcing, como imagens de mídias sociais, podem ser usadas para aumentar a velocidade de resposta de autoridades em situações de emergência. Imagens retiradas da área médica podem auxiliar médicos em suas atividades diárias, como no diagnostico de pacientes. Sistemas de recuperação de imagens baseada em conteúdo (CBIR – do inglês *Content-Based Image Retrieval*) são capazes de recuperar as imagens mais similares, sendo uma etapa importante para a descoberta de conhecimento. Entretanto, em alguns domínios de imagens, apenas partes da imagem são relevantes para o problema de recuperação.

Essa pesquisa de doutorado se baseia na seguinte hipótese: a integração de métodos de segmentação de imagens em sistemas CBIR através de características locais aumenta a precisão na recuperação de imagens. As propostas dessa pesquisa de doutorado foram avaliadas em dois domínios de imagem: detecção de fogo em imagens de situações de emergência urbana e imagens de úlcera cutânea crônica. As principais contribuições dessa pesquisa de doutorado podem ser divididas em quatro partes. Primeiro foi proposto o BoWFire, um método para detectar e segmentar fogo em situações de emergência. Foi explorada a combinação das características de cor e textura através de superpixeis para a detecção de fogo em imagens estáticas. A segunda contribuição foi o método BoSS, que explora o uso de superpixeis para extrair características locais. O método BoSS é uma abordagem de *Bag-of-Visual-Words* (BoVW) baseada em assinaturas visuais. Para integrar os métodos de segmentação com sistemas CBIR, foi proposto o *framework* ICARUS para a recuperação de imagens de úlcera cutânea. O ICARUS integra métodos se segmentação baseados em superpixel com BoVW. Também foi proposto o *framework* ASURA para a segmentação de úlceras cutâneas baseado em técnicas de *deep learning*. Além de segmentar as úlceras cutâneas, o ASURA é capaz de estimar a área da lesão em unidades de medida reais. Para tanto, o ASURA analisa os objetos presentes nas imagens. Os experimentos mostraram que as propostas dessa pesquisa de doutorado alcançaram uma melhor precisão ao recuperar as imagens mais similares em comparação às abordagens existentes na literatura.

Palavras-chave: Recuperação de Imagens Baseada em Conteúdo, CBIR, Segmentação de

Imagens, Bag-of-Visual-Words, Detecção de Fogo, Úlceras Cutâneas Crônicas.

ABSTRACT

CHINO, D. Y. T. **Managing feature extraction, mining and retrieval of complex data: applications in emergency situations and medicine**. 2019. 142 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 size and complexity of the data generated by social media and medical images has increased in a fast pace. Unlike traditional data, images cannot be dealt with in its original domain, leading to rising challenges in knowledge discovery tasks. The image analysis can aid on several decision making tasks. Crowdsourcing images such as social media images can be used to increase the speed of authorities to take action in emergency situations. Images taken from the medical domain can support on daily activities of physicians to diagnose their patients. Content-Based Image Retrieval (CBIR) systems are built to retrieve similar images, being an important step for the knowledge discovery. However, in some image domains, only parts of the image are relevant to the problem.

This PhD research is based on the following hypothesis: the integration of image segmentation methods with local feature CBIR system improves the precision of the retrieved images. We evaluate our proposals in two images domain: fire detection on urban emergency situations and chronic skin ulcer images. The main contributions of this PhD research can be divided in four parts. First, we propose BoWFire to detect and segment fire in emergency situations. We explore the combination of color and texture features through superpixels to detect fire in still images. Then, we explore the use of superpixels to extract local features with BoSS. BoSS is a Bag-of-Visual-Words (BoVW) approach based on visual signatures. To integrate segmentation methods with CBIR, we propose ICARUS, a skin ulcer image retrieval framework. ICARUS integrate segmentations methods based on superpixels with BoVW. We also propose ASURA, a deep learning segmentation method for skin ulcer lesions. Besides segmenting skin ulcer lesions, ASURA is able to estimate the area of the lesion in real-world units by analyzing real-world objects present in the images. Our experiments show that our proposals achieved a better precision while retrieving the most similar images in comparison with the existing approaches.

Keywords: Content-Based Image Retrieval, CBIR, Image Segmentation, Bag-of-Visual-Words, Fire Detection, Skin Ulcer.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ASURA	Automatic Skin Ulcer Region Assessment
BIC	Border/Interior pixel Classification
BoSS	Bag-of-Superpixels Signatures
BoSS-CT	Bag-of-Superpixels Color and Texture Signatures
BoVW	Bag-of-Visual-Words
BoW	Bag-of-Words
BoWFire	Best of both Worlds Fire detection
C-BoVW	Cluster-Based Bag-of-Visual-Words
CBIR	Content-Based Image Retrieval
CL-Measure	Counting-Labels Similarity Measure
CNN	Convolutional Neural Network
CPMC	Constrained Parametric Min-Cut
CT	Computerized Tomography
DELF	DEep Local Feature
ELU	Exponential Linear Unit
FCN	Fully Convolutional Network
FiSmo	Fire and Smoke Dataset
GUI	Graphical User Interface
ICARUS	Imaging Content Analysis for the Retrieval of Ulcer Signatures
ICARUS-Seg	Imaging Content Analysis for the Retrieval of Ulcer Signatures Through Segmentation
k-NN	k-Nearest Neighbors
LBP	Local Binary Patterns
LSC	Linear Spectral Clustering
MAP	Mean Average Precision
MLE	maximum-likelyhood estimation
MRI	Magnetic Resonance Imaging
RAFIKI	Retrieval-based Application for Imaging and Knowledge Investigation
ROI	Region Of Interest
S-BoVW	Signature-Based Bag-of-Visual-Words
SCBIR	Semantic Content-Based Image Retrieval

SDLC	Sorted Dominant Local Color
SDLCT	Sorted Dominant Local Color and Texture
SEEDS	Superpixels Extracted via Energy-Driven Sampling
SIFT	Scale-Invariant Feature Transform
SLIC	Simple Linear Iterative Clustering
SURF	Speeded Up Robust Features
TBIR	Tag-Based Image Retrieval
VSM	Vector Space Model

CONTENTS

CHAPTER 1

INTRODUCTION

Nowadays, data generation has increased in size and complexity in a fast pace. The number of image data generated by social media (ORNAGER; LUND, 2018) and medical systems (ANWAR *et al.*, 2018) can be counted to billions. This can present a great challenging for computational systems, such as information retrieval systems. Experts can analyze similar situations and use them to aid in decision making tasks. For example, crowdsourcing images such as social media images can be used to aid authorities in emergency situations (BEDO *et al.*, 2015a; CAZZOLATO *et al.*, 2016; SHARMA *et al.*, 2017), while medical images can support physicians in the diagnosis of diseases (Oliveira *et al.*, 2017; GHOLAMI *et al.*, 2018; CAZZOLATO *et al.*, 2019). One way to deal with similarity queries in images is through Content-Based Image Retrieval (CBIR) systems (LIU *et al.*, 2007; ZHENG; YANG; TIAN, 2018).

1.1 Motivation

When searching for similar images, the human perception is able to describe the image in details (ALZU'BI; AMIRA; RAMZAN, 2015). Humans can properly describe the objects of an image and interpret their meaning and interactions, *e.g.*, the location of a fire in an emergency situation, or the severity of a wound in a patient image. On the other hand, computer systems see digital images as a set of numerical values in a matrix with no semantic. Usually, this type of data is called complex data. The distance between the richness of details of the human perception and the machine view of digital images is called the "semantic gap" (HARE *et al.*, 2006).

To overcame this distance, many proposal tries to describe the visual properties of the images through numerical features (DESELAERS; KEYSERS; NEY, 2008). These features can describe visual properties of the image as a whole (TORRES; FALCAO, 2006) or they can describe each region of the image separately (SIVIC; ZISSERMAN, 2003; SANTOS *et al.*, 2017). However, sometimes only a part of the image contain relevant information regarding the semantic of the problem (PEREYRA *et al.*, 2014; BLANCO *et al.*, 2016).

In this PhD research, we aim at answering the following research question: *"How can we improve image retrieval systems when only parts of the image are relevant to the problem?"*. The main challenge towards solving this problem is to correctly detect the relevant regions of the image and to find the best way to extract features to represent these regions. One way of dealing with these tasks is by implying segmentation algorithms and using local features.

1.2 Problem Statement

As previously discussed, the main goal of this PhD research was to propose a method able to retrieve similar images based only on the relevant parts of the image. Our proposal aims at integrating segmentations methods to discover the relevant regions and local features retrieval systems. On this context, we propose the following thesis:

Thesis. *The integration of segmentation methods with local feature extraction improves the precision of similarity-based image retrieval tasks, consequently lowering the semantic gap between the computer knowledge representations and human perception.*

In order to support the stated thesis, this PhD research focused on analyzing two image domains: fire emergency situations and chronic skin ulcers in lower limbs.

1.2.1 Emergency Fire Events

An intense flow of information is gathered in a short period of time 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 emergency situations. Cameras embedded in mobile devices can provide visual information of wider spaces, which can be used by authorities to better understand the situation (CHEN *et al.*, 2006). Dealing with such amount of information in real-time is a challenging task.

On this context, the RESCUER[1] Project developed an emergency system to support Crisis Control Committees (CCC) during a crisis situation. The system developed in the RESCUER Project allows witnesses, victims or rescue staff, present at the emergency location, to send images and videos of the incident to a crowdsourcing mobile framework. Part of this framework involved an automated data analysis solution to detect fire regions in images. 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 (HUANG; CHENG; CHIU, 2013).

[1] Project FP7-ICT-2013-EU-Brazil - "RESCUER - Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management" – <http://www.rescuer-project.org/>

To the best of our knowledge, the fire detection methods on the literature are focused on video data (CHEN; WU; CHIOU, 2004; CELIK; DEMIREL, 2009; ROSSI; AKHLOUFI; TISON, 2011; RUDZ *et al.*, 2013). They explore the temporal component of videos to improve the fire detection. However, when the temporal component is not available (still images), the presence of false-positives increases. In this PhD research we aimed at fire detection methods on still images while avoiding the presence of false-positives.

1.2.2 Chronic Skin Ulcer

On this PhD research, we also explored images of chronic skin ulcers in lower limbs. Acute wounds have a well understood healing steps, however, in chronic skin ulcers these steps are disrupted (MORTON; PHILLIPS, 2016). These lesions may be caused by different reasons, such as poor blood circulation in lower extremities, injuries, infections, tumors and other skin conditions (DORILEO *et al.*, 2010; MORTON; PHILLIPS, 2016). To diagnose skin ulcers, physicians observe visual aspects of the wound, such as location, size, color, texture, and shape(MORTON; PHILLIPS, 2016).

One way the physicians follow-up the healing rate of chronic skin ulcers is by regularly taking their photographs. By making this temporal comparison, the physician can diagnose if the wound is healing, *e.g.*, the area of the lesion decreased with time. This photographs can be taken by digital cameras or mobile devices. In this context, there is a need to create applications that can aid physicians on the follow-up of the wounds (EVANS; LOBER, 2017; NAVARRO; ESCUDERO-VINOLO; BESCOS, 2018) Among the tasks needed on this applications are the detection of the lesion region/area (DORILEO *et al.*, 2010; GHOLAMI *et al.*, 2018) and the retrieval of similar images (BEDO *et al.*, 2015b; BLANCO *et al.*, 2016).

1.3 Contributions

This PhD research resulted in four main contributions, each of them addressing one of the research problems previously listed. The contributions are summarized as follows:

1. **The BoWFire method:** BoWFire is a method to detect and segment fire in still images of fire emergency situations in urban areas. Since the majority of fire detection methods are based on videos, they fail to detect the fire when we remove the temporal aspect of the data. BoWFire overcame this problem by using color and texture features of superpixels.

2. **The BoSS method:** BoSS is a CBIR system based on Bag-of-Visual-Words (BoVW). BoSS dismisses the need to create a visual word dictionary beforehand. BoSS extracts local histograms of superpixels, instead of using a visual word, BoSS maps the local histogram into a visual signature. The visual signature is described by the dominant colors of the superpixels. Another aspect of BoSS, is that it uses fractal theory to automatically discover

the dominant colors. We also explored the combination of color and texture features to map the visual signatures.

3. **The ICARUS framework:** ICARUS is a CBIR system for skin ulcer images. We applied the skin ulcer segmentation methods in the feature extraction of a CBIR. By extracting only the features from the relevant regions of the image, we were able to increase the precision of the retrieved images.

4. **The ASURA framework:** ASURA is a framework to assess the area in real-world units (*e.g.* cm^2) of skin ulcer lesions. ASURA takes advantage of deep learning techniques to detect and segment the lesion and the measurement tool. A later step, process the segmented measurement tool to detect the ticks and estimate the relationship between number of pixels and a real-world unit.

1.4 Summary

The remainder of this PhD thesis is organized as follows. We give a brief overview of the relevant literature and the background concepts in Chapter 2. In Chapter 3 we explore the color and texture aspects of superpixels to detect and segment fire in emergency situations through the BoWFire method. In Chapter 4 we explore the use of superpixels in BoVW approaches and propose BoSS, a signature based BoVW. In Chapter 5 we introduce ICARUS, a CBIR system for skin ulcer images. In Chapter 6 we propose ASURA to segment and measure the lesion areas of chronic skin ulcers. And in Chapter 7, we present the conclusions and suggestions of future work. We also present VolTime in Appendix A, an analysis of the user activities based on timestamps and event volume.

CHAPTER

2

BACKGROUND AND RELATED WORKS

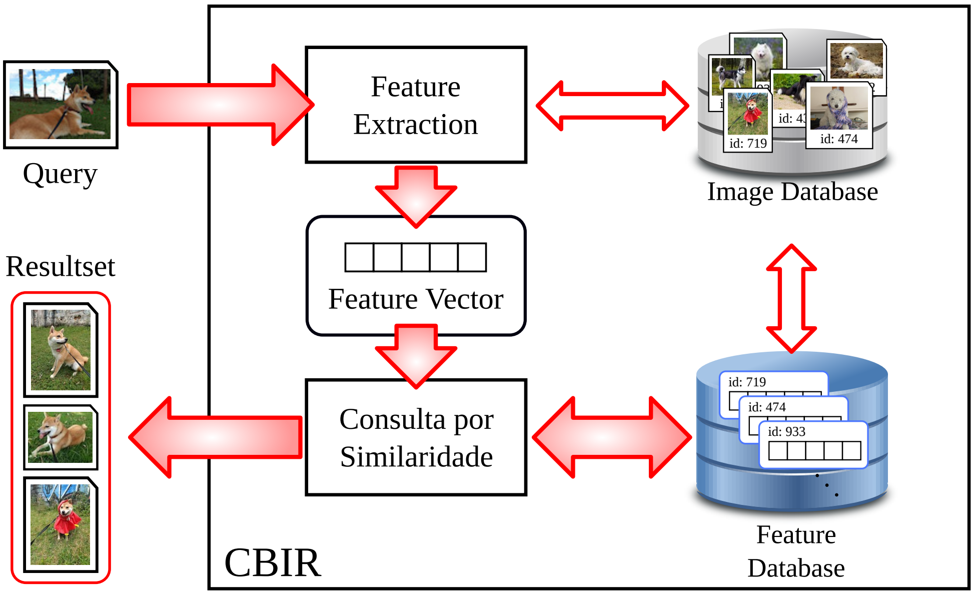
In this Chapter we briefly present the main concepts needed for the understanding of this PhD research. In Section 2.1, we provide an introduction to Content-Based Image Retrieval (CBIR) systems. In Section 2.2, we discuss the use of local feature through the Bag-of-Visual-Words (BoVW) approach. An overview of image segmentation is given in Section 2.3. We also show how the introduction of deep learning techniques can be used on these tasks in Section 2.4. Section 2.5 presents the state-of-the-art methods in the fire image and skin ulcer domains.

2.1 Content-Based Image Retrieval Systems

When dealing with complex data, such as images, the analysis is not made on the original domain of the data. Traditionally, the complex data is analyzed using attributes/features that describe the visual features of the image. There are two approaches for image retrieval. The first approach is the Tag-Based Image Retrieval (TBIR). TBIR is based on the context of the image, text attributes describing scenes or objects of the image, as tags and labels (WU *et al.*, 2012). Usually such descriptions are manually made by a user or are automatically extracted from text informations near the image as well as the content of the image itself. The latter approach is the CBIR. CBIR is based on the content of the image, where the images are described by their intrinsic characteristics (WELTER *et al.*, 2012; DESERNO; ANTANI; LONG, 2009; NEVEOL *et al.*, 2009; VARGHESE *et al.*, 2014; SHRIVASTAVA; TYAGI, 2014). Usually, these characteristics are obtained automatically. On this PhD research, we will focus on the CBIR approach.

A CBIR system needs three modules to retrieve information (TORRES; FALCAO, 2006): data storage (knowledge database), feature extraction, query processing. Figure 1 shows a generic CBIR architecture. The CBIR system receives an image as input. The image passes through the feature extraction module where its visual features are extracted. The image is now represented by a set of numerical attributes which denotes visual properties. These numerical attributes are known as feature vectors. The query processing module uses the feature vectors to access the

Figure 1 – CBIR system architecture.



Source: Elaborated by the author.

knowledge database and return the most similar images. To avoid excessive processing, CBIR systems can storage the feature vectors of the knowledge database in a specific database (Feature Database). This way, the CBIR system extracts the features from the images and store them in the knowledge database only once. Afterwards, the feature vectors are used in the place of the images for indexing, querying and retrieval, making the process more efficient.

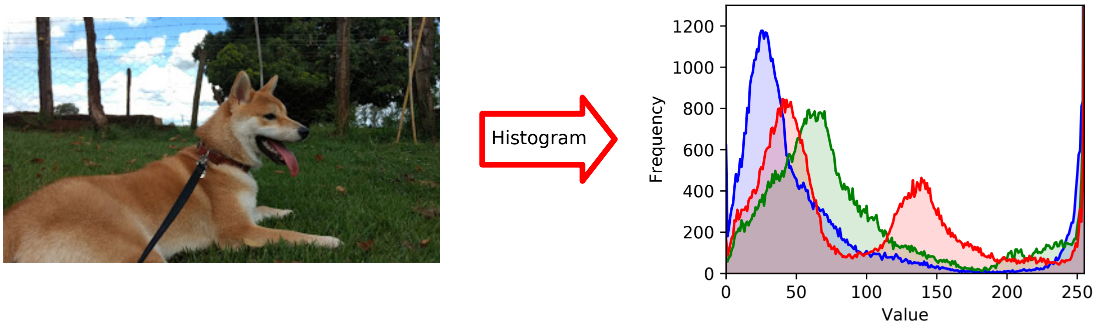
2.1.1 Feature Extraction

In order for a CBIR system to retrieve the images, they should be represented by a numerical representation that captures visual properties of the image. This numerical representation is obtained through feature extraction methods.

As mentioned before, the attributes extracted from the images are placed in feature vectors. The feature vectors extracted from a given image correspond to numerical measurements that describe the image's visual properties. Such properties are able to discover connections between pixels of the whole image (global) (TORRES; FALCAO, 2006), or of small regions of the image (local) (SHABAN *et al.*, 2013). Low-level descriptors (DESELAERS; KEYSERS; NEY, 2008), as those based on color, texture and shape, are frequently used.

The color distribution is one of the most basic visual property of an image. One of the most common methods is the color histogram (HAFNER *et al.*, 1995), which extracts the color distribution properties of the image. For a monochromatic image, the histogram describes the frequency of the grayscale values. On the other hand, the histogram of a color image (RGB images) can be the concatenation of the histogram of each color channel. Figure 2 shows the histogram of an RGB image. Through the color histogram, it is also possible to compute the color moments (DATTA *et al.*, 2008) to describe the probability distribution of the image colors. A variation of the color histogram is the Border/Interior pixel Classification (BIC) (STEHLING; NASCIMENTO; FALCAO, 2002), which computes two histograms, one for the border pixels and another one for the interior pixels.

Figure 2 – Histogram extraction on a RGB image.



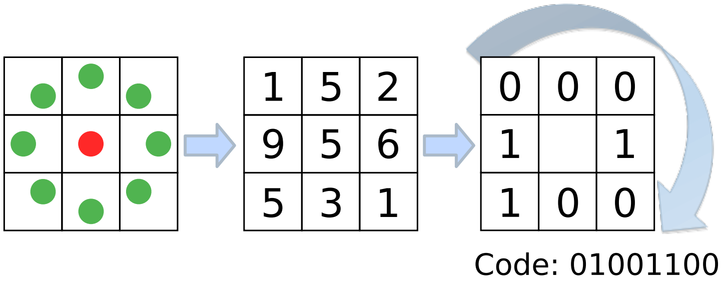
Source: Elaborated by the author.

There are other color extraction methods, such as the methods presented in the MPEG-7 standard (SIKORA, 2001; KIM *et al.*, 2011). The Dominant Color method clusters the colors in regions into a small number of representative colors. The colors are represented by their value, their percentage in the image and their variance. The Color Structure method captures the spatial distribution of the colors using a sliding window to compute a histogram. For each step, Color Structure computes the frequency of the colors in each position of the sliding window. The Scalable Color method uses the HSV color space to compute the histogram and a Haar wavelet transformation. The Color Layout method divides the image in rectangular regions and calculate the mean value of the color in each region. Then, it uses a discrete cosine transformation to describe the space distribution of the colors. There is also combinations of MPEG-7 extractors. The Dominant Color Structure Descriptor (WONG; PO; CHEUNG, 2007) combines the Dominant Color and the Color Structure methods. The Weighted Dominant Color Descriptor (TALIB *et al.*, 2013) weights the dominant colors according to their contribution on the edge of objects.

Besides color, texture is a common feature in image processing. It is important because, together with color, it describes the surface of naturally-occurring phenomena. These objects' textures can be described, for example, by the roughness and homogeneity of their surfaces (SAIPULLAH; KIM, 2012). Unlike color extractors, which can be computed by analyzing only one pixel, texture patterns occurs along a region of the image. One way to capture texture features is through statistical measures, such as mean, standard deviation and entropy of the pixel values (GONZALEZ; WOODS, 2008). These statistical measures can also be computed from the co-occurrence matrix of the image as in the Haralick feature extractor (HARALICK; SHANMUGAM; DINSTEIN, 1973).

One of the most used texture features is the LBP (OJALA; PIETIKAINEN; MAENPAA,

Figure 3 – The Local Binary Patterns (LBP) process to code the textures.



1	5	2
9	5	6
5	3	1

0	0	0
1		1
1	0	0

Source: Elaborated by the author.

2002; GUO; ZHANG; ZHANG, 2010). The LBP method codes the texture according to the region of the image. LBP defines a neighborhood region for each pixel and compares the value of each neighbor pixel with the central pixel. Neighbors with a value greater than the central pixel are coded with the value 1, otherwise, they are coded with value 0. At the end of the process, LBP creates a code by looking clockwise the values of the neighbors. Figure 3 shows the steps LBP uses to code the texture. There are some variations of the LBP, as the Centralized Binary Pattern (FU; WEI, 2008), the Completed LBP (GUO; ZHANG; ZHANG, 2010), the Local Ternary Pattern (LIAO, 2010) and the Structural Difference Histogram Representation (FENG *et al.*, 2017).

Shape information is considered the closest approximation to the human perception of an object's image (YANG *et al.*, 2008). Feature extractors of this depend on a pre-processing step that segments and detects the border of the objects. The shape extractors describes visual properties of the objects like translation, rotation, and scale invariance (ZHANG; LU, 2004; KAZMI; YOU; ZHANG, 2013). There are various methods to extract shape features, as the Zernike moments (HOSNY, 2008), Fourier descriptors (CHEN; YEH; YIN, 2009) and the contour salience descriptors (TORRES; FALCAO, 2007).

Local features extractors summarize the visual properties of certain regions of the image (TUYTELAARS; MIKOLAJCZYK, 2008). Ideally, the local features extracted from the regions may have some semantic meaning, such as edges or small objects (LOWE, 1999). Local feature extraction can be used on object/scene detection (ONEATA *et al.*, 2014) and tracking (BUONCOMPAGNI *et al.*, 2015). The features are extract in the neighborhood of key points. The key point detection can be made in various way, one way is by using a regular grid to divide the image (TUYTELAARS; SCHMID, 2007). It is also possible to use superpixels as a key point detection (JUNEJA *et al.*, 2013; CHINO *et al.*, 2018). Another approach uses salient points (edges and corners) to detect the key points. One of the most used local feature extractor is the Scale-Invariant Feature Transform (SIFT) (LOWE, 1999), which uses difference of gaussians to detect the key points. A similar method is the Speeded Up Robust Features (SURF) (BAY *et al.*, 2008), which detects key points by using of Hessian matrix approximations.

2.1.2 Similarity Measures

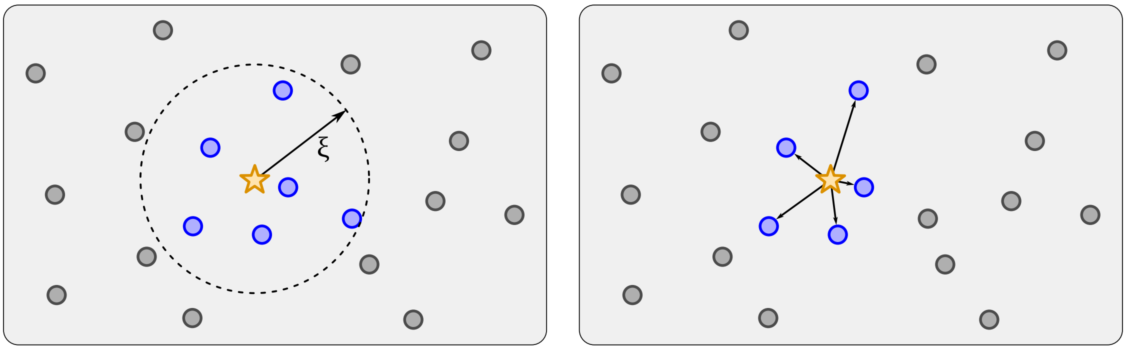
When processing a query, the CBIR system must retrieve images with similar visual properties. To do so, the CBIR systems must have a way to compare the query image with the images in the knowledge database. One way to compare two images is by calculating a distance function between their feature vectors. The distance function measures the dissimilarity between two objects. That is, similar (or close) objects have a smaller distance, while different objects have a larger distance.

The most common distance functions are the ones from the Minkowski family (L_p) (WILSON; MARTINEZ, 1997), as the Euclidian (L_2) or the Manhattan (L_1) distances. Another distance function is the Cosine Angle distance (QIAN *et al.*, 2004), which calculates the inner product between to feature vectors. There are also distance functions based on set theory, as the Jaccard index (ARASU; GANTI; KAUSHIK, 2006) and the distance of Hausdorff (ZHAO; SHI; DENG, 2005). The latter three distance functions are very useful to compare the similarity between text documents (BRODER, 1997).

2.1.3 Similarity Queries

On the final step of a CBIR system, the query processing module must return the most similar images. The query processing module uses the similarity measure (distance functions) to determine the most similar images. Usually, there are two similarity queries: the range query and the k-Nearest Neighbors (k-NN). Given a query image and a radius ξ, the range query returns all the images which the distances to the query image are within ξ. On the other hand, given a query image and an integer k, the k-NN query returns the top k closest images to the query image. Figure 4 shows examples of both range and k-NN queries.

Figure 4 – Visual representation of both similarities queries on a 2D space. The query image is represented by a star shape and the retrieved images are represented by the blue circles.



(a) Range query with a given radius ξ,

(b) k-NN query for $k = 5$

Source: Elaborated by the author.

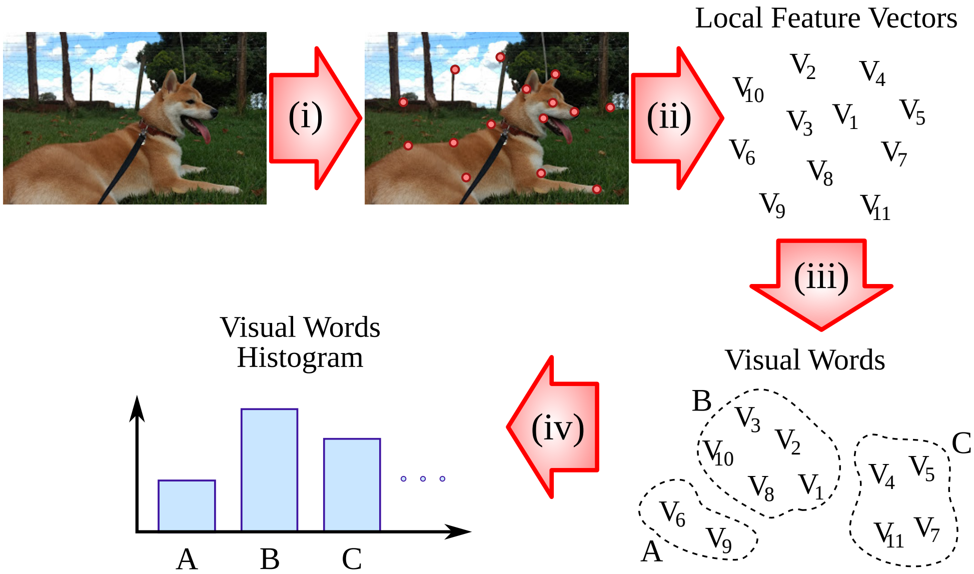
2.2 CBIR Using Local Features

Unlike global feature extractors, local feature extractors get more than one feature vector from an image. This way, it is not possible to compare two images using the similarity measures presented in the previous Section, because the number of elements to be compare may differ, as well as their positioning. Local features can be represented in various way: BoVW (SIVIC; ZISSERMAN, 2003), fisher vector representation (PERRONNIN *et al.*, 2010) or vector locally aggregated descriptors (JEGOU *et al.*, 2012). On this PhD research we will be focusing on BoVW approaches.

2.2.1 Bag-of-Visual-Words

The BoVW method (SIVIC; ZISSERMAN, 2003) is inspired on the Bag-of-Words (BoW) (JOACHIMS, 1998) approach developed to mine information from long texts. The BoW is a text representation through a histogram of words. Similarly, the BoVW represents the image as a histogram of visual words. Thus, the local features are summarized in a non-ordered way into a single feature vector. The BoVW can be used in various applications: similar fragment retrieval in animations (SUN; KISE; CHAMPEIL, 2012); classification of histopathological images (KUMAR *et al.*, 2017); handwritten signature verification (OKAWA, 2018); and building detection in pictures (RADENOVIC *et al.*, 2018).

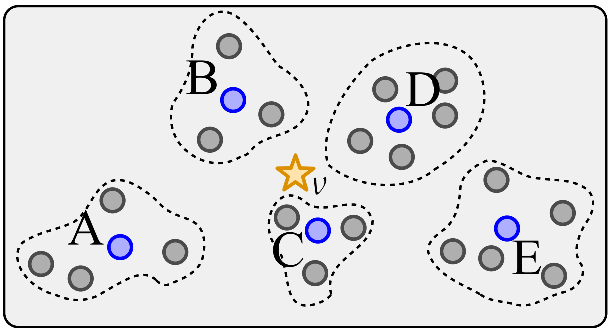
Figure 5 – BoVW approach steps: (i) Points of interest detection; (ii) Local feature extraction from the points of interest; (iii) Local feature mapping into visual words through a visual dictionary; (iv) Visual word histogram.


Local Feature Vectors
(i)
(ii)
V_{10} V_2 V_4 V_3 V_1 V_5 V_6 V_7 V_8 V_9 V_{11}
(iii)
Visual Words
B V_3 V_2 V_{10} V_8 V_1
C V_4 V_5 V_{11} V_7
A V_6 V_9
(iv)
Visual Words Histogram
A B C

Source: Elaborated by the author.

On a BoVW approach the local features are mapped into visual words using a visual dictionary. Figure 5 shows the process to represent an image using a BoVW approach. The steps (i) and (ii) are the ones discussed on the previous Sections. During step (iii), the local features are mapped into visual words using a visual dictionary. Usually, the visual dictionary are created using the local features extracted from the knowledge database. One way to create the visual dictionary is by clustering the local features from the knowledge database (SIVIC; ZISSERMAN, 2003). By clustering, the local features are partitioned into regions with similar visual properties. The visual dictionary is then defined by the clusters' representatives. This approach of creating the visual dictionary is also known as Cluster-Based Bag-of-Visual-Words (C-BoVW). It is also possible to create a visual dictionary by random sampling local features from the knowledge database (SANTOS *et al.*, 2010). The random sampling allows a faster building time without losing too much information. One important aspect of the visual dictionary is its size (number of visual words). A small visual dictionary has little discriminative power, while a large visual dictionaries lacks generalization. The size of the dictionary is strongly related to the application, varying between 500 (PAPADOPOULOS *et al.*, 2011) to 10,000 (SIVIC; ZISSERMAN, 2003) visual words.

Figure 6 – Mapping local feature v into visual word.



Source: Elaborated by the author.

The final step – step (iv) – is the computation of the visual words histogram. To compute the histogram, the local features must be assigned to the visual words in the visual dictionary. The local features can be assigned using three approaches: hard, multiple and soft assignment. On the hard assignment, each local feature is assigned to the closest visual word in the visual dictionary (SIVIC; ZISSERMAN, 2003). Therefore, each local feature has the same weight on the visual word histogram. On the example shown in Figure 6, the local feature v would be assigned to the visual word C. On the multiple assignment (JEGOU; HARZALLAH; SCHMID, 2007), each local feature can be assigned to more than one visual word. Each visual word assigned to the local feature count as one on the visual word histogram. In Figure 6, the local feature v would be assigned to B, C and D if we considered up to three visual words. Finally, on the soft assignment (JIANG; NGO; YANG, 2007), each local feature can also be assigned to multiple visual words. However, the visual words are weighted according to their distance on the visual word histogram. On the example shown in Figure 6, the visual word C would weight more than the visual word B and D.

There are also variations of the BoVW approach that consider the spatial distribution of the visual words (PENATTI; VALLE; TORRES, 2011). Avni *et al.* (AVNI *et al.*, 2011) incorporate the spatial position of the visual words on the histogram. Pedrosa *et al.* (PEDROSA *et al.*, 2014) introduced the n-grams concept, which represents the co-occurrence of spatially near visual words. On this PhD research we will not be exploring the spatial distribution of local features, but we will explore BoVW approaches that dismiss the need to build a visual dictionary beforehand.

2.2.2 Signature-Based Bag-of-Visual-Words

Previously, we described a C-BoVW visual dictionary, which is created by using a clustering technique over a set of local features. However, despite the strategies adopted to speed up the clustering process (PHILBIN *et al.*, 2007; DIMITROVSKI *et al.*, 2016), determining the visual words can still take a lot of processing time. One way to overcome this problem is using Signature-Based Bag-of-Visual-Words (S-BoVW) approaches, which use map functions to represent the local features in visual signatures (VIDAL *et al.*, 2012; SANTOS, 2016).

Visual signatures summarize information directly from the local feature, eliminating the need of clustering techniques to create a dictionary. One method of this approach is the Sorted Dominant Local Color (SDLC) (SANTOS *et al.*, 2015), which extracts visual signatures based only on the image's color. In this method, the image is separated into rectangular partitions at fixed positions, where each labeled partition is then separated into squared blocks. By doing this, the method generates a signature of each block by selecting the most frequent color values up to a threshold. Later, the signature of each block is assigned to its unique partition label. The main limitation of SDLC is the requirement of multiple parameters, such as the number of partitions, blocks and the threshold value.

Following, the Sorted Dominant Local Color and Texture (SDLCT) (SANTOS *et al.*, 2017) is an extension of the SDLC by including textural properties of the images. In this method, not only color signatures are generated for each block, but also the signatures for their respective textures. Both signatures types (color and texture) are processed separately during the query execution, and the similarity among images is determined by combining the result achieved for each type. This combination requires an additional parameter to determine the weight of each signature type when retrieving. The main problem with these approaches is the lack of semantic meaning to their parameters (thresholds), making them difficult to tune up.

2.2.3 Retrieval Models Using BoVW

The final step of a CBIR system using BoVW is the retrieval model used for the queries. Since BoVW is a visual word histogram representation of the image, it is possible to compare two images using similarity measures, such as the Cosine Angle and Jaccard distances presented on the previous Section. However, it is also possible to use textual based retrieval models such as the Vector Space Model (VSM) (SIVIC; ZISSERMAN, 2003; SANTOS *et al.*, 2017). On the VSM, given a visual dictionary S, each image I is represented by a vector of weights $W_I = \{w_1, w_2, \ldots, w_{|S|}\}$, where $|S|$ is the size of the visual dictionary. The VSM compare two images by calculating the normalized scalar product between the query vector W_q and the image vector W_I. It is important to note that different weights can be used depending on the application, some of the used weights are shown in Table 1.

Table 1 – Weighting schema evaluated in the queries.

Weight type	Definition
w_1	$w_{s,I} = tf_{s,I}$
w_2	$w_{s,I} = idf_s$
w_3	$w_{s,I} = tf_{s,I} \times idf_s$
w_4	$w_{s,I} = tf_{s,I} \times idf_s \times match_{I',I}$
w_5	$w_{s,I} = idf_s \times match_{I',I}$
w_6	$w_{s,I} = match_{I',I}$
w_7	$w_{s,I} = tf_{s,I} \times match_{I',I}$

Source: Santos *et al.* (2017).

where s is a visual world in S, I' is the query image, I is the compared image and $tf_{s,I}$, idf_s and $match_{I',I}$ are defined as follows:

- $tf_{s,I}$ represents the frequency of the visual word s in image I;
- idf_s represents the importance of the visual word s in the knowledge database and can be computed using $idf_s = \log(\frac{n}{n_s})$, where n is the number of images in the knowledge database and n_s is the number of images in which the visual word s appears;
- $match_{I',I}$ measures how many visual signatures of query image I' were found in image I. It is important to note that $match_{I',I}$ is query dependent and is computed during queries.

2.3 Image Segmentation

As mentioned in Chapter 1, when dealing with images, humans can describe and interpret the content of the image, by detecting the objects and the interaction between them in the image. Although the feature extraction methods can translate the images in numerical values, there is no semantics in these values. It is possible to close this semantic gap by analyzing only certain regions of the images (JING *et al.*, 2004; ALZU'BI; AMIRA; RAMZAN, 2015; BLANCO *et al.*, 2016). One way of detecting these regions is through image segmentation. Image segmentation is the process to divide an image into meaningful regions (ZHU *et al.*, 2016). Usually such regions correspond to actual objects in real world scenarios, such as obstacles in traffic (PFEIFFER;

FRANKE, 2010), fire in emergency situations (RUDZ *et al.*, 2013) or chronic wounds in medical images (DORILEO *et al.*, 2010).

One of the most basic segmentation algorithm is the watershed (VINCENT; SOILLE, 1991). The watershed considers the morphological surface of the image and floods the local minimums until different components meet. Another approach is the thresholding (OHLANDER; PRICE; REDDY, 1978). The thresholding uses the image histogram to discover a cutting point value. This process can be done recursively to split the regions into subregions. Another method uses contour detector algorithms and merge regions according to the edge strength (ARBELAEZ *et al.*, 2011).

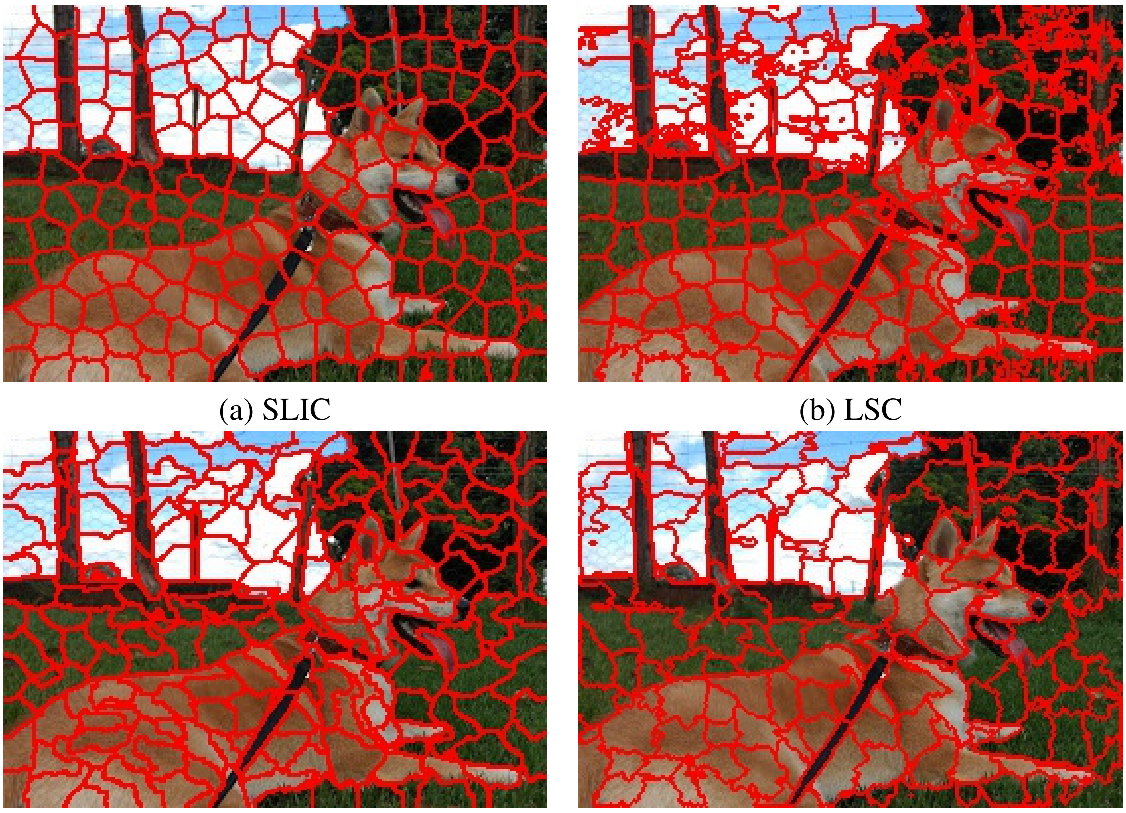
It is also possible to use machine learning algorithms to aid on the image segmentation task. One approach uses clustering methods such as the K-Means (ZHU *et al.*, 2016). Park *et al.* (PARK; YUN; LEE, 1998) used the K-Means on a 3D space of the RGB coordinates. Weeks and Hague (WEEKS; HAGUE, 1997) also used the K-Means, however, they applied on the HSI color space. Another clustering algorithm used for segmentation is the mixed gaussians (RAO *et al.*, 2009; PEREYRA *et al.*, 2014). The Constrained Parametric Min-Cut (CPMC) (CARREIRA; SMINCHISESCU, 2012) uses regular grids to train a gaussian mixture model and ranks how well each region is. The Category Independent Object Proposal (ENDRES; HOIEM, 2014) uses a random forest classifier to segment objects.

One important technique used in image segmentation is the superpixels. Superpixels have being applied to a variety of applications, such as image segmentation (LI; WU; CHANG, 2012), retrieval (WANG *et al.*, 2017) and BoVW techniques (JUNEJA *et al.*, 2013). A superpixel is defined as an atomic region of an image in which their pixels share similar homogeneity, *i.e.*, each group of pixels is coherent to some visual aspect (ACHANTA *et al.*, 2012). In practice, superpixels are useful to capture redundancies on the image and, more importantly, to reduce the complexity of subsequent image processing tasks. Moreover, a superpixel generation needs to comply with the visual boundaries of an image.

One of the most used superpixel generation algorithm is the Simple Linear Iterative Clustering (SLIC) (ACHANTA *et al.*, 2012). The SLIC is an adaptation of the K-Means algorithm for superpixel generation that is fast and memory efficient. The SLIC starts using a regular grid and then adjusts the superpixel boundaries using a distance function based on the values of pixels using the Lab color space and their geometric position. The superpixels produced by SLIC tends to have a more regular shape. Another superpixel generation algorithm based on K-Means is the Linear Spectral Clustering (LSC)(LI; CHEN, 2015). However, instead of using the color space, LSC maps each pixel of the image in a ten dimensional feature space.

Another method used to produce superpixels is the Superpixels Extracted via Energy-Driven Sampling (SEEDS) (BERGH *et al.*, 2012). SEEDS uses a multi-resolution grid, starting with a large regular grid, and then uses an energy function to adjust the boundaries. The energy function is based on enforcing the color similarity between the boundaries and the superpixel

Figure 7 – Examples of the output of the superpixels algorithms.



(a) SLIC

(b) LSC

(c) SEEDS

(d) Compact Watershed

Source: Elaborated by the author.

color histogram. SEEDS uses hill-climbing to evaluate the energy function and move the boundaries. This way, according to the authors, SEEDS is able to produce superpixels in real time at 30Hz. Neubert *et al.* proposed the Compact Watershed (NEUBERT; PROTZEL, 2014), a variation of the watershed segmentation to produce superpixels. The Compact Watershed is based on the seeded watershed. However, instead of producing segmentations with irregular size/shape, it uses the geometric position to guarantee the compactness of the superpixels. Figure 7 shows examples of the representative superpixels algorithms aforementioned.

2.4 Deep Learning Techniques

Since the introduction of deep learning techniques on the ImageNet[1] in 2012, deep learning techniques have been largely explored in image processing tasks. One of the most famous deep learning techniques, are the Convolutional Neural Networks (CNNs). CNNs have been used in other image processing tasks, such as image classification (KAWAHARA; HAMARNEH, 2016; Yu *et al.*, 2017), object recognition (REDMON *et al.*, 2016), fire detection (SHARMA *et al.*, 2017) and segmentation (Yuan; Chao; Lo, 2017). It is important to note that, since the objective

[1] http://image-net.org/

of this PhD research is not the development of deep learning techniques, but to take advantage of the ones already provided in the literature. We will give a brief discussion on deep learning techniques on image processing.

2.4.1 Convolutional Neural Networks

The architecture of a CNN consists a series of convolutions, pooling operations (downsampling), activation functions, and fully-connected layers, which are similar to the hidden layers of a multilayer perceptron (PONTI *et al.*, 2017). The basic idea of a CNN is to use a series of convolutions and downsamplings which encodes an image to a feature map, which can be used in different tasks.

So far, several architectures have been proposed. AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) was the champion of the ImageNet 2012, it consists of five convolution layers and two fully-connected layers to classify the images. The VGG-Net (SIMONYAN; ZISSERMAN, 2014) increased the depth of the CNN, winning the ImageNet in 2014. The Residual Network (ResNet) (HE *et al.*, 2016) introduced the residual blocks, which preserve the characteristics of the input tensor before applying transformations. The Inception (SZEGEDY *et al.*, 2017) used small parallel convolutions instead of adding depth. One important aspect of these models is the large amount of parameters, which requires a large amount of data to learn in the training phase (ANWAR *et al.*, 2018). However, it is possible to use deep learning techniques in smaller datasets by using transfer learning (OQUAB *et al.*, 2014). The idea of the transfer learning is to use the weights learned in large datasets in different tasks. This way, it is possible to use these pre-trained models, such as the ones previously mentioned, in tasks as feature extraction and segmentation.

2.4.2 Deep Learning Features

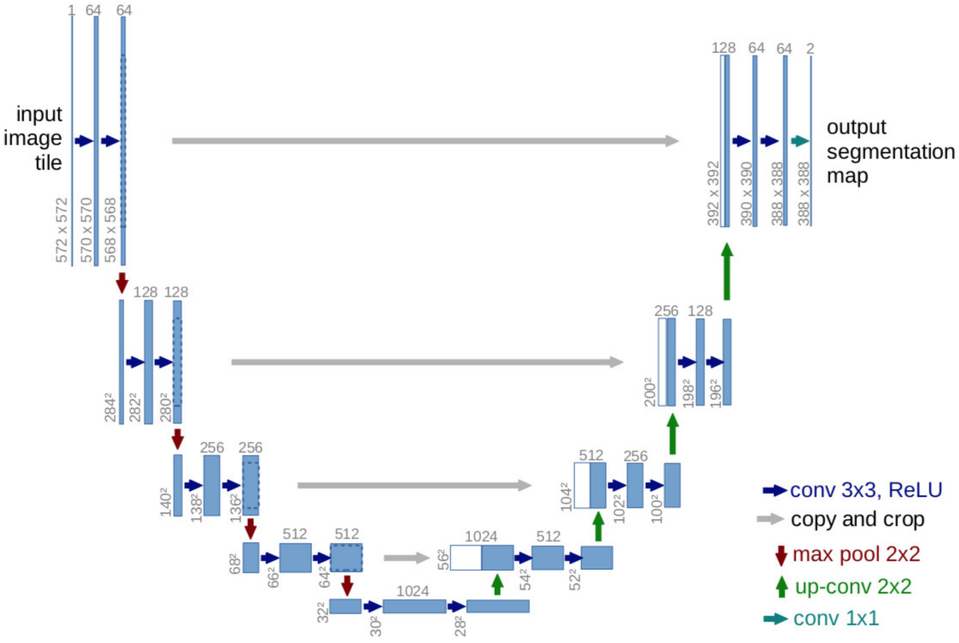
Classification CNNs can be used to extract features from images. An initial approach to use CNN in CBIR systems considered the fully connect layers as global features (BABENKO *et al.*, 2014; GONG *et al.*, 2014). Razavian *et al.* (RAZAVIAN *et al.*, 2014) uses the output of the first fully connected layer. Razavian *et al.* extracts CNN features from sub-patches of the image with different size and locations. Yandex *et al.* (Yandex; Lempitsky, 2015) considers the activations of convolution layers as local features and aggregate them in a global feature using a sum pooling.

A local feature extractor based in CNN is the DEep Local Feature (DELF) (NOH *et al.*, 2017). DELF constructs an image pyramid and applies a pre-trained ResNet until the fourth convolution layer for each level independently. DELF uses the feature maps as local features and is able to detect keypoints using the receptive fields.

2.4.3 Image Segmentation Using Deep Learning

Several models using CNNs were proposed for image segmentation. The Fully Convolutional Network (FCN) (LONG; SHELHAMER; DARRELL, 2015) uses an convolutional network to encode the image and then learns to make a pixel wise prediction of the pixel class. The Deconvolutional Network (NOH; HONG; HAN, 2015) uses an encoder/decoder architecture, in which the encoder consists of the fully connected layers of the VGG-Net. To do the upscaling, the Deconvolutional Network uses an "unpooling". The unpooling is made by recording the locations of maximum activations while doing the pooling operations. A similar approach was used in the SegNet (Badrinarayanan; Kendall; Cipolla, 2017), which also uses an enconder/decoder architecture. However, the decoder layers of the SegNet have a corresponding enconding layer.

Figure 8 – The U-Net architecture for a gray scale input image of size 572x572 and two classes on the output layer.



Source: Ronneberger, Fischer and Brox (2015).

One disadvantage of these networks is that they require thousands of annotated training samples. To overcome this problem, Ronneberger *et al.* (RONNEBERGER; FISCHER; BROX, 2015) proposed a simpler architecture, the U-Net. The U-Net is an encoder/decoder FCN able to deal with a smaller training set. On the U-Net, the decoder receives a copy of the output of the activation layers and concatenate with the upscaling tensor. In this way, U-Net can pass the spatial information lost in the encoder step to the corresponding decoder layers, improving the segmentation output. Figure 8 shows the architecture used by the U-Net model. The blue boxes represent the feature maps (tensors) and the size of the tensors are represented by the numbers on the lower left and on top of the box. The white boxes are the copied feature maps that are concatenated on its corresponding decoder layer. The arrows denote different operations, such as convolutions, poolings and up-convolutions (upscaling convolutions).

2.5 Related Works

As mentioned in Chapter 1, the focus of this PhD research is aimed at images in two distinct situations: urban emergency situations with fire and chronic skin ulcers. In this Section we will discuss some of the state-of-the-art methods in these application domains.

2.5.1 Fire Detection in Emergency Images

There are an extensive literature on forest and urban fire detection on videos (CELIK; DEMIREL, 2009; RUDZ *et al.*, 2013; AVALHAIS; RODRIGUES; TRAINA, 2016; BENJAMIN *et al.*, 2016; MUHAMMAD; AHMAD; BAIK, 2018). The majority of these methods are based only on the color aspects of the images, which can lead to the presence of false-positives in their output. To dismiss false-positives the authors uses the temporal aspects of the video. Since we are interested only on fire detection on still images, we will focus our analysis only on the image parts of the method.

A fire detection method based on rules was proposed in the work of Chen *et al.* (CHEN; WU; CHIOU, 2004). They defined a set of three rules using a combination of the RGB and the HSI color spaces; the user, in turn, must set two threshold parameters to detect fire pixels. Another method based on color was proposed by Celik *et al.* (CELIK; DEMIREL, 2009), who conducted a wide-ranging study regarding the color of fire pixels to define a model. This method defines a set of five mathematical rules based on the YCbCr color space; this was because the YCbCr has a better discrimination regarding fire (CELIK; DEMIREL, 2009; RUDZ *et al.*, 2013). These rules compare the intensity of the YCbCr channels and the user must define a threshold parameter.

Rossi *et al.* (ROSSI; AKHLOUFI; TISON, 2011) proposed a method to extract geometric fire characteristics using stereoscope videos. One of the steps is a segmentation based on a clustering algorithm, in which the image is divided into two clusters based on the channel V of the YUV color space. The cluster with the highest value of V corresponds to fire. Thereafter, Rossi *et al.* used a 3D-Gaussian model to classify pixels as fire. In this method, the accuracy of the classification depends on a parameter provided by the user. This method presents limitations, since the authors assume that the fire is registered in a controlled environment.

Rudz *et al.* (RUDZ *et al.*, 2013) proposed another method based on clustering. Instead of using the YUV color space, Rudz *et al.* computes four clusters using the blue chrominance Cb of the YCbCr color space. The cluster with the lowest value of Cb refers to a fire region. A second step eliminates false-positive pixels using a reference dataset. The method treats small and large regions with different approaches; small regions are compared with the mean value of a reference region, while large regions are compared to the reference histogram. This comparison is made for each RGB color channel. The user must set three constants for the small regions, and three thresholds for the large regions, resulting in a total of six parameters.

Benjamin *et al.* (BENJAMIN *et al.*, 2016) proposed a forest fire segmentation based on rules. Benjamin *et al.* used the RGB, YCbCr and HSV to create rules and improved the segmentation by extracting texture features from the co-occurrence matrix. Avalhais *et al.* (AVALHAIS; RODRIGUES; TRAINA, 2016) proposed the SPATFIRE to detect fire in hand-held device videos. SPATFIRE uses a color model based on the HSV color space and used motion flow to reduce the motion of the video. Deep learning techniques have been employed to detect fire on images and videos. Sharma *et al.* (SHARMA *et al.*, 2017) proposed an FCN segmentation based on the VGG16 and the ResNet50 for fire images. While Muhammad *et al.* (MUHAMMAD; AHMAD; BAIK, 2018) proposed an FCN segmentation based on the AlexNet for fire images and videos.

As mentioned earlier, these methods are only based on the color aspect of the images. Thus, when dealing with still images, they output a high rate of false-positives. Another downside of these methods is the presence of lots of parameters, which are very sensitive with respect to their tuning. Another problem of their parameters is the lack of physical significance. The majority of these parameters are based on color intensity or multiple empirical constants, making the fine tuning of their methods very troublesome.

2.5.2 Venous Skin Ulcers

To the best of our knowledge, there are few works that deal with skin ulcer images. We will first discuss skin ulcer segmentation methods. Dorileo *et al.* (DORILEO *et al.*, 2010) proposed an image segmentation method. Its segmentation is based on the analysis of the RGB channels of the image. Dorileo *et al.* took advantage of the controlled environment of the images to process the images. Since all images had a blue background, they discarded the blue channel and also used the intensity channel of the HSI (hue, saturation, intensity) color space. Each channel is used to find a type of tissue: fibrin, granulation and necrotic. For each channel, the method automatically finds thresholds and process the discovered regions by focusing on blobs near the center of the image. One problem of this method is the need for a controlled environment. Another skin ulcer segmentation method was proposed by Seixas *et al.* (SEIXAS; BARBON; MANTOVANI, 2015). Seixas *et al.* employed off-the-shelf classifiers to segment ulcer images. They extracted pixel-wise color features, the mean value of the neighborhood of the pixel, and the difference of the pixel value and the mean beforehand mentioned. They manually segmented a training set of images to isolate the wound region.

There are also works that use CBIR systems on skin ulcer images. Dorileo *et al.* (DORILEO *et al.*, 2008) proposed a CBIR system for skin ulcer images. The images were manually segmented in two regions, the lesion and the background. For each region, the images were decomposed in 5 gray scale images, the RGB channels, and two images based on hue and saturation. For each channel image were extracted Haralick texture features.

Pereyra *et al.* (PEREYRA *et al.*, 2014) proposed a CBIR method using only the lesion regions of the skin ulcer images. As a processing step, Pereyra *et al.* proposed a segmentation step based on a multivariate gaussian mixture mode. The clusters were manually selected in a Graphical User Interface (GUI) to output the segmentation mask. Using this masks, Pereyra *et al.* extracted color and texture features only on the lesion regions of the image. Pereyra *et al.* used the average of each channel of the RGB, HSI, Luv, and Lab color spaces as color features and extracted Haralick texture features. Bedo et al. (BEDO *et al.*, 2015a) used the same segmentation step based on mixed gaussians and concatenated the Color Layout, Color Structure, Scalable Color, Edge Histogram, Texture-Spectrum and Haralick features. A feature selection step was needed to reduce the dimensionality. Both Pereyra *et al.* and Bedo *et al.* approaches segment the lesion regions and extract features from the whole segmented image.

Since a skin ulcer may have more than one class at a time, it is interesting to separate the regions of the image. To explore this possibility, Blanco et al. (BLANCO *et al.*, 2016) proposed the Counting-Labels Similarity Measure (CL-Measure) to compare the images. CL-Measure segments the image into superpixels, then extracts color and shape features, and finally classifies the tissue of each superpixel using supervised learning algorithms. CL-Measure compares the images according to the labels of their superpixel. CL-Measure calculate the similarity of each tissue severity (fibrin, granulation and necrotic) and weights them accordingly to their area in the image. However, its similarity measure has a high computational cost, since its based on the similarity Jaccard, which has a quadratic complexity. Although the results are promising, their proposed distance measure with the highest precision is not metric, and is computationally costly for high-resolution images, since it extracts fixed-size superpixels.

2.6 Final Thoughts

The goal of this Chapter was to present the basic background and related works that are relevant to this PhD research. We discussed the basic concepts of CBIR and BoVW. Then, we discussed ways to improve CBIR system by using image segmentation methods. We also presented some deep learning methods for feature extraction and image segmentation. Finally, we presented the state-of-the-art methods in the domains of this PhD research.

It is important to note that the literature covered in this chapter is quite broad in the subjects presented. It was not the author's intent to exhaustively discuss all of these subjects. Rather, the objective was to present the reader the required background and knowledge to understand the contributions of this PhD research.

CHAPTER

3

FIRE DETECTION IN URBAN SCENARIO

In this Chapter we explore the use of superpixel techniques to detect fire on the context of emergency situations. We also explore the use of color and texture features to improve precision and reduce the false positive rate of fire segmentation. The organization of this Chapter is as follows. We give a brief introduction on the problem of fire detection on Section 3.1. Section 3.2 introduces the Best of both Worlds Fire detection (BoWFire) and Section 3.3 shows its results. Finally, Section 3.4 show our final thoughts on the fire detection problem. This Chapter is based on the work presented in the 28th Conference on Graphics, Patterns and Images (SIBGRAPI2015) (CHINO *et al.*, 2015).

3.1 Introduction

Emergency situations can cause economic losses, environmental disasters or serious damage to human life. In particular, accidents involving fire and explosion, have attracted interest to the development of automatic fire detection systems. Existing solutions are based on ultraviolet and infrared sensors, and usually explore the chemical properties of fire and smoke in particle samplings (CHEN; WU; CHIOU, 2004). However, the main constraint of these solutions is that sensors must be set near to the fire source, which brings complexity and cost of installation and maintenance, especially in large open areas. Alternative to sensors, cameras can provide visual information of wider spaces, and have been increasingly embedded in a variety of portable devices such as smartphones.

Several methods regarding to fire detection on videos have been proposed in the last years. These methods use two steps to detect fire. First, they explore the visual features extracted from the video frames (images); second, they take advantage of the motion and other temporal features of the videos (KIM; JEONG, 2014). In the first step, the general approach is to create a mathematical/rule-based model, defining a sub-space on the color space that represents all the fire-colored pixels in the image. There are several empirical models using different color spaces as RGB (CHEN; WU; CHIOU, 2004), YCbCr (CELIK; DEMIREL, 2009), CIE Lab (HA *et al.*, 2012) and HSV (ZHAO *et al.*, 2011). In these cases, the limitation is the lack of correspondence of these models to fire properties beyond color. The problem is that high illumination value or reddish-yellowish objects lead to a higher false-positive rate. These false-positives are usually eliminated on the second step through temporal analysis.

In contrast to such methods, our proposal is to detect fire in still images, without any further (temporal) information, using only visual features extracted from the images. To overcome the problems aforementioned, we propose a new method to detect fire in still images that is based on the combination of two approaches: pixel-color classification and texture classification. The use of color is a traditional approach to the problem; whilst, the use of texture is promising, because fire traces present particular textures that permit to distinguish between actual fire and fire-like regions. We show that, even with just the information present in the images, it is possible to achieve a high accuracy level in such detection.

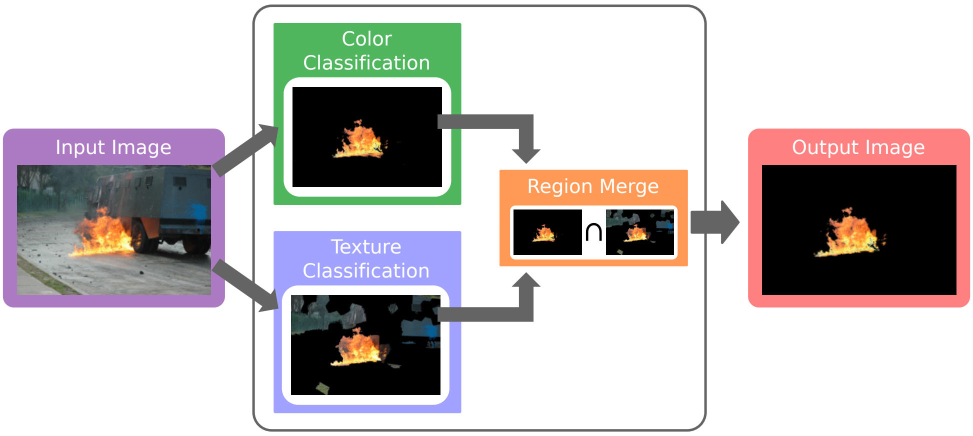
The main contribution of this research is the proposal of BoWFire, a novel method to detect fire in still images. By merging color and texture information, our method showed to be effective in detecting true-positive regions of fire in real-scenario images, while discarding a considerable quantity of false-positives. Our method uses fewer parameters than former works, what leads to a more intuitive process of fine tuning the automated detection. Regarding these claims, in the experiments, we systematically compare *BoWFire* with four works that currently define the state-of-the-art, that is, the works of Celik *et al.* (CELIK; DEMIREL, 2009), Chen *et al.* (CHEN; WU; CHIOU, 2004), Rossi *et al.* (ROSSI; AKHLOUFI; TISON, 2011), and Rudz *et al.* (RUDZ *et al.*, 2013).

3.2 Best of both Worlds Fire detection

We propose BoWFire, a novel method for fire detection in emergency-situation images. We explore the fact that color combined with texture can improve the detection of fire, reducing the number of false-positives as compared to related works from the literature. We show that such combination can distinguish actual fire from fire-like regions (reddish/yellowish) of a given image. The goal is to provide a more effective automated detection of fire scenes in the context of the crisis situations, as those of the *RESCUER* Project. Figure 9 shows the basic architecture of our proposal. The *BoWFire* method consists of three basic steps: *Color Classification*, *Texture Classification*, and *Region Merge*. As shown in Figure 9, the two first steps occur in parallel to produce images in which fire-classified pixels are marked. Then, the output from both classifications is merged into a single output image by the *Region Merge* step.

Different from other methods, usually based on mathematical models, the use of a Color Classification step avoids the need of a great number of parameters. Any machine learning classification algorithm could be used, specifically, in this work, we use Naive-Bayes and *KNN*.

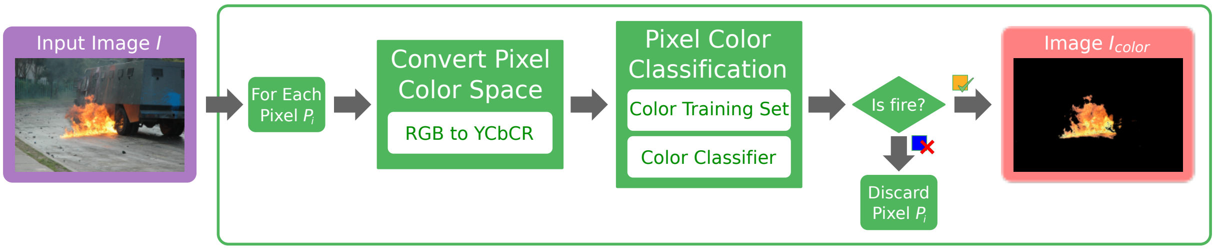
Figure 9 – Architecture of the *BoWFire* method.



Source: Adapted from Chino *et al.* (2015).

By doing so, we also avoid the use of the global information of the image to classify only one pixel as opposed to other approaches; this is a desired feature because the semantics of the image may vary according to the emergency situation (small/large fire regions or day/night time). Figure 10 presents more details of the color-based classification. Given an image I with n pixels P_i, $0 \leq i < n$. Each pixel $P_i = (R_i, G_i, B_i)$ of the image is converted to $P_i' = (Y_i, Cb_i, Cr_i)$ in the YCbCr color space, since this color space provides a better discrimination of fire regions. Then P_i' goes through a *Pixel-Color Classification* (*pixelClass*), which consists of a *Color Training Set* and a *Color Classifier*. Then, if $pixelClass(P_i') = \langle \text{fire} \rangle$, P_i is used to build the output image I_{color}, otherwise P_i is discarded.

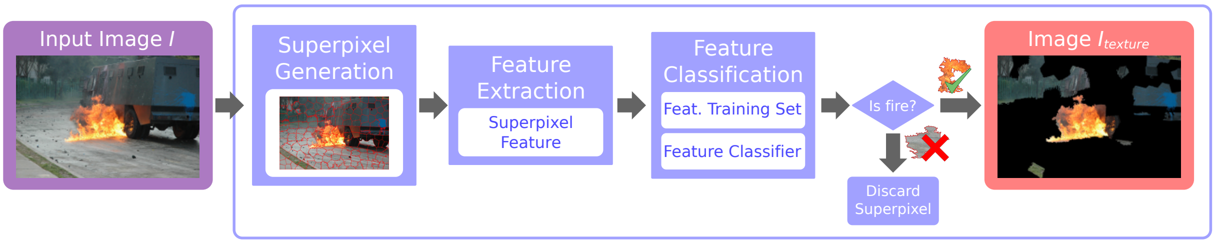
Figure 10 – Color-based classification step.



Source: Adapted from Chino *et al.* (2015).

As mentioned earlier, the *Texture Classification* step allows for a more accurate detection; however, it brings a challenge. Since there may be a variety of fire images according to the emergency situation, it is not possible to extract global features of the image because the small fire regions would vanish in the global context. Therefore, we extract only local features from regular shaped regions with similar patterns automatically detected by superpixel methods. Figure 11 presents details of the *Texture Classification* step. Given the same image I, we use a superpixel method $extractSuperPixels(I, K_{sp})$ to generate a set of K_{sp} superpixels Sp_j, where $0 \leq j < K_{sp}$. Next, each superpixel Sp_j passes through a local *Feature Extraction* process, resulting in a feature vector $V_j = (v_{j0}, \ldots, v_{j(d-1)})$ of size d. Then, V_j is classified using a *Feature Classification* ($featClass$), which consists of a *Feature Training Set* and a *Feature Classifier*. If $featClass(V_j) = \langle \text{fire} \rangle$, all pixels $P_i \in Sp_j$ are used to build the output image $I_{texture}$, otherwise they are discarded. After this, the superpixel region is no longer necessary since the method is performed in pixel-level only.

Figure 11 – Texture-based classification step.



Source: Adapted from Chino *et al.* (2015).

With the outputs from the *Color Classification* (image I_{color}), and from the *Texture Classification* (image $I_{texture}$), it is still necessary to join the results in an output image $I_{classified}$. We perform this task in the *Region Merge* step. According to our hypothesis, if a pixel is simultaneously classified as fire following color and texture classification, then there is a higher chance that this pixel is actual fire. Therefore, given an image I and its color and texture classifications I_{color} and $I_{texture}$, the final classified image $I_{classified}$ is defined as $I_{classified} = \{P_i | P_i \in I_{color} \text{ and } P_i \in I_{texture}\}$. That is, a given pixel is added to the final output only if it was detected in both color and texture classifications, otherwise it is discarded. Consequently, we dismiss false-positives from both approaches, taking advantage of the best of both worlds. Algorithm 1 shows the algorithm used by BoWFire.

The *BoWFire* method was developed in a modularized scheme, allowing an easy way to add and set different feature extraction algorithms, as well as different classifiers. We note that, since the *BoWFire* method is fully customizable, the number of parameters is dependent only on the algorithms used in the intermediate steps.

3.3 Experiments

In this section we show the performance of BoWFire to segment fire regions in emergency images. In this section, we present the results of three experiments: (i) the impact of parameter K_{sp}; (ii) the *Color Classification* Evaluation; and (iii) the BoWFire Evaluation. We implemented BoWFire in C/C++11 and all experiments were carried out on a 3.40GHz Intel Core i7-4770 CPU with 16GB RAM and a NVIDIA GeForce GTX 645 with 1GB GDDR5, running Ubuntu 14.04.

To reduce the number of parameters, we used the following algorithms for the BoWFire

Algorithm 1 – BoWFire method

```
Input: Image $I$, $K_{sp}$: number of superpixels
Output: Image $I_{classified}$: Mask with fire regions
$I_{color} \leftarrow \emptyset$
$I_{texture} \leftarrow \emptyset$
$I_{classified} \leftarrow \emptyset$
for all $P_i \in I$ do
    $P_i' \leftarrow RGB2YCbCr(P_i)$
    if $pixelClass(P_i')$ is ⟨fire⟩ then
        Add $P_i$ to $I_{color}$
    end if
end for
$SP \leftarrow extractSuperPixels(I, K_{sp})$
for all $Sp_j \in SP$ do
    $V_j \leftarrow extractTextureFeature(Sp_j)$
    if $featClass(V_j)$ is ⟨fire⟩ then
        for all $P_i \in Sp_j$ do
            Add $P_i$ to $I_{texture}$
        end for
    end if
end for
for all $P_i \in I$ do
    if $P_i \in I_{color}$ and $P_i \in I_{texture}$ then
        Add $P_i$ to $I_{classified}$
    end if
end for
```

intermediate steps. The *Pixel-Color Classification* is done by a Naive-Bayes classifier, using an automatic discretization method; the superpixel algorithm was the SLIC method with a modification. Instead of using the Lab color space, we used the YCbCr space due to its discriminative property. Since we wanted to add texture information, our implementation uses the *uniform patterns* LBP. The features were classified using the *k-NN* classification with the Manhattan Distance.

Considering the configuration given by the choice of intermediate algorithms, the BoWFire method needs only 3 parameters: K_{sp}, m and K. For all experiments we empirically evaluated the best values for parameters m and K; for parameter K, we used the value $K = 11$. Regarding to parameter m, we observed that a more compact superpixel generates a more regular region, which leads to a better representation of the texture feature. In this case, the best value was $m = 40$. With these parameters, each method was executed on three different datasets: only fire images, only non-fire images, and a complete dataset with both fire and non-fire. For each execution, we computed the confusion matrix for the classification of all pixels and calculated four measures: Precision (Equation 3.1), Recall (Equation 3.2), F1-Score (Equation 3.4), and

False-Positive Rate (FPR) (Equation 3.3).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.2}$$

$$\text{FPR} = \frac{FP}{FP+TN} \tag{3.3}$$

where TP, FP, TN and FN stand for true positive, false positive, true negative and false negative respectively.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

3.3.1 The BoWFire dataset

We performed experiments using a dataset of fire images. Since at the time we were proposing the BoWFire there was no urban fire dataset, we proposed the BoWFire-dataset. The BoWFire-dataset consists of 226 images of emergency situations with fire in urban scenarios with various resolutions[1]. The images were collect from Flickr through a crawler using Flickr API[2], in August of 2014. All images were downloaded under the Creative Commons license. The crawler used the textual keywords shown in Table 2.

Table 2 – Keywords used by the crawler to collect the images.

Keywords				
fire	smoke	emergency	flames	burning
protest	boston marathon	car fire accident	criminal fire	fire department
firefighter	urban fire	house burning	criminal fire	fire car accident

Source: Research data.

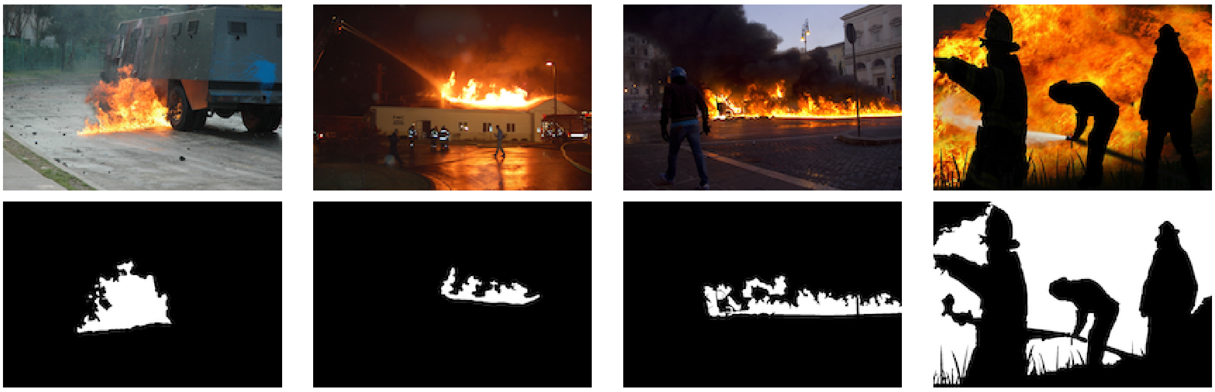
The BoWFire-dataset was divided in two categories: 119 images containing fire, and 107 images without fire. The fire images consist of emergency situations with different fire incidents, as buildings on fire, industrial fire, car accidents, and riots. These images were manually cropped by human experts. The remaining images consist of emergency situations with no visible fire and also images with fire-like regions, such as sunsets, and red or yellow objects. Figures 12 and 13 show some samples of this dataset. Since we are using supervised machine learning, we also created a training dataset. The training dataset consists of 240 images of 50×50 pixels

[1] Available at <http://chinodyt.github.io/>

[2] <https://www.flickr.com/services/api/>

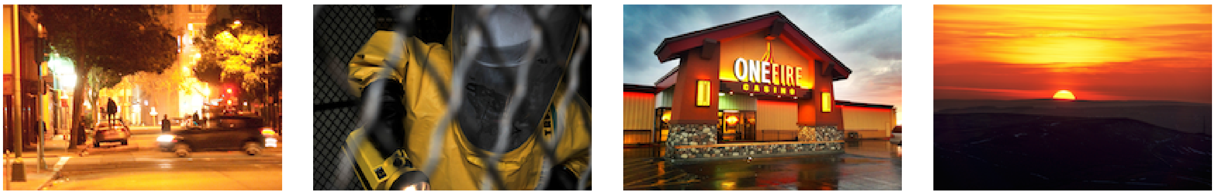
resolution; 80 images classified as fire, and 160 as non-fire. Figure 14 shows some samples of this dataset. It is important to note that the non-fire images also contain red or yellow objects. The training dataset was used for both classification steps, *Pixel Color Classification* and *Feature Classification*.

Figure 12 – Examples of image containing fire emergencies. The first lines are images containing fire and the second line is the ground truth.



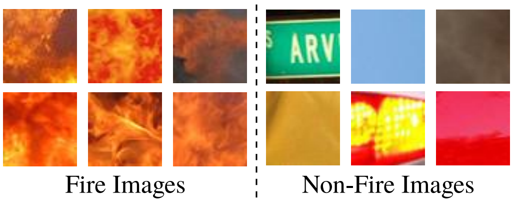
Source: Elaborated by the author.

Figure 13 – Examples of non emergency image containing fire like colors.



Source: Elaborated by the author.

Figure 14 – Sample images of the training dataset.

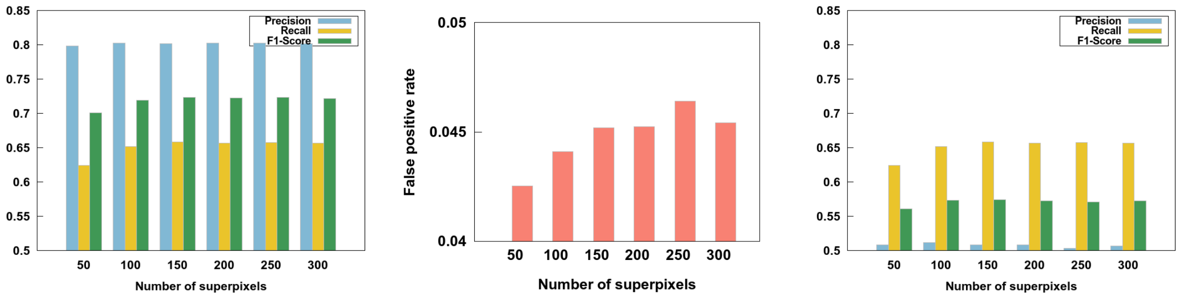


Source: Chino *et al.* (2015).

3.3.2 Impact of K_{sp}

The first experiment evaluates the impact of the number of superpixels on the BoWFire performance. We vary the number of superpixels K_{sp} with the following values: 50, 100, 150, 200, 250 and 300. Figure 15 shows the results obtained while varying the number of superpixels. Figure 15(a) shows the results for the fire dataset. In this case, there was a slight increase of all measure until $K_{sp} = 150$, then for greater values they had a similar behavior. The Precision obtained was around 0.8, Recall around 0.65 and F1-Score around 0.72. Figure 15(b) shows the results for the non-fire dataset. For this dataset we computed only the False-Positive Rate. There is a slight increasing of FPR as the number of superpixels increases, except for $K_{sp} = 300$. It is important to notice that although FPR increases, the values remain around 0.045, that is, less than 5% of false-positives. And Figure 15(c) shows the results combining both datasets. Again, there is a similar behavior regarding K_{sp}, except for $K_{sp} = 50$. The Precision obtained was around 0.5, Recall around 0.65 and F1-Score around 0.57. The FPR values were not shown on both Figures 15(a) and 15(c) due to their low values for all K_{sp}. There is also a slight increasing of FPR as K_{sp} increases, but with lower values. On the fire dataset, FPR went from 0.0169 to 0.0175, and on the complete dataset it varied from 0.0305 to 0.0323.

Figure 15 – Impact evaluation of the number of superPixel K_{sp} in three different datasets.



(a) Fire dataset (b) Non-fire dataset (c) Complete dataset

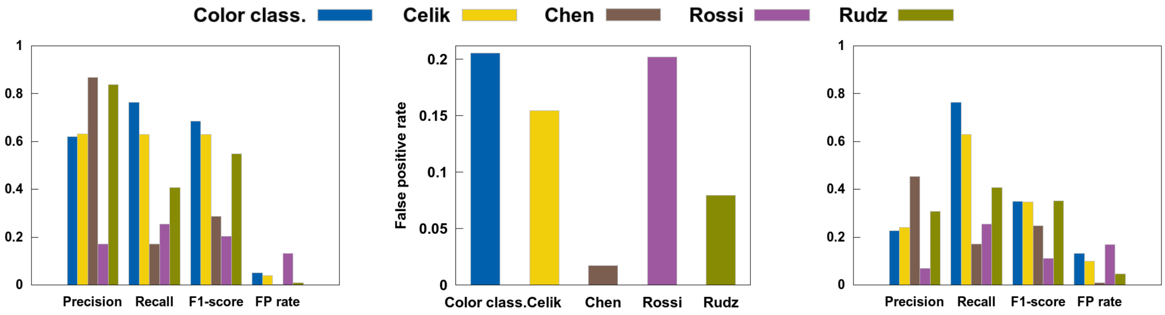
Source: Chino *et al.* (2015).

The main goal of the BoWFire is to decrease the FPR while maintaining a good performance. With that in mind, we evaluated that the best result is achieved when the number of superpixels $K_{sp} = 150$. This number presented better results while dealing with just the fire and complete dataset (fire and non-fire), as showed by F1-score. Also, the value of FPR for this K_{sp} is close to the lowest FPR value.

3.3.3 Color Classification Evaluation

In this experiment, we aim at evaluating the capability of the *Color Classification* step proposed in this paper. Since BoWFire is based on a combination of two different approaches, it is important that the color-based method recovers as many fire pixels as possible. So, Recall is the measure that closely meets this need. Also, on this step FPR is not so important, since it will be handled on the *Texture Classification* step. We evaluated the behavior of our proposed *Color Classification* in comparison with the state-of-the-art, as in the works of Celik (CELIK; DEMIREL, 2009), Chen (CHEN; WU; CHIOU, 2004), Rossi (ROSSI; AKHLOUFI; TISON, 2011) and Rudz (RUDZ *et al.*, 2013).

Figure 16 – Evaluation of the Color Classification method with the state-of-the-art methods.



(a) Fire dataset

(b) Non-fire dataset

(c) Complete dataset

Source: Adapted from Chino *et al.* (2015).

Figure 16 shows the results for the *Color Classification* step. Considering only color-based approaches, *Color Classification*, Celik and Rudz presented the best overall performance. Although Chen obtained the highest value of Precision, its Recall got the lowest value. As seen in Figures 17 and 18, Chen missed too many true-positive pixels and Rossi has the lowest overall performance. We observed that in outdoor emergency situations, fire was not in the cluster with the higher values of V, as shown in Figures 17 and 18. From now on, we will focus our analysis only on the methods with the best overall performance.

Regarding to Precision, Rudz achieved the best value, 0.84 on the fire dataset and 0.31 on the complete dataset, while *Color Classification* and Celik had similar behavior with values around 0.62 on the fire dataset and 0.24 on the complete. On the other hand, *Color Classification* achieved the highest value of Recall, 0.77 on both fire and complete dataset, followed by Celik, 0.63 on both fire and complete, and Rudz, 0.41 on both fire and complete. Analyzing the F1-Score, *Color Classification* and Celik methods outperformed Rudz by at most 23.6% on the fire dataset with values of 0.68 to *Color Classification*, 0.63 to Celik and 0.55 to Rudz. On the complete dataset, all methods achieved similar F1-Score with the value of 0.35.

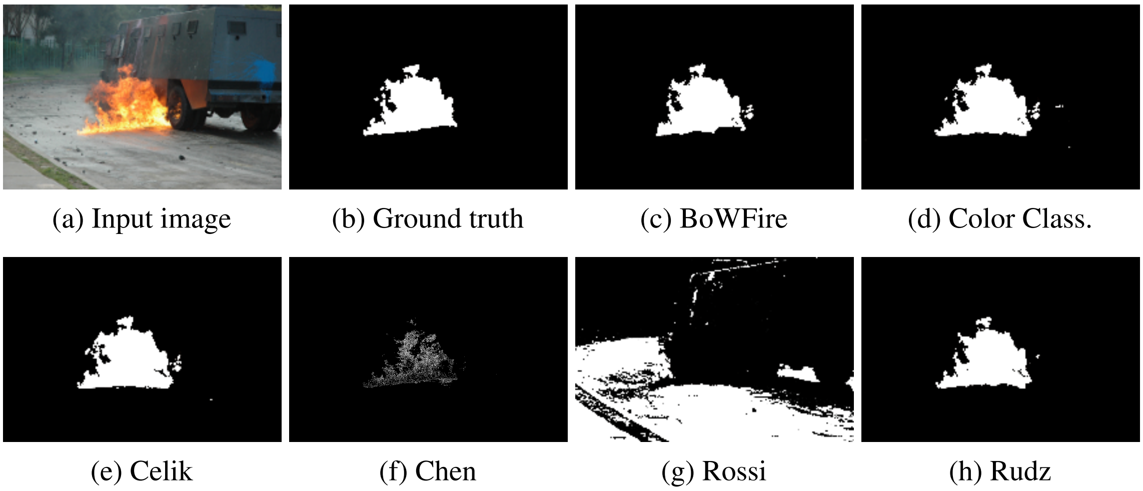
On the fire dataset, *Color Classification* and Celik achieved similar values of FPR (0.05 and 0.04) and Rudz method achieved 0.01 FPR. On the non-fire dataset, *Color Classification*, Celik and Rudz achieved respectively 0.21, 0.15 and 0.08. And on the complete dataset, *Color Classification*, Celik and Rudz methods achieved respectively 0.13, 0.10 and 0.05. On all datasets, Rudz achieved the best FRP value, less than 9% of the pixels was incorrectly classified. However, while discarding more false-positives, Rudz also discarded true-positives, reducing its Recall capability. Except for FPR, *Color Classification* had a similar behavior of Celik, but had better values of Recall and F1-Score. Therefore, the *Color Classification* outperformed the other methods.

3.3.4 BoWFire Evaluation

After evaluating only color, we evaluate the impact of considering texture together with color, as defined in our proposal. The most important aspect of this step is to reduce false-positives without affecting the overall performance. In this context, we analyze the BoWFire performance, which is the combination of the *Color Classification* step with the *Texture Classification* step. We also evaluated the performance of the state-of-the-art methods combined with the *Texture Classification*. We used the best value of K_{sp} as obtained in the experimentation.

Figures 17, 18 and 19 show visual samples of output images from three different situations. Figure 17 shows an emergency situation with fire and low percentage of possible false-positives. On this input image it is possible to note that *Color Classification*, Celik and Rudz methods had similar outputs. The BoWFire method was able to detect the same fire region as these methods, but discarded the fire reflection on the ground. Rossi was not able to correctly detect fire regions, while Chen discarded more than half of the true-positives. Figure 18 also shows an emergency situation with fire with a higher percentage of false-positives. In this case, all methods detected false-positives, with the exception of BoWFire. It is also possible to note that in some cases, Rudz discards more fire pixels than necessary. This image also shows the problem with the Rossi method, since no fire region was detected as fire. Once again, Chen discarded almost every true-positive. Figure 19 shows a sunset skyline image. For this input image, excluding BoWFire, all methods detected a high rate of false-positives. Chen had a lower rate of false-positives, however, as seen in the previous examples, it also has the same behavior with yellowish fire regions. Meanwhile, when adding texture information to the *Color Classification*, BoWFire was capable of discarding all false-positives for this image.

Figure 17 – Output from the methods with an input image with fire.



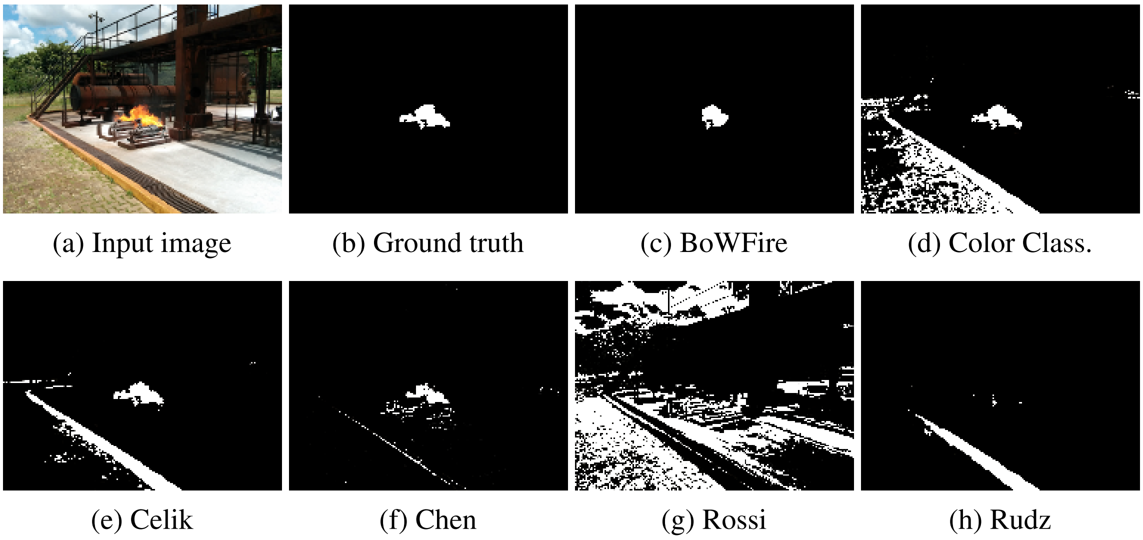
(a) Input image (b) Ground truth (c) BoWFire (d) Color Class.

(e) Celik (f) Chen (g) Rossi (h) Rudz

Source: Chino *et al.* (2015).

Figure 20 shows the results when added texture information. It is possible to note that there was an overall improvement for all methods. Regarding Precision, with the exception of

Figure 18 – Output from the methods with an input image with fire and possible false-positives pixels.

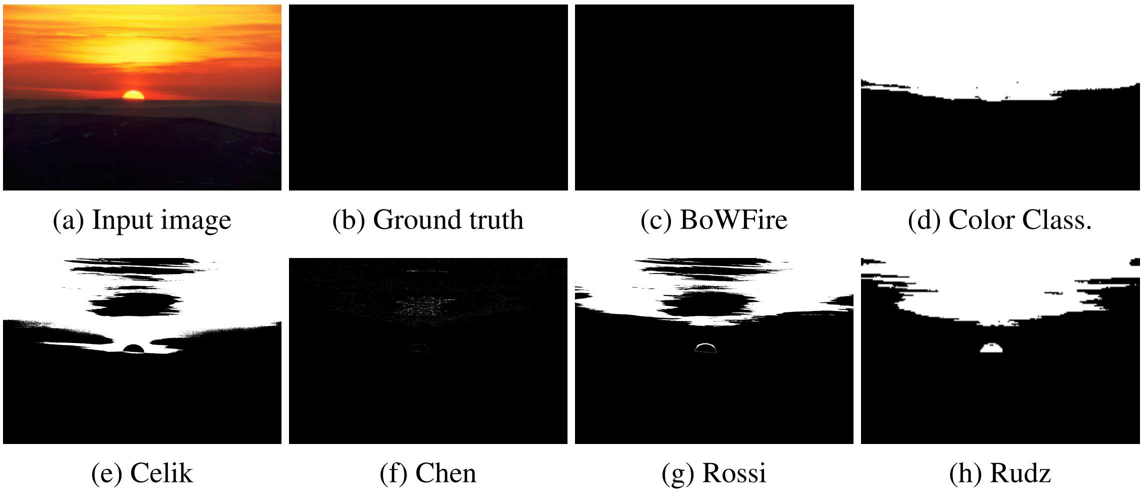


(a) Input image (b) Ground truth (c) BoWFire (d) Color Class.

(e) Celik (f) Chen (g) Rossi (h) Rudz

Source: Chino *et al.* (2015).

Figure 19 – Output of a non-fire image.



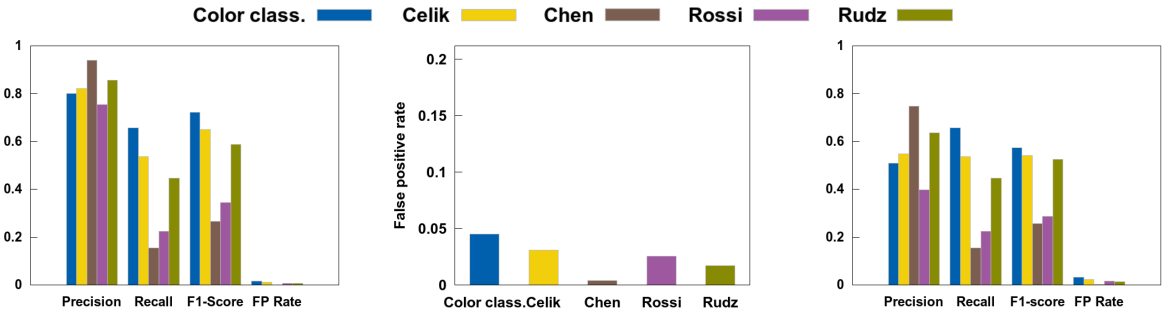
(a) Input image (b) Ground truth (c) BoWFire (d) Color Class.

(e) Celik (f) Chen (g) Rossi (h) Rudz

Source: Adapted from Chino *et al.* (2015).

Rudz and Chen, all methods had a considerable improvement. *Color Classification* and Celik had a Precision improvement of up to 1.30 times on the fire dataset and 2.28 times on the complete dataset. Rossi had the greatest improvement, 4.43 times on fire dataset and 5.65 times on the complete dataset. This high improvement was due to the fact that Rossi, on outdoor images, detected other regions than fire, as shown on Figures 17 and 18. When adding texture information, these false-positive regions were discarded. For both Chen and Rudz, there was a slightly improvement on the fire dataset, but it is due to the fact that they already had low FPR. On the other hand, on the complete dataset, there was an improvement of 1.64 times to Chen and 2.06 times to Rudz.

Figure 20 – Evaluation of the BoWFire method with the state-of-the-art methods.



(a) Fire dataset (b) Non-fire dataset (c) Complete dataset

Source: Adapted from Chino *et al.* (2015).

There was a decreasing on the Recall value of up to 15% less for all methods, except Rudz. This is due to the fact that the combination of both approaches discarded a few true-positives. However, the considerable gain on the precision can justify this drawback. Analyzing the F1-Score, there was a slight increase of up to 7% for all methods, except Rossi, on the fire dataset, which had an improvement of 69%. On the complete dataset, all methods had a considerable improvement, up to 65%.

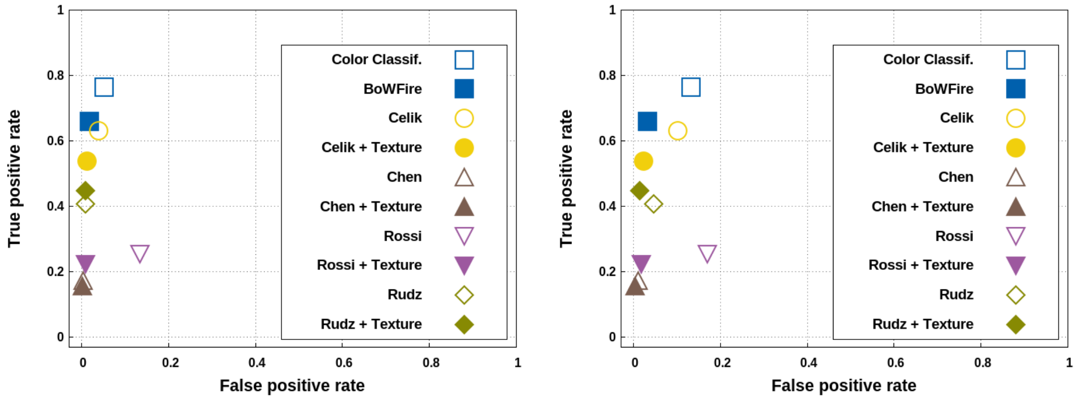
As one of the goals of BoWFire is to reduce the number of false-positives, it is important to analyze FPR. On the fire dataset, there was a reduction of up to 68% of FPR for *Color Classification*, using Celik and Chen methods. Rossi had 94% less false-positives. Rudz was the least affected by this step, reducing 5% false-positives, since they had already dismissed false-positives on a post processing step. On both the non-fire and complete dataset, all methods reduced FPR by up to 80%. This result confirms that the *Texture Classification* step is capable of discarding false-positives without compromising the overall performance.

We can now use the Receiver Operating Characteristic (ROC) space to analyze the performance behavior between all methods. The ROC Space shows the relation between FPR and the true-positive rate (Recall). Figure 21 shows the ROC Space on the fire and the complete datasets for all methods. On both ROC Spaces, it is possible to note that all methods move to the left, i.e., achieve less FPR when texture information is added. The *Color Classification* and the BoWFire method presented the best classification results among the other methods, followed by Celik. Also, the BoWFire achieved a similar Recall value as Celik without texture information, but with a smaller FPR.

3.4 Final Thoughts

In this Chapter, we introduced the BoWFire method, a novel approach for fire detection on images in emergency context. Our results showed that BoWFire was capable of detecting fire with a performance similar to what is observed in the works of the state-of-the-art, but with less

Figure 21 – ROC Space of all methods.



(a) Fire dataset (b) Complete dataset

Source: Adapted from Chino *et al.* (2015).

false-positives. We systematically compared our work with four former works, demonstrating that we achieved consistent improvements.

The course of action of BoWFire was that, by simultaneously using color and texture information, it was able to dismiss false-positives relying solely on the information present in the images; as opposed to former methods that use temporal information. Furthermore, since BoWFire is based on classification methods, rather than on mathematical modeling, it was able to solve the problem with only three parameters. In addition, these parameters were more intuitive for tuning, unlike those of previous works, which are based on thresholds and color-based values. Given its performance, we conclude that BoWFire is suitable to integrate a crisis management system as the one that motivates this work.

Another contribution present in this Chapter was the proposal of the BoWFire-dataset. The BoWFire-dataset is an image dataset of urban fire images aimed at fire segmentation problems. One important aspect of the BoWFire-dataset is that its focus is to measure how well a fire segmentation method is able to avoid false-positive rates.

CHAPTER

4

BAG OF SUPERPIXELS SIGNATURES

In this Chapter we introduce a Signature-Based Bag-of-Visual-Words (S-BoVW) technique based on superpixels. S-BoVW introduces the visual signature concept, which skips the visual dictionary building step of BoVW approaches. We also explore the Fractal Theory analysis to estimate parameters of our proposal. The organization of this Chapter is as follows. We give a brief introduction on Bag-of-Visual-Words on Section 4.1. Section 4.2 introduces some concepts needed for the understanding of this Chapter. On Section 4.3, we explore the image datasets used in this Chapter to find useful patterns. Section 4.4 introduces the Bag-of-Superpixels Signatures (BoSS) and Section 4.5 shows its results. Finally, Section 4.6 concludes this Chapter. This Chapter is based on works published in the 33rd ACM/SIGAPP Symposium On Applied Computing(CHINO *et al.*, 2018).

4.1 Introduction

Advances in technology, such as gadgets and cell phones, enabled not only a massive capture and storage of images and videos, but also the sharing of these complex data through social media (SANTOS *et al.*, 2017). Due to the increasing omnipresence of such data, performing decision-making in a timely manner and image retrieval tasks has been a challenge (BEDO *et al.*, 2015a). Computer systems can support those tasks with the CBIR approach. There are many techniques to extract an image description and to compare it with other images. One of them is by using the BoVW approach.

BoVW is an extension of Bag-of-Words techniques from the textual domain applied to the images domain. BoVW techniques represent an image as a set of visual words, which are extracted from the image's local features. BoVW techniques are widely employed in image retrieval over large databases (CAETANO *et al.*, 2014; SIVIC; ZISSERMAN, 2003). There are three main reasons to adopt this representation (JEGOU *et al.*, 2012): (i) these techniques benefit from enabling the use of robust local image descriptors, such as SIFT or SURF; (ii) the comparisons among the images can be performed with standard distance functions; and (iii) it can deal with high dimensional vectors, where words can be indexed with inverted indexes to perform an efficient search.

Since BoVW uses local features extraction techniques, there must be a way to map the numerical features into visual words. This can be done by using a clustering process to generate the visual words (SIVIC; ZISSERMAN, 2003) – C-BoVW. However, it is also possible to map the local features using visual signatures (SANTOS *et al.*, 2015; SANTOS *et al.*, 2017). In comparison to C-BoVW, the latter approach – S-BoVW – enables the identification of visual signatures at a low cost. The two major drawbacks of the existing S-BoVW techniques are that they only employ fixed squared regions and require some unintuitive parameters to tune the algorithms.

In this Chapter, we introduce the BoSS approach, which was designed to overcome the afore mentioned drawbacks, by including superpixels in the existing S-BoVW techniques. The superpixels enable the adjustment of region's boundaries in an image, allowing the extraction of visual signatures from flexible and more meaningful regions. Moreover, we applied statistical analysis using the Zipf and power laws, as well as concepts from the Fractal theory (SCHROEDER, 2012) to drastically reduce the required parameters of the existing S-BoVW techniques. The main contributions of our approach are:

- **Self-contained:** we propose a visual signature extraction method, which does not demand pre-computed knowledge, such as visual dictionaries.
- **Intuitive parameter:** BoSS is designed to have as few parameters as possible. We only need to set the expected number of visual signatures to be extracted.
- **Scalability:** we propose a scalable algorithm for extracting visual signatures.
- **Effectiveness:** we show that the visual signatures extracted using BoSS retrieved images successfully, being up to 12.46% better than the state-of-the-art.

4.2 Basic Concepts

In this section, we present some concepts needed for the understanding of this Chapter.

Power Law and Zipf Law: Power law distributions allow explaining data behaviors and can be often observed in computer and social sciences (DEVINENI *et al.*, 2015). The Zipf distribution is a particular type of power law commonly used in text analysis (ZENG *et al.*, 2012). Zipf distribution is based on the Zipf's law, which states that the frequency of any word is inversely proportional to its rank. Yang *et al.* (YANG *et al.*, 2007) showed that the distribution of visual words in BoVW approximately follows a Zipf distribution. Moreover, the distribution of words is usually described by its frequency, with the exception of rare words (which produces tiny clusters).

Fractal Theory: A fractal is an object that presents similar characteristics when analyzed in different resolutions, *i.e.*, they are self-similar (SCHROEDER, 2012). Fractals can be found in geometric shapes, such as the Sierpinski triangle, as well as in nature, like shapes of mountains and clouds. Data analysis tasks can take advantage of the fractal theory, since real datasets have also shown to exhibit fractal behavior, since many times the datasets present the self-similarity property (FRAIDEINBERZE; RODRIGUES; CORDEIRO, 2016). Fractal theory has being used to feature selection (ZHANG *et al.*, 2016), clustering (BARBARA; CHEN, 2009) and data stream analysis (ZHANG *et al.*, 2015). When applied to data analysis, one important concept in fractal theory is the intrinsic dimension. The intrinsic dimension provides the minimum number of attributes needed to represent a point in a given dataset, regardless of the number of attributes present in the data, *i.e.*, embedded dimension (TRAINA *et al.*, 2010). The intrinsic dimension can be approximated by the Correlation Fractal Dimension D_2, which can be calculated with linear complexity on the data size by the box-counting approach (TRAINA *et al.*, 2010).

4.3 Patterns in Local Features

The existing S-BoVW approaches require a predefined threshold to define a signature for a local color histogram. In this section, we discuss the patterns we found on local color histograms and how it can be used on our proposal. We analyzed five image datasets commonly used on image retrieval, which are described in detail as follows:

Corel1000 (`Corel`) (WANG; LI; WIEDERHOLD, 2001): A dataset of 1,000 images of the Corel stock photo[1], uniformly divided into 10 classes.

Caltech Buildings (`Caltech`) (ALY *et al.*, 2009): A dataset of 250 buildings images around Caltech[2]. Each one of the 50 buildings is considered as a class, whereas there are 5 images taken from different angles and distances.

Flickr-Fire (`Flickr-Fire`) (BEDO *et al.*, 2015a): A dataset with 1,984 images related to fire emergency situations. The images were divided into two classes: 984 pictures containing fire and 1,000 without it.

INRIA Holidays (`Inria`) (JEGOU; DOUZE; SCHMID, 2008): A dataset of 1,491 images taken of personal holiday photos[3]. The authors proposed 500 images classes, with one representative image for each class and a list of the images retrieved. Each image class

[1] <http://wang.ist.psu.edu/docs/related/>

[2] <http://www.vision.caltech.edu/malaa/datasets/caltech-buildings/>

[3] <http://lear.inrialpes.fr/~jegou/data.php#holidays>

represents a distinct scenario or object that include a variety of types, such as natural scenes, man-made items and fire effects.

Describable Textures (`Texture`) (LAZEBNIK; SCHMID; PONCE, 2005): This dataset consists of 1,000 grayscale images of textures[4]. There are a total of 25 classes with exactly 40 samples each one.

As discussed on Section 2.1, the ranked distribution of visual words follows a Zipf distribution. We intend to extrapolate this observation to the color histogram of the local features of an image. For all datasets, we quantize the image colors in $q = 140$ color bins. Then, we partitioned the images in $m = 600$ blocks (superpixels) and extracted their respective color histogram. Both values of q and m were obtained experimentally. We sorted the color histogram by frequency in descending order, *i.e.*, we have the most frequent color in the first position, the second most frequent color in the second position and so on. For this step, we are not interested in the color values, since only the ranked frequency distribution is used to fit a Zipf distribution. Figure 22 shows the aggregate distribution of the ranked histogram of all blocks for each dataset.

We evaluated the fitting quality using Kolmogorov-Smirnov test for all datasets and discovered that for the first 15 elements, they all fit a Zipf distribution. Previous works on S-BoVW rely on a threshold to estimate the dominant colors of a histogram. Knowing that the ranked color histogram follows as Zipf distribution, we can propose a more intuitive way to get the dominant colors. Instead of using a less semantic threshold, we can summarize the color histogram by the γ most frequent colors. This is possible since, accordingly to the Pareto principle, roughly 80% of the pixels come from only 20% of the colors, in our case, the sum of the frequency of the top values the ranked histogram is the majority of the sum of all frequencies.

4.4 Bag-of-Superpixel Signatures

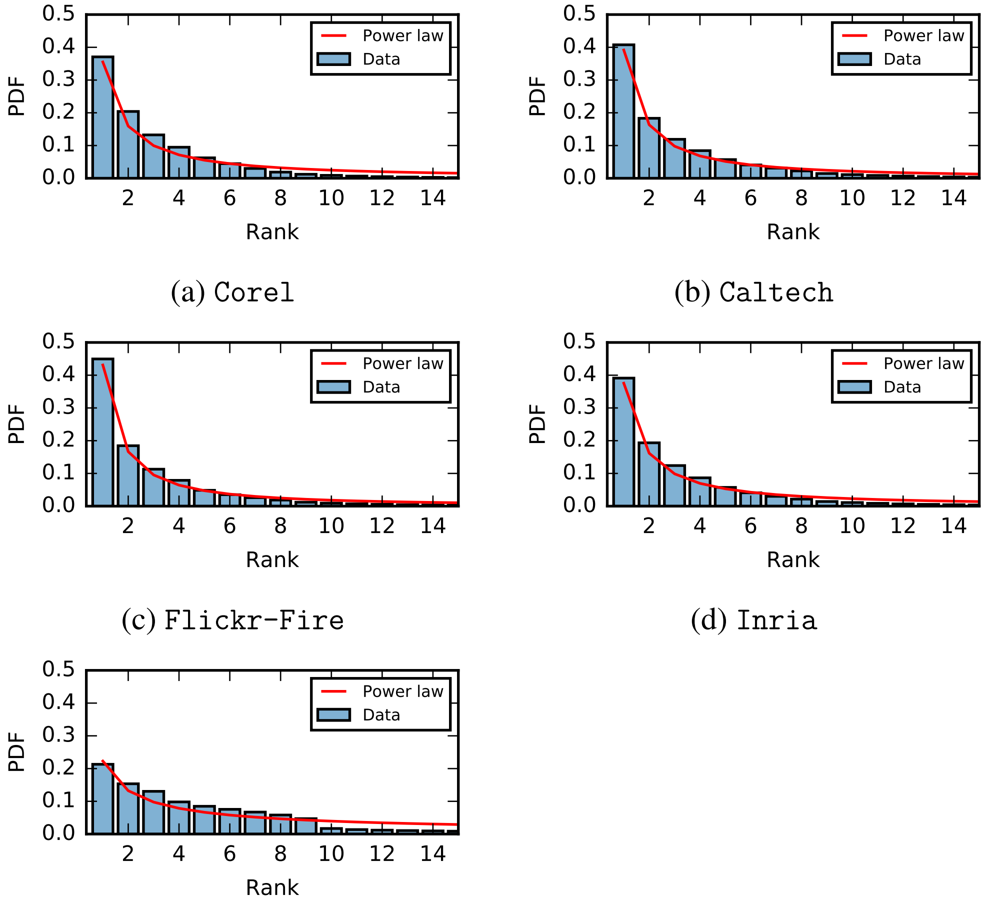
In this section we introduce the BoSS method, an S-BoVW based on superpixels and dominant colors. BoSS is able to extract visual signatures from images without demanding a visual dictionary beforehand, *i.e.*, there is no clustering involved. We proposed BoSS to have as few parameters as possible, the only parameter is the expected number m of visual signatures to be extracted. We also proposed two variations of BoSS: one based solely on color (BoSS) and one combining color and texture (Bag-of-Superpixels Color and Texture Signatures (BoSS-CT)).

4.4.1 The BoSS's Idea

The main idea of BoSS consists of three steps: region detection, feature extraction and signature generation. Figure 23 shows an overview of the BoSS method. Let I be an image, the first step

[4] <http://www-cvr.ai.uiuc.edu/ponce_grp/>

Figure 22 – Distribution of the ranked histograms on all datasets.



(a) `Corel`

(b) `Caltech`

(c) `Flickr-Fire`

(d) `Inria`

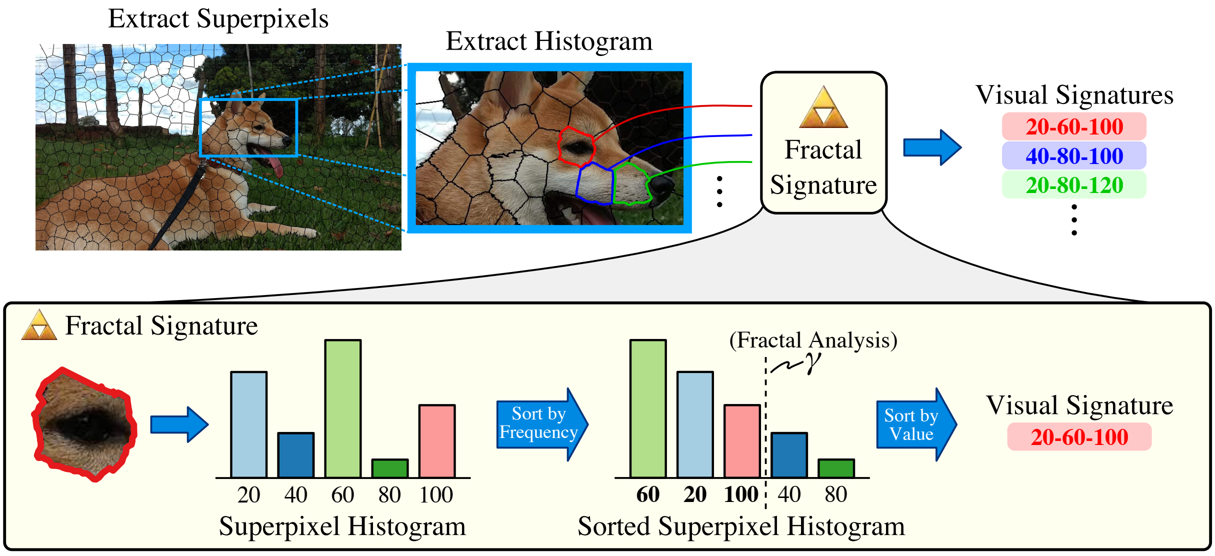
(e) `Texture`

Source: Elaborated by the author.

is to generate the blocks that will be converted to visual signatures. In our approach, we use a superpixel algorithm to generate the list $B = \{b_1, b_2, \ldots, b_m\}$ of m superpixels. BoSS then quantize the pixel values of I in q values, *e.g.*, from 256 intensity levels to 16 levels. Next, for each region $b_i \in B$, we extract its color histogram $h_i = \{(f_1, c_1), (f_2, c_2), \ldots, (f_q, c_q)\}$, where f_l is the frequency of the color c_l. Then, the histogram h_i go through the `Fractal Signature` module to be converted to a visual signature.

In Section 4.3, we observed that, when we sort the histogram h_i in descending order of the frequency, the sorted histogram $\hat{h}$ follows a Zipf's law. The basic idea of the `Fractal Signature` module is to summarize h_i by its γ most frequent values (dominant colors). To do so, `Fractal Signature` first sort h_i by the frequency of values in descending order to create $\hat{h}_i = \{(\hat{f}_1, \hat{c}_1), (\hat{f}_2, \hat{c}_2), \ldots, (\hat{f}_q, \hat{c}_q)\}$, where $\hat{f}_l \geq \hat{f}_{l+1}$ and $1 \leq l < q$. Then, `Fractal Signature` selects only the γ elements of $\hat{h}$ to create $\bar{h}_i = \{(\bar{f}_1, \bar{c}_1), (\bar{f}_2, \bar{c}_2), \ldots, (\bar{f}_\gamma, \bar{c}_\gamma)\}$, where $\bar{c}_l < \bar{c}_{l+1}$ and $1 \leq l < \gamma$. Finally, the superpixel can be represented as a visual signature $s_i =$ "$\bar{c}_1 - \bar{c}_2 - \cdots - \bar{c}_\gamma$". It is important to note that $\bar{h}_i$ is sorted by the color value to reduce the size of the visual signature vocabulary. In the example shown in Figure 23, using $\gamma = 3$, the m dominant colors

Figure 23 – Overview of the BoSS method.



Source: Adapted from Chino *et al.* (2018).

are $\{60, 20, 100\}$, following the `Fractal Signature` algorithm, the visual signature is "20-60-80". All steps of BoSS are shown in Algorithm 2, where $extractFeatures(I, m)$ is a function that receives a quantized image I, generates m superpixels and returns their histograms. The complexity of BoSS is linear with the dataset size (n) and the number of superpixels m, *i.e.*, $\mathcal{O}(mn)$.

Algorithm 2 – BoSS method

```
Input: Image $I$, $m$: number of superpixels, $\gamma$: number of dominant colors
Output: List of visual signatures $S$
$H \leftarrow extractFeatures(I, m)$
$S \leftarrow \emptyset$
for all $h_i \in H$ do
    $\hat{h}_i \leftarrow$ Sort $h_i$ by frequency
    $\bar{h}_i \leftarrow$ Select the $\gamma$ most frequent values of $\hat{h}_i$
    $s_i \leftarrow$ Sort $\bar{h}_i$ by value
    Add $s_i$ to $S$
end for
```

4.4.2 Freeing BoSS of Parameters

The algorithm described on the previously section needs as parameter the number of superpixels m and the number γ of dominant colors to use as visual signatures. However, since it is desired to have as few parameters as possible, in this section we propose a method to estimate the value of γ using the Fractal theory. We considered the set of all sorted histograms of an image dataset as a dataset F with q attributes (embedded dimension), *i.e.*, each sorted histogram is considered as a point in F. By calculating the intrinsic dimension D_2 of F, we can determine the minimum number of attributes needed to represent a histogram. Thus, a good value to γ must be at least or greater than D_2 ($\gamma = \lceil D_2 \rceil$). Algorithm 3 shows the main idea of this step. The complexity to estimate the intrinsic dimension D_2 is $\mathcal{O}(n)$ (TRAINA *et al.*, 2010). Once γ is estimated for the knowledge base, we can use this value to extract visual signatures of the query images.

Algorithm 3 – Fractal estimation of γ

```
Input: List of images $\{I_1, I_2, \ldots, I_n\}$, $m$: number of dominant colors
Output: Estimated $\gamma$
Initialize box-counting
for all $I_j \in \{I_1, I_2, \ldots, I_n\}$ do
    $H_j \leftarrow extractFeatures(I_j, m)$
    for all $H_i \in H_j$ do
        $\hat{h}_i \leftarrow$ Sort $h$ by frequency
        Add $\hat{h}$ in box-counting
    end for
end for
$D_2 \leftarrow$ Calculate intrinsic dimension using box-counting
$\gamma \leftarrow \lceil D_2 \rceil$
```

4.4.3 Integrating Color and Texture in BoSS

We also proposed BoSS-CT, a variation of BoSS based on Color and Texture. Given an image I, BoSS-CT extract color signatures the same way BoSS does. However, BoSS-CT also extracts signatures based on texture histograms. The steps to extract the texture signatures are the same steps described on the previous two subsections. However, instead of receiving a quantized color image, it receives a texture image (an image with values that describe the texture on each position). The texture image can be obtained using, for example, the LBP descriptor. The image is then represented by a set of color signatures and another set of texture signatures. BoSS-CT adds the prefix “C” to the color signatures and “T” to the texture signatures.

4.5 Experiments and Discussion

In this section we show the performance of BoSS on retrieving images. We present the results of two sets of experiments: BoSS parameter analysis and comparison with the BoVW techniques. When using color and texture, we are considering both color rank score and texture rank score to be equally important, as proposed by Santos *et al.* (SANTOS *et al.*, 2017). We implemented BoSS in Python and all experiments were carried on a 2.67GHz Intel Xeon X5650 CPU with 32GB RAM, running Ubuntu 16.04.

We ran the experiments on the datasets described on Section 4.3. We used the leave-one-out cross-validation for all datasets, except for `Inria`, where we used the representative of the 500 classes as queries. To evaluate the effectiveness of all methods we used the Mean

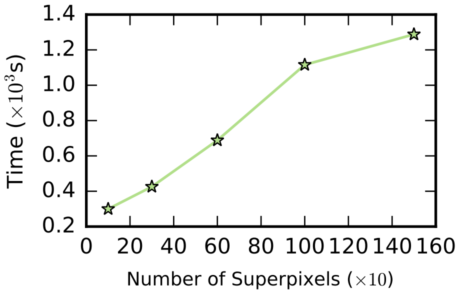
Average Precision (MAP) measurement. We also analyzed the precision and recall curves when comparing BoSS with the baseline approaches.

As baseline we used C-BoVW (SIVIC; ZISSERMAN, 2003) and S-BoVW (SANTOS *et al.*, 2017) methods. For the C-BoVW, we extract local features using SURF and two different dictionary sizes: 1,000 words and 20,000 words. To create the C-BoVW dictionaries, we used the *Mini-Batch K-Means* (SCULLEY, 2010) with *k-means++* (ARTHUR; VASSILVITSKII, 2007) to select the initial seeds. From now on, these methods will be referred to as BoVW1k and BoVW20k respectively. The S-BoVW methods used were SDLC and SDLCT, for these methods we used the author's recommended parameters. For all S-BoVW (including BoSS), we used a color quantization of $q = 140$ colors. The algorithm BoSS used to extract the superpixels was SLIC (ACHANTA *et al.*, 2012). For both SDLCT and BoSS-CT, the texture extractor used is the rotation invariant LBP.

4.5.1 BoSS Parameter Analysis

The first set of experiments aims at analyzing the influence of parameter m, the number of superpixels. As retrieval models, we used similarity distances (Cosine and Jaccard) and the VSM using the weights defined in Table 1. We ran all experiments in the `Corel` dataset 5 times, and the results showed are the average value of all runs. First, we evaluated how m influences on the time complexity of BoSS. We measured the wall-clock time needed to extract the visual signatures, what includes the time needed to estimate γ by calculating the dataset intrinsic dimension. Figure 24 shows the time needed to extract the signatures when m varies. As expected, BoSS is linear in relation with m.

Figure 24 – Time consumed to extract the visual signatures in the BoSS method versus m.



Source: Elaborated by the author.

Then, we evaluated how m impacts on the quality of the retrieved images. We used five different values of m and calculated the MAP. Table 3 shows the MAP, the first two lines were obtained while using similarity measures and the following lines were using VSM. It is possible to observe that MAP values increase as m grows for all retrieval models. On our evaluation, we considered $m = 600$ as the best value to be used, because there is only a slight variation when m varies from 600 to 1,500 and remembering that BoSS is linear with m, the lower the better..

There is no need to choose a larger value of m to have a slightly better result. The best results were achieved when using w_3 weight. From now on all results are using VSM and w_3 to retrieve the images.

Table 3 – BoSS MAP measures for each weight schema on Corel dataset. The **bold** values are the best results for each m and the value marked with a * is the best result.

Weight	MAP				
	$m = 100$	$m = 300$	$m = 600$	$m = 1{,}000$	$m = 1{,}500$
Cosine	0.099	0.100	0.141	0.190	0.212
Jaccard	0.125	0.130	0.184	0.248	0.278
w_1	0.437	0.483	0.505	0.513	0.517
w_2	0.392	0.420	0.505	0.434	0.4363
w_3	**0.469**	**0.514**	**0.530**	**0.533**	**0.534***
w_4	0.468	0.511	0.522	0.517	0.516
w_5	0.450	0.473	0.460	0.447	0.446
w_6	0.440	0.458	0.440	0.424	0.424
w_7	0.456	0.504	0.522	0.524	0.525

Source: Research data.

Lastly, we analyzed if BoSS is able to correctly estimate the best value of γ. Table 4 shows the MAP while varying γ. The intrinsic dimension of the ranked histogram on the Corel dataset is $D_2 = 4.38$. The best value of MAP was achieved while using $\gamma = 5$. Since BoSS estimates the best value using $\gamma = \lceil D_2 \rceil = \lceil 4.38 \rceil = 5$, BoSS was able to correctly estimate the best γ value.

Table 4 – How well BoSS chooses γ. The highlighted line is the estimated γ and the **bold** value is the best result.

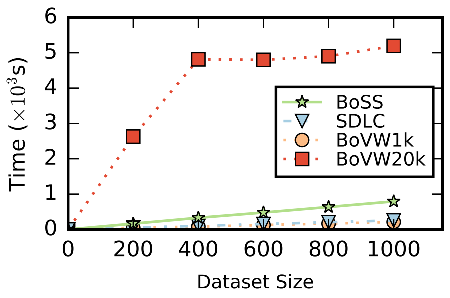
γ	MAP	γ	MAP
1	0.454	6	0.526
2	0.518	7	0.517
3	0.528	8	0.511
4	0.527	9	0.500
5	**0.530**	10	0.489

Source: Research data.

4.5.2 Comparison with the state-of-the-art

We compared BoSS and BoSS-CT with the state-of-the-art methods. First, we measured the time needed to extract the visual signature words. For the BoVW approaches, we included the time needed to create the dictionary. We also take into account the wall-clock time needed for BoSS to calculate the intrinsic dimension to estimate γ. We ran all experiments in the Corel dataset 5 times, and the results showed are the average value of all runs. Figure 25 shows the time needed to extract the signatures when n varies.

Figure 25 – Time comparison to extract the visual signatures versus the size of the dataset.



Source: Elaborated by the author.

The time BoSS spent to extract the visual signatures increases linearly with the size of the dataset. However, BoSS spent more time when compared to SDLC and BoVW1k, but less time than BoVW20k. To extract the whole Corel dataset, BoSS spent around 791 seconds, SDLC spent 256 seconds, BoVW1k spent 211 seconds and BoVW20k spent 5,196 seconds. The reason BoSS is slower than SDLC is because SDLC can extract the visual signatures in $\mathcal{O}(1)$, since it uses rectangular grids instead of superpixels. BoVW20k spent more time than the others due to the fact that, even though we used fast clustering algorithms, the number of clusters makes the algorithm very expensive. For both, BoSS-CT and SDLCT, the time needed to extract the visual signatures is approximately the double of BoSS and SDLC respectively, since all they need to do is to also extract signatures using a texture image.

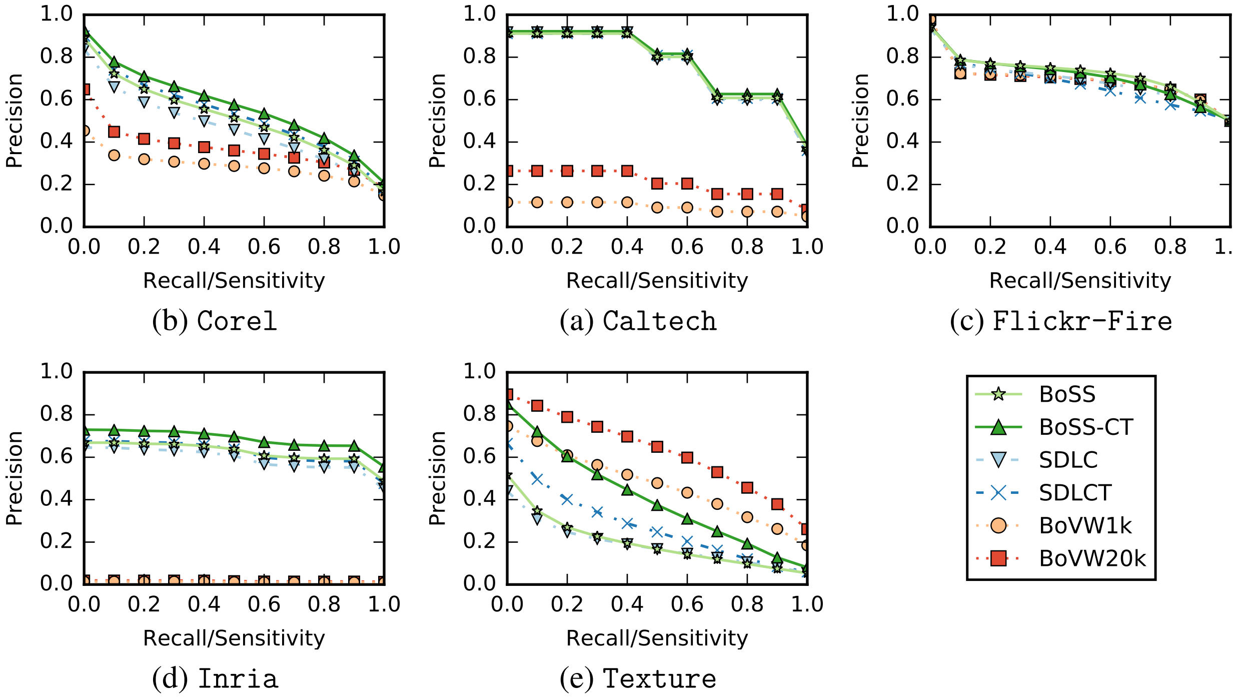
Table 5 – MAP measures for each dataset. The first four lines are S-BoVW methods, while the last two are C-BoVW. The highlighted lines are our proposal and the **bold** values are the best results.

Weight	MAP				
	Corel	Caltech	Flickr-Fire	Inria	Texture
BoSS	0.530	0.694	**0.703**	0.580	0.180
BoSS-CT	**0.563**	**0.713**	0.700	**0.638**	0.389
SDLC	0.452	0.662	0.654	0.538	0.171
SDLCT	0.522	0.670	0.643	0.568	0.258
BoVW1k	0.264	0.075	0.601	0.010	0.455
BoVW20k	0.351	0.168	0.614	0.014	**0.617**

Source: Research data.

Next, we compared the quality of the results obtained by BoSS, BoSS-CT, SDLC, SDLCT, BoVW1k and BoVW20k. We run the queries in all datasets and calculated the MAP measurement. Table 5 shows the MAP for all methods. Both our proposals, BoSS and BoSS-CT, had the best results on Corel, Caltech, Flickr-Fire and Inria, however, they were

Figure 26 – Precision and recall curves of all methods.



(b) Corel (a) Caltech (c) Flickr-Fire

(d) Inria (e) Texture

Source: Chino *et al.* (2018).

outperformed by BoVW20k on the Texture dataset. On the Corel dataset, BoSS-CT had a MAP up to 7.83% more precise than SDLCT. On the Caltech dataset, BoSS-CT was 6.03% better than SDLCT. On the Flickr-Fire dataset, BoSS had the best MAP, being up to 6.97% better than SDLC. On the Inria dataset, BoSS-CT was up to 10.97% better than SDLCT. On the Texture dataset, BoSS was outperformed by BoVW20k. However, when comparing between S-BoVW, BoSS-CT was 33.68% better than SDLCT.

We also compared the Precision and Recall curves on all datasets (Figure 26). Once again, except for Texture, BoSS-CT was similar or better than the competition on all datasets. On the Texture dataset, it is possible to see better results from BoVW20k. However, it is important to note that since BoSS and BoSS-CT are based on signatures, they do not need a pre-computed dictionary, skipping a lot of processing (as shown in Figure 25) as BoVW20k does. Our results showed that both BoSS and BoSS-CT have similar or better results than the SDLC and SDLCT. More importantly, SDLC needs a tuning of at least three parameters and SDLCT needs at least four. For example, the threshold both methods demands is not intuitive, rendering the method hard to tune. Conversely, the only parameter of BoSS and BoSS-CT is the number of expected visual signatures *m*.

4.6 Final Thoughts

In this Chapter we proposed BoSS, an intuitive, self-contained, scalable and effective approach for signature-based bags-of-visual-words (S-BoVW). BoSS extracts visual signatures from images' regions, which are given by superpixels. The signatures are taken from local features dominant colors and textures. Moreover, our proposal employs a fractal analysis to extract information about the domain application and also automatically estimate one of the parameters.

CHAPTER

5

SKIN ULCER IMAGE RETRIEVAL

In this Chapter we take advantage of image segmentation methods to improve image retrieval techniques. The idea is to introduce the semantics of segmentation methods on BoVW. Therefore we can improve the precision while retrieving skin ulcer images. The organization of this Chapter is as follows. We give a brief introduction on the skin ulcer image retrieval problem on Section 5.1. Section 5.2 shows some of the challenges to retrieve skin ulcer images. Section 5.3 introduces Imaging Content Analysis for the Retrieval of Ulcer Signatures (ICARUS) and Section 5.4 shows its results. Finally, Section 5.6 concludes this Chapter. The results of this Chapter were based on works presented in the IEEE 31st International Symposium on Computer-Based Medical Systems (CHINO *et al.*, 2018)

5.1 Introduction

In the last decades, the advances in technologies such as cameras, clinical equipment, and storage infrastructure, have motivated a big increase in the amount of clinical data available (PEREYRA *et al.*, 2014). Computer-Aided Diagnosis comprises a set of processes to support specialists in the analysis of medical images. However, some health care units do not have access to specialized equipments for image acquisition, *e.g.*, computed tomography scanners and multi-spectral cameras. On the other hand, mobile devices, such as smartphones, can acquire high quality images which can be a feasible alternative for image acquisition (DORILEO *et al.*, 2008).

One scenario where these images are especially useful is the analysis of chronic skin lesions, often referred as ulcers. These lesions may be caused by different reasons, such as poor blood circulation in lower extremities, injuries, infections, tumors and other skin conditions (DORILEO *et al.*, 2010). The visual appearance of these wounds provides clinical signs that may help physicians in the diagnosis. This scenario highlights the necessity of accurately processing images at a fast pace, giving support to image retrieval tasks (BEDO *et al.*, 2015b). One way to unravel this problem is through CBIR, which provides similar cases based on historical data (PIRAS; GIACINTO, 2017).

The analysis of the colors and texture present in ulcers is of particular interest, since they may indicate the healing process stage. Overall, simple lesions may present different characteristics during its healing process: inflammation (mainly characterized by redness in the limb region), grown of granulated tissue, followed by the final stage of healing and re-epithelialization. However, the healing pattern of chronic lesions is not well defined (BEDO *et al.*, 2015b; ODUNCU *et al.*, 2004).

After the first stage, the wound presents a coverage of yellow fibrin, sometimes containing small parts of necrosis, generating a non-uniform mix of granulation (reddish pixels), fibrin (yellowish pixels) and necrotic tissue (blackish pixels). Particularly, in neuropathic ulcers, callous lesions may appear, which are mainly composed of white tissue, presenting uniform thickness on the extremities. In this work, we aim at detecting these four variations of tissue composition in skin ulcers: granulation, fibrin, callous and necrotic tissue.

While the problem of segmenting ulcer regions from images have already been addressed in the literature (DORILEO *et al.*, 2010; SEIXAS; BARBON; MANTOVANI, 2015), most of the proposed approaches do not classify ulcer regions according to the skin patterns. On the other hand, when classifying the type of tissue composition, many approaches perform global classification in images, considering not only the wound regions but also the background and other objects in the image (BEDO *et al.*, 2015a; PEREYRA *et al.*, 2014). Global approaches badly influence the resulting set of similar images retrieved by a CBIR application, since they have to deal with a mixture of color and texture patterns in ulcer regions and the remaining parts of the images. Blanco *et al.* (BLANCO *et al.*, 2016) proposed the CL-Measure method, which employs superpixels to obtain homogeneous regions of pixels when performing image retrieval. CL-Measure segments the images using a superpixel-based approach and off-the-shelf classifiers, and then a CBIR task is executed based on the wound region using a label-scaled similarity measure. Although the results are promising, their proposed distance measure is not metric, and is computationally costly for high-resolution images, since it extracts fixed-size superpixels.

In order to overcome the aforementioned drawbacks, we propose ICARUS, which uses a bag-of-visual words approach considering only the relevant regions of the image. We evaluate our proposal in a real-world dataset, containing 217 images from four classes: granulation, fibrin, callous and necrosis. Our main contributions are as follows:

- **Relevant Signatures:** ICARUS process only relevant regions, discarding background and healthy skin;
- **Fast:** since ICARUS discards the non-relevant regions, it enables faster queries processing, being up to 5 orders of magnitude faster than the state-of-the-art; and
- **Effectiveness:** by adding semantic to the feature extraction, ICARUS increases the preci-

Table 6 – Distribution of the classes in the ULCER-DATASET, Granulation (G), Fibrin (F), Callous (C) and Necrosis (N). The combination of two or more letters means the presence of two or more types of lesion, *e.g.*, GF means the lesion has granulation and fibrin tissues.

Class	G	F	C	N	GF	GC	GN	FC	FN	CN	FGN
Frequency	69	40	5	6	68	11	10	1	3	1	3

Source: Research data.

sion of the retrieved images by up to 7% over the state-of-the-art.

5.2 Challenges on Retrieving Skin Ulcer Images

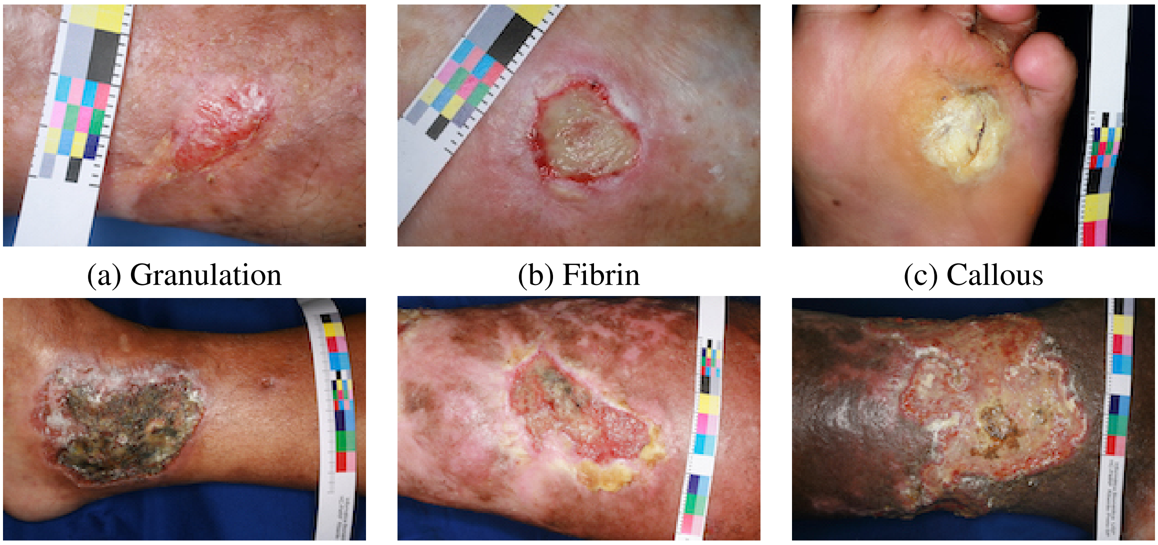
As discussed on Sections 2.5.2 and 6.2, venomous skin ulcers are lesions with different healing stages: fibrin, granulation, callous and necrosis. Each of these healing stages have a visual characteristic, *e.g.*, granulation is reddish, fibrin is yellowish and necrosis is black. On this Chapter we will be analyzing images from the ULCER-DATASET (DORILEO *et al.*, 2008). The images were manually labeled by experts and the dataset has a distribution of lesions, as follows: 161 granulation (G), 115 fibrin (F), 18 callous (C) and 23 necrotic tissue (N). It is important to note that some images have more than one type of lesion, e.g., both fibrin and granulation. There are a total of 97 images with a mixture of two or three types of lesions. Table 6 shows the distribution of each class in the ULCER-DATASET. The ULCER-DATASET also has 15 images manually segmented and labeled by experts (BLANCO *et al.*, 2016). Each segmentation region was labeled as healthy/background, granulation, fibrin and necrosis. From the 15 ground-truth images, there are a total of 30,426 superpixels (24,357 healthy, 2,333 granulation, 3,557 fibrin and 179 necrosis).

Figure 27 shows examples of the healing stages of skin ulcer images. When considering a CBIR system for skin ulcer images, it is important to notice that these images have elements that may negatively impact the results of a similarity query. Although each healing stage has a specific visual characteristic, every image also has elements that are not relevant for retrieving similar lesions, *e.g.*, healthy skin, a background, or a measurement tape. On this context, a CBIR system for skin ulcer image must consider this problem while extracting features, or comparing two images. One way to solve this problem is by considering only the relevant regions of the image while extracting features.

5.3 Our Proposal: ICARUS

In this section we introduce ICARUS, an image retrieval method for skin ulcer images through S-BoVW. ICARUS receives as input an ulcer image and retrieves the most similar images in a dataset. Since one of the problems with ulcer images is that they have elements, such as background and skin regions without lesion, they are not relevant to the image retrieval

Figure 27 – Examples of venomous skin ulcer images and the different types of healing stages.

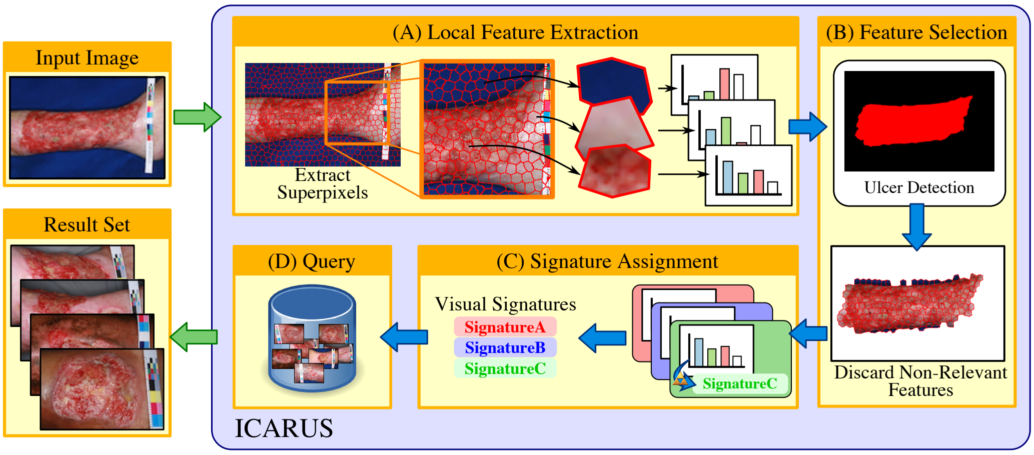


(a) Granulation (b) Fibrin (c) Callous

(d) Necrosis (e) Granulation and necrosis (f) Gran., fibrin and necrosis

Source: Elaborated by the author.

Figure 28 – How ICARUS flies: we retrieve the most similar images by extracting signatures using only the relevant regions.



Source: Adapted from Chino *et al.* (2018).

task. ICARUS extracts local features from the images and checks their relevancy. By doing so, ICARUS is able to generate only signatures from relevant regions, improving the result set. The main idea of ICARUS is to describe the ulcer images as a set of visual words (signatures). Therefore, ICARUS quickly recovers the most similar images. ICARUS performs four steps: (A) Local Feature Extraction, (B) Feature Selection, (C) Signature Assignment and (D) Query Processing. Figure 28 shows an overview of ICARUS. We also propose a variation of ICARUS based on segmentation algorithm, the Imaging Content Analysis for the Retrieval of Ulcer Signatures Through Segmentation (ICARUS-Seg).

ICARUS receives an image I and the number of superpixels m as input. On step (A), ICARUS generates the list B of m superpixels. Then ICARUS extracts the features h_i of each superpixel $b_i \in B$. The features extracted by ICARUS are color histogram and texture histogram. These features and the image I are passed as input to step (B), where they will be classified as relevant or non-relevant, this can be done by a supervised learning classification algorithm or by using an image segmentation algorithm. While using a classification algorithm, both color and texture features are concatenated into a single feature. For the segmentation algorithm used by ICARUS-Seg, we are using the segmentation used in the ASURA framework, which will be introduced with more details in Chapter 6. We consider the lesion regions as relevant, while the background and healthy skin are discarded.

The features from relevant regions will proceed to step (C), where they are assigned to visual signatures. ICARUS uses the dominant values of color and texture to assign the visual words. Each feature will be assigned to two signatures, one for color and another for texture. In the end of step (C), the image is represented by a set of visual signatures. Finally, step (D) uses the visual signature representation to retrieve the most similar images calculating the similarity using Jaccard/Cosine similarity measures or using the VSM weight schema. Algorithm 4 shows the basic idea of ICARUS.

Algorithm 4 – The ICARUS algorithm.

```
Input: $I$: input image, $m$: number of superpixels, $k$
Output: $RS$: list of the $k$ most similar images
$B \leftarrow extractSuperpixels(I,m)$
$H \leftarrow extractFeatures(I,B)$
$S \leftarrow \emptyset$
for all $h_i \in H$ do
    if $classifyAsRelevant(h_i)$ is $True$ then
        $s_i \leftarrow assignSignature(h_i)$
        Add $s_i$ to $S$
    end if
end for
$RS \leftarrow Searcher.query(S,k)$ return $RS$
```

5.4 Experiments and Discussion

In this section we show the performance of ICARUS and ICARUS-Seg to retrieve images. We present the results of three experiments: (i) classification method evaluation; ICARUS and ICARUS-Seg parameter analysis; and comparison with the state-of-the-art techniques. We implemented ICARUS and ICARUS-Seg in Python and all experiments were carried out on a 3.40GHz Intel Core i7-4770 CPU with 16GB RAM and a NVIDIA GeForce GTX 645 with 1GB GDDR5, running Ubuntu 16.04. To evaluate the overall effectiveness of all methods, we performed queries centered at each image of the `ULCER-DATASET`, using the leave-one-out strategy. For each query we calculated the MAP, precision and recall. Since the images in

ULCER-DATASET is multi-labeled, we calculated the MAP values using a label-based approach with macro-averaging (ZHANG; ZHOU, 2014).

5.4.1 Feature Classification Evaluation

As described on Section 5.3, ICARUS needs a feature classifier for the Feature Selection step. On this first experiment we evaluated the best classification algorithm. Since ICARUS needs to classify the regions as relevant and non-relevant, we considered the 24,357 superpixels of healthy tissues as non-relevant and the remaining 6,069 superpixels (granulation, fibrin and necrosis) as relevant. We evaluated 8 classifiers using 5-fold cross-validation: k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), Decision Tree, Random Forest, Multi Layer Perceptron, AdaBoost, Naive Bayes and Quadratic Discriminant Analysis (QDA). To evaluate the classifiers, we measured the accuracy and the F-Measure (harmonic average of the precision and recall). Table 7 shows the accuracy and the F-Measure values of every classifier. Since the Random Forest classifier achieved the best accuracy and F-Measure, we chose it as the feature classifier.

Table 7 – Evaluation of the Feature Selection classifier. The **bold** values are the best results.

Classifier	Accuracy (%)	F-Measure
k-NN ($k = 5$)	84.70	0.6127
SVM	87.81	0.6567
Decision Tree	86.54	0.6234
Random Forest	**89.87**	**0.7173**
MLP	80.86	0.3427
AdaBoost	88.09	0.6993
Naive Bayes	49.26	0.4428
QDA	49.82	0.4485

Source: Chino *et al.* (2018).

5.4.2 ICARUS Parameter Analysis

The second set of experiments aimed at analyzing the influence of parameter m, the number of superpixels, and the different retrieval models. We used six different values of m and two different retrieval models. As the retrieval model, we used similarity measures (Jaccard and Cosine) and VSM with the weights described on Table 1. We used a color quantization of $q = 140$ colors and the rotation invariant LBP as texture descriptor. The superpixel algorithm used by all methods to extract the superpixels was SLIC (ACHANTA *et al.*, 2012).

Tables 8 and 9 show the MAP while varying the parameters of ICARUS and ICARUS-Seg, respectively. While using similarity measures, ICARUS achieved the best result when using Jaccard similarity with $m = 300$, and ICARUS-Seg achieved the best result also when using Jaccard similarity, but with $m = 2{,}000$. When using VSM, ICARUS achieved the best result using $m = 1,000$ and ICARUS-Seg achieved the best result using $m = 1,500$. Both ICARUS and ICARUS-Seg achieved the best result using VSM with weight w_3. From now on, all results are presented using w_3, and $m = 1,000$ for ICARUS and $m = 1,500$ for ICARUS-Seg.

Table 8 – MAP measures achieved by ICARUS while varying m and the retrieval model. The **bold** value is the best result.

Weight	MAP				
	$m = 300$	$m = 600$	$m = 1,000$	$m = 1,500$	$m = 2,000$
Cosine	0.699	0.698	0.698	0.693	0.698
Jaccard	0.707	0.704	0.703	0.702	0.704
w_1	0.721	0.720	0.726	0.726	0.722
w_2	0.730	0.732	0.733	0.731	0.732
w_3	0.732	0.731	**0.735**	0.733	0.730
w_4	0.730	0.731	0.732	0.732	0.730
w_5	0.733	0.734	0.733	0.732	0.733
w_6	0.734	0.734	0.733	0.733	0.734
w_7	0.722	0.722	0.727	0.727	0.723

Source: Research data.

Table 9 – MAP measures achieved by ICARUS-Seg while varying m and the retrieval model. The **bold** value is the best result.

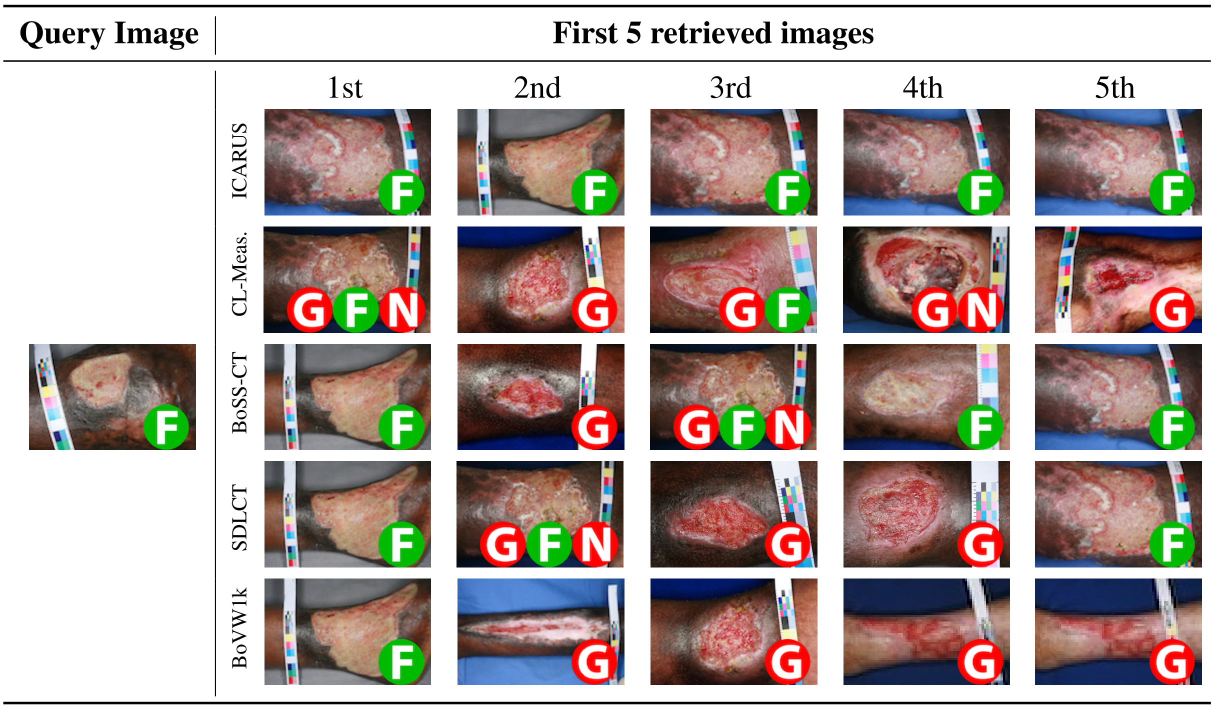
Weight	MAP				
	$m = 300$	$m = 600$	$m = 1,000$	$m = 1,500$	$m = 2,000$
Cosine	0.690	0.696	0.702	0.704	0.708
Jaccard	0.710	0.713	0.716	0.718	0.719
w_1	0.733	0.733	0.733	0.734	0.733
w_2	0.735	0.734	0.736	0.734	0.736
w_3	0.739	0.739	0.739	**0.740**	0.739
w_4	0.737	0.738	0.737	0.736	0.737
w_5	0.738	0.738	0.737	0.736	0.737
w_6	0.739	0.738	0.737	0.736	0.737
w_7	0.732	0.735	0.733	0.733	0.733

Source: Research data.

5.4.3 The Flight of ICARUS

We compared ICARUS and ICARUS-Seg with 5 methods. The CL-Measure (BLANCO *et al.*, 2016), a similarity measure for ulcer images. Since ICARUS is based on S-BoVW techniques, we also compared it with BoSS-CT (CHINO *et al.*, 2018), SDLCT (SANTOS *et al.*, 2017) and C-BoVW (SIVIC; ZISSERMAN, 2003). We are considering both color and texture rank scores to be equally important, as proposed by Santos *et al.* (SANTOS *et al.*, 2017). For the C-BoVW,

Figure 29 – Query example of ICARUS and the state-of-the-art. The green letters represent the query image class and the red letters are incorrect classes. The letters G, F and N stand for granulation, fibrin and necrosis respectively.



Source: Elaborated by the author.

we used two different configurations. One using a vocabulary of 20,000 visual words using the SIFT descriptor (JEGOU; DOUZE; SCHMID, 2008) and another 1,000 visual words using the deep learning based descriptor, DELF (NOH *et al.*, 2017). We will be referring to these methods as BoVW20k-SIFT and BoVW1k-DELF respectively. For the BoVW20k-SIFT, the visual words were learned from Flickr60K dataset using SIFT descriptor (JEGOU; DOUZE; SCHMID, 2008). The BoVW1k-DELF visual words dictionary were learned from a K-Means clusterings of the DELF descriptors extracted from the knowledge dataset. For all methods we used the author's recommended parameters.

Figure 29 shows a query example of all methods. The query image shows an image of a skin ulcer with fibrin tissue. ICARUS was able to retrieve all 5 images also containing fibrin tissue lesions. On the other hand, the competitors also retrieved images with granulation and necrosis tissues. Although CL-Measure also classifies the superpixels of the images according to their lesions it incorrectly retrieved images without fibrin tissue. One reason for this low precision was due to the fact that CL-Measure classified the yellow regions of some images as fibrin (4th and 5th images). However, the yellow region on these images are tissues in an advanced healing stage. The other methods use the whole image to extract features and were mainly influenced by the color of the skin and the background.

We calculated the MAP for each class and the average of all classes for all methods.

Table 10 shows the results achieved for each method in each class. Analyzing each class individually, ICARUS achieved the best result for granulation and ICARUS-Seg was the second best. And for fibrin and callous, ICARUS-Seg had the best results. CL-Measure achieved the best results on the necrosis class. One reason for ICARUS worst performance on the necrosis class is due to the fact that the majority of the images with necrosis tissues have a mixture of lesions. Usually these images have only a small portion of necrosis tissues, which leads to the presence of features extracted from different regions. Since CL-Measure weights each lesion according to the ratio of the lesion area on the image, it considers images with a similar ratio of lesions as more similar. On the other hand, ICARUS does not take this ratio in consideration when using the VSM weights. On the VSM w_3 weight, the most frequent word have a higher relevance. For this class in specific, ICARUS-Seg had a better result when using the Cosine similarity, achieving a MAP of 0.830.

Table 10 – MAP for all methods. The highlighted line is our proposal and the **bold** values are the best results.

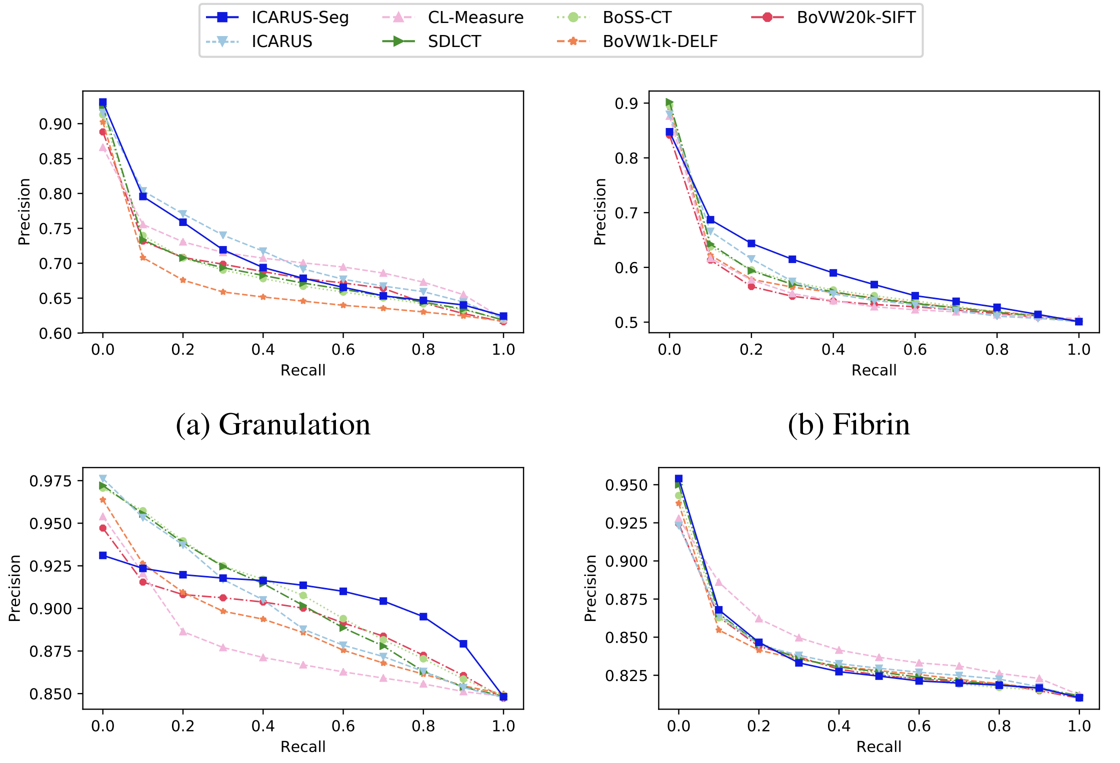
Method	Granulation	Fibrin	Callous	Necrosis	Average
ICARUS-Seg	0.681	**0.554**	**0.902**	0.822	**0.740**
ICARUS	**0.692**	0.539	0.885	0.823	0.735
CL-Measure	0.679	0.522	0.862	**0.834**	0.724
BoSS-CT	0.656	0.538	0.893	0.816	0.726
SDLCT	0.654	0.537	0.892	0.818	0.725
BoVW20k-SIFT	0.602	0.519	0.870	0.830	0.705
BoVW1k-DELF	0.632	0.532	0.864	0.816	0.711

Source: Research data.

While considering the average of all classes, ICARUS-Seg achieved the best result and ICARUS was the second best. CL-Measure, BoSS-CT and SDLCT achieved similar results and the BoVW achieved the worst results. ICARUS-Seg was 2.12% better than CL-Measure, 2.05% better than SDLCT, 1.93% better than BoSS-CT, 4.04% better than BoVW1k-DELF and 4.93% better than BoVW20k-SIFT.

We also analyzed the precision and recall curves when comparing ICARUS with the other methods. On Precision and Recall curves, the closer the curve to the top (precision of 1.0), the better the method. Figure 30 shows the Precision and Recall curves of all methods for each class. For the granulation class (Figure 30(a)), ICARUS had the best performance, ICARUS achieved a precision of 0.771 with a recall of 0.2, being up to 5.5%, 8.9%, 8.9%, 8.82% and 14.0% better than CL-Measure, SDLCT, BoSS-CT, BoVW20k-SIFT and BoVW1k-DELF respectively. However, for recall values greater than 0.5, CL-Measure had a better precision. For the fibrin class (Figure 30(b)), ICARUS-Seg had the best performance, ICARUS-Seg achieved a precision of 0.6444 with a recall of 0.2, being up to 11.5%, 8.4%, 8.1%, 13.9% and 11.3% better than CL-Measure, SDLCT, BoSS-CT, BoVW20k-SIFT and BoVW1k-DELF respectively. Both granulation and fibrin are the classes with more elements. For the callous class (Figure 30(c)), ICARUS, BoSS-CT and SDLCT had a similar performance up to a recall value of 0.2. For recall values greater than 0.4, ICARUS-Seg performed better than the other methods. It is important to note that although ICARUS-Seg did not had the best performance, it was able to achieve a precision of 0.916 with a recall of 0.4 and a precision of 0.85 with a recall of 1.0. CL-Measure worst performance can be explained by the fact that the authors did not consider callous lesions when proposing this similarity. Finally, for the necrosis class (Figure 30(d)), CL-Measure achieved the best result, while the other methods achieved a similar results. All methods were able to achieve a precision greater than 0.8 while retrieving all elements from this class (recall of 1.0).

Figure 30 – Precision and recall curve comparison of all methods for each class. The closer the curve to the top the better the method is.



(a) Granulation

(b) Fibrin

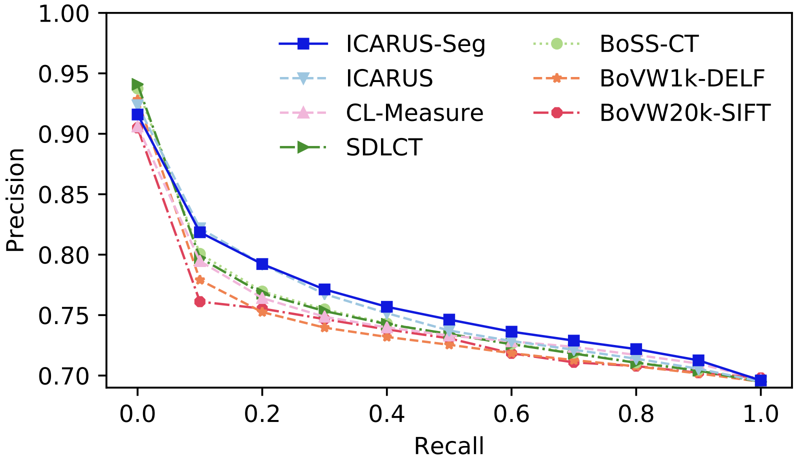
(c) Calous

(d) Necrosis

Source: Elaborated by the author.

Figure 31 shows the average Precision and Recall curves for all methods. Both ICARUS and ICARUS-Seg had similar results, however, ICARUS-Seg had a better precision with a recall value greater than 0.2. ICARUS-Seg had a precision of 0.792 with a recall value of 0.2. CL-Measure, BoSS-CT and SDLCT had similar results and, BoVW1k-DELF and BoVW20k had the worst result. Considering the BoVW approaches, ICARUS was up to 3% more precise than BoSS-CT/SDLCT, 4.9% more precise than BoVW20k-SIFT and 5.2% more precise than BoVW1k-DELF. Moreover, ICARUS-Seg was 7.5% more precise than BoVW20k-SIFT with a recall value of 0.1. ICARUS was 3.7% more precise than CL-Measure when the recall value was 0.2. ICARUS and ICARUS-Seg were more precise because it discarded the non-relevant regions to extract visual signatures, while the other methods used the whole image. On the other hand, although ICARUS used the same training set, we considered the diseases only as relevant and non-relevant, thus avoiding to label some regions incorrectly.

Figure 31 – Precision and recall curve comparison of all methods. The closer the curve to the top the better the method is.



Source: Elaborated by the author.

Finally, we compared the time needed to execute a query on all methods. We measured the time to extract features and then the time to process the query. We ran this process for every image in the dataset 5 times. The results showed are the average time value of all runs. Table 11 shows the average time needed to extract and to query for one image. Since CL-Measure is based on similarity measures, we are also measuring the time ICARUS needed to process queries using the Jaccard similarity. The top two lines uses similarity measures and the bottom six lines uses the VSM with w_3 weight as a retrieval method.

While extracting features/visual signatures, ICARUS was faster than most of the methods. ICARUS was 14 times faster than CL-Measure to extract features and 4 times faster than BoVW20k-SIFT and BoVW1k-DELF. CL-Measure was a lot slower since it extracts superpixels with a fixed size (number of pixels). Depending on the dimension of the image, it may extract more local features to be classified. Both BoVW20k-SIFT and BoVW1k-DELF was slower because they use a more complex extractor, SIFT and DELF respectively. Additionally, they also need to query a visual dictionary to assign the visual words. Both ICARUS and BoSS extracted visual signatures with similar times, but ICARUS is slightly slower because it has the classification step. ICARUS-Seg was 2 times slower than ICARUS due to the fact that ICARUS-Seg uses a segmentation algorithm based on a deep neural network to classify the features as relevant. ICARUS was 3 times slower than SDLCT, since SDLCT uses a regular

Table 11 – Elapsed time to extract features and execute one query of each method. The highlighted lines are our proposal and the **bold** values are the best results.

Method	Extraction Time (s)	Query Time (s)
ICARUS (Jaccard)	**3.362781**	**0.002563**
CL-Measure	46.68183	348.6101
ICARUS	3.362781	**0.003479**
ICARUS-Seg	6.319333	0.008442
BoSS-CT	3.038659	0.015619
SDLCT	**1.155796**	0.010344
BoVW20k-SIFT	13.188789	0.101450
BoVW1k-DELF	13.552783	0.004101

Source: Research data.

grid to extract the local features and ICARUS uses a superpixel approach. However, ICARUS and ICARUS-Seg is much faster for querying, which is what really matters, since querying is executed several times, while the feature extraction is made just once.

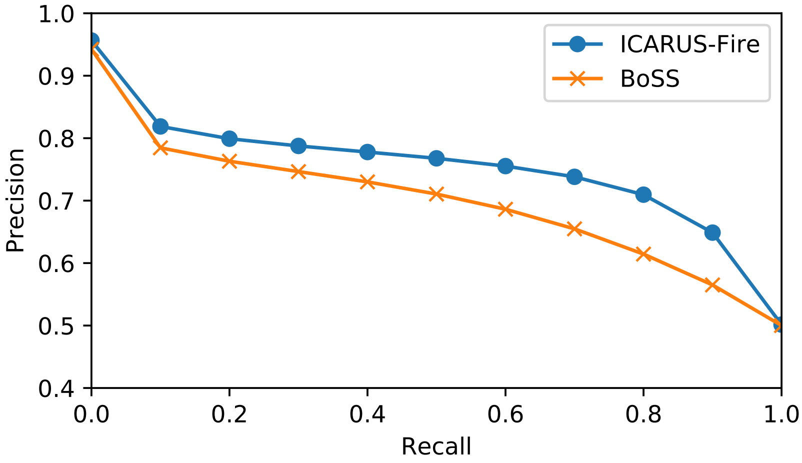
Regarding the query processing time, ICARUS was faster than all the state-of-the-art methods. ICARUS was 4.5, 3 and 29 times faster than BoSS, SDLCT and BoVW20k-SIFT respectively. ICARUS was faster than the other BoVW approaches, since it extracts less visual signatures after discarding the local features from non-relevant regions. BoVW1k-DELF had a similar query time than ICARUS due to the fact that DELF was trained to extract fewer local features than SIFT. In addition, ICARUS was 5 orders of magnitude faster than CL-Measure. CL-Measure is slower because it extracts more local features and is the only approach which uses numerical features, demanding more calculations. Also, CL-Measure is quadratic in the number of local features per class. The other approaches are based only on the frequency of the visual words. It is important to note that ICARUS-Seg was 2 times slower than ICARUS. This was due to the fact that the segmentation method used on ICARUS-Seg (Automatic Skin Ulcer Region Assessment (ASURA)) was more accurate than the feature classification used on ICARUS (see Section 6.4). This difference on the feature classification leads to fewer visual words extracted when using ICARUS.

5.5 ICARUS-Fire

In this section we explore how well the ICARUS approach can be used on images in different domains, such as images containing emergency situations with fire. In order to adjust ICARUS to support fire images (ICARUS-Fire), we replaced the segmentation algorithm with the BoWFire (Chapter 3) method trained on the BoWFire dataset. We implemented ICARUS-Fire in Python and all experiments were carried out on a 3.40GHz Intel Core i7-4770 CPU with 16GB RAM and a NVIDIA GeForce GTX 645 with 1GB GDDR5, running Ubuntu 16.04. To evaluate

ICARUS-Fire, we used the `Flickr-Fire` dataset and performed queries centered at each image, using the leave-one-out strategy. For each query we calculated the MAP, precision and recall.

Figure 32 – Precision and recall curve comparing the ICARUS-Fire and BoSS on the `Flickr-Fire` dataset. The closer the curve to the top the better the method is.



Source: Elaborated by the author.

We compared ICARUS-Fire with BoSS, Figure 32 shows the precision and recall curves achieved by both methods. One can see that ICARUS-Fire presented a better behavior than BoSS, ICARUS-Fire had a precision of 0.755 with a recall of 60% while BoSS had a precision of 0.686. When measuring the MAP, ICARUS-Fire was 4% better than BoSS. ICARUS-Fire achieved a MAP of 0.729, while BoSS achieved 0.703. These results show that the ICARUS approach can also be used on different domains of application to improve the precision on retrieval tasks.

5.6 Final Thoughts

In this Chapter we presented ICARUS, a CBIR method based on the bag-of-visual-words approach, which focuses only at the relevant regions of each image. ICARUS extracts local features from the image regions using superpixels. The local features are classified as either relevant or non-relevant. The relevant features are then assigned to visual signatures based on the dominant colors and textures.

CHAPTER 6

SKIN ULCER SEGMENTATION

In this Chapter we imply the use of deep convolutional neural networks to segment lesions on skin ulcer images. We also use image processing techniques to detect ticks on measurement rulers/tapes to estimate the pixel density of the images. By doing so, we can estimate the area of the lesions in real-world units, which can aid physicians on patients' healing follow-up. The organization of this Chapter is as follows. We give a brief introduction on the problem of skin ulcer segmentation on Section 6.1 and show the challenges of processing skin ulcer images on Section 6.2. Section 6.3 introduces the ASURA framework and Section 6.4 shows its results. Finally, Section 6.5 show our final thoughts on the skin ulcer segmentation problem. This Chapter is based on the work submitted to the Journal of Computer Methods and Programs in Biomedicine (CHINO *et al.*, 2019).

6.1 Introduction

Clinical environments are increasingly improving the capacity of generating large amounts of images, exams, and related information. Advances in technologies such as cameras, storage infrastructure, and clinical equipment are improving the quality of such information (PEREYRA *et al.*, 2014). However, many health care facilities do not have access to specialized equipment for image acquisition, such as computerized tomography scanners. Patients bedridden due to specific health conditions need to be examined at home. This is the case of patients presenting chronic skin lesions, referred to as skin ulcers.

Skin ulcers appear due to different reasons, including poor blood circulation, injuries, infections, tumors and other skin conditions (DORILEO *et al.*, 2008; DORILEO *et al.*, 2010). The lesion visual appearance can provide clinical signs that lead physicians during the diagnosis process. Venous skin ulcers are lesions with different healing stages, namely fibrin, granulation, callous and necrosis. Physicians and caretakers follow-up the healing evolution of lesions on patients by regularly taking photographs of the lesion. The healing time of each wound depends on multiple factors, including depth, location, patient age, local and systemic disease, and wound size (MORTON; PHILLIPS, 2016).

The introduction of deep learning techniques (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) has motivated several works to use CNNs on the medical domain. Photograph images have being used to recognize melanomas by segmenting (Yuan; Chao; Lo, 2017) or classifying (KAWAHARA; HAMARNEH, 2016; Yu *et al.*, 2017) them, and for foot ulcer segmentation (GOYAL *et al.*, 2017). However, to the best of our knowledge, there are little works that deal with skin ulcer images.

This task requires two major steps: the ulcer region segmentation and the wound area measurement. The image segmentation task refers to locating the boundary between the lesion and the surrounding skin (NAVARRO; ESCUDERO-VINOLO; BESCOS, 2018). By measuring the ulcer size, physicians are able to deem the healing evolution of the patient, compared to previous measurements. The ulcer area estimation is usually performed manually, which can be a time consuming and inaccurate task (BLANCO *et al.*, 2016), also causing discomfort to the patients. However, accurate and automatic lesions measurement strongly relies on well-segmented regions. Existing works lack on accurate segmentation, as they focus more in the retrieval and classification tasks (DORILEO *et al.*, 2010; PEREYRA *et al.*, 2014; SEIXAS; BARBON; MANTOVANI, 2015; BLANCO *et al.*, 2016; CHINO *et al.*, 2018).

Based on such scenario, in this work we propose the ASURA framework. ASURA uses CNNs and is able to detect and segment the ulcer lesions and the measurements tools depicted in the images, such as measurement rulers and/or tapes. With these information, ASURA automatically computes the lesion size, helping physicians and caretakers in the analysis of the patient's image. Accordingly, the contributions of ASURA are two-fold:

- **Precision**: ASURA accurately segments regions depicting skin ulcers and measurement rulers/tapes using deep CNN; and
- **Area measurement**: ASURA computes the area of lesion in real world units, based on the pixel density information and the segmented measurement ruler/tape.

We provide an extensive experimental analysis, comparing our proposal to state-of-the-art methods for the segmentation of skin ulcers. Regarding the wound area measurement, we compare the obtained results of ASURA to manually annotated segmentations, showing the high accuracy of our method.

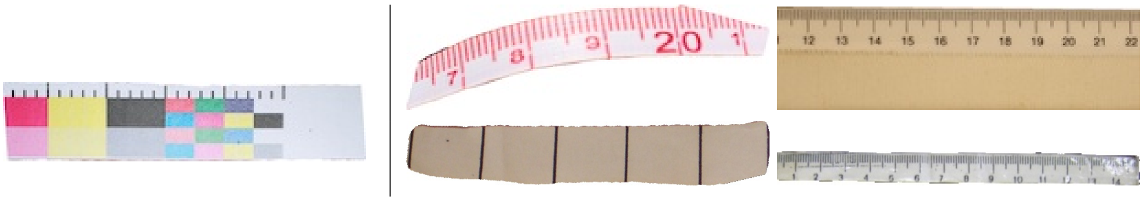
6.2 Area Estimation Challenges

One problem with skin ulcer images is the lack of standards to register and analyze them. Different from other medical images, such as Computerized Tomography (CT) scans, Magnetic

Resonance Imaging (MRI) and ultrasound, there is no specific equipment to capture images of skins lesions. Physicians take pictures from the whole lesion and perform comparisons spanning distinct healing stages at different periods of time. Usually, the pictures are taken using digital cameras or smartphones.

One way to overcome this problem is by creating a standard protocol to take the pictures, regarding the distance of the camera to the lesion, lighting conditions and background. However, it is not always possible to follow this protocol. The size of the lesion may guide the distance that the picture must be taken, or the picture may be taken at different locations (different lighting or background). It is also difficult to take pictures of lesions located in parts of the body with low access, mainly due to mobility deficiencies of patients. In these cases, the angle can harden even further the analysis of the image. Consequently, the size of the reference object may vary depending on the image. Figure 33 shows examples of distinct measurement rulers/tapes that can be found on skin ulcer images.

Figure 33 – Example of measurement rulers/tapes.



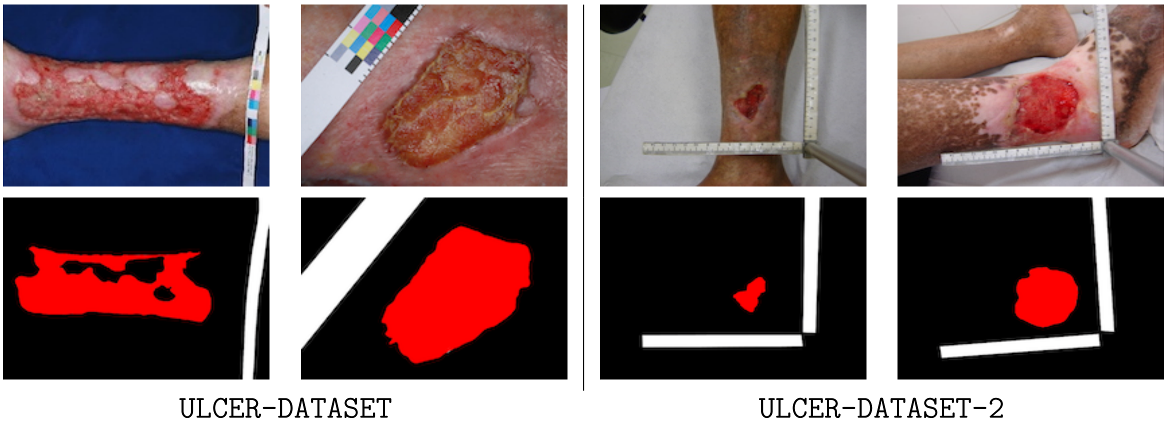
Source: Elaborated by the author.

On this Chapter we will be analyzing images from the `ULCER-DATASET` (DORILEO *et al.*, 2008) and the `ULCER-DATASET-2`. The `ULCER-DATASET` (DORILEO *et al.*, 2008) is composed of 217 dermatological images originated from both venous or arterial insufficiencies. The lesions were located on the inferior limbs with different sizes and healing stages. Only one lesion per patient was included and the majority of the patients skin color was white. The `ULCER-DATASET-2` is composed of 229 dermatological images, also originated from both venous or arterial insufficiencies. The lesions were also located on the inferior limbs with different sizes and healing stages. For each patient, a series of images were taken along a period of 90 days. On both datasets the images were taken using a digital camera. Personal data were deleted by an anonymization process. For both datasets, experts manually segmented the lesion region and the measurement ruler/tape to create a ground truth mask. Figure 34 shows some examples of the images and their respective masks. The red region on the ground truth mask is the lesion area and the white region is the measurement ruler/tape.

6.3 The ASURA Framework

In this section, we introduce Automatic Skin Ulcer Region Assessment (ASURA) framework to measure the area of lesions on skin ulcer images. ASURA uses a deep learning approach

Figure 34 – Example of skin ulcer images. The dataset images are on the top row and the ground truth masks are on the bottom row.

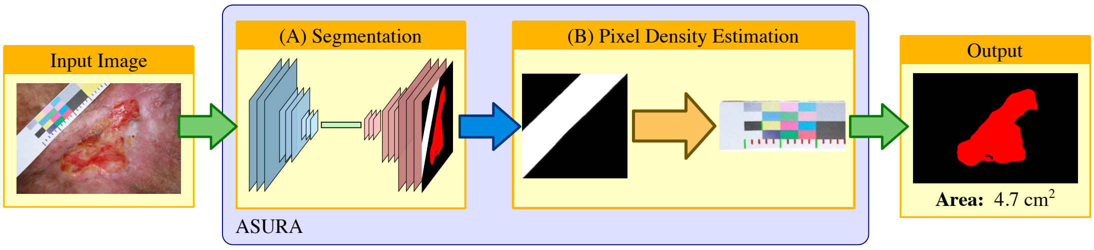


ULCER-DATASET ULCER-DATASET-2

Source: Elaborated by the author.

to segment the skin ulcer lesion. By analyzing objects like measurement rulers/tapes on the images, ASURA is able to estimate the area of the ulcer lesion in real world units. ASURA works performing two steps: (A) Ulcer Segmentation and (B) Pixel Density Estimation. Figure 35 shows the ASURA's architecture. ASURA also offers an interactive GUI in which the user can analyze the provided automatic Pixel Density estimation. Alternatively, the user also has the option of manually mark a more accurate Pixel Density.

Figure 35 – ASURA framework.



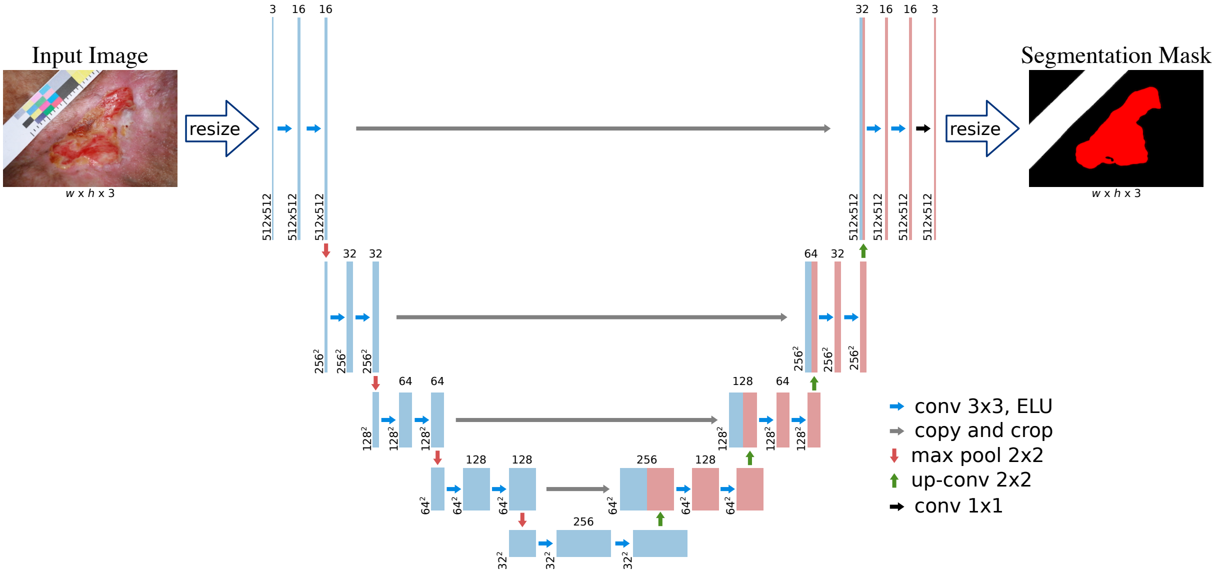
Source: Elaborated by the author.

6.3.1 Ulcer Segmentation

In the segmentation step, ASURA receives an skin ulcer RGB image and outputs a segmentation mask with both the lesion and the measurement ruler/tape. Ulcer Segmentation is based on a CNN for image segmentation. Since the size of the training dataset is limited, ASURA uses an architecture model based on the U-Net (RONNEBERGER; FISCHER; BROX, 2015).

Figure 36 shows the model of the architecture used. The network consists of an encoder and a decoder. First, ASURA receives as input an RGB image with arbitrary resolution. Since the input layer of the network is a tensor of size (512x512x3), the image is resized to a 512x512 resolution. The encoding phase consists of repeatedly applying two 3x3 padded convolutions

Figure 36 – Architecture of the network used by ASURA. The blue tensors are the encoders and the red tensors are the decoders.



Source: Adapted from Ronneberger, Fischer and Brox (2015).

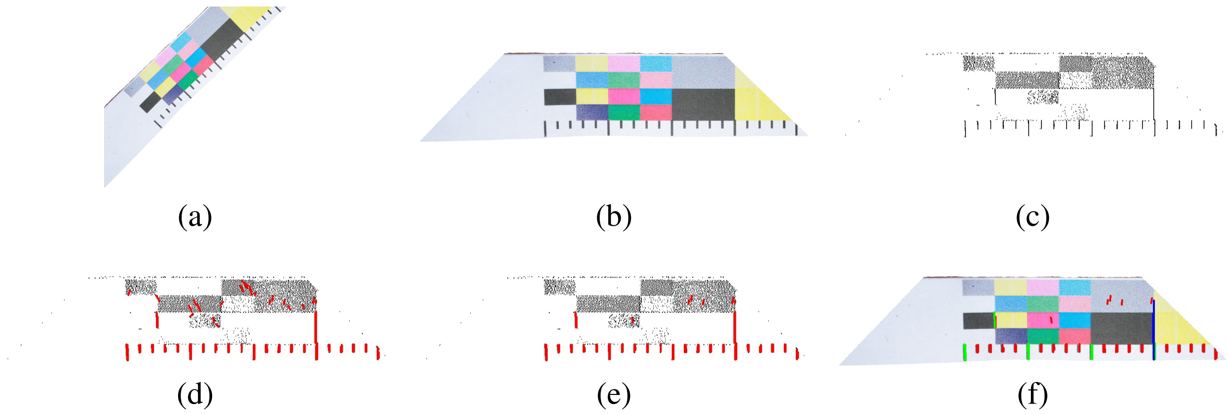
followed by an exponential linear unit (ELU). The tensor size is then halved by a 2x2 max pooling. The decoder consists of the repeatedly applying a 2x2 deconvolution which halves the depth of the tensor. This tensor is then concatenated with the corresponding tensor on the encoding phase. Then again, two 3x3 padded convolutions followed by an Exponential Linear Unit (ELU) are applied. The output layer of the network is a 1x1 sigmoid convolution to map the 16 layers of the decoded tensor into the three classes (lesion, measurement ruler/tape, background). In the final step, the output tensor is resized to the resolution of the input image. A heaviside step function is applied on the resized segmentation mask.

6.3.2 Pixel Density Estimation

After the Ulcer Segmentation step, ASURA process objects such as measurement rulers/tapes to estimate the Pixel Density (λ) of the image. With the segmentation mask and knowing the Pixel Density of the image, it is possible to estimate the area of the lesion in real world units. Figure 37 shows the steps used by ASURA to estimate λ.

Pixel Density Estimation receives as input the image and a segmentation mask of the ulcer lesion, as well as the measurement ruler/tape. (a) Using the segmentation mask, ASURA crops the measurement ruler/tape and delete the image's background (Figure 37(a)). (b) To simplify the measurement ruler/tape processing, ASURA finds the orientation of the ruler and rotates the image horizontally (Figure 37(b)). (c) On the next step, ASURA binarize the image (Figure 37(c)). The image is converted to gray scale and passes through an auto-threshold method. Then, a vertical edge detector filter is applied on the binarized image. (d) After the edge detector filter, ASURA uses a line detector to detect the ticks on the measurement ruler/tape (Figure 37(d)

Figure 37 – Pixel Density Estimation steps. (a) Find and crop the measurement ruler/tape in the image. (b) Rotate the measurement ruler/tape. (c) Binarize the image. (d) Detect lines in the image. (e) Keep only the most frequent parallel lines. (f) Group ticks by size and calculate distance between ticks.


(a) (b) (c)

(d) (e) (f)

Source: Elaborated by the author.

depicted in red). (e) With the ticks detected as line segments, ASURA group the ticks by its angle and keeps only the most frequent parallel ticks (Figure 37(e) shown in red). (f) On the last step, ASURA group the ticks by size and then calculate the distance in pixels between ticks of each group (Figure 37(f)). It is important to note that some images may have more then one measurement ruler/tape. In these cases, the steps are repeated for each segmented measurement ruler/tape. Algorithm 5 shows the algorithm used by ASURA to estimate λ.

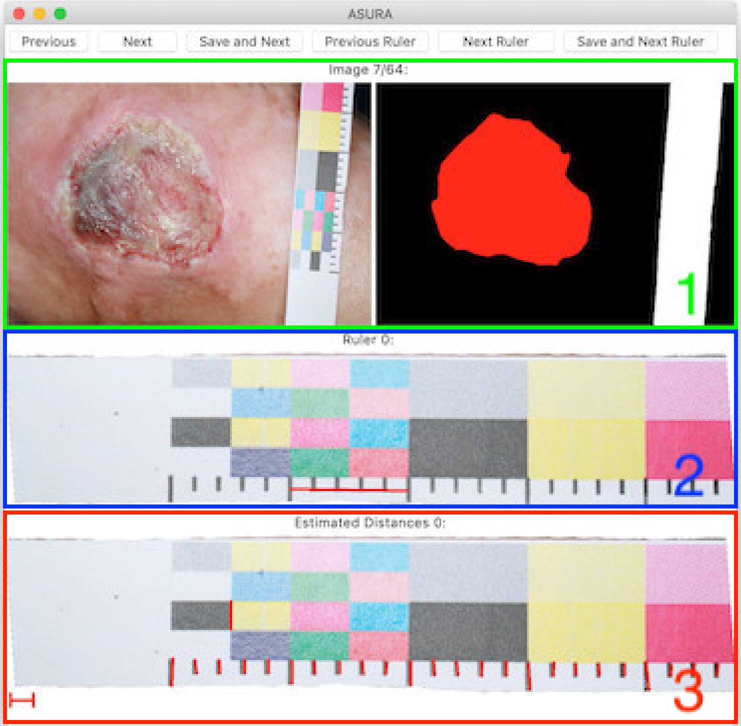
Algorithm 5 – Pixel Density Estimation algorithm.

```
Input: $I$: input image, $Mask$: segmentation mask
Output: $\lambda$: distances in pixels between ticks
$ruler_I, ruler_{mask} \leftarrow$ cropRuler($I$, $Mask$)
$ruler_I \leftarrow$ findAngleAndRotate($ruler_I, ruler_{mask}$)
$ruler_{bw} \leftarrow$ binarizeImage($ruler_I$)
$ruler_{edge} \leftarrow$ verticalLineFilter($ruler_{bw}$)
$lines \leftarrow$ detectLines($ruler_{edge}$)
$lines_0 \leftarrow$ getMostFrequentParallel($lines$)
$L \leftarrow$ groupBySize($lines_0$)
$\lambda \leftarrow \emptyset$
for all $l_i \in L$ do
    $\lambda_i \leftarrow$ calculateDistance($l_i$)
    Add $\lambda_i$ to $\lambda$
end forreturn $\lambda$
```

6.3.3 Graphical User Interface

ASURA has an interactive GUI that allows the user to browse the image and to analyze the segmentation output mask and the Pixel Density estimation. The interactive interface also allows the user to indicate a better Pixel Density estimation when needed.

Figure 38 – ASURA’s interactive graphical user interface.



Source: Elaborated by the author.

Figure 38 shows the ASURA’s GUI. On the area highlighted in green (1), the user can see the input image and the output of the Ulcer Segmentation. On the area highlighted in blue (2), the user can draw a line on the measurement ruler/tape image to indicate the length of the real world unit he/she desires to employ. Finally, on the area highlighted in red (3), ASURA shows the Pixel Density estimation. ASURA marks the detected ticks and assumes the red line below the measurement ruler/tape as the estimated distance between the ticks. If ASURA detects more than one tick size, it uses different colors for each tick size.

6.4 Experiments

In this section we show the performance of ASURA to estimate the area of the lesions in ulcer images. To evaluate ASURA, we run two sets of experiments: Ulcer Segmentation and Pixel Density estimation. We implemented ASURA in Python with the Keras libraries[1] using TensorFlow[2] backend. All experiments were carried out on a 4.20GHz Intel Core i7-7700k CPU with 16GB RAM and an NVIDIA Titan Xp with 12GB GDDR5X, running CentOS 7.

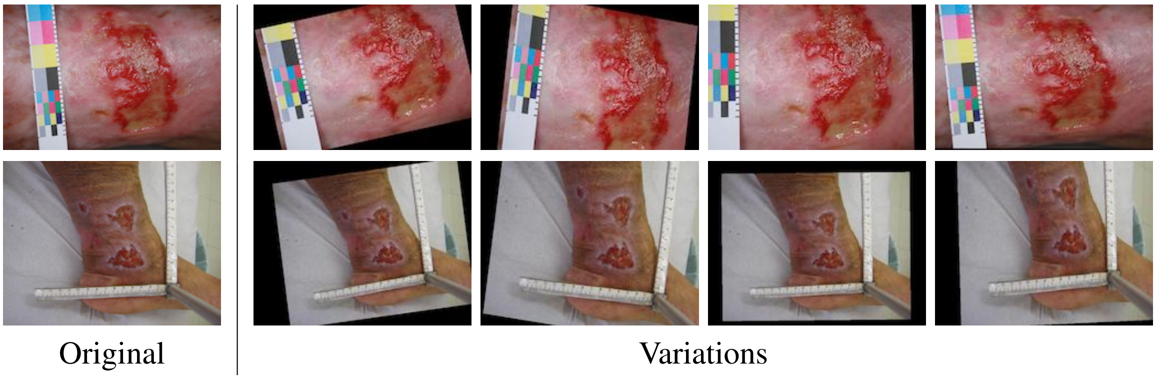
[1] https://keras.io/

[2] https://www.tensorflow.org/

6.4.1 Datasets and Data Augmentation

We evaluated ASURA on two skin ulcer datasets: ULCER-DATASET and ULCER-DATASET-2. We also considered the combination of both datasets (ULCER-BOTH) to evaluate ASURA. To evaluate ASURA, we split the datasets in test and training by a ratio of 70:30. The images were randomly split between the test and training sets such that images from one patient are present in both sets.

Figure 39 – Examples of the some images produced after the data augmentation.



Source: Elaborated by the author.

As mentioned in Section 2.4, deep learning models require a large amount of data to correctly learn patterns. Since both ULCER-DATASET and ULCER-DATASET-2 are small datasets, we imply the use of data augmentation to increase the robustness of ASURA. The images and masks were augmented using a series of random geometric transformations (translation, scale and rotation). Each image was translated by a random value up to 10% of the width/height of the image. Each image was rotated by a random angle between -15° and 15°. And finally each image was scaled up/down by a random value between 0.8 and 1.2. Points outside the image that are now visible were filled with a background color (black). Figure 39 shows examples of these transformations. Each mask received the same geometric transformations of its respective image. Table 12 shows the number of images in the test, training and augmentation for each dataset.

Table 12 – Number of images of each datasets.

Dataset	Size	Test	Training	Augmentation
ULCER-DATASET	217	64	153	1671
ULCER-DATASET-2	229	68	161	1558
ULCER-BOTH	446	132	314	1560

Source: Research data.

6.4.2 Ulcer Segmentation Evaluation

The first set of experiments evaluated how well ASURA segmented the skin ulcer images. We compared ASURA with CL-Measure (BLANCO *et al.*, 2016), ICARUS (Chapter 5) and a pixel color based segmentation (Color Classification). For both CL-Measure and ICARUS we considered the superpixel classification step as a segmentation algorithm. The Color Classification is the same algorithm introduced in Chapter 3, using the skin ulcer image datasets as training set. Although Dorileo *et al.* (DORILEO *et al.*, 2010) and Pereyra *et al.* (PEREYRA *et al.*, 2014) proposed ulcer segmentation algorithms, they were not directly compared with ASURA. Dorileo *et al.* was designed in a controlled environment, all images need a blue background. Since we are also considering images outside this scope, the results obtained by Dorileo *et al.* would be harmed. Pereyra *et al.* was not considered since it requires manual selection of the correct clusters. For this step we aimed at automatic segmentation methods.

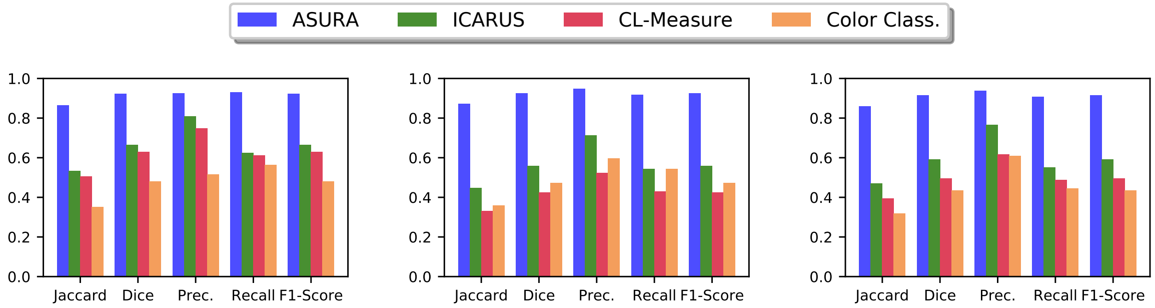
To evaluate the overall effectiveness of all methods, we run each segmentation method in every image of each test dataset. Then, we calculated five measures: Jaccard Coefficient, Dice Score, Precision, Recall, F1-Score. These measurements are given by Equations 6.1, 6.2, 3.1, 3.2 and 3.4 respectively. All results shown in this Section are the average of all images.

$$\text{Jaccard Coefficient}(GT,Seg) = \frac{|GT \cap Seg|}{|GT \cup Seg|} \tag{6.1}$$

$$\text{Dice Score}(GT,Seg) = \frac{2 \cdot |GT \cap Seg|}{|GT| + |Seg|} \tag{6.2}$$

where GT is the ground truth region and Seg is the region yielded by the segmentation algorithm.

Figure 40 – Evaluation considering the five indexes of the segmentation methods on each dataset.



(a) ULCER-DATASET (b) ULCER-DATASET-2 (c) ULCER-BOTH

Source: Elaborated by the author.

Tables 13, 14 and 15 show the results obtained by all methods on the ULCER-DATASET, ULCER-DATASET-2 and ULCER-BOTH, respectively. Figure 40 shows a summary of the results. Our experiments showed that ASURA outperformed the competitors on all datasets. On all datasets, ASURA achieved values above 86% for the Jaccard Coefficient and above 90% for the other measurements. On the ULCER-DATASET, ASURA was 38% better than the second best method (ICARUS), 41% better than CL-Measure and 59% better than Color Classification on the Jaccard Coefficient. When comparing the Dice Score and F1-Score, ASURA was 28% better than the second best method (ICARUS), 32% better than CL-Measure and 48% better than Color

Classification. According to the authors, for this dataset, Pereyra *et al.* (PEREYRA *et al.*, 2014) achieved a Jaccard Coefficient of 56%.

Table 13 – Evaluation of the segmentation methods on the `ULCER-DATASET`. The highlighted line is our proposal and the **bold** values are the best results.

Method	Jaccard	Dice	Precision	Recall	F1-Score
ASURA	**86.5**	**92.4**	**92.5**	**93.1**	**92.4**
CL-Measure	50.7	62.8	74.9	61.1	62.8
ICARUS	53.2	66.5	80.8	62.5	66.5
Color Class.	35.3	48.0	51.6	56.5	48.0

Source: Research data.

A similar behavior occurred with the `ULCER-DATASET-2`. ASURA was 49% better than the second best method (ICARUS), 62% better than CL-Measure and 59% better than Color Classification on the Jaccard Coefficient. When comparing the Dice Score and F1-Score, ASURA was 40% better than the second best method (ICARUS), 54% better than CL-Measure and 49% better than Color Classification.

Table 14 – Evaluation of the segmentation methods on the `ULCER-DATASET-2`. The highlighted line is our proposal and the **bold** values are the best results.

Method	Jaccard	Dice	Precision	Recall	F1-Score
ASURA	**87.2**	**92.6**	**94.7**	**91.9**	**92.6**
CL-Measure	33.2	42.4	52.3	42.9	42.4
ICARUS	44.7	55.9	71.3	54.4	55.9
Color Class.	35.9	47.2	59.6	54.2	47.2

Source: Research data.

While processing the combined dataset (`ULCER-BOTH`), ASURA was still able to correctly segment the lesion regions in the skin ulcer images. ASURA was 45% better than the second best method (ICARUS), 54% better than CL-Measure and 63% better than Color Classification on the Jaccard Coefficient. When comparing the Dice Score and F1-Score, ASURA was 35% better than the second best method (ICARUS), 46% better than CL-Measure and 52% better than Color Classification.

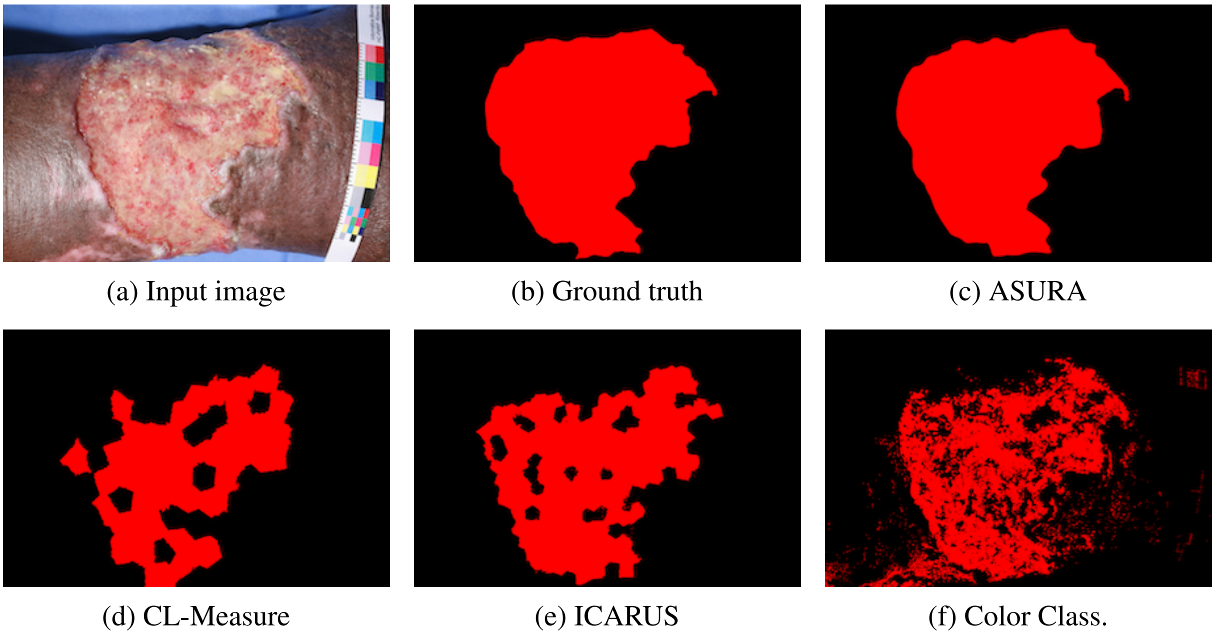
Figures 41 and 42 show one example of the segmentation output from `ULCER-DATASET` and `ULCER-DATASET-2`, respectively. On both datasets, ASURA had an output similar to the Ground Truth (GT). We can note that both CL-Measure and ICARUS had problems to segment the lesions because of the miss-classification of their superpixels. Since Color Classification is pixel-wise, it is unable to correctly segment the whole lesion, thus some pixels inside the region were not considered as part of the lesion.

Table 15 – Evaluation of the segmentation methods on the ULCER-BOTH. The highlighted line is our proposal and the **bold** values are the best results.

Method	Jaccard	Dice	Precision	Recall	F1-Score
ASURA	**86.0**	**91.4**	**93.7**	**90.7**	**91.4**
CL-Measure	39.4	49.7	61.7	48.8	49.7
ICARUS	47.1	59.2	76.6	55.1	59.2
Color Class.	31.9	43.5	61.0	44.6	43.5

Source: Research data.

Figure 41 – Example of the ulcer segmentations of an image from ULCER-DATASET.


(a) Input image

(b) Ground truth

(c) ASURA

(d) CL-Measure

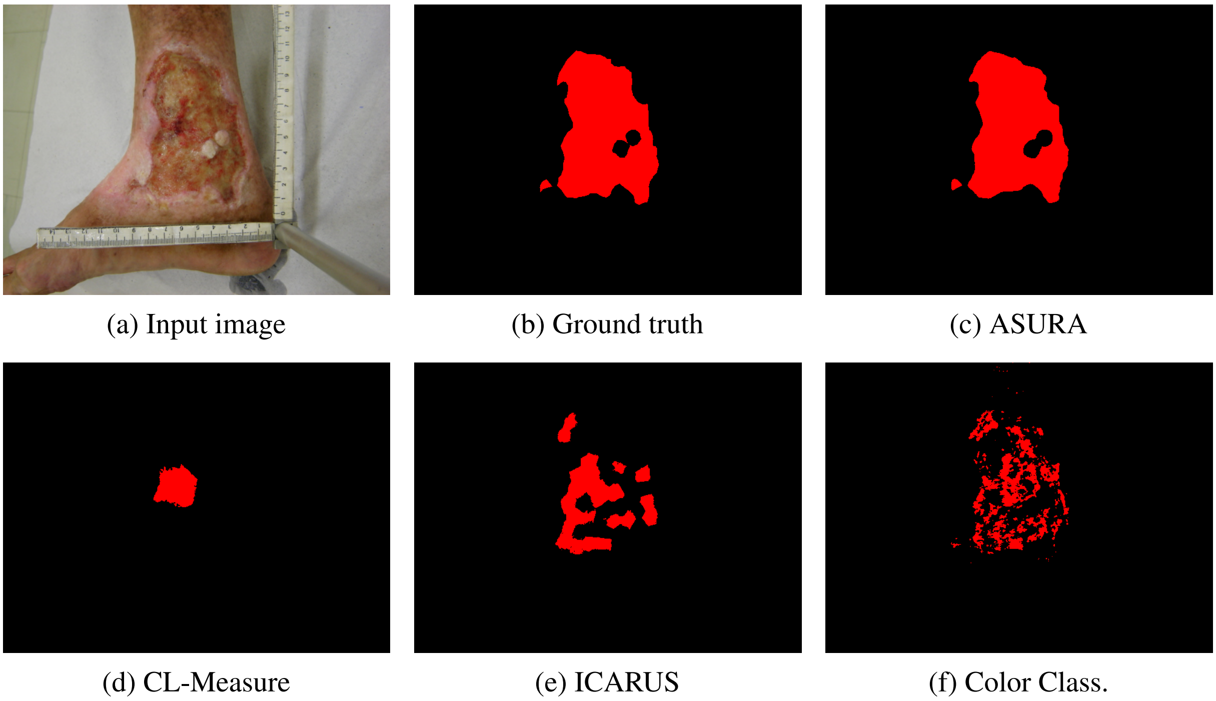
(e) ICARUS

(f) Color Class.

Source: Elaborated by the author.

Figures 43 and 44 show examples of bad segmentation outputs generated by ASURA for the ULCER-DATASET and ULCER-DATASET-2. In Figure 43, ASURA considered parts of the lesion as healthy/background, one reason for this error was due to the fact that this image has a shine region on that part of the lesion. Color Classification had problems with this image because the lighting of the image made the skin looks reddish. However, it is important to note that although this is one of the worst segmentations, ASURA achieved a Jaccard Coefficient of 67.81% and a Dice Score of 80.82%. ASURA also had a bad performance while segmenting images with small lesions (Figure 44). For this image, ASURA achieved a Jaccard Coefficient of 25.85% and a Dice Score of 41.08%. One reason for this bad performance on images with small lesions are due to the fact that ASURA has to resize the images to the input tensor size (512 × 512) and later resize the output tensor to the original size. The lesion on this image has 147 pixels and the ASURA output has 38 pixels. Small lesions also show the problem with the methods based on superpixels, both CL-Measure and ICARUS were not able to segment

Figure 42 – Example of the ulcer segmentations of an image from `ULCER-DATASET-2`.



(a) Input image

(b) Ground truth

(c) ASURA

(d) CL-Measure

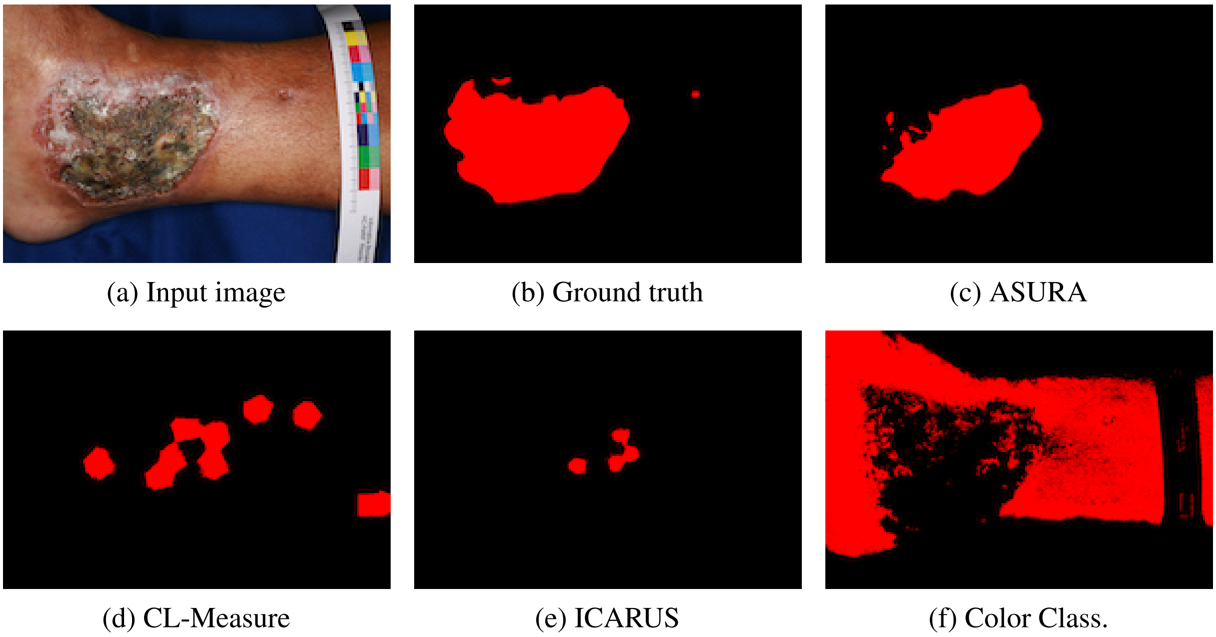
(e) ICARUS

(f) Color Class.

Source: Elaborated by the author.

anything, since the size of the superpixel on both cases are larger than the size of the lesion. As can be seen, all the cases where ASURA produced bad segmentation lead the other methods to produce bad results too.

Figure 43 – Bad segmentation of a lesion with necrosis on the `ULCER-DATASET`.



(a) Input image

(b) Ground truth

(c) ASURA

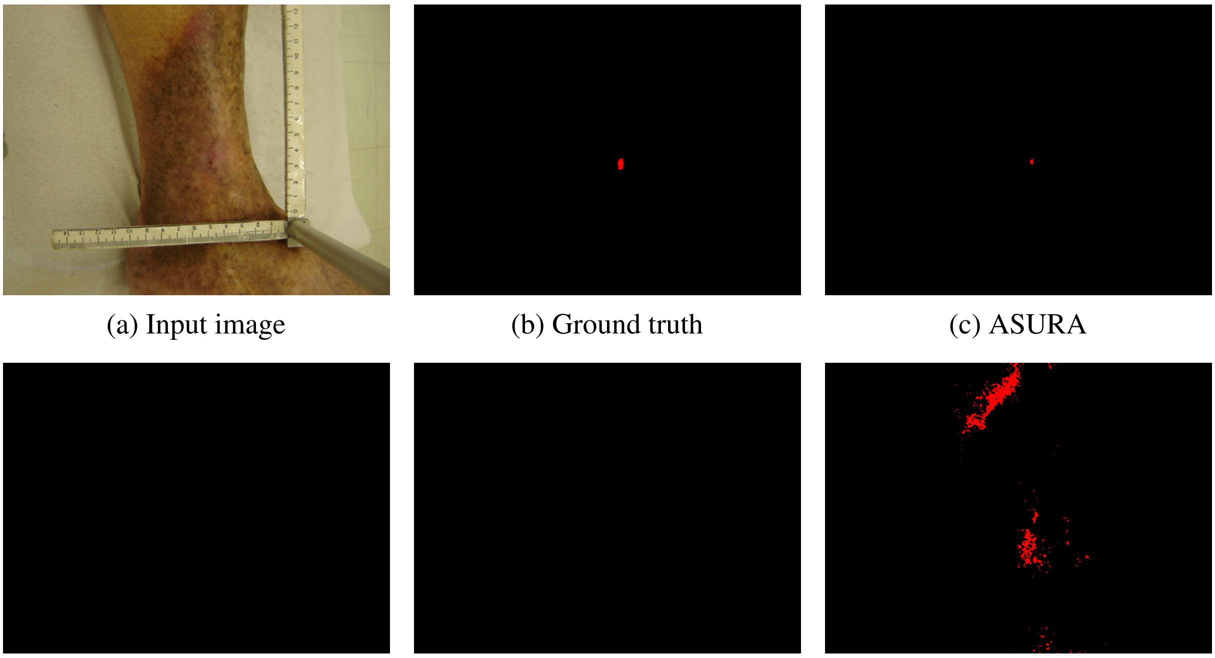
(d) CL-Measure

(e) ICARUS

(f) Color Class.

Source: Elaborated by the author.

Figure 44 – Segmentation of an image with a small lesion on the ULCER-DATASET-2.



(a) Input image (b) Ground truth (c) ASURA

(d) CL-Measure (e) ICARUS (f) Color Class.

Source: Elaborated by the author.

6.4.3 Pixel Density Estimation Evaluation

In this experiment, we evaluate how well ASURA is able to estimate the area of the lesion on a real world unit of measurement, *e.g.*, squared centimeters (cm^2). The area A of a lesion can be computed using the Pixel Density (λ), thus, the area can be obtained using $A = |Mask|/\lambda^2$, where $Mask$ is a segmentation mask of the lesion. On this experiment we estimated the Pixel Density in pixels per centimeter (*pixel* / cm). For evaluation purposes, an expert drew a one-centimeter line in each image with the ASURA GUI. By measuring the length in pixels of this line, it is possible to obtain the real Pixel Density (λ_{real}) of each image. Examples of these lines can be seen on the top row of Figure 45, the one centimeter line is marked by a red line.

Since ASURA can estimate more than one Pixel Density (λ) per measurement ruler/tape (distance between small ticks or distance between large ticks), the best estimation was chosen (λ_{est}). It is important to note that sometimes the chosen λ_{chosen} can be equivalent to a fraction of the desired unit. In this case, λ_{chosen} must be multiplied by the corresponding factor, *e.g.*, if λ_{chosen} is equivalent to one millimeter, the estimated Pixel Density for one centimeter is $\lambda_{est} = 10 \times \lambda_{chosen}$. With the values of λ_{real} and λ_{est}, we can calculate three different areas in real world unit of measurement:

- **Ground Truth Area** (Equation 6.3): using the size of the lesion on the ground truth mask

($|GT|$) and the real Pixel Density (λ)

$$A_{gt} = \frac{|GT|}{\lambda_{real}^2} \tag{6.3}$$

- **Real Area** (Equation 6.4): using the size of the lesion on the ground truth mask ($|GT|$) and the estimated Pixel Density (λ_{est})

$$A_{real} = \frac{|GT|}{\lambda_{est}^2} \tag{6.4}$$

- **[Estimated Area** (Equation 6.5): using the size of the lesion on the ASURA segmentation mask ($|Seg|$) and the estimated Pixel Density (λ_{est})

$$A_{est} = \frac{|Mask|}{\lambda_{est}^2} \tag{6.5}$$

To evaluate the results obtained by ASURA, we calculated the relative error E in percentage, which can be calculated using Equation 6.6. We calculated the relative error for all images in the test set. The results shown in this Section are the average over all images.

$$E_v = \frac{|\bar{v} - v|}{\bar{v}} \times 100\% \tag{6.6}$$

where v can be any variable (λ_{est}, A_{real} and A_{est}), $\bar{v}$ is the true value of the variable and v is the estimated value of the variable.

Table 16 shows the relative error of the estimated Pixel Density (λ_{est}), real area (A_{real}) and estimated area (A_{est}). By calculating the relative error of A_{real}, we can estimate the impact of the ASURA's Pixel Density estimation and the relative error of A_{est} shows how well ASURA can estimate the lesion area in cm^2. ASURA had the best result on the ULCER-DATASET-2, estimating λ_{est} with a relative error of 5.6%, the error while calculating the area A_{real} was of 14.3% and the area A_{est} had an error of 18.0%. On the ULCER-DATASET, ASURA was able to estimate λ_{est} with a relative error of 7.9%. Using the λ_{est} to calculate the A_{real}, ASURA had a relative error of 19.1% and the estimated area A_{est} had an error of 23.9%. The ULCER-BOTH had errors between ULCER-DATASET-2 and ULCER-DATASET.

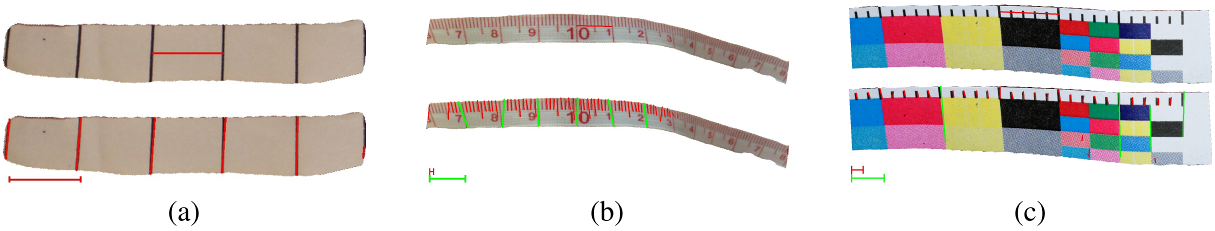
Table 16 – Relative error of the Pixel Density and area estimation.

Dataset	Relative Error (%)		
	λ_{est}	A_{real}	A_{est}
ULCER-DATASET	7.9	19.1	23.9
ULCER-DATASET-2	5.6	14.3	18.0
ULCER-BOTH	6.7	16.6	21.3

Source: Research data.

Figure 45 shows examples of some Pixel Density estimations. Figure 45(a) shows a measurement tape where ASURA correctly estimated a λ equivalent to 1.0 cm. The estimated Pixel Density is $\lambda_{est} = 116$ pixels/cm and the ground truth is $\lambda_{real} = 115$ pixels/cm. Figure 45(b) shows a measurement tape where ASURA estimated two different λ, one for the smaller ticks (λ_{red}) and one for the larger ticks (λ_{green}). On this measurement tape, the smaller ticks are the millimeters (mm) and the green ticks are the centimeters. For this measurement tape $\lambda_{real} = 100$ pixels/cm, the estimated $\lambda_{red} = 11$ pixels/mm and $\lambda_{green} = 101$ pixels/cm. Figure 45(c) shows a measurement tape with more details (colored squares). Even with a more complex object, ASURA was able to correctly estimate the Pixel Density. Once again, ASURA estimated two Pixel Density, $\lambda_{red} = 47$ pixels/(2×mm) and $\lambda_{green} = 231$ pixels/cm, the ground truth is $\lambda_{real} = 240$ pixels/cm.

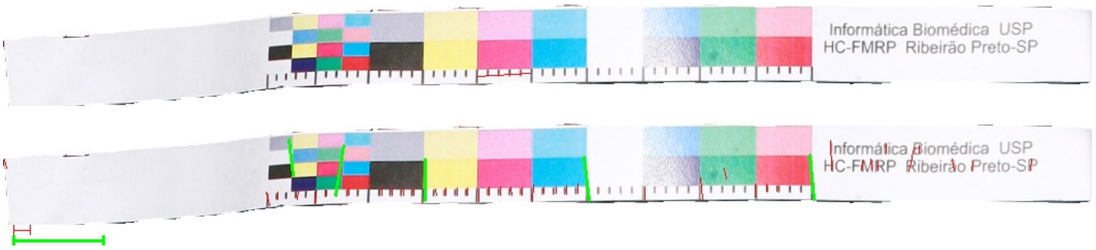
Figure 45 – Example of correct Pixel Density estimation on different measurement tapes.


(a) (b) (c)

Source: Elaborated by the author.

However, ASURA was not able to correctly estimate the Pixel Density on some images. Figure 46 shows an example of a bad estimation. While trying to detect the ticks on this measurement tape, ASURA wrongly detected the vertical lines on the text as ticks. Also, this image has a bright spot on a region of the measurement tape, leading ASURA to fail to detect the ticks on this region. The ground truth on this measurement tape is $\lambda_{real} = 52$ pixels/cm, due to this problems, ASURA estimated $\lambda_{red} = 15$ pixels/2mm for the red ticks and $\lambda_{green} = 87$ pixels per unknown unit for the green ticks. Since we cannot use λ_{green}, we can use λ_{red} to estimate $\lambda_{est} = 5 \times \lambda_{red} = 80$ pixels/cm, giving a relative error of 44.0%.

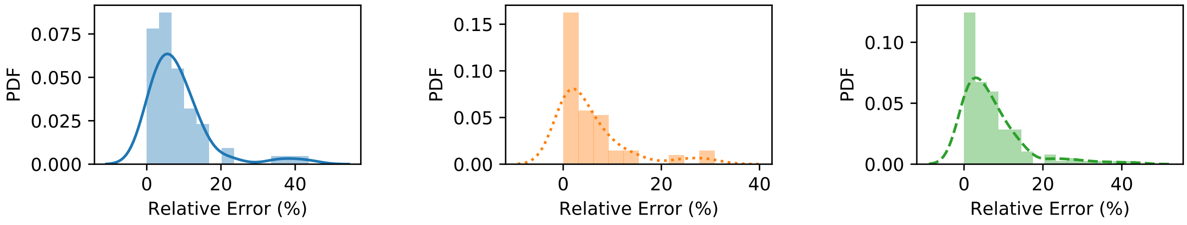
Figure 46 – Wrong Pixel Density estimation.


Informática Biomédica USP
HC-FMRP Ribeirão Preto-SP
Informática Biomédica USP
HC-FMRP Ribeirão Preto-SP

Source: Elaborated by the author.

When using a λ_{est} with a high relative error, the relative error on both areas A_{real} and A_{est} grows even more. To overcome this problem, we can take advantage of the interactive GUI of ASURA. If the user is not satisfied with the estimated λ, the user can manually mark a more accurate λ. Figure 47 shows the relative error distribution on all datasets. It is possible to note that, the majority of the estimated λ have a relative error up to 20%.

Figure 47 – Pixel Density's relative error distribution on all datasets. The bars are the histogram and the lines are the probability density function (PDF).



(a) ULCER-DATASET (b) ULCER-DATASET-2 (c) ULCER-BOTH

Source: Elaborated by the author.

Measurements of the Pixel Density performed by ASURA with a high relative error can be manually fixed by the user (expert). If the user opts to replace the estimations with a relative error greater than 20%, the new relative errors are shown in Table 17. By replacing the wrong estimations, we had a reduction of up to 2% on the relative error of the Pixel Density on all datasets. We reduced the relative error on the area A_{real} by 5.3%, 6.4% and 5.9% on the ULCER-DATASET, ULCER-DATASET-2 and ULCER-BOTH respectively. The relative error on the area A_{est} is reduced by 5.0%, 5.9% and 5.5% on the ULCER-DATASET, ULCER-DATASET-2 and ULCER-BOTH respectively.

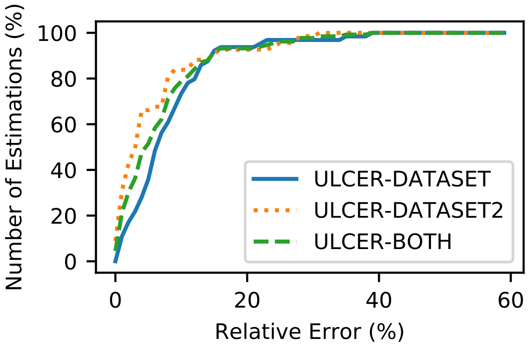
Table 17 – Relative error of the Pixel Density and area estimation with the aid of an expert.

Dataset	Relative Error (%)		
	λ_{est}	A_{real}	A_{est}
ULCER-DATASET	6.1	13.8	18.9
ULCER-DATASET-2	3.6	7.8	12.1
ULCER-BOTH	4.8	10.7	15.8

Source: Research data.

Although the user must input a new value of Pixel Density on some estimations, the majority of the images are below 20% of relative error. Figure 48 shows the percentage of estimations with a relative error lower than the values on the x axis. All datasets have more than 90% of estimations with a relative error lower than 20%, the ULCER-DATASET have 93.8%, the ULCER-DATASET-2 have 92.6% and the ULCER-BOTH have 93.2%. This shows that by replacing only a few bad estimations we can improve the area estimation by up to 5.9%.

Figure 48 – Proportion of estimations with a relative error lower than the values on the *x* axis.



Source: Elaborated by the author.

6.5 Final Thoughts

In this Chapter we presented the ASURA framework aimed at estimating the lesion area of a skin ulcer image. ASURA uses an encoder/decoder deep neural network to segment the lesion on the image. The comparison of ASURA with methods built on traditional image processing algorithms, such as CL-Measure and ICARUS, have shown that the deep learning approach presented way better results. ASURA also detects the measurement ruler/tape present in the image and automatically estimates the image's pixel density. This allows an accurate size measurement of the lesions.

CHAPTER 7

CONCLUSION

The large amount of image data generated by social media, crowdsourcing and medical systems can be useful for decision making tasks. The knowledge extracted from these images can aid authorities in emergency situations or assist on the daily activities of physicians. In this PhD research, we explored the use of these images from fire emergency situations and skin ulcer images.

We explored the integration of segmentation methods with local features from CBIR systems. First we focused on segmentation algorithms using superpixels. Then we expanded the use of superpixels for local feature extractions and proposed an S-BoVW approach. Finally, we integrate the segmentation methods with a CBIR system. The remaining of this chapter is organized as follows. In Section 7.1 we discuss the contributions resulted by this PhD research. In Section 7.2 we discuss the future lines of research. Finally, in Section 7.3, we list the publications resulted by this PhD research.

7.1 Contributions of this PhD Research

The contributions of this PhD dissertation were results of four research problems investigated during this PhD research. The first problem was to detect and segment fire in emergency situation images, which can be provided by crowdsourcing. The second one was the proposal of a method to represent local features based on S-BoVW using superpixels. The third problem consisted of retrieving similar images on a chronic skin ulcer image dataset. And finally we explored better ways to segment and assess the lesions on skin ulcer images. These contributions will be discussed with more details in the following sections.

Another contribution of this PhD research, was the creation of two image segmentation datasets. To the best of our knowledge, until the publication of the BoWFire method, fire segmentation datasets focused only on forest fires. So, we proposed the BoWFire-Dataset, a fire segmentation dataset for urban emergency situations with fire. We collected images from Flickr and manually segmented the fire regions of the images. The same problem was faced when dealing with the skin ulcer image datasets. A team of specialists manually created the lesion masks used on this PhD research. As additional contribution, we also analyzed the users' behavior in social media, e-commerce and phone calls based on the timestamps and the activity volume. All implementations and the BoWFire-dataset are available in <http://chinodyt.github.io/>.

7.1.1 BoWFire

BoWFire, as described in Chapter 3, is a fire detection method in emergency situation images. We explored the use of superpixels to reduce the complexity of the image, and analyzed two modalities of the visual properties of the image. By analyzing both color and texture features, BoWFire was able to correctly segment fire, and dismiss false-positives on still images.

7.1.2 BoSS

BoSS, as described in Chapter 4, is a S-BoVW that uses superpixels to extract local features. By analyzing the fractal dimension of these features, BoSS can automatically estimate the best parameters to map these features into visual signatures. Since BoSS is based on signatures, it does not require the creation of visual dictionaries beforehand. The main contributions of the BoSS are summarized as follows:

- **Self-contained:** by analyzing the patterns of the local histograms, we proposed a summarization of the color histograms based on the most dominant colors and textures to generate the visual signature.
- **Intuitive parameter:** BoSS estimates the γ parameter by using fractal analysis. The intrinsic dimension provides the minimum number of attributes needed to represent the color histogram in the application domain.
- **Scalability:** we showed that BoSS was scalable, being linear with the size of the dataset.
- **Effectiveness:** we showed that the visual signatures extracted using BoSS retrieved images successfully, being up to 10.97% better than the state-of-the-art approaches.

7.1.3 ICARUS

The ICARUS framework, introduced in Chapter 5, is a CBIR system for skin ulcer images. ICARUS integrates segmentation methods with local features. We imply superpixels to extract the local features and map them in visual signatures. We introduced two approaches to segment the lesion, one based on superpixel classification and one based on CNN (Chapter 6). By integrating the segmentation with the feature extraction, ICARUS discards the non-relevant regions of the image, such as background and health skin. The main contributions of ICARUS are summarized as follows:

- **Relevant Signatures:** ICARUS uses only the local features from relevant regions to represent the image. Our results showed an accuracy greater than 89% when ICARUS classifies the local features.

- **Fast:** by discarding the non-relevant regions, ICARUS was able to process queries up to 5 orders of magnitude faster than the state-of-the-art methods.

- **Effectiveness:** lastly, the addition of semantic to the image representation resulted in an increased precision while retrieving the similar images, being up to 7% better than the state-of-the-art.

7.1.4 ASURA

ASURA, proposed in Chapter 6, is a skin ulcer assessment framework able to measure the area of a skin ulcer lesion in real-world units. To segment the lesion and detect the measurement ruler/tape, we used a CNN architecture. Them we used image processing techniques to detect the ticks on the measurement ruler/tape and estimate the density of pixels/cm in the image. ASURA can be used to aid physicians in the follow-up of the healing process of their patients. Accordingly, its main contributions are as follows:

- **Precision**: ASURA was able to correctly segment the lesion regions with a precision greater than 92.5%, having a precision up to 63% better than the competitors.

- **Area measurement**: ASURA was able to automatically estimate the pixel density of the images with a relative error of 5.6% and semi-automatically able to estimate the area of the lesion in cm^2 with a relative error of 12.1%. This is a novel resource, previously not available.

7.2 Future Work

After studying the methods developed in this research, it is possible to suggest several lines of research that can continue and deepen the results obtained so far. A few suggestions are presented as follows:

1. Explore the use of signatures based BoVW using local features extracted by deep learning techniques.

2. On the skin ulcer problem, considering that the physicians take photographs regularly of patients. Is is also possible to analyze the temporal aspect of the lesions (increasing/decreasing size), and aid physicians to intervene on the patient treatment.

3. On this PhD research, we only explored the segmentation of the whole lesion without taking in account the healing stages of the tissue (fibrin, granulation, callous and necrotic). Knowing the evolution of these tissues can aid physicians even more. So, one future work is the detection and assessment of these healing stages in skin ulcer images.

4. An additional future work is the development of a mobile application able to assess the skin ulcer in real time. Since mobile devices have lower processing power, this can also imply in changes on the protocol to take photographs of the skin ulcer. Instead of using measurement tools, such as ruler and measurement tapes, physicians can use objects with a known length or area, such as cards with patterns or QR codes.

7.3 List of Resulted Publications

The summary of the publications resulted from this PhD research are as follows:

- BoWFire (CHINO *et al.*, 2015) was published in the 28th Conference on Graphics, Patterns and Images (SIBGRAPI2015).

- An additional contribution of this work was the publication of a fire image segmentation dataset, which is also part of a larger dataset, the Fire and Smoke Dataset (FiSmo) (CAZZOLATO *et al.*, 2017). FiSmo is a dataset containing images and videos of fire and smoke emergency situations. FiSmo was published in the Dataset Showcase Workshop, a satellite event of the 32nd Brazilian Symposium on Databases (SBBD2017).

- BoSS (CHINO *et al.*, 2018) was published in the 33rd ACM/SIGAPP Symposium On Applied Computing (SAC2018).

- ICARUS (CHINO *et al.*, 2018) resulted in a publication on the IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS2018).

- Finally the ASURA (CHINO *et al.*, 2019) was submitted to the Journal of Computer Methods and Programs in Biomedicine.

- As another contribution (see Appendix A), the user activity analysis method VolTime (CHINO *et al.*, 2017) was published in the SIAM International Conference on Data Mining (SIAM-SDM2017).

In addition to the mentioned contributions, the PhD candidate also collaborated with his colleagues at the Database and Images Group[1]. A full paper was published at CBMS2018, in this paper was proposed Retrieval-based Application for Imaging and Knowledge Investigation (RAFIKI) (NESSO *et al.*, 2018). RAFIKI is an infrastructure to automatically extract features indexing and organizing all medical information in a relational data base management system. RAFIKI also allows an integration of analytical tools.

[1] GBdI (http://gbdi.icmc.usp.br/)

BIBLIOGRAPHY

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SUSSTRUNK, S. Slic superpixels compared to state-of-the-art superpixel methods. **IEEE TPAMI**, IEEE, v. 34, n. 11, p. 2274–2282, 2012. Citations on pages 40, 68, and 78.

AKOGLU, L.; CHANDY, R.; FALOUTSOS, C. Opinion Fraud Detection in Online Reviews by Network Effects. In: **ICWSM 2013**. [S.l.: s.n.], 2013. p. 2–11. Citation on page 133.

ALY, M.; WELINDER, P.; MUNICH, M. E.; PERONA, P. Towards automated large scale discovery of image families. In: **IEEE CVPR**. [s.n.], 2009. p. 9–16. Available: <https://doi.org/10.1109/CVPRW.2009.5204177>. Citation on page 63.

ALZU'BI, A.; AMIRA, A.; RAMZAN, N. Semantic content-based image retrieval: A comprehensive study. **Journal of Visual Communication and Image Representation**, Elsevier, v. 32, p. 20–54, 2015. Citations on pages 27 and 39.

ANWAR, S. M.; MAJID, M.; QAYYUM, A.; AWAIS, M.; ALNOWAMI, M.; KHAN, M. K. Medical image analysis using convolutional neural networks: a review. **Journal of medical systems**, Springer, v. 42, n. 11, p. 226, 2018. Citations on pages 27 and 42.

ARASU, A.; GANTI, V.; KAUSHIK, R. Efficient exact set-similarity joins. In: **Proceedings of the 32Nd International Conference on Very Large Data Bases**. VLDB Endowment, 2006. (VLDB '06), p. 918–929. Available: <http://dl.acm.org/citation.cfm?id=1182635.1164206>. Citation on page 35.

ARBELAEZ, P.; MAIRE, M.; FOWLKES, C.; MALIK, J. Contour detection and hierarchical image segmentation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 33, n. 5, p. 898–916, 2011. Citation on page 40.

ARTHUR, D.; VASSILVITSKII, S. K-means++: The advantages of careful seeding. In: **ACM-SIAM SODA**. [s.n.], 2007. p. 1027–1035. ISBN 978-0-898716-24-5. Available: <http://dl.acm.org/citation.cfm?id=1283383.1283494>. Citation on page 68.

AVALHAIS, L. P. S.; RODRIGUES, J.; TRAINA, A. J. M. Fire detection on unconstrained videos using color-aware spatial modeling and motion flow. In: **2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)**. [S.l.: s.n.], 2016. p. 913–920. ISSN 2375-0197. Citations on pages 44 and 45.

AVNI, U.; GREENSPAN, H.; KONEN, E.; SHARON, M.; GOLDBERGER, J. X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words. **Medical Imaging, IEEE Transactions on**, v. 30, n. 3, p. 733–746, March 2011. ISSN 0278-0062. Citation on page 38.

BABENKO, A.; SLESAREV, A.; CHIGORIN, A.; LEMPITSKY, V. Neural codes for image retrieval. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 584–599. Citation on page 42.

Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 12, p. 2481–2495, Dec 2017. ISSN 0162-8828. Citation on page 43.

BARABASI, A. The origin of bursts and heavy tails in human dynamics. **Nature**, v. 435, n. 7039, p. 207–211, 2005. Citation on page 129.

BARBARA, D.; CHEN, P. Fractal mining-self similarity-based clustering and its applications. In: **Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer, 2009. p. 573–589. Citation on page 63.

BAY, H.; ESS, A.; TUYTELAARS, T.; GOOL, L. V. Speeded-up robust features (surf). **Computer Vision and Image Understanding**, v. 110, n. 3, p. 346 – 359, 2008. ISSN 1077-3142. Similarity Matching in Computer Vision and Multimedia. Available: <http://www.sciencedirect.com/science/article/pii/S1077314207001555>. Citation on page 34.

BEDO, M. V. N.; BLANCO, G.; OLIVEIRA, W. D.; CAZZOLATO, M. T.; COSTA, A. F.; JR., J. F. R.; TRAINA, A. J. M.; Traina Jr., C. Techniques for effective and efficient fire detection from social media images. In: **ICEIS**. [s.n.], 2015. p. 34–45. Available: <http://dx.doi.org/10.5220/0005341500340045>. Citations on pages 27, 46, 61, 63, and 74.

BEDO, M. V. N.; SANTOS, L. F. D.; OLIVEIRA, W. D. de; BLANCO, G.; TRAINA, A. J. M.; FRADE, M. A.; MARQUES, P. M. de A.; JR., C. T. Color and texture influence on computer-aided diagnosis of dermatological ulcers. In: **IEEE CBMS**. [S.l.: s.n.], 2015. p. 109–114. Citations on pages 29, 73, and 74.

BENJAMIN, S. G.; RADHAKRISHNAN, B.; NIDHIN, T.; SURESH, L. P. Extraction of fire region from forest fire images using color rules and texture analysis. In: IEEE. **2016 International Conference on Emerging Technological Trends (ICETT)**. [S.l.], 2016. p. 1–7. Citations on pages 44 and 45.

BERGH, M. Van den; BOIX, X.; ROIG, G.; CAPITANI, B. de; GOOL, L. V. Seeds: Superpixels extracted via energy-driven sampling. In: SPRINGER. **European conference on computer vision**. [S.l.], 2012. p. 13–26. Citation on page 40.

BESSI, A.; PETRONI, F.; VICARIO, M. D.; ZOLLO, F.; ANAGNOSTOPOULOS, A.; SCALA, A.; CALDARELLI, G.; QUATTROCIOCCHI, W. Viral misinformation: The role of homophily and polarization. p. 355–356, 2015. Citation on page 127.

BLANCO, G.; BEDO, M. V. N.; CAZZOLATO, M. T.; SANTOS, L. F. D.; JORGE, A. E. S.; TRAINA, C.; AZEVEDO-MARQUES, P. M.; TRAINA, A. J. M. A label-scaled similarity measure for content-based image retrieval. In: **IEEE ISM**. [S.l.: s.n.], 2016. p. 20–25. Citations on pages 27, 29, 39, 46, 74, 75, 79, 88, and 94.

BRODER, A. On the resemblance and containment of documents. In: **Compression and Complexity of Sequences 1997. Proceedings**. [S.l.: s.n.], 1997. p. 21–29. Citation on page 35.

BUONCOMPAGNI, S.; MAIO, D.; MALTONI, D.; PAPI, S. Saliency-based keypoint selection for fast object detection and matching. **Pattern Recognition Letters**, v. 62, p. 32 – 40, 2015. ISSN 0167-8655. Available: <http://www.sciencedirect.com/science/article/pii/S0167865515001464>. Citation on page 34.

CAETANO, C.; AVILA, S.; aES, S. G.; ARAúJO, A. d. A. Representing local binary descriptors with bossanova for visual recognition. In: **ACM SAC**. [s.n.], 2014. p. 49–54. ISBN 978-1-4503-2469-4. Available: <http://doi.acm.org/10.1145/2554850.2555058>. Citation on page 61.

CARREIRA, J.; SMINCHISESCU, C. Cpmc: Automatic object segmentation using constrained parametric min-cuts. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 34, n. 7, p. 1312–1328, 2012. Citation on page 40.

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: **SBBD Proceedings - Satellite Events of the 32nd Brazilian Symposium on Databases**. [S.l.]: SBC, 2017. p. 213–223. ISBN 978-85-7669-399-4. Citation on page 108.

CAZZOLATO, M. T.; BEDO, M. V. N.; COSTA, A. F.; SOUZA, J. A. de; TRAINA JR., C.; RODRIGUES JR., J. F.; TRAINA, A. J. M. Unveiling smoke in social images with the smokeblock approach. In: **Proceedings of the 31st Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2016. (SAC '16), p. 49–54. ISBN 978-1-4503-3739-7. Available: <http://doi.acm.org/10.1145/2851613.2851634>. Citation on page 27.

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. Citation on page 27.

CELIK, T.; DEMIREL, H. Fire detection in video sequences using a generic color model. **Fire Safety Journal**, v. 44, n. 2, p. 147–158, 2009. ISSN 0379-7112. Citations on pages 29, 44, 48, and 54.

CHEN, C.-S.; YEH, C.-W.; YIN, P.-Y. A novel fourier descriptor based image alignment algorithm for automatic optical inspection. **J. Visual Communication and Image Representation**, v. 20, n. 3, p. 178–189, 2009. Citation on page 34.

CHEN, T.; YIN, Y. H.; HUANG, S. F.; YE, Y. T. The smoke detection for early fire-alarming system base on video processing. In: **IIH-MSP**. [S.l.: s.n.], 2006. p. 427–430. Citation on page 28.

CHEN, T.-H.; WU, P.-H.; CHIOU, Y.-C. An early fire-detection method based on image processing. In: **ICIP**. [S.l.: s.n.], 2004. v. 3, p. 1707–1710. ISSN 1522-4880. Citations on pages 29, 44, 47, 48, and 54.

CHENG, H.; TAN, P.-N.; POTTER, C.; KLOOSTER, S. Detection and characterization of anomalies in multivariate time series. In: SIAM. **SDM**. [S.l.], 2009. v. 9, p. 413–424. Citations on pages 129 and 130.

CHINO, D. Y.; COSTA, A. F.; TRAINA, A. J.; FALOUTSOS, C. Voltime: Unsupervised anomaly detection on users' online activity volume. In: SIAM. **Proceedings of the 2017 SIAM International Conference on Data Mining**. [S.l.], 2017. p. 108–116. Citations on pages 108, 127, 128, 130, 131, 133, 134, 135, 137, 139, 140, and 142.

CHINO, D. Y. T.; AVALHAIS, L. P. S.; JR., J. F. R.; TRAINA, A. J. M. Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. In: **28th Conference on Graphics, Patterns and Images (SIBGRAPI2015)**. [S.l.: s.n.], 2015. p. 95–102. Citations on pages 47, 49, 50, 53, 54, 55, 56, 57, 58, 59, and 108.

CHINO, D. Y. T.; SCABORA, L. C.; CAZZOLATO, M. T.; JORGE, A. E. S.; TRAINA, C.; TRAINA, A. J. M. Icarus: Retrieving skin ulcer images through bag-of-signatures. In: **2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)**. [S.l.: s.n.], 2018. p. 82–87. ISSN 2372-9198. Citations on pages 73, 76, 78, 88, and 108.

____. Segmenting skin ulcers and measuring the wound area using deep convolutional networks [submitted]. **Journal of Computer Methods and Programs in Biomedicine**, p. 1–22, 2019. Citations on pages 87 and 108.

CHINO, D. Y. T.; SCABORA, L. C.; TRAINA JR., C.; TRAINA, A. J. M. Boss: Image retrieval using bag-of-superpixels signatures. In: **Proceedings of the 33rd Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2018. (SAC '18), p. 309–312. ISBN 978-1-4503-5191-1. Available: <http://doi.acm.org/10.1145/3167132.3167374>. Citations on pages 34, 61, 66, 71, 79, and 108.

CHO, J.; GARCIA-MOLINA, H. Estimating frequency of change. **ACM TOIT**, v. 3, n. 3, p. 256–290, 2003. ISSN 15335399. Citation on page 129.

COSTA, A. F.; YAMAGUCHI, Y.; TRAINA, A. J. M.; Traina Jr., C.; FALOUTSOS, C. RSC: Mining and Modeling Temporal Activity in Social Media. In: **KDD**. [S.l.: s.n.], 2015. p. 269–278. Citations on pages 129, 130, and 132.

COSTA, A. F.; YAMAGUCHI, Y.; TRAINA, A. J. M.; FALOUTSOS, C. *et al.* Modeling temporal activity to detect anomalous behavior in social media. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM, v. 11, n. 4, p. 49, 2017. Citation on page 129.

DATTA, R.; JOSHI, D.; LI, J.; WANG, J. Z. Image retrieval: Ideas, influences, and trends of the new age. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 40, n. 2, p. 5:1–5:60, May 2008. ISSN 0360-0300. Available: <http://doi.acm.org/10.1145/1348246.1348248>. Citation on page 33.

DESELAERS, T.; KEYSERS, D.; NEY, H. Features for image retrieval: an experimental comparison. **Information Retrieval**, Springer Netherlands, v. 11, n. 2, p. 77–107, 2008. ISSN 1386-4564. Available: <http://dx.doi.org/10.1007/s10791-007-9039-3>. Citations on pages 27 and 32.

DESERNO, T.; ANTANI, S.; LONG, R. Ontology of gaps in content-based image retrieval. **Journal of Digital Imaging**, Springer-Verlag, v. 22, n. 2, p. 202–215, 2009. ISSN 0897-1889. Available: <http://dx.doi.org/10.1007/s10278-007-9092-x>. Citation on page 31.

DEVINENI, P.; KOUTRA, D.; FALOUTSOS, M.; FALOUTSOS, C. If walls could talk: Patterns and anomalies in facebook wallposts. In: **IEEE/ACM ASONAM**. [s.n.], 2015. p. 367–374. Available: <http://doi.acm.org/10.1145/2808797.2808880>. Citations on pages 62 and 134.

DIMITROVSKI, I.; KOCEV, D.; LOSKOVSKA, S.; DZEROSKI, S. Improving bag-of-visual-words image retrieval with predictive clustering trees. **Information Sciences**, Elsevier, v. 329, p. 851–865, 2016. Citation on page 38.

DORILEO, E. A. G.; FRADE, M. A. C.; RANGAYYAN, R. M.; AZEVEDO-MARQUES, P. M. Segmentation and analysis of the tissue composition of dermatological ulcers. In: **CCECE**. [S.l.: s.n.], 2010. p. 1–4. Citations on pages 29, 40, 45, 73, 74, 87, 88, and 95.

DORILEO, E. A. G.; FRADE, M. A. C.; ROSELINO, A. M. F.; RANGAYYAN, R. M.; AZEVEDO-MARQUES, P. M. Color image processing and content-based image retrieval techniques for the analysis of dermatological lesions. In: **IEEE EMBC**. [S.l.: s.n.], 2008. p. 1230–1233. ISSN 1094-687X. Citations on pages 45, 73, 75, 87, and 89.

DOW, P. A.; ADAMIC, L. A.; FRIGGERI, A. The Anatomy of Large Facebook Cascades. In: **ICWSM**. [S.l.: s.n.], 2013. p. 145–154. Citation on page 127.

ECKMANN, J.-P.; MOSES, E.; SERGI, D. Entropy of dialogues creates coherent structures in e-mail traffic. **PNAS**, v. 101, n. 7, p. 14333–14337, 2004. ISSN 0027-8424. Citation on page 129.

ENDRES, I.; HOIEM, D. Category-independent object proposals with diverse ranking. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 36, n. 2, p. 222–234, 2014. Citation on page 40.

EVANS, H. L.; LOBER, W. B. A pilot use of patient-generated wound data to improve post-discharge surgical site infection monitoring. **JAMA surgery**, American Medical Association, v. 152, n. 6, p. 595–596, 2017. Citation on page 29.

FAKHRAEI, S.; FOULDS, J.; SHASHANKA, M.; GETOOR, L. Collective Spammer Detection in Evolving Multi-Relational Social Networks. In: **KDD**. [S.l.: s.n.], 2013. p. 1769–1778. ISBN 9781450336642. Citation on page 127.

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. Citation on page 34.

FRAIDEINBERZE, A. C.; RODRIGUES, J. F.; CORDEIRO, R. L. F. Effective and unsupervised fractal-based feature selection for very large datasets: Removing linear and non-linear attribute correlations. In: **IEEE ICDMW**. [S.l.: s.n.], 2016. p. 615–622. Citation on page 63.

FU, X.; WEI, W. Centralized binary patterns embedded with image euclidean distance for facial expression recognition. In: **Proceedings of the 2008 Fourth International Conference on Natural Computation - Volume 04**. Washington, DC, USA: IEEE Computer Society, 2008. (ICNC '08), p. 115–119. ISBN 978-0-7695-3304-9. Available: <http://dx.doi.org/10.1109/ICNC.2008.94>. Citation on page 34.

GHOLAMI, P.; AHMADI-PAJOUH, M. A.; ABOLFTAHI, N.; HAMARNEH, G.; KAYVANRAD, M. Segmentation and measurement of chronic wounds for bioprinting. **IEEE Journal of Biomedical and Health Informatics**, v. 22, n. 4, p. 1269–1277, July 2018. ISSN 2168-2194. Citations on pages 27 and 29.

GONG, Y.; WANG, L.; GUO, R.; LAZEBNIK, S. Multi-scale orderless pooling of deep convolutional activation features. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 392–407. Citation on page 42.

GONZALEZ, R.; WOODS, R. **Digital Image Processing**. Pearson/Prentice Hall, 2008. ISBN 9780131687288. Available: <https://books.google.com.br/books?id=8uGOnjRGEzoC>. Citation on page 33.

GOYAL, M.; YAP, M. H.; REEVES, N. D.; RAJBHANDARI, S.; SPRAGG, J. Fully convolutional networks for diabetic foot ulcer segmentation. In: IEEE. **2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.], 2017. p. 618–623. Citation on page 88.

GUERRA, P. H. C.; VELOSO, A.; Meira Jr., W.; ALMEIDA, V. From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In: **KDD**. [S.l.: s.n.], 2011. p. 150–158. ISBN 9781450308137. Citation on page 127.

GUNNEMANN, S.; GUNNEMANN, N.; FALOUTSOS, C. Detecting Anomalies in Dynamic Rating Data: a Robust Probabilistic Model for Rating Evolution. In: **KDD**. [S.l.: s.n.], 2014. p. 841–850. ISBN 9781450329569. Citations on pages 129 and 130.

GUO, Z.; ZHANG, D.; ZHANG, D. A completed modeling of local binary pattern operator for texture classification. **Image Processing, IEEE Transactions on**, v. 19, n. 6, p. 1657–1663, June 2010. ISSN 1057-7149. Citation on page 34.

HA, C.; HWANG, U.; JEON, G.; CHO, J.; JEONG, J. Vision-based fire detection algorithm using optical flow. In: **CISIS**. [S.l.: s.n.], 2012. p. 526–530. Citation on page 48.

HAFNER, J.; SAWHNEY, H.; EQUITZ, W.; FLICKNER, M.; NIBLACK, W. Efficient color histogram indexing for quadratic form distance functions. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 17, n. 7, p. 729–736, Jul 1995. ISSN 0162-8828. Citation on page 32.

HARALICK, R.; 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. Citation on page 33.

HARE, J. S.; LEWIS, P. H.; ENSER, P. G.; SANDOM, C. J. Mind the gap: another look at the problem of the semantic gap in image retrieval. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Multimedia Content Analysis, Management, and Retrieval 2006**. [S.l.], 2006. v. 6073, p. 607309. Citation on page 27.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. Citation on page 42.

HOEL, P. G.; PORT, S. C.; STONE, C. J. **Introduction to Stochastic Processes**. [S.l.]: Waveland Pr. Inc., 1986. 203 p. ISBN 0-88133-267-4. Citation on page 129.

HOOI, B.; SHAH, N.; BEUTEL, A.; GUNNEMAN, S.; AKOGLU, L.; KUMAR, M.; MAKHIJA, D.; FALOUTSOS, C. Birdnest: Bayesian inference for ratings-fraud detection. In: SIAM. **SDM**. [S.l.], 2016. v. 16, p. 495–503. Citations on pages 127, 130, and 140.

HOSNY, K. M. Fast computation of accurate zernike moments. **J. Real-Time Image Processing**, v. 3, n. 1-2, p. 97–107, 2008. Citation on page 34.

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. Citation on page 28.

IHLER, A.; HUTCHINS, J.; SMYTH, P. Adaptive event detection with time-varying poisson processes. In: **KDD**. [S.l.: s.n.], 2006. p. 207–216. ISBN 1595933395. Citation on page 129.

JEGOU, H.; DOUZE, M.; SCHMID, C. Hamming embedding and weak geometric consistency for large scale image search. In: **ECCV**. [s.n.], 2008. p. 304–317. ISBN 978-3-540-88681-5. Available: <http://dx.doi-org.ez67.periodicos.capes.gov.br/10.1007/978-3-540-88682-2_24>. Citations on pages 63 and 80.

JEGOU, H.; HARZALLAH, H.; SCHMID, C. A contextual dissimilarity measure for accurate and efficient image search. In: **Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on**. [S.l.: s.n.], 2007. p. 1–8. ISSN 1063-6919. Citation on page 37.

JEGOU, H.; PERRONNIN, F.; DOUZE, M.; SANCHEZ, J.; PEREZ, P.; SCHMID, C. Aggregating local image descriptors into compact codes. **IEEE TPAMI**, v. 34, n. 9, p. 1704–1716, 2012. Available: <https://doi.org/10.1109/TPAMI.2011.235>. Citations on pages 36 and 61.

JIANG, Y.-G.; NGO, C.-W.; YANG, J. Towards optimal bag-of-features for object categorization and semantic video retrieval. In: **Proceedings of the 6th ACM International Conference on Image and Video Retrieval**. New York, NY, USA: ACM, 2007. (CIVR '07), p. 494–501. ISBN 978-1-59593-733-9. Available: <http://doi.acm.org/10.1145/1282280.1282352>. Citation on page 37.

JING, F.; LI, M.; ZHANG, H.-J.; ZHANG, B. An efficient and effective region-based image retrieval framework. **IEEE Transactions on Image Processing**, IEEE, v. 13, n. 5, p. 699–709, 2004. Citation on page 39.

JOACHIMS, T. Text categorization with suport vector machines: Learning with many relevant features. In: **Proceedings of the 10th European Conference on Machine Learning**. London, UK, UK: Springer-Verlag, 1998. (ECML '98), p. 137–142. ISBN 3-540-64417-2. Available: <http://dl.acm.org/citation.cfm?id=645326.649721>. Citation on page 36.

JUNEJA, M.; VEDALDI, A.; JAWAHAR, C.; ZISSERMAN, A. Blocks that shout: Distinctive parts for scene classification. In: **IEEE CVPR**. [S.l.: s.n.], 2013. p. 923–930. Citations on pages 34 and 40.

KAWAHARA, J.; HAMARNEH, G. Multi-resolution-tract cnn with hybrid pretrained and skin-lesion trained layers. In: SPRINGER. **International Workshop on Machine Learning in Medical Imaging**. [S.l.], 2016. p. 164–171. Citations on pages 41 and 88.

KAZMI, I. K.; YOU, L.; ZHANG, J. J. A Survey of 2D and 3D Shape Descriptors. In: **Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference**. [S.l.: s.n.], 2013. p. 1–10. Citation on page 34.

KIM, A. K. Y.-H.; JEONG, H.-Y. Rgb color model based the fire detection algorithm in video sequences on wireless sensor network. **International Journal of Distributed Sensor Networks**, p. 10, 2014. Citation on page 47.

KIM, J.; BAIK, S.; KIM, K.; JUNG, C.; KIM, W. A cartoon image classification system using mpeg-7 descriptors. In: DENG, H.; MIAO, D.; LEI, J.; WANG, F. (Ed.). **Artificial Intelligence and Computational Intelligence**. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 7003). p. 368–375. ISBN 978-3-642-23886-4. Available: <http://dx.doi.org/10.1007/978-3-642-23887-1_46>. Citation on page 33.

KLEINBERG, J. Bursty and Hierarchical Structure in Streams. In: **KDD**. Edmonton, Alberta, Canada: [s.n.], 2003. p. 373–397. ISBN 158113567X. ISSN 13845810. Citation on page 130.

KRISHNAN, N. C.; COOK, D. J. Activity recognition on streaming sensor data. **PMC**, v. 10, p. 138–154, 2014. ISSN 15741192. Citation on page 129.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1**. USA: Curran Associates Inc., 2012. (NIPS'12), p. 1097–1105. Available: <http://dl.acm.org/citation.cfm?id=2999134.2999257>. Citations on pages 42 and 88.

KUMAR, M. D.; BABAIE, M.; ZHU, S.; KALRA, S.; TIZHOOSH, H. R. A comparative study of cnn, bovw and lbp for classification of histopathological images. In: IEEE. **2017 IEEE Symposium Series on Computational Intelligence (SSCI)**. [S.l.], 2017. p. 1–7. Citation on page 36.

KUMAR, S.; HOOI, B.; MAKHIJA, D.; KUMAR, M.; FALOUTSOS, C.; SUBRAHMANIAN, V. Rev2: Fraudulent user prediction in rating platforms. In: ACM. **Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining**. [S.l.], 2018. p. 333–341. Citation on page 129.

LAPPAS, T.; VIEIRA, M. R.; GUNOPULOS, D.; TSOTRAS, V. J. On the spatiotemporal burstiness of terms. In: **VLDB**. [S.l.: s.n.], 2012. p. 836–847. ISBN 2150-8097. ISSN 2150-8097. Citations on pages 129 and 130.

LAZEBNIK, S.; SCHMID, C.; PONCE, J. A sparse texture representation using local affine regions. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 27, n. 8, p. 1265–1278, Aug 2005. ISSN 0162-8828. Citation on page 64.

LESKOVEC, J.; BACKSTROM, L.; KLEINBERG, J. Meme-tracking and the dynamics of the news cycle. In: **KDD**. [S.l.]: ACM Press, 2009. p. 497–505. ISBN 9781605584959. Citation on page 127.

LI, Z.; CHEN, J. Superpixel segmentation using linear spectral clustering. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2015. p. 1356–1363. Citation on page 40.

LI, Z.; WU, X. M.; CHANG, S. F. Segmentation using superpixels: A bipartite graph partitioning approach. In: **CVPR**. [S.l.: s.n.], 2012. p. 789–796. ISSN 1063-6919. Citation on page 40.

LIAO, W.-H. Region description using extended local ternary patterns. In: **Pattern Recognition (ICPR), 2010 20th International Conference on**. [S.l.: s.n.], 2010. p. 1003–1006. ISSN 1051-4651. Citation on page 34.

LIU, Y.; ZHANG, D.; LU, G.; MA, W.-Y. A survey of content-based image retrieval with high-level semantics. **Pattern Recognition**, v. 40, n. 1, p. 262 – 282, 2007. ISSN 0031-3203. Available: <http://www.sciencedirect.com/science/article/pii/S0031320306002184>. Citation on page 27.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 3431–3440. Citation on page 43.

LOWE, D. Object recognition from local scale-invariant features. In: **Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on**. [S.l.: s.n.], 1999. v. 2, p. 1150–1157 vol.2. Citation on page 34.

MALMGREN, R. D.; HOFMAN, J. M.; AMARAL, L. A. N.; WATTS, D. J. Characterizing Individual Communication Patterns. In: **KDD**. [S.l.]: ACM Press, 2009. p. 607–616. ISBN 9781605584959. Citations on pages 129 and 130.

MATSUBARA, Y.; SAKURAI, Y.; PRAKASH, B. A.; LI, L.; FALOUTSOS, C. Rise and fall patterns of information diffusion: model and implications. In: **KDD**. [S.l.: s.n.], 2012. p. 6–14. ISBN 9781450314626. Citation on page 127.

MORTON, L. M.; PHILLIPS, T. J. Wound healing and treating wounds: Differential diagnosis and evaluation of chronic wounds. **Journal of the American Academy of Dermatology**, v. 74, n. 4, p. 589 – 605, 2016. ISSN 0190-9622. Citations on pages 29 and 88.

MUHAMMAD, K.; AHMAD, J.; BAIK, S. W. Early fire detection using convolutional neural networks during surveillance for effective disaster management. **Neurocomputing**, v. 288, p. 30 – 42, 2018. ISSN 0925-2312. Learning System in Real-time Machine Vision. Available: <http://www.sciencedirect.com/science/article/pii/S0925231217319203>. Citations on pages 44 and 45.

NAVARRO, F.; ESCUDERO-VINOLO, M.; BESCOS, J. Accurate segmentation and registration of skin lesion images to evaluate lesion change. **IEEE Journal of Biomedical and Health Informatics**, p. 2168–2194, 2018. ISSN 2168-2194. Citations on pages 29 and 88.

NESSO, M. R.; CAZZOLATO, M. T.; SCABORA, L. C.; OLIVEIRA, P. H.; SPADON, G.; SOUZA, J. A. de; OLIVEIRA, W. D.; CHINO, D. Y.; RODRIGUES, J. F.; TRAINA, A. J. *et al.* Rafiki: Retrieval-based application for imaging and knowledge investigation. In: IEEE. **2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)**. [S.l.], 2018. p. 71–76. Citation on page 109.

NEUBERT, P.; PROTZEL, P. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In: IEEE. **2014 22nd International Conference on Pattern Recognition**. [S.l.], 2014. p. 996–1001. Citation on page 41.

NEVEOL, A.; DESERNO, T. M.; DARMONI, S. J.; GULD, M. O.; ARONSON, A. R. Natural language processing versus content-based image analysis for medical document retrieval. **Journal of the American Society for Information Science and Technology**, Wiley Subscription Services, Inc., A Wiley Company, v. 60, n. 1, p. 123–134, 2009. ISSN 1532-2890. Available: <http://dx.doi.org/10.1002/asi.20955>. Citation on page 31.

NOH, H.; ARAUJO, A.; SIM, J.; WEYAND, T.; HAN, B. Largescale image retrieval with attentive deep local features. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 3456–3465. Citations on pages 42 and 80.

NOH, H.; HONG, S.; HAN, B. Learning deconvolution network for semantic segmentation. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1520–1528. Citation on page 43.

ODUNCU, H.; HOPPE, A.; CLARK, M.; WILLIAMS, R. J.; HARDING, K. G. Analysis of skin wound images using digital color image processing: A preliminary communication. **IJEW**, v. 3, n. 3, p. 151–156, 2004. Citation on page 74.

OHLANDER, R.; PRICE, K.; REDDY, D. R. Picture segmentation using a recursive region splitting method. **Computer Graphics and Image Processing**, Elsevier, v. 8, n. 3, p. 313–333, 1978. Citation on page 40.

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**, v. 24, n. 7, p. 971–987, Jul 2002. ISSN 0162-8828. Citation on page 34.

OKAWA, M. From bovw to vlad with kaze features: Offline signature verification considering cognitive processes of forensic experts. **Pattern Recognition Letters**, v. 113, p. 75 – 82, 2018. ISSN 0167-8655. Integrating Biometrics and Forensics. Available: <http://www.sciencedirect.com/science/article/pii/S016786551830206X>. Citation on page 36.

Oliveira, P. H.; Scabora, L. C.; Cazzolato, M. T.; Oliveira, W. D.; Traina, A. J. M.; Traina, C. Efficiently indexing multiple repositories of medical image databases. In: **2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)**. [S.l.: s.n.], 2017. p. 286–291. ISSN 2372-9198. Citation on page 27.

ONEATA, D.; REVAUD, J.; VERBEEK, J.; SCHMID, C. Spatio-temporal object detection proposals. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). **Computer Vision – ECCV 2014**. Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8691). p. 737–752. ISBN 978-3-319-10577-2. Available: <http://dx.doi.org/10.1007/978-3-319-10578-9_48>. Citation on page 34.

OQUAB, M.; BOTTOU, L.; LAPTEV, I.; SIVIC, J. Learning and transferring mid-level image representations using convolutional neural networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 1717–1724. Citation on page 42.

ORNAGER, S.; LUND, H. Images in social media: Categorization and organization of images and their collections. **Synthesis Lectures on Information Concepts, Retrieval, and Services**, Morgan & Claypool Publishers, v. 10, n. 1, p. i–101, 2018. Citation on page 27.

OTTONI, R.; CASAS, D. L.; PESCE, J. P.; Meira Jr., W.; WILSON, C.; MISLOVE, A.; ALMEIDA, V. Of Pins and Tweets: Investigating How Users Behave Across Image-and Text-Based Social Networks. In: **ICWSM**. [S.l.: s.n.], 2014. p. 386–395. ISBN 9781577356578. Citation on page 129.

PAN, J.; LIU, Y.; LIU, X.; HU, H. Discriminating bot accounts based solely on temporal features of microblog behavior. **Physica A**, 2016. ISSN 03784371. Citations on pages 129 and 130.

PAPADOPOULOS, S.; ZIGKOLIS, C.; KOMPATSIARIS, Y.; VAKALI, A. Cluster-based landmark and event detection for tagged photo collections. **MultiMedia, IEEE**, v. 18, n. 1, p. 52–63, Jan 2011. ISSN 1070-986X. Citation on page 37.

PARK, S. H.; YUN, I. D.; LEE, S. U. Color image segmentation based on 3-d clustering: morphological approach. **Pattern Recognition**, Elsevier, v. 31, n. 8, p. 1061–1076, 1998. Citation on page 40.

PEDROSA, G.; RAHMAN, M.; ANTANI, S.; DEMNER-FUSHMAN, D.; LONG, L.; TRAINA, A. Integrating visual words as bunch of n-grams for effective biomedical image classification. In: **Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on**. [S.l.: s.n.], 2014. p. 431–436. Citation on page 38.

PENATTI, O.; VALLE, E.; TORRES, R. da S. Encoding spatial arrangement of visual words. In: MARTIN, C. S.; KIM, S.-W. (Ed.). **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications**. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 7042). p. 240–247. ISBN 978-3-642-25084-2. Available: <http://dx.doi.org/10.1007/978-3-642-25085-9_28>. Citation on page 38.

PEREYRA, L. C.; PEREIRA, S. M.; SOUZA, J. P.; FRADE, M. A. C.; RANGAYYAN, R. M.; AZEVEDO-MARQUES, P. M. Characterization and pattern recognition of color images of dermatological ulcers - a pilot study. **The Computer Science Journal of Moldova**, v. 22, n. 2, p. 211–235, 2014. Available: <http://www.math.md/publications/csjm/issues/v22-n2/11672/>. Citations on pages 27, 40, 46, 73, 74, 87, 88, 95, and 96.

PERRONNIN, F.; LIU, Y.; SANCHEZ, J.; POIRIER, H. Large-scale image retrieval with compressed Fisher vectors. In: **Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on**. [S.l.: s.n.], 2010. p. 3384–3391. ISSN 1063-6919. Citation on page 36.

PFEIFFER, D.; FRANKE, U. Efficient representation of traffic scenes by means of dynamic stixels. In: IEEE. **2010 IEEE Intelligent Vehicles Symposium**. [S.l.], 2010. p. 217–224. Citation on page 40.

PHILBIN, J.; CHUM, O.; ISARD, M.; SIVIC, J.; ZISSERMAN, A. Object retrieval with large vocabularies and fast spatial matching. In: **CVPR**. [s.n.], 2007. p. 1–8. Available: <https://doi.org/10.1109/CVPR.2007.383172>. Citation on page 38.

PIRAS, L.; GIACINTO, G. Information fusion in content based image retrieval: A comprehensive overview. **Information Fusion**, Elsevier, v. 37, p. 50–60, 2017. Citation on page 74.

PONTI, M. A.; RIBEIRO, L. S. F.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: IEEE. **2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)**. [S.l.], 2017. p. 17–41. Citation on page 42.

QIAN, G.; SURAL, S.; GU, Y.; PRAMANIK, S. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In: ACM. **Proceedings of the 2004 ACM symposium on Applied computing**. [S.l.], 2004. p. 1232–1237. Citation on page 35.

QIU, T.; YAN, Y.; LU, G. An autoadaptive edge-detection algorithm for flame and fire image processing. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 61, n. 5, p. 1486–1493, 2012. No citation.

RADENOVIC, F.; ISCEN, A.; TOLIAS, G.; AVRITHIS, Y.; CHUM, O. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 5706–5715. Citation on page 36.

RAO, S. R.; MOBAHI, H.; YANG, A. Y.; SASTRY, S. S.; MA, Y. Natural image segmentation with adaptive texture and boundary encoding. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2009. p. 135–146. Citation on page 40.

RAYANA, S.; AKOGLU, L. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In: **KDD**. [S.l.: s.n.], 2015. p. 985–994. ISBN 9781450336642. Citations on pages 127 and 129.

RAZAVIAN, A. S.; AZIZPOUR, H.; SULLIVAN, J.; CARLSSON, S. Cnn features off-the-shelf: an astounding baseline for recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2014. p. 806–813. Citation on page 42.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 779–788. Citation on page 41.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **International Conference on Medical image computing and computer-assisted intervention**. [S.l.], 2015. p. 234–241. Citations on pages 43, 90, and 91.

ROSSI, L.; AKHLOUFI, M.; TISON, Y. On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics. **Fire Saf. J.**, v. 46, n. 1–2, p. 9–20, 2011. ISSN 0379-7112. Citations on pages 29, 44, 48, and 54.

RUDZ, S.; CHETEHOUNA, K.; HAFIANE, A.; LAURENT, H.; SéRO-GUILLAUME, O. Investigation of a novel image segmentation method dedicated to forest fire applications. **IET Meas. Sci. Technol.**, v. 24, n. 7, p. 75403, 2013. Citations on pages 29, 40, 44, 48, and 54.

SAIPULLAH, K.; KIM, D.-H. A robust texture feature extraction using the localized angular phase. **Multimedia Tools and Applications**, Springer US, v. 59, n. 3, p. 717–747, 2012. ISSN 1380-7501. Citation on page 33.

SANTOS, E. R. S.; LOPES, A. P. B.; VALLE, E. A.; ARAUJO, A. A. Vocabularios visuais para para recuperacao de informacao multimidia. In: **Simposio Brasileiro em Sistemas Multimidia e Web (WebMedia)**. [S.l.: s.n.], 2010. p. 21–24. Citation on page 37.

SANTOS, J. M. dos. **Descritores de imagens baseados em assinatura textual**. Phd Thesis (PhD Thesis) — Universidade Federal do Amazonas, Manaus, AM, 11 2016. Citation on page 38.

SANTOS, J. M. dos; MOURA, E. S. de; SILVA, A. S. da; CAVALCANTI, J. M. B.; TORRES, R. da S.; VIDAL, M. L. A. A signature-based bag of visual words method for image indexing and search. **Pattern Recognition Letters**, Elsevier, v. 65, p. 1–7, 2015. Citations on pages 38 and 62.

SANTOS, J. M. dos; MOURA, E. S. de; SILVA, A. S. da; TORRES, R. da S. Color and texture applied to a signature-based bag of visual words method for image retrieval. **MTA**, v. 76, n. 15, p. 16855–16872, 2017. Available: <https://doi.org/10.1007/s11042-016-3955-4>. Citations on pages 27, 38, 39, 61, 62, 67, 68, and 79.

SCHROEDER, M. **Fractals, chaos, power laws: Minutes from an infinite paradise**. [S.l.]: Dover Publications, Incorporated, 2012. Citations on pages 62 and 63.

SCULLEY, D. Web-scale k-means clustering. In: **WWW**. [s.n.], 2010. p. 1177–1178. ISBN 978-1-60558-799-8. Available: <http://doi.acm.org/10.1145/1772690.1772862>. Citation on page 68.

SEIXAS, J. L.; BARBON, S.; MANTOVANI, R. G. Pattern recognition of lower member skin ulcers in medical images with machine learning algorithms. In: **IEEE CBMS**. [S.l.: s.n.], 2015. p. 50–53. Citations on pages 45, 74, and 88.

SHABAN, A.; RABIEE, H.; FARAJTABAR, M.; GHAZVININEJAD, M. From local similarity to global coding: An application to image classification. In: **Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on**. [S.l.: s.n.], 2013. p. 2794–2801. ISSN 1063-6919. Citation on page 32.

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. ISBN 978-3-319-65172-9. Citations on pages 27, 41, and 45.

SHRIVASTAVA, N.; TYAGI, V. Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching. **Information Sciences**, v. 259, p. 212–224, 2014. ISSN 0020-0255. Available: <http://www.sciencedirect.com/science/article/pii/S0020025513006105>. Citation on page 31.

SIA, K. C.; CHO, J.; CHO, H.-K. Efficient monitoring algorithm for fast news alerts. **TKDE**, v. 19, n. 7, p. 950–961, 2007. Citation on page 129.

SIKORA, T. The mpeg-7 visual standard for content description-an overview. **Circuits and Systems for Video Technology, IEEE Transactions on**, v. 11, n. 6, p. 696–702, Jun 2001. ISSN 1051-8215. Citation on page 33.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **CoRR**, abs/1409.1556, 2014. Available: <http://arxiv.org/abs/1409.1556>. Citation on page 42.

SIVIC, J.; ZISSERMAN, A. Video google: a text retrieval approach to object matching in videos. In: **Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on**. [S.l.: s.n.], 2003. p. 1470–1477 vol.2. Citations on pages 27, 36, 37, 39, 61, 62, 68, and 79.

STEHLING, R. O.; NASCIMENTO, M. A.; FALCAO, A. X. A compact and efficient image retrieval approach based on border/interior pixel classification. In: **Proceedings of the Eleventh International Conference on Information and Knowledge Management**. New York, NY, USA: ACM, 2002. (CIKM '02), p. 102–109. ISBN 1-58113-492-4. Available: <http://doi.acm.org/10.1145/584792.584812>. Citation on page 33.

SUN, W.; KISE, K.; CHAMPEIL, Y. Similar fragment retrieval of animations by a bag-of-features approach. In: **Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on**. [S.l.: s.n.], 2012. p. 140–144. Citation on page 36.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Thirty-First AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2017. Citation on page 42.

TALIB, A.; MAHMUDDIN, M.; HUSNI, H.; GEORGE, L. E. A weighted dominant color descriptor for content-based image retrieval. **Journal of Visual Communication and Image Representation**, v. 24, n. 3, p. 345 – 360, 2013. ISSN 1047-3203. Available: <http://www.sciencedirect.com/science/article/pii/S1047320313000084>. Citation on page 33.

TORRES, R. D. S.; FALCAO, A. X. Content-based image retrieval: Theory and applications. **Revista de Informatica Teórica e Aplicada**, v. 13, p. 161–185, 2006. Citations on pages 27, 31, and 32.

TORRES, R. da S.; FALCAO, A. X. Contour salience descriptors for effective image retrieval and analysis. **Image Vision Comput.**, v. 25, n. 1, p. 3–13, 2007. Citation on page 34.

TRAINA, A. J. M.; ROMANI, L. A. S.; CORDEIRO, R. L. F.; SOUSA, E. P. M.; RIBEIRO, M. X.; AVILA, A. A. M. H.; JR, J. Z.; JR., J. F. R.; JR., C. T. How to find relevant patterns in climate data: an efficient and effective framework to mine climate time series and remote sensing images. In: **SIAM Annual Meeting 2010 (B. L. Keyfitz and L. N. Trefethen, eds.)**. Pittsburh, PA, USA: [s.n.], 2010. (SIAM 2010), p. 6. Citations on pages 63 and 67.

TSYTSARAU, M.; PALPANAS, T.; CASTELLANOS, M. Dynamics of News Events and Social Media Reaction. In: **KDD**. [S.l.]: ACM, 2014. p. 901–910. ISBN 9781450329569. Citations on pages 129 and 130.

TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: A survey. **Found. Trends. Comput. Graph. Vis.**, Now Publishers Inc., Hanover, MA, USA, v. 3, n. 3, p. 177–280, Jul. 2008. ISSN 1572-2740. Available: <http://dx.doi.org/10.1561/0600000017>. Citation on page 34.

TUYTELAARS, T.; SCHMID, C. Vector quantizing feature space with a regular lattice. In: **Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on**. [S.l.: s.n.], 2007. p. 1–8. ISSN 1550-5499. Citation on page 34.

VAHDATPOUR, A.; SARRAFZADEH, M. Unsupervised discovery of abnormal activity occurrences in multi-dimensional time series, with applications in wearable systems. In: SIAM. **SDM**. [S.l.], 2010. v. 10, p. 641–652. Citations on pages 129 and 130.

VARGHESE, A.; BALAKRISHNAN, K.; VARGHESE, R. R.; PAUL, J. S. Content-Based Image Retrieval of Axial Brain Slices Using a Novel LBP with a Ternary Encoding. **The Computer Journal**, Br Computer Soc, v. 57, n. 9, p. 1383 – 1394, 2014. Citation on page 31.

Vaz de Melo, P. O. S.; AKOGLU, L.; FALOUTSOS, C.; LOUREIRO, A. A. F. Surprising Patterns for the Call Duration Distribution of Mobile Phone Users. In: **PKDD**. [S.l.: s.n.], 2010. p. 354–369. Citations on pages 129, 130, and 134.

Vaz de Melo, P. O. S.; FALOUTSOS, C.; ASSUNÇÃO, R.; ALVEZ, R.; LOUREIRO, A. A. F. Universal and Distinct Properties of Communication Dynamics: How to Generate Realistic Inter-event Times. **TKDD**, v. 9, n. 3, p. 24:1–24:31, 2015. Citations on pages 129 and 130.

VIDAL, M. L. A.; CAVALCANTI, J. M.; MOURA, E. S. de; SILVA, A. S. da; TORRES, R. da S. Sorted dominant local color for searching large and heterogeneous image databases. In: IEEE. **Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)**. [S.l.], 2012. p. 1960–1963. Citation on page 38.

VINCENT, L.; SOILLE, P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, n. 6, p. 583–598, 1991. Citation on page 40.

WANG, J. Z.; LI, J.; WIEDERHOLD, G. Simplicity: Semantics-sensitive integrated matching for picture libraries. **IEEE TPAMI**, v. 23, n. 9, p. 947–963, 2001. Available: <https://doi.org/10.1109/34.955109>. Citation on page 63.

WANG, K.; YANG, L.; YANG, G.; LUO, X.; SU, K.; YIN, Y. Finger vein image retrieval via coding scale-varied superpixel feature. In: **ICMR**. [s.n.], 2017. p. 375–382. ISBN 978-1-4503-4701-3. Available: <http://doi.acm.org/10.1145/3078971.3078975>. Citation on page 40.

WEEKS, A. R.; HAGUE, G. E. Color segmentation in the hsi color space using the k-means algorithm. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Nonlinear Image Processing VIII**. [S.l.], 1997. v. 3026, p. 143–155. Citation on page 40.

WELTER, P.; FISCHER, B.; GUNTHER, R. W.; LEHMANN), T. M. D. (ne. Generic integration of content-based image retrieval in computer-aided diagnosis. **Computer Methods and Programs in Biomedicine**, v. 108, n. 2, p. 589 – 599, 2012. ISSN 0169-2607. Available: <http://www.sciencedirect.com/science/article/pii/S0169260711002288>. Citation on page 31.

WILSON, D. R.; MARTINEZ, T. R. Improved heterogeneous distance functions. **J. Artif. Intell. Res. (JAIR)**, v. 6, p. 1–34, 1997. Citation on page 35.

WONG, K.-M.; PO, L.-M.; CHEUNG, K.-W. Dominant color structure descriptor for image retrieval. In: **Image Processing, 2007. ICIP 2007. IEEE International Conference on**. [S.l.: s.n.], 2007. v. 6, p. VI – 365–VI – 368. ISSN 1522-4880. Citation on page 33.

WU, L.; HUA, X.-S.; YU, N.; MA, W.-Y.; LI, S. Flickr distance: A relationship measure for visual concepts. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 34, n. 5, p. 863–875, May 2012. ISSN 0162-8828. Citation on page 31.

Yandex, A. B.; Lempitsky, V. Aggregating local deep features for image retrieval. In: **2015 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015. p. 1269–1277. ISSN 2380-7504. Citation on page 42.

YANG, J.; JIANG, Y.; HAUPTMANN, A. G.; NGO, C. Evaluating bag-of-visual-words representations in scene classification. In: **ACM SIGMM MIR**. [s.n.], 2007. p. 197–206. Available: <http://doi.acm.org/10.1145/1290082.1290111>. Citation on page 62.

YANG, M.; KPALMA, K.; RONSIN, J. *et al.* A survey of shape feature extraction techniques. **Pattern recognition**, p. 43–90, 2008. Citation on page 34.

Yu, L.; Chen, H.; Dou, Q.; Qin, J.; Heng, P. Automated melanoma recognition in dermoscopy images via very deep residual networks. **IEEE Transactions on Medical Imaging**, v. 36, n. 4, p. 994–1004, April 2017. ISSN 0278-0062. Citations on pages 41 and 88.

Yuan, Y.; Chao, M.; Lo, Y. Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance. **IEEE Transactions on Medical Imaging**, v. 36, n. 9, p. 1876–1886, Sep. 2017. ISSN 0278-0062. Citations on pages 41 and 88.

ZENG, J.; DUAN, J.; CAO, W.; WU, C. Topics modeling based on selective zipf distribution. **Expert Syst. Appl.**, v. 39, n. 7, p. 6541–6546, 2012. Available: <https://doi.org/10.1016/j.eswa.2011.12.051>. Citation on page 62.

ZHANG, C.; NI, Z.; NI, L.; TANG, N. Feature selection method based on multi-fractal dimension and harmony search algorithm and its application. **IJSS**, Taylor and Francis, v. 47, n. 14, p. 3476–3486, 2016. Available: <http://dx.doi.org/10.1080/00207721.2015.1086931>. Citation on page 63.

ZHANG, C. M.; PAXSON, V. Detecting and Analyzing Automated Activity on Twitter. **LNCS**, v. 6579, p. 102–111, 2011. Citation on page 130.

ZHANG, D.; LU, G. Review of shape representation and description techniques. **Pattern Recognition**, v. 37, n. 1, p. 1 – 19, 2004. ISSN 0031-3203. Available: <http://www.sciencedirect.com/science/article/pii/S0031320303002759>. Citation on page 34.

ZHANG, M. L.; ZHOU, Z. H. A review on multi-label learning algorithms. **IEEE TKDE**, v. 26, n. 8, p. 1819–1837, Aug 2014. ISSN 1041-4347. Citation on page 78.

ZHANG, S.; YANG, M.; COUR, T.; YU, K.; METAXAS, D. N. Query Specific Rank Fusion for Image Retrieval. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 37, n. 4, p. 803–815, 2015. ISSN 0162-8828. Citation on page 63.

ZHAO, C.; SHI, W.; DENG, Y. A new hausdorff distance for image matching. **Pattern Recognition Letters**, v. 26, n. 5, p. 581 – 586, 2005. ISSN 0167-8655. Available: <http://www.sciencedirect.com/science/article/pii/S0167865504002466>. Citation on page 35.

ZHAO, J.; ZHANG, Z.; HAN, S.; QU, C.; YUAN, Z.; ZHANG, D. Svm based forest fire detection using static and dynamic features. **Computer Science and Information Systems**, v. 8, n. 3, p. 821–841, 2011. Citation on page 48.

ZHENG, L.; YANG, Y.; TIAN, Q. Sift meets cnn: A decade survey of instance retrieval. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 40, n. 5, p. 1224–1244, 2018. Citation on page 27.

ZHU, H.; MENG, F.; CAI, J.; LU, S. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. **Journal of Visual Communication and Image Representation**, v. 34, p. 12 – 27, 2016. ISSN 1047-3203. Available: <http://www.sciencedirect.com/science/article/pii/S1047320315002035>. Citations on pages 39 and 40.

APPENDIX

A

VOLTIME

The ability of following and analyzing the users behavior in specific situations can bring relevant insight for detecting bias or even forecast changing of a tendency. Is it possible to spot review frauds and spamming on social media and online stores? In this Chapter we analyze the joint distribution of the inter-arrival times and volume of events such as comments and online reviews and show that it is possible to accurately rank and detect suspicious users such as spammers, bots and fraudsters. We propose VolTime, a generative model that fits well the inter-arrival time distribution (IAT) of real users. Thus, VolTime automatically spots and ranks suspicious users. The results on this Appendix were presented in the SIAM International Conference on Data Mining (SIAM-SDM2017) (CHINO *et al.*, 2017) and were obtained during an internship at the Carnegie Mellon University (CMU), in the United States of America.

A.1 Introduction

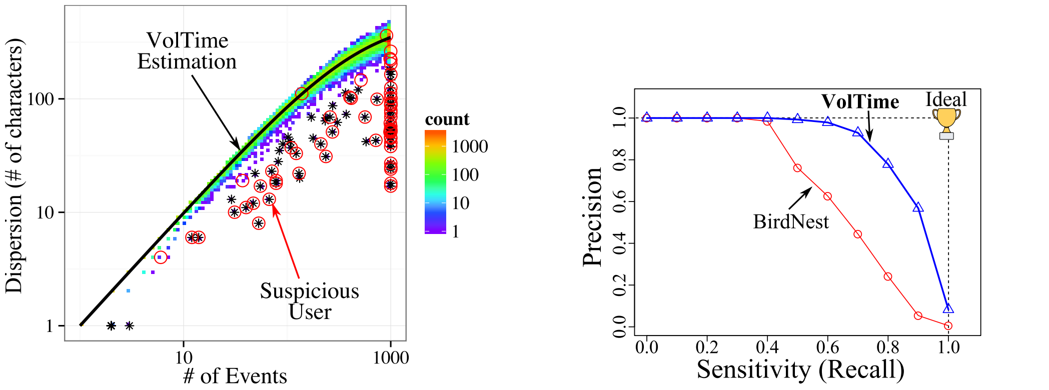
Suppose that user 'Alice' uploaded 20 reviews to an app-store, all exactly 85 characters long - is this suspicious? How about 'Bob', who uploaded 30 reviews, one every 10 minutes (but of variable length)? Most people would agree that both 'Alice' and 'Bob' are suspicious, and deserve further investigation. The reason for this agreement is that, for such a high count of events, a real human's activity would have higher variety ("dispersion") - that is, higher count of distinct values. This is one of the main insights behind this paper, and we show how to use it, to model real user behavior, and to spot impostors.

Social media services and online review platforms influence opinions (LESKOVEC; BACKSTROM; KLEINBERG, 2009; GUERRA *et al.*, 2011; MATSUBARA *et al.*, 2012; DOW; ADAMIC; FRIGGERI, 2013) and even purchasing decisions (HOOI *et al.*, 2016). This has created issues such as spam (FAKHRAEI *et al.*, 2013), spreading of rumors (BESSI *et al.*, 2015) and fake reviews (RAYANA; AKOGLU, 2015). Detecting these issues is important to improve user's experience. Thus, given the activities of a large number of users, can we find the user with the strangest behavior? Specifically, we present two inter-related problems:

- **Modeling:** How can we model human behavior across different platforms (social media, online stores)? We want to model both the *temporal*-aspect (such as the inter-arrival time of the events of a user), jointly with the *volume* of activity (number of characters in the review, or phone call duration, among other aspects).
- **Anomaly Detection:** How can we use these models to detect anomalies such as spammers, bots and fraudsters, like ‘Alice’, and ‘Bob’ in our earlier example.

To answer these questions, we analyze data from different domains, including comments from a social media service (Reddit), reviews from an online store (Flipkart) and phone calls from a large Asian city. From each platform, we analyze the joint distribution of inter-arrival times (IAT) and volume (comment and review length; phone call duration) of communication events.

Figure 49 – **VolTime detects anomaly successfully**: (a) The DISPERSION-PLOT reveals strange behaviors as outliers. (b) VolTime outperforms competitors on accuracy.



(a) DISPERSION-PLOT (b) Bot-Detection using DispersionScore

Source: Chino *et al.* (2017).

Our first contribution is the introduction of the *dispersion* metric, which we use to measure the variability of users’ behavior. Considering the example of ‘Alice’, her dispersion would be 1 since all her comments have the same length. However, users with a larger variability in the comment length would have a large dispersion. We also propose a visualization named DISPERSION-PLOT, which illustrates the relationship between the dispersion and number of events for different users. Figure 49(a) shows the DISPERSION-PLOT of `Reddit` users. While typical users form a single cluster, suspicious users (indicated by red circles), clearly deviate from this pattern.

The second contribution of this paper is VolTime, a model that generates synthetic inter-arrival times (IATs) and event volumes. An important property of VolTime is that it closely matches the typical users' dispersion. In Figure 49(a), the black line, which corresponds to the expected dispersion of our VolTime, accurately follows the behavior of typical users. This allow us to use VolTime to generate a score that measures users' suspiciousness. That is, users whose dispersion deviate most from VolTime's dispersion will have a higher VolTime score. Our main contributions are summarized as follows:

- **Patterns - Population behavior:** We proposed the dispersion (Equations A.5 and A.9) to analyze how the joint distribution of inter-arrival times and volume changes as users produce more events. By analyzing the dispersion across several diverse datasets through the DISPERSION-PLOT, we show that normal users present a similar behavior while bots, fraudsters and spammers clearly deviate from this pattern;
- **VolTime - Generative model:** Based on the patterns observed using DISPERSION-PLOT we propose VolTime, a generative model that is able to describe the inter-arrival times of communication events across all the studied domains;
- **Usefulness - Anomaly detection:** We used VolTime to automatically rank users according to their suspiciousness. VolTime was able to detect bots using only time-stamp and event volume data;

A.2 Background and Related Work

Modeling Human Dynamics: The dynamics of human activity is a widely studied topic (OTTONI *et al.*, 2014; KRISHNAN; COOK, 2014; COSTA *et al.*, 2017; KUMAR *et al.*, 2018), as it has applications that range from resource management (IHLER; HUTCHINS; SMYTH, 2006) and user clustering (ECKMANN; MOSES; SERGI, 2004; MALMGREN *et al.*, 2009) to anomaly detection (RAYANA; AKOGLU, 2015). A well-known model for the timing of human activity is the Poisson-Process (HOEL; PORT; STONE, 1986; CHO; GARCIA-MOLINA, 2003; SIA; CHO; CHO, 2007). Other works argue that IAT distribution of human activities can be better modeled by heavy-tailed distributions such as power-laws (BARABASI, 2005). Recent models for human dynamics include the Self-Feeding Process (SFP) (Vaz de Melo *et al.*, 2015), Cascading Non-homogeneous Poisson Process (MALMGREN *et al.*, 2009) and Rest-Sleep-and-Comment model (RSC) (COSTA *et al.*, 2015). There are also works that focus on the activity volume (number of characters or call duration) as Truncated Lazy Contractor (TLAC) (Vaz de Melo *et al.*, 2010) for call duration. In this paper we propose a model for human activity (VolTime) that describe both the timing of activities and the volume of an event, what to the best of our knowledge was not done so far.

Anomaly Detection: There are many works devoted to detect anomalies based on user activity (CHENG *et al.*, 2009; VAHDATPOUR; SARRAFZADEH, 2010; LAPPAS *et al.*, 2012; GUNNEMANN; GUNNEMANN; FALOUTSOS, 2014; TSYTSARAU; PALPANAS; CASTEL-

LANOS, 2014; PAN *et al.*, 2016). In (ZHANG; PAXSON, 2011) the authors proposed a method that consists in creating a scatter-plot of the minute vs. the second for all comment time-stamps of a user. This plot is then used to spot users from Twitter that are suspicious of being bots. In (HOOI *et al.*, 2016) the authors proposed BIRDNEST, which consist of two steps. A model named BIRD (Bayesian Inference for Rating Data), which describes the statistical properties of the timing and ratings in online commerce using a Bayesian model. Based on BIRD, the authors introduced NEST (Normalized Expected Surprise Total), a suspiciousness metric to detect fraudsters. Table 18 compares our proposed method with existing methods.

Table 18 – Summary of different models for human dynamics.

	Kleinberg (KLEINBERG, 2003)	Poisson (MALMGREN *et al.*, 2009)	SFP (Vaz de Melo *et al.*, 2015)	RSC (COSTA *et al.*, 2015)	TLAC (Vaz de Melo *et al.*, 2010)	BIRDNEST (HOOI *et al.*, 2016)	Zhang/Paxson (ZHANG; PAXSON, 2011)	**VolTime**
Models IAT	√	√	√	√		√		√
Models Volume					√	√		√
Models Both						?		√
Visualization							√	√
Spots anomalies	√	√		√	√	√	√	√

Source: Chino *et al.* (2017).

A.3 Problem Formulation

In this Section, we outline the problem of modeling the user behavior on online social media. Table 19 gives the list of symbols used throughout the paper.

A.3.1 How do individual users behave online?

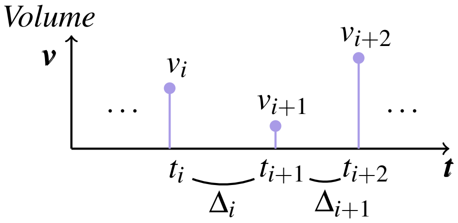
On social services, users interact with each other by posting comments on a Reddit forum or making mobile phone calls on a certain timestamp. We are given a user, as shown on Figure 50,

Table 19 – Concepts and Symbols

Concepts	**Interpretation**
Activity Volume	Characteristic of the online activity.
Dispersion	Number of non-empty bins.
DISPERSION-PLOT	Visualization tool of population behavior.
DispersionScore	Suspicioness of a user.
Symbols	**Definitions**
n	Number of events of a user.
$\mathscr{T} = \{t_1, t_2, \dots\}$	Multiset of timestamps of a user.
$\boldsymbol{\Delta} = \{\Delta_1, \Delta_2, \dots\}$	Multiset of inter-arrival times of a user.
$\mathscr{V} = \{v_1, v_2, \dots\}$	Multiset of activity volumes of a user.
$e_i = (\Delta_i, v_i)$	Event at instant t_i.
$\mathscr{E} = \{e_1, e_2, \dots\}$	Multiset of events of a user.
$D(\mathscr{E})$	Dispersion of events $\mathscr{E}$.
$\hat{D}(n)$	Expected dispersion of n events.
τ	DispersionScore.
p_s	Probability of user entering state s.
LL	Log-logistic distribution.
$\theta_{k,s} = \{\alpha_{k,s}, \beta_{k,s}\}$	Log-logistic parameters of attribute k and state s.
Θ	Set of VolTime parameters.

Source: Chino *et al.* (2017).

Figure 50 – Users can post at any time t_i an activity of volume v. The volume may correspond to the number of characters (e.g. textual comments) or duration (e.g. phone calls).



Source: Chino *et al.* (2017).

with a multiset of activities timestamps $\mathscr{T} = \{t_1, t_2, \dots\}$, where $t_i \leq t_{i+1}$. As the user interacts with a social service, he/she can generate an activity volume v_i at every t_i. The activity volume v is an attribute that describes the amount of the user interaction, for example, v can describe the length (number of characters) of comments/reviews or the duration of a phone call.

For simplicity, we will denote each user interaction with social services as an event e_i represented by the ordered pair (Δ_i, v_i), where $\Delta_i = t_{i+1} - t_i$. A user that interacts n times will generate a multiset of activities events $\mathscr{E} = \{e_1, \dots, e_n\}$. It is important to note that among the infinite possibilities of describing a user, on this paper we will be using the inter-arrival time Δ (IAT) between events. The issue of using timestamps directly is that it is not able to generalize the behavior of users that are more active during different times of day. With these considerations in mind, the first problem can be stated as follows:

Problem 1. (MODELING STATISTICAL PROPERTIES) Given a multiset of events $\mathscr{E} = \{e_1, \ldots, e_n\}$, where each event $e_i = (\Delta_i, v_i)$, $1 \leq i \leq n$, is a pair of IAT (Δ_i) and activity volume (v_i). What is the joint distribution of the multiset?

A.3.2 How can we spot anomalies?

A right community on online social services helps the users to have better experience. With that in mind, is it possible to describe the behavior of the community of users? Do the more active users have the same behavior of the less active? These questions bring the main problems of this paper:

Problem 2. (SUCCINCT FEATURE EXTRACTION) Given a multiset of n events (Δ, v), find few features to describe its behavior.

Problem 3. (SPOT SUSPICIOUS USERS) Given several multisets of events from different users, find a score describing how suspicious a specific user is.

Our ultimate goal is to solve the Problem 3. To achieve this goal, we first handle with the Problem 1 by understanding and describing how the majority of users behave in terms of the joint distribution of IAT and volume (Section A.4). Then in Section A.5, we answer Problem 2 by extracting two features from a user's behavior (multiset of events) (see Equation A.5). In the same Section A.5, we answer Problem 3 using Equation A.6.

A.4 Modeling Statistical Properties

Is it possible to model the patterns of the users' behavior? In this section we discuss the patterns found on users' online activity on real-world datasets. We also point the implications of our findings and how to model their behavior.

A.4.1 Datasets Description

We analyzed four real-world datasets of user's activity events, such as social media posts, e-commerce reviews and mobile phone calls. The datasets are summarized in Table 20 and described in details as follows.

Reddit: The `Reddit` dataset consists of comments posted by users on Reddit. Reddit allows users to submit content, as text posts or URL links. The dataset was originally collected and used in (COSTA *et al.*, 2015). Out of the 94 thousand users, 60 users are known bots inserted by the authors. Since the authors aimed at bot detection, we also checked the dataset for spammers and users that now got their account deleted or banned.

Table 20 – Summary of real-world datasets.

Dataset	# of Users	# of Events
Reddit	94,739	35,979,723
Flipkart	158,638	409,679
SWM	113,145	163,873
LAC	1,696,602	280,814,170

Source: Chino *et al.* (2017).

Flipkart: The Flipkart dataset consists of reviews written by users on the Flipkart e-commerce network, which provides a platform for sellers to market products to costumers. Users can write reviews of products using between 100 and 5000 characters.

Software Marketplace (SWM): The SWM dataset contains reviews in an anonymous online software marketplace. For this dataset the timestamp has a granularity of a day, there is no information about the time the review was posted. The dataset was originally collected by (AKOGLU; CHANDY; FALOUTSOS, 2013).

Large Asian City (LAC): The LAC dataset has information of phone calls made on a large anonymous Asian city. For this dataset, it was collected the timestamp of the beginning of a call and its duration.

For the Reddit, Flipkart and SWM datasets the activity volume represents the length of the text comment (number of characters). The IAT is calculated as the difference between timestamps of consecutive events. For the LAC dataset the activity volume represents the duration of a phone call in seconds. Since phone calls have a different nature, the IAT was calculated as the difference of the end of call timestamp and the beginning of the next call.

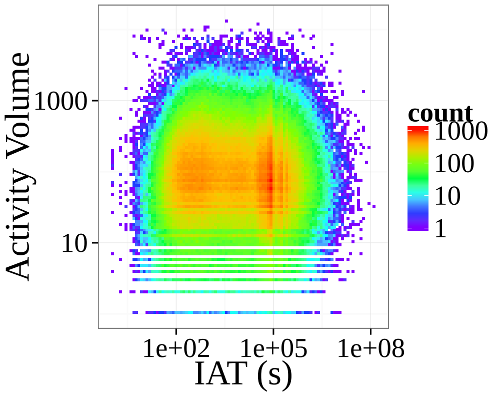
A.4.2 Online Activity Event Patterns

The focus of this paper is to analyze the behavior of the user's online activity events. As stated in the beginning of Section A.3, an activity event is the ordered pair of IAT and activity volume. The activity events of a user can be seen by his/her heatmap, a visualization that shows the relationship between the IAT and the activity volume. The frequency of (Δ, v) is shown using a color coding, more frequent events are reddish and less frequent are bluish. The heatmap can show the behavior of a single user or show how the entire population behaves.

Figures 51 and 54 show the heatmap for the population of each dataset. When analyzing the activity volume, we make the following observation:

Observation 1. The Activity Volume can be accurately modeled by a mixture of log-logistic distribution.

Figure 51 – The heatmap of the Reddit dataset showing the behavior of users. Users have two distinct behaviors, an in-session with activities in short bursts and an out-session with a larger IAT.



Source: Chino *et al.* (2017).

The log-logistic (LL) distribution was previously used to model human activity, as phonecall duration (Vaz de Melo *et al.*, 2010) and users' activity on social media (number of posts, likes and photos) (DEVINENI *et al.*, 2015). The log-logistic PDF is:

$$LL_{PDF}(x;\alpha,\beta) \sim \frac{(\beta/\alpha)(x/\alpha)^{\beta-1}}{(1+(x/\alpha)^{\beta})^{2}} \tag{A.1}$$

where α is a scale parameter and β is a shape parameter.

It is also possible to notice that there are two modes on the activity events for all datasets (see Figure 54). During the first mode, users appear to be more active, generating events with inter-arrival times between 5 to 10 minutes. On the other hand, during the second mode (around 3 hours), they make a post and rest before generating a new event. We summarize these observations as follows:

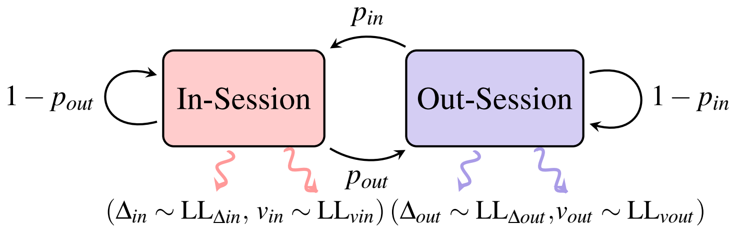
Observation 2. The events' IAT can be described by a mixture of two log-logistics. The first log-logistic corresponds to short intervals, generated by bursts of activity. The second log-logistic is generated when users are less active or resting.

A.4.3 VolTime Model

How can we generate a model capable of following the Observations 1 and 2? In this section we introduce VolTime, a generative model that is capable to describe the interval and volume of human communication in different media. The goal of VolTime is to describe two aspects of human communication: (i) the inter-arrival times (IAT) between events and (ii) the volume of each event. VolTime is a generative model that creates pairs of synthetic IAT and event volumes that matches statistical properties from real data. With VolTime, we can answer the Problem 1.

In order to respect Observation 2, VolTime uses a Markov chain to transition between two states: in-session and out-session. Figure 52 shows the state digram for VolTime. If VolTime is in the in-session state, there is a probability p_{out} to transition to the out-session state and a probability $1 - p_{out}$ to remain in the in-session state. Similarly, if VolTime is in the out-session state, the transition probability to the in-session state is p_{in}.

Figure 52 – State diagram of VolTime. After each state transition VolTime generates an IAT Δ and an event volume v sampled from independent log-logistic distributions.


p_{in}
$1 - p_{out}$
In-Session
Out-Session
$1 - p_{in}$
p_{out}
$(\Delta_{in} \sim \mathrm{LL}_{\Delta in}, v_{in} \sim \mathrm{LL}_{vin})$
$(\Delta_{out} \sim \mathrm{LL}_{\Delta out}, v_{out} \sim \mathrm{LL}_{vout})$

Source: Chino *et al.* (2017).

As noted on Observation 1, VolTime uses a LL distribution to model volume and IAT. After each state transition, VolTime generates an event tuple $e = (\Delta_s, v_s)$ for the current state s (either in-session or out-session). In each state, VolTime waits a time interval Δ_i sampled from a LL distribution with parameters $\theta_{\Delta,s}$ and generates an event volume v_i sampled from a LL with parameters $\theta_{\Delta,s}$.

To estimate the parameters of VolTime we are given an observed input multiset of IAT and activity volumes. We start by finding the probabilities $P(s_i = in)$ and $P(s_i = out)$ that the i-th event is in the in-session and out-session states, respectively. We assume that the distribution of IAT is mixture of two log-logistics with two components corresponding to the in-session and out-session. This allows us to estimate $P(s_i = in)$ and $P(s_i = out)$ using an expectation-maximization (EM) algorithm.

In order to estimate the log-logistic parameters $\theta_{\Delta,in}$ and $\theta_{v,in}$ that will be used to generate the IAT and event volumes for the in/out-session state, we randomly sample the IAT and volumes from the input sequences while weighting according to the probabilities $P(s_i = in)/P(s_i = out)$. Finally, the sampled IAT and event volumes are used to estimate the log-logistic parameters using the maximum-likelyhood estimation (MLE) method. The complexity of the EM algorithm is linear on the size of the multiset of events. Similarly, the complexity of the MLE algorithm is linear on the number of samples used to estimate the parameters of the log-logistic distributions. Now, let $\mathrm{LL}(X;\theta)$ denote a log-logistic distribution with random variable X and parameters θ.

Lemma 1 (VolTime PDF)**.** The joint probability distribution $f(\Delta, v)$ of the events IAT and volume generated by VolTime is given by:

$$\begin{aligned} f(\Delta, v) = {} & w_{in} \cdot \mathrm{LL}(\Delta;\theta_{\Delta,in}) \cdot \mathrm{LL}_{(}v;\theta_{v,in}) \\ & + w_{out} \cdot \mathrm{LL}(\Delta;\theta_{\Delta,out}) \cdot \mathrm{LL}(v;\theta_{v,out}) \end{aligned} \tag{A.2}$$

where:

$$w_{in} = \frac{p_{in}}{p_{in} + p_{out}}, w_{out} = \frac{p_{out}}{p_{in} + p_{out}} \tag{A.3}$$

A.5 Spotting Suspicious Activities

How can we spot suspicious users by analyzing their behavior? In Section A.4 we proposed VolTime to model users' behavior. However, instead of using all 10 parameters of VolTime to spot anomalies, we propose a succinct feature extraction, allowing us to visually spot anomalies.

A.5.1 Population behavior

In Section A.1 we introduced 'Alice' and 'Bob' who have suspicious behaviors. How could we describe them? If we consider the multiset of activity volume $\{85, 85, \ldots, 85\}$ of the 20 reviews that 'Alice' wrote, a natural feature is the size of the multiset ($n = 20$). What other features can we extract? Entropy? Second moment? We now introduce you the definition of dispersion. The dispersion summarizes how the users behave online and can be used to spot anomalies. Suspicious users will have lower values of dispersion than typical users. And this is our proposed answer to the Problem 2, for each user, with a multiset of events, we extract two features: (a) the number n of events and (b) the dispersion, as defined below:

Definition 1. (DISPERSION) Given a multiset of n integer numbers $\mathscr{X} = \{x_1, \ldots, x_n\}$. The dispersion D_{1d} of the multiset $\mathscr{X}$ is the count of distinct values ('vocabulary').

For example, given a multiset of integers $\{1, 2, 1, 5, 5, 2, 5\}$, $n = 7$ and $D_{1d} = 3$. Formally, given x_i an integer in $(1, 2, \ldots, \infty)$, let I_j denote an indicator variable such that $I_j = 1$ if there is at least one i so that $x_i = j$. The dispersion D_{1d} is given by:

$$D_{1d} = \sum_{j=1}^{\infty} I_j \tag{A.4}$$

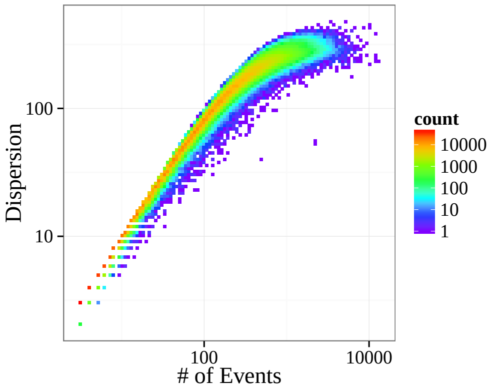
The same idea can be applied to a multiset of 2-d points.

Definition 2. (DISPERSION 2-D) Given a multiset $\mathscr{Y}$ of n two-dimensional points (x, y), where both x and y are integers, the dispersion is calculated as follows:

$$D_{2d} = \sum_{j=1}^{\infty} \sum_{i=1}^{\infty} I_{i,j} \tag{A.5}$$

For example, the multiset $\{(1,1), (1,3), (1,1)\}$ has dispersion $D_{2d} = 2$. In our case, the pairs correspond to events (Δ, v). Since Δ and v have a continuous nature, the vocabulary would be huge and we may lose information. To overcome this, we make them integers using bucketization. We partitionate them in log-bins, because we expect skewed distributions in both of them. This concludes our response to Problem 2: For a given user, with a multiset of (IAT,

Figure 53 – DISPERSION-PLOT shows the relationship between the number of events and dispersion on LAC dataset.



Source: Chino *et al.* (2017).

volume) pairs, we map him/her to a 2-d point: (n, D_{2d}). For the remaining of this text we will denote the 2-d dispersion as D.

We are ready to tackle Problem 3, namely, *how strange is a given user "X", as compared to a large set of users*. The intuition behind our response, is to map all those users (including user "X"), to such 2-d points, as shown in Figure 53. We propose to name such a plot as a DISPERSION-PLOT, and, since there is heavy over-plotting, we make it a heatmap. We expect to see a clear trend, and specifically, a (non-linear) correlation between dispersion and event-count n; this correlation would of course depend on the joint distribution of (IAT, volume). The upcoming Lemmas 2 and 3 quantify this correlation, between n and expected dispersion, which gives the black line on Figure 49(a).

Our final answer to the question '*how strange is user "X"?*' is intuitively the distance of the 2-d image of user "X", from the expectation ("black line" in Figure 49(a)).

Formally, we have the following: Let $D(\mathscr{E})$ denote the dispersion (Equation A.5) of the event multiset $\mathscr{E} = \{e_1, \cdots, e_n\}$. Let $\hat{D}(n)$ denote the expected dispersion from n samples randomly sampled from a joint probability distribution of VolTime. The DispersionScore is computed as follows:

$$\tau = |\log \hat{D}(n) - \log D(\mathscr{E})| \tag{A.6}$$

A.5.2 Expected Dispersion

The only missing part is how to estimate the expected dispersion $\hat{D}$, as a function of the sample size n, and given the joint distribution of (IAT, volume). The answer is Equation A.9, but we need some lemmas first. We start by showing the Expected Dispersion lemma for one dimension:

Lemma 2 (Expected Dispersion 1d)**.** Given a multiset of n integers $\mathscr{X} = \{x_1, \dots, x_n\}$ and P_i the probability of an $x \in \mathscr{X}$ to be equal i. The expected dispersion is:

$$\hat{D}_{1d}(n) = \sum_{i=1}^{\infty} [1 - (1 - P_i)^n] \tag{A.7}$$

Proof. Let $\mathscr{X}$ and P_i be as described in Lemma 2. Let I_i denote an indicator variable such that $I_i = 1$, if there is at least one w where $x_w = i$. The expected value of I_i is:

$$E(I_i) = 1 - (1 - P_i)^n \tag{A.8}$$

Equation A.7 can be obtained combining Equations A.4 and A.8.

□

Lemma 2 can be extended for a 2-d multiset.

Lemma 3 (Expected Dispersion)**.** Given a multiset of n 2-d points $\mathscr{Y} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and $P_{i,j}$ the probability of a $(x, y) \in \mathscr{Y}$ to be equal (i, j). The expected dispersion is:

$$\hat{D}(n) = \sum_{j=1}^{\infty} \sum_{i=1}^{\infty} \left[1 - (1 - P_{i,j})^n\right] \tag{A.9}$$

Proof. Let $\mathscr{Y}$ and $P_{i,j}$ be as described in Lemma 3. Let $I_{i,j}$ denote an indicator variable such that $I_{i,j} = 1$, if there is at least one w where $(x_w, y_w) = (i, j)$. The expected value of $I_{i,j}$ is:

$$E(I_{i,j}) = 1 - \left(1 - P_{i,j}\right)^n \tag{A.10}$$

Equation A.9 can be obtained combining Equations A.5 and A.10. □

Notice that if we have a continuous 2-d distribution, we can always digitize it to an integer-valued 2-d distribution. Formally, for our setting, the joint probability $P_{i,j}$ of an event falling in the (i, j) bin is computed as follows:

$$P_{i,j} = \int_{\Delta'_j}^{\Delta'_{j+1}} \int_{v'_i}^{v'_{i+1}} f(\Delta, v) d\Delta dv \tag{A.11}$$

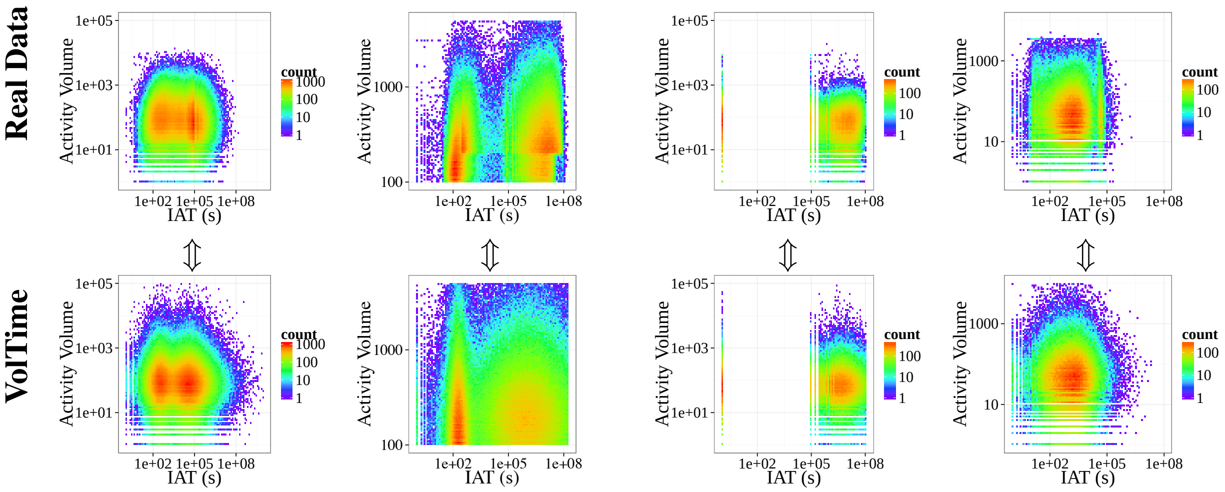
where $f(\Delta, v)$ is the VolTime PDF described by Equation A.2.

The complexity to calculate the DispersionScore is the complexity to calculate the expected dispersion $\hat{D}$ and the user's dispersion D. Considering that we already have the VolTime PDF, the complexity of $\hat{D}$ is $\mathscr{O}(m)$, where m is the total number of discrete bins. The Expected Dispersion can be calculated only once for each number of events n. The complexity to calculate D is $\mathscr{O}(n)$, where n is the user's number of events. Since we only need to count the total number of events and the number of distinct events. The complexity to compute the dispersion for each user is linear to the size of the dataset.

A.5.3 Activity Event Generation with VolTime

In this section we show how well VolTime can fit real data. To the best of our knowledge, there is no work aimed at modeling the joint distribution of IAT and volume. The parameters were estimated using the algorithm described in Section A.5 on all datasets. Figure 54 shows the heatmap for the synthetic data generated by VolTime.

Figure 54 – Heatmap of synthetic data generated by VolTime model for each dataset. On all datasets, VolTime was able to correctly model the In/Out-sessions behavior, showing the capability of correctly modeling the activity events of users.



(a) `Reddit` (b) `Flipkart` (c) `SWM` (d) `LAC`

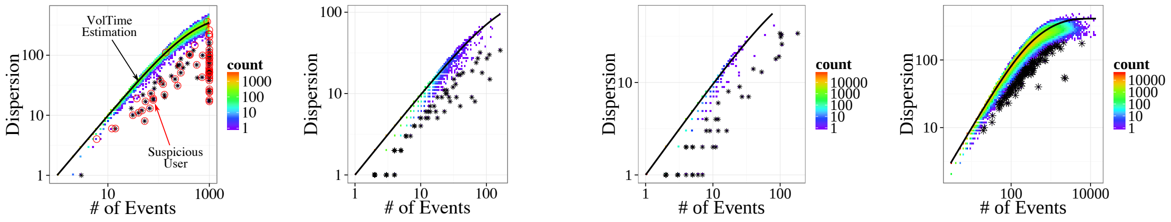
Source: Chino *et al.* (2017).

For all datasets, VolTime managed to model the in/out-session behavior. We made a modification on the activity volume generation of VolTime due to the `Flipkart` dataset limitation on the number of characters. We also modified VolTime to only generate IAT with intervals of one day for the `SWM` dataset, due to its granularity. The correctness of VolTime shows its robustness to different granularities. The `LAC` dataset has a different behavior than the other datasets. The VolTime model was able to generate the in-session correctly, but did not manage to model the less intense out-session spike. Although VolTime presents this issue, Section A.6 shows that VolTime can predict the behavior of the population.

A.6 Spotting Suspicious Activities with the DispersionScore

In this section we show how well the DispersionScore can spot suspicious users. We used Equation A.9 to estimate the expected dispersion and calculate the DispersionScore (τ) for a given number n of events. Figure 55 shows the DISPERSION-PLOT for each dataset. The solid black line is the expected dispersion $\hat{D}(n)$, where n is the number of events. For all datasets, the

Figure 55 – **DISPERSION-PLOT spots outliers:** DISPERSION-PLOT showing the usefulness of VolTime. The solid black line is the expected dispersion. The black stars are the spotted suspicious users ($\tau \geq 1$). (a) The red circles are the confirmed suspicious users.



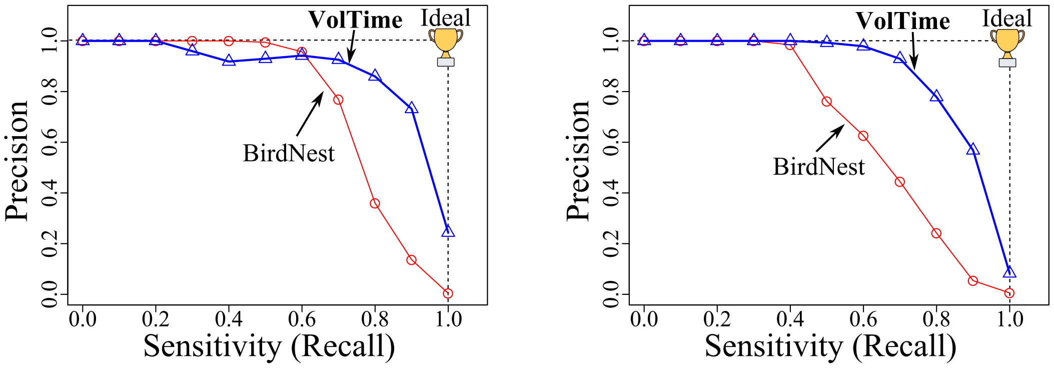
(a) `Reddit` (b) `Flipkart` (c) `SWM` (d) `LAC`

Source: Chino *et al.* (2017).

$\hat{D}(n)$ falls on the behavior of typical users, represented by the green and red areas, showing that VolTime was able to correctly predict the dispersion given the number of activity events. The black stars (*) represent the suspicious users spotted by VolTime and known suspicious users are marked as red circles. Since only the `Reddit` dataset has a ground truth, experiments on the other datasets discuss the top suspicious users found by our method. The results will be detailed as follows.

`Reddit`: The result obtained by VolTime on `Reddit` users are shown in Figure 55(a). More than 80% of the known suspicious users are marked with a black star, showing the correctness of VolTime. We compared VolTime with BIRDNEST (HOOI *et al.*, 2016), but considering the activity volume as its ratings. The activity volume was log-binned to better adapt to BIRDNEST. Figure 56(a) shows the precision vs sensitivity (recall) obtained by VolTime and BIRDNEST. VolTime spotted **80%** of the suspicious users with a precision greater than **85%**, being up to **2.39 times more accurate** than BIRDNEST.

Figure 56 – Precision of VolTime spotting suspicious users on the `Reddit` dataset. VolTime in blue is closer to ideal.



(a) Ground truth (b) Relabeled ground truth

Source: Chino *et al.* (2017).

Note that on Figure 55(a), there are some black star users that were not labeled as suspicious on the ground truth. We manually checked these users and spotted suspicious activities:

users that only post URL or spammers or had their accounts deleted/banned. The same procedure was done with the BIRDNEST output. Figure 56(b) shows the result considering the new suspicious users. This time, VolTime spotted **70%** of the suspicious users with a **precision greater than 90%**, while BIRDNEST had a precision of **44%**.

`Flipkart` and `SWM`: On both datasets, VolTime spotted spammer users. On `Flipkart`, the majority of the spam reviews do not add too much information for future buyers, since it has generic adjectives. Usually the top suspicious users post all their reviews in short bursts and in a short time span. One of the top suspicious user wrote the same 60 reviews on different products in less than 1 hour. VolTime was also able to spot users that use a variety of template review texts on different products. One user wrote the same review for different movies of the same actor, just changing the title of the movie. Also, we noted that different users sometimes used the same review text to review different products. All of the top 20 reviewers spotted by VolTime had this same behavior.

On `SWM`, the majority of the top suspicious users just promote some kind of code associated with an app. They usually promote these codes to their own benefit by saying that new users that use their codes will get free points or cash. Usually the top suspicious users posts all their reviews on the same day or in less than a week. Every user from the top 20 users spotted by VolTime have similar review texts that promote their codes, always offering promises of free points and cash. We listed below some reviews:

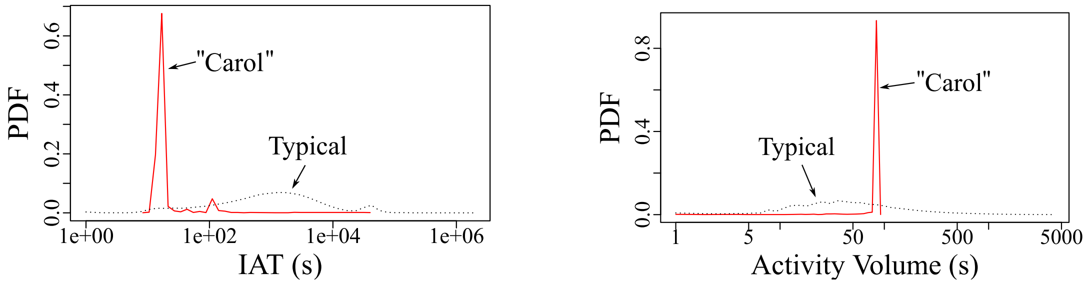
- `Flipkart`: *"The item quality is very good and its look is very well really appreciate. Highly Recommended item buy again. Fast shipping."*
- `SWM`: *"Download [redacted] for some free cash!!! Sign up using [redacted] for some points."*

`LAC`: VolTime spotted users with suspicious behavior, like "`Carol`". The behavior of "`Carol`" is shown on Figure 57, the solid red line is "`Carol`"'s behavior and the dotted black line the typical user behavior. "`Carol`" has over two thousand calls in short bursts to the same person. Notice that the typical user has a smoother distribution of IAT and activity volume. The majority of the top suspicious users found by the VolTime also have this same behavior.

A.7 Final Thoughts

In this text we analyzed the online activity events of 2M users from online social services as Reddit, e-commerce reviews and mobile phonecalls. We proposed VolTime, which is able to mimic the human online activity event behavior. We also showed how VolTime can be used to spot users with suspicious behavior, like bots and spammers. The contributions of this paper are as follows:

Figure 57 – **Closer inspection of a suspicious user:** The top suspicious user (“`Carol`”) found by VolTime on the `LAC` dataset. The red line shows the behavior of the suspicious users and the black dotted line is the typical user behavior.



(a) “`Carol`”’s IAT distribution.

(b) “`Carol`”’s activity volume distribution.

Source: Chino *et al.* (2017).

- **Patterns:** We proposed dispersion (Definitions 1 and 2) to quantify the variability of inter-arrival times and volume of events generated by users of different platforms, such as social media services and phone networks.
- **Model:** We introduced VolTime, a model for the joint distribution of IAT and volume of events generated by users (Figure 52). We show that our model can accurately fit real data (Figure 54), and, more importantly, match the dispersion metric of human users (Figure 55).
- **Anomaly Detection:** We used VolTime to calculate DispersionScore that measures users’ suspiciousness (Equation A.6). Users whose dispersion deviate most from VolTime’s dispersion will have a higher score. Taking advantage of DispersionScore, we managed to spot **70%** of the suspicious users with a precision higher than **90%** on the `Reddit` dataset (Figure 56).

ICMC USP

SÃO CARLOS