Fire detection from social media images by means of instance-based learning

Marcos Vinicius Naves Bedo, William Dener de Oliveira, Mirela Teixeira Cazzolato, Alceu Ferraz Costa, Gustavo Blanco, Jose F. Rodrigues Jr., Agma Juci Machado Traina, and Caetano Traina Jr.

Institute of Mathematics and Computer Science,
University of São Paulo, P.O. Box 668, Brazil,
{bedo,willian,alceufc,junio,agma,caetano}@icmc.usp.br,
{blanco,mirelac}@usp.br,
Home page: gbdi.icmc.usp.br

Abstract. Social media can provide valuable information to support decision making in crisis management, such as in accidents, explosions, and fires. However, much of the data from social media are images, which are uploaded at a rate that makes it impossible for human beings to analyze them. To cope with that problem, we design and implement a database-driven architecture for fast and accurate fire detection named FFireDt. The design of FFireDt uses the instance-based learning through indexed similarity queries expressed as an extension of the relational Structured Query Language. Our contributions are: (i) the design of the Fast-Fire Detection ($FFireDt$), which achieves efficiency and efficacy rates that rival to the state-of-the-art techniques; (ii) the sound evaluation of 36 image descriptors, for the task of image classification in social media; (iii) the evaluation of content-based indexing with respect to the construction of instance-based classification systems; and (iv) the curation of a ground-truth annotated dataset of fire images from social media. Using real data from Flickr, the experiments showed that system $FFireDt$ was able to achieve a precision for fire detection comparable to that of human annotators. Our results are promising for the engineering of systems to monitor images uploaded to social media services.

Key words: Fire Detection, Image Descriptors, Social Media, Extended-SQL

1 INTRODUCTION

Fire incidents represent a serious threat to industries, crowded events and densely populated areas. This kind of incident may cause impacts in property, environment, and danger to human life. Therefore, a fast response of the authorities is essential to prevent and reduce injuries and financial losses, when crises situations strike. The management of such situations is a challenge that requires fast and effective decisions using all available data, because decisions based on incomplete information may lead to more damages [Russo, 2013]. Software systems

can be used to support expert and rescue forces in the decision making during crises, by improving information correctness and availability [Kudyba, 2014].

Systems aimed at supporting salvage and rescue teams often rely on images to understand the crisis scenario and to determine the actions that will reduce losses [Villela et al., 2014]. Crowdsourcing and social media, as massive sources of images, possess a great potential to assist such systems. Users from social media such as Flickr, Twitter, and Facebook upload pictures from mobile devices in many situations, generating a flow of images with valuable information. Such information may reduce the time spent to make decisions, especially when used along with other information sources. In order to benefit from massive volume of social media images, automatic image analysis is an important technique to grasp the dimensions, type, objects and people involved in an incident.

Despite the potential benefits, we observed that there is still a lack of studies concerning automatic content-based processing of crisis images [Villela et al., 2014]. In the specific case of fire – which is observed during explosions, car accidents, forest and building fire, to name a few, there is an absence of studies to identify the most adequate content-based retrieval techniques (image descriptors) able to identify and retrieve relevant images captured during crises. Therefore, our objective in this work is to fill this gap with the aid of a prototype system for fire monitoring. Our system sets up an extension of the relational model [Sumathi and Esakkirajan, 2007] to store images collected from social media at the same time that it employs similarity queries to classify them.

This work reports on one of the steps of the project *Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management – Rescuer*[1]. The project goal is to use crowdsourcing data (image, video, and text captured with mobile devices) to assist in rescue missions. Here, we evaluate a set of techniques for fire detection by image content, one of the project targets. We use real images from Flickr[2], a well-known social media website from where we collected a large set of images that were manually annotated as having fire, or not. We used this dataset as a ground-truth to evaluate image descriptors in the task of detecting fire. Our contributions are the following:

1. **Design of FFireDt:** we introduce the Fast-Fire Detection architecture, a scalable system for automatic fire detection that rivals the state-of-the-art methods and whose accuracy compares to that of human annotation;
2. **Evaluation:** we soundly compare the precision and performance of 36 image descriptors for image classification. Additionally, we show that indexing similarity querying is necessary to make instance-based learning scalable;
3. **Flickr-Fire Dataset:** a vast human-annotated dataset of real images, suitable as ground-truth for the development of content-based techniques for fire detection – distributed online at `http://icmc.usp.br/pessoas/junio/DatasetFlicker/DatasetFlickr.htm`.

---

[1] http://www.rescuer-project.org/

[2] https://www.flickr.com/

The groundwork of the $FFireDt$ architecture is described in [Bedo et al., 2015]. This paper's contribution extends the latter publication with a number of new results. Specifically, we introduce: *(i)* an extended description of the $FFireDt$ logical design; *(ii)* the $FFireDt$ schema fully integrated into the relational model; *(iii)* an extended analysis of the $FFireDt$ image classification, which includes the use of indexing techniques; and *(iv)* a comprehensive presentation of the experimental results.

The rest of this paper is organized as follows. Section 2 introduces the related work; Section 3 presents the main concepts regarding fire-detection in images; Section 4 presents the proposed architecture and the $FFireDt$ relational schema. Section 5 describes the experiments and discusses their results; finally, Section 6 presents the conclusions.

2 RELATED WORK

In this paper we are interested in the following problem:

Problem 1. Given a collection of photos, possibly obtained from a social media service, how can we efficiently and effectively detect fire?

There are interesting approaches related to this problem concerning motion analysis on video, but they are not applicable for still images [Chunyu et al., 2010], and most of these approaches do not present satisfactory performance [Celik et al., 2007, Ko et al., 2009, Liu and Ahuja, 2004]. Other works have a different focus, as in the case of [Tamura et al., 2012], whose work aims at mining information from sets of images originated from social events at the same time that its corresponding related topics are detected (e.g. touristic attractions).

In the literature, there are studies proposing particular color models focused on fire detection, based on Gaussian differences [Celik et al., 2007] or in spectral characteristics to identify fire, smoke, heat or radiation. The spectral color model has been used along with spatial correlation and stochastic modeling in order to capture fire motion [Liu and Ahuja, 2004]. However, such technique requires a sequence of images and is not suitable for individual images, which are common in social media.

Other studies employ a variation of the combination given by a color model transform and a classifier. This combination is employed in the work of Dimitropoulos *et al.* [Dimitropoulos et al., 2014], which represents each frame of a video according to the most prominent texture and shape features. It also combines such representation with spatio-temporal motion features to employ SVM classification to detect fire in videos. However, this approach is neither scalable nor suitable for fire detection on still images.

On the other hand, the feature extraction methods available in the MPEG-7 Standard have been used for image representation in fast-response systems that deal with large amounts of data [Doeller and Kosch, 2008, Ojala et al., 2002, Tjondronegoro and Chen, 2002]. Nevertheless, to the best of our knowledge,

there is no study employing those extractors for fire detection. Despite the multiple approaches seen in this section, there is no conclusive work about which image descriptors are suitable to identify fire in images. Thus, it is a demand to have means to accurately evaluate the most appropriate method, as we pursue in the present research.

3 BACKGROUND

3.1 Content-based model for retrieval and classification

The use of queries based on similarity is a well-known approach to retrieve images by content, relying on the representation of such data through feature extraction techniques [Silva et al., 2010, Guyon et al., 2006]. After the extraction of a vector of representative numerical features, the images can be retrieved by comparing their vectors using a distance function, which is usually a metric or a divergence function; the Euclidean distance is the most common one. Such a comparison is a necessary step in image retrieval systems. However, it is possible to employ this same method, the similarity-based retrieval, in the task of image classification. This is called Instance-Based Learning (IBL). IBL classifiers label a given image according to the labels of its most similar images as informed by a similarity query [Aha et al., 1991]. Formally, these concepts can be expressed as follows:

Definition 1. Feature Extraction Method (FEM): *A feature extraction method is a non-bijective function that, given an image domain $\mathbb{I}$, is able to represent any image $i_q \in \mathbb{I}$ in a domain $\mathbb{F}$ as f_q. Each value f_q is called a feature vector (FV) and represents characteristics of the image i_q.*

We use FEMs to represent images in multidimensional domains. Therefore, the image feature vectors can be compared according to the next definition:

Definition 2. Evaluation Function (EF): *Given the feature vectors f_i, f_j and $f_k \in \mathbb{F}$, an evaluation function $\delta \; : \; \mathbb{F} \times \mathbb{F} \to \mathbb{R}$ is able to compare any two elements from $\mathbb{F}$. The EF is said to be a metric distance function if it complies with the following properties:*

– *Symmetry: $\delta(f_i, f_j) = \delta(f_j, f_i)$.*
– *Non-negativity: $0 < \delta(f_i, f_j) < \infty$.*
– *Triangular inequality: $\delta(f_i, f_j) \leq \delta(f_i, f_k) + \delta(f_k, f_j)$.*

The FEM defines the element distribution in the multidimensional space. On the other hand, the evaluation function defines the behavior of the searching functionalities. Therefore, the combination of FEM and EF is the main parameter to improve or decrease the accuracy and quality for both classification and retrieval. Formally, this association can be defined as:

Definition 3. Image Descriptor (ID): *An image descriptor is a pair $\langle \epsilon, \; \delta \rangle$, where ϵ is a composition of FEM and δ is a weighted EF.*

By employing a suitable *image descriptor*, it is possible to inspect the neighborhood of a given element considering previously labeled cases. This course of action is the principle of the Instance-Based Learning algorithms, which rely on previously labeled data to classify new elements according to their nearest neighbors. The sense of what "nearest" means is provided by the EF. Formally, the nearest elements correspond to the following definition:

Definition 4. k-Nearest-Neighbors - kNN: *Given an image i_q summarized as $f_q \in \mathbb{F}$, an image descriptor $ID = \langle \epsilon, \delta \rangle$, a number of neighbors $k \in \mathbb{N}$ and a set F of represented images, the k-Nearest Neighbors set is, generically, the subset of $F \subset \mathbb{F}$ such that $kNN = \{f_n \in \mathbb{F} \mid \forall\ f_i \in F;\ \delta(f_n,\ f_q) < \delta(f_i,\ f_q)\}$.*

The fast execution of kNN queries is crucial to a system that handles images arriving from a social media service. Similarity kNN queries using extended-SQL [Barioni et al., 2009] satisfies this need as they lead to huge performance gains [Silva et al., 2013, Kaster et al., 2010] if planned as an extension of a Relational Database Management System. Once part of a database system, the similarity retrieval mechanism is also able to store the automatic classification and the results of the kNN queries. Similarity queries rely on the properties of the *image descriptor* to define the image representations and the search space. Therefore, the choice of suitable descriptors becomes the critical point to be defined in a fire detection system. The next section reviews the possibilities of image descriptors for this task as well as how similarity queries can be used to support IBL over the relational model.

3.2 MPEG-7 Feature Extraction Methods

The MPEG-7 standard was proposed by the ISO/IEC JTC1 [IEEE MultiMedia, 2002]. It defines expected representations for images regarding color, texture and shape. The set of proposed feature extraction methods was designed to process the original image as fast as possible, without taking into account specific image domains. The original proposal of MPEG-7 is composed of two parts: high and low-level values, both intended to represent the image. The low-level value is the representation of the original data by a FEM. On the other hand, the high-level feature requires examination by an expert.

The goal of MPEG-7 is to standardize the representation of *streamed* or stored images. The low-level FEMs are widely employed to compare and to filter data, based purely on content. These FEMs are meaningful in the context of various applications according to several studies [Doeller and Kosch, 2008, Tjondronegoro and Chen, 2002]. They are also supposed to define objects by including color patches, shapes or textures. The MPEG-7 standard defines the following set of low-level extractors [Sato et al., 2010]:

– **Color:** Color Layout, Color Structure, Scalable Color and Color Temperature, Dominant Color, Color Correlogram, Group-of-Frames;

– **Texture:** Edge Histogram, Texture Browsing, Homogeneous Texture;
– **Shape:** Contour Shape, Shape Spectrum, Region Shape;

We highlight that shape FEMs' usually depends on previous object definitions. As the goal of this study relies on defining the best setting for automatic classification and retrieval for fire detection, user interaction in the extraction process is not suitable for our proposal. Thus, we focus only on the color and texture extractors. In this study we employ the following MPEG-7 extractors: Color Layout, Color Structure, Scalable Color, Color Temperature, Edge Histogram, and Texture Browsing. They are explained in the next Sections.

Color Layout The MPEG-7 Color Layout (CL) [Kasutani and Yamada, 2001] describes the image color distribution considering spatial location. It splits the image in square sub-regions (the number of sub regions is a parameter) and label each square with the average color of the region. Next, the average colors are transformed to the YCbCr space and a Discrete Cosine Transformation is applied over each band of the YCbCr region values. The low-frequency coefficients are extracted through a zig-zag image reading. In order to reduce dimensionality, only the most prominent frequencies are employed in the feature vector.

Scalable Color The MPEG-7 Scalable Color (SC) [Manjunath et al., 2001] aims at capturing the prominent color distribution. It is based on four stages. The first stage converts all pixels from the RGB color-space to the HSV space and a normalized color histogram is constructed. The color histogram is quantized using 256 levels of the HSV space. Finally, a Haar wavelet transformation is applied over the resulting histogram [Ojala et al., 2002].

Color Structure The MPEG-7 Color Structure (CS) expresses both spatial and color distribution [Sikora, 2001]. This paper splits the original image in a set of color structures with fixed-size windows. Each fixed-size window selects equally spaced pixels to represent the local color structure. The window size and the number of local structures are parameters of CS [Manjunath et al., 2001]. For each color structure, a quantization based on the HMMD - a color-space derived from HSV that represents color differences - is executed. Then a local "histogram" based on HMMD is built. It stores the presence or absence of the quantized color instead of its distribution along with the window. The resulting feature vector is the accumulated distribution of the local histograms according to the previous quantization.

Edge Histogram The MPEG-7 Edge Histogram (EH) aims at capturing local and global edges. It defines five types of edges regarding $N \times N$ blocks, where N is an extractor parameter. Each block is constructed by partitioning the original image into square regions. After applying the square masks to an image, it is possible to compute the local edge histograms. At this stage, the entire histogram is composed of $5 \times N$ bins, but it is biased by local edges. To circumvent this problem, a variation [Park et al., 2000] was proposed to capture

also semi-local edges. The resulting feature vector is composed of N plus thirteen edge-histograms, which represents the local and the semi-local distribution, respectively.

Color Temperature The main hypothesis supporting the MPEG-7 Color Temperature (CT) is that there is a correlation between the "feeling of image temperature" and illumination properties. Formally, the proposal considers a theoretical object called *black body*, whereupon its color depends on the temperature [Wnukowicz and Skarbek, 2003]. According to Planck's formula, the locus of the theoretical black body changes from 2,000 Kelvin (red) to 25,000 Kelvin (blue). The feature vector represents the linearized pixels in the XYZ space. This is performed by interactively discarding every pixel with luminance Y above the given threshold – a FEM's parameter. Thereafter, the average color coordinates in XYZ are converted to UCS. Finally, the two closest isotemperature lines is calculated from the given color diagrams [Wnukowicz and Skarbek, 2003]. The formula for the resulting color temperature depends on the average point, the closest isolines and the distances among them.

Texture Browsing The MPEG-7 Texture Browsing extractor (TB) is obtained from Gabor filters applied to the image [Lee and Chen, 2005]. This FEM parameters' are the same used in Gabor filtering. The Texture Browsing feature vector is composed of 12 positions: 2 to represent regularity, 6 for directionality and 4 for coarseness. The regularity features represent the degree of regularity of the texture structure as a more/less regular pattern, in such a way that the more regular a texture, the more robust the representation of the other features is. The directionality defines the most dominant texture orientation. This feature is obtained providing an orientation variation for the Gabor filters. Finally, the coarseness represents the two dominant scales of the texture.

3.3 Evaluation Functions

An Evaluation Function expresses the proximity between two feature vectors. We are interested in feature extractor that generates the same amount of features for each image, thus we account only for evaluation functions for multidimensional spaces. Particularly, we employed distance functions (metrics) and divergences as evaluation functions. Suppose two feature vectors $X = \{x_1, \ x_2, \ \ldots, \ x_n\}$ and $Y = \{y_1, \ y_2, \ \ldots, \ y_n\}$ of dimensionality n. Table 1 shows the EFs implemented, according to their evaluation formulas. The most widely employed metric distance functions are those related to the Minkowski family: the Manhattan, Euclidean and Chebyshev [Zezula et al., 2006]. A variation of the Manhattan distance is the Canberra distance that results in distances in the range $[0, 1]$. These four EFs satisfy the properties of Definition 2. Therefore, they are metric distance functions. However, there are non-metric distance functions that are useful for image classification and retrieval. The Kullback-Leibler Divergence, for instance, does not follow the triangular inequality neither the symmetry properties. A symmetric variation of the Kullback-Leibler is the Jeffrey Divergence, yet it still is not a metric due to the lack of the triangular inequality compliance.

Table 1: Evaluation functions: their classification as metric distance functions and respective formulas.

Name	Metric	Formula
City-Block	Yes	$\sum_{i=1}^{n} |x_i - y_i|$
Euclidean	Yes	$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2)}$
Chebyshev	Yes	$\lim_{p \to \infty} (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$
Canberra	Yes	$\sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$
Kullback Leibler Divergence	No	$\sum_{i=1}^{n} x_i \ln(\frac{x_i}{y_i})$
Jeffrey Divergence	No	$\sum_{i=1}^{n} (x_i - y_i) \ln(\frac{x_i}{y_i})$

3.4 Similarity Queries Support on RDBMS

Relational Database Management Systems (RDBMS) do not offer support to content-retrieval and similarity queries. Accordingly, several works have proposed an extension of the relational model to provide this support [Silva et al., 2013, Kaster et al., 2010]. In this research, we use the *framework* SimbA [Bedo et al., 2014], of our authorship, for the content-based retrieval functionalities. In accordance to the aforementioned benefits of this approach, this design choice improves the fire detection in our architecture, which focus on social media images. This extended-RDBMS tool enables the improvement of the logical design and capabilities of the original proposal of the *FFireDt* architecture. It defines a relational schema to store the images arriving from a social network as well as their classifications and nearest neighbors. In this arrangement, the SimbA *framework* permits to create image descriptors and to store the images and their descriptive data into tables. It also enables to query images by content using extended SQL according to a given *image descriptor* and to employ the `GROUP BY` clause to perform the classification task. The advantages of using SimbA are: *(i)* the definition of a new *image descriptor* through extended-SQL as well new feature extractor methods and evaluation functions; *(ii)* query images by content through Selection and Join operations; and *(iii)* automatic indexing according to the *image descriptor*. Unlike its competitors FMI-SiR and Sim-DB, SimbA does not depend on a specific RDBMS and is able to handle regular and extended-SQL queries alike. Moreover, SimbA is flexible enough to provide support for several image file formats.

3.5 Instance-Based Learning - IBL

The main hypothesis for IBL classification is that the unlabeled feature vectors (FV) pertain to the same class of its k Nearest-Neighbors, according to a predefined rule. Such classifier relies on three components:

1. An evaluation function, which evaluates the proximity between two FVs;
2. A classification function, which receives the nearest FVs to classify the unlabeled one – commonly considering the majority of retrieved FVs;
3. A concept description updater, which maintains the record of previous classifications.

Variations of this strategy define different IBL versions. For instance, the IB1 adopts the majority of the retrieved elements as the classification rule and keeps no record of previous classifications. kNN similarity queries are sufficient to achieve IB1 classification. Moreover, it can be seamlessly expressed as an extended-SQL statement to be submitted to *framework* SimbA. This database-driven approach reduces the total time to execute the IB1 processing by using indexing techniques.

3.6 Dataset Flickr-Fire

For experimentation, we built the Flickr-Fire dataset, which consists of 5,962 images (no duplicates), obtained under license Creative Commons, using the Flickr API[3]. To retrieve the images we used textual queries such as: "fire car accident", "criminal fire", and "house burning". Figures 1a and 1b illustrate samples of the obtained images. After the retrieval, we excluded images with low quality.

<center>(a)   (b)</center>

Fig. 1: Sample images labeled as (a) `fire` and (b) `not_fire` from dataset `Flickr-Fire`.

Even with queries related to fire, some of the images did not present visual traces of fire, so each image was manually annotated in order to construct a coherent ground-truth dataset. To perform the annotation, we selected 7 subjects, all of them aging between 20 and 30 years, and non-color-blinded. The annotation was planned in a manner that each image was classified for at least two subjects as containing, or not, traces of fire. For the images with divergence in the annotation process, we asked a third subject to act as a tiebreaker. The average disagreement between the subjects was of 7.2%. Aiming at balancing

---

[3] The Flickr API is available at: `www.flickr.com/services/api/`

the class distribution of the dataset, we randomly removed images so to end up with 1,000 images containing fire and 1,000 images without fire. This dataset was named `Flickr-Fire` and employed in the experiments of Section 5.

4 FFireDt PHYSICAL DESIGN

Here we introduce the Fast-Fire Detection ($FFireDt$) architecture, which depends on the *framework* SimbA, proposed in a previous work of ours, to perform image retrieval and classification. Figure 2 illustrates the relationship among the modules, their communication, and how they relate to SimbA. The feature extraction methods module (the FEM module) accepts any kind of feature extractor. For this work, we implemented six extractors following the MPEG-7 standard: Color Layout, Scalable Color, Color Structure, Edge Histogram, Color Temperature, and Texture Browsing – explained in Section 3.2. The evaluation functions module (the EF module) is also designed for general implementations; for this work, we implemented six functions: City-Block, Euclidean, Chebyshev, Jeffrey Divergence, Kullback-Leibler Divergence, and Canberra. The feature extractors methods and evaluation functions acronyms are listed in Tables 2(a) and 2(b).

Table 2: Acronyms used in the experiments for (a) Feature Extracted Methods, and (b) Evaluation Functions.

<div>

(a)

Feature Extractor Method	Acronym
Color Layout	CL
Scalable Color	SC
Color Structure	CS
Color Temperature	CT
Edge Histogram	EH
Texture Browsing	TB

(b)

Evaluation Function Name	Acronym
City-Block	CB
Euclidean	EU
Chebyshev	CH
Canberra	CA
Kullback Leibler	KU
Jeffrey Divergence	JF

</div>

The $FFireDt$ system heavily relies on image descriptors, which must be first defined and integrated into the system. By using the extended-SQL syntax provided by SimbA, the $FFireDt$ image descriptors can be expressed as high-level commands as follows:

```
CREATE METRIC ID_<fem>_<ef> USING <ef>
FOR STILLIMAGE (<fem> (<fem_setting>));
```

The above command creates an *image descriptor*, a ``METRIC'' named ``ID_<fem>_<ef>'', by associating one feature extractor method (`<fem>`) and

one evaluation function (`<ef>`). Additionally, it is possible to define the parameters of the FEM through the `<fem_setting>` command option. The tuple corresponding to the metric just created is stored inside the user-transparent SimbA data-dictionary (which is not a table of the $FFireDt$ system) and is used as basis for the creation of further tables: the "KnowledgeBase" (`Know_Base`) table that contains the ground-truth set of images of `Flickr-Fire`, and the `Data` table to store the incoming flow of images arriving from the social network service. The tables can be defined through the extended-SQL commands as follows:

```
CREATE TABLE <Table_Name>(
    id INTEGER,
    img STILLIMAGE,
    label INT,
    PRIMARY KEY (id),
    METRIC (img)
        USING (<ID_fem1_ef1> DEFAULT, <ID_fem2_ef1>, <...>)
);
```

The differences between the attributes of tables `Know_Base` and `Data` is that `class` on `Know_Base` cannot be `null`, while `label` on `Data` is the system's classification. SimbA enables the inclusion of the `STILLIMAGE` data type, which stores the image content. Notice that, it is also necessary to specify how the images will be compared. Therefore, here we use the previously created image descriptors (metrics) as a comparison constraint to the image. According to this construction, it is possible to query by the k-nearest neighbors using one of the employed image descriptors. The $FFireDt$ architecture is organized in modules as depicted by Figure 2. The system is fed by the flow of images arriving from a social media service and the image analysis is carried out by the Evaluation Module. In this module, a query using extended-SQL is composed using an *image descriptor* and an unlabeled image. The SQL command is executed by the SimbA Engine as a kNN query and the output is the set of grouped elements:

```
SELECT COUNT(class)/k, class
FROM Know_Base R
WHERE R.img NEAR <Image i> BY <ID_fem1_ef1>
    STOP AFTER <k>
GROUP BY (R.class)
ORDER BY (R.class) DESC;
```

In this query, the `NEAR` predicate corresponds to the kNN query. The query output is used for the IB1 classification, in which the image is labeled as {`fire`, `not_fire`} according to the predominant class of its neighbors. In the core module, the $FFireDt$ system stores the image, the system classification, and the result of the kNN query. Therefore, users can use a visual interface to verify the results of system $FFireDt$ and of the automated classification; as the system stores the results of a kNN query, experts can also visualize the set of most similar images without the need of performing the search a second time. All images submitted for analysis are labeled and stored.
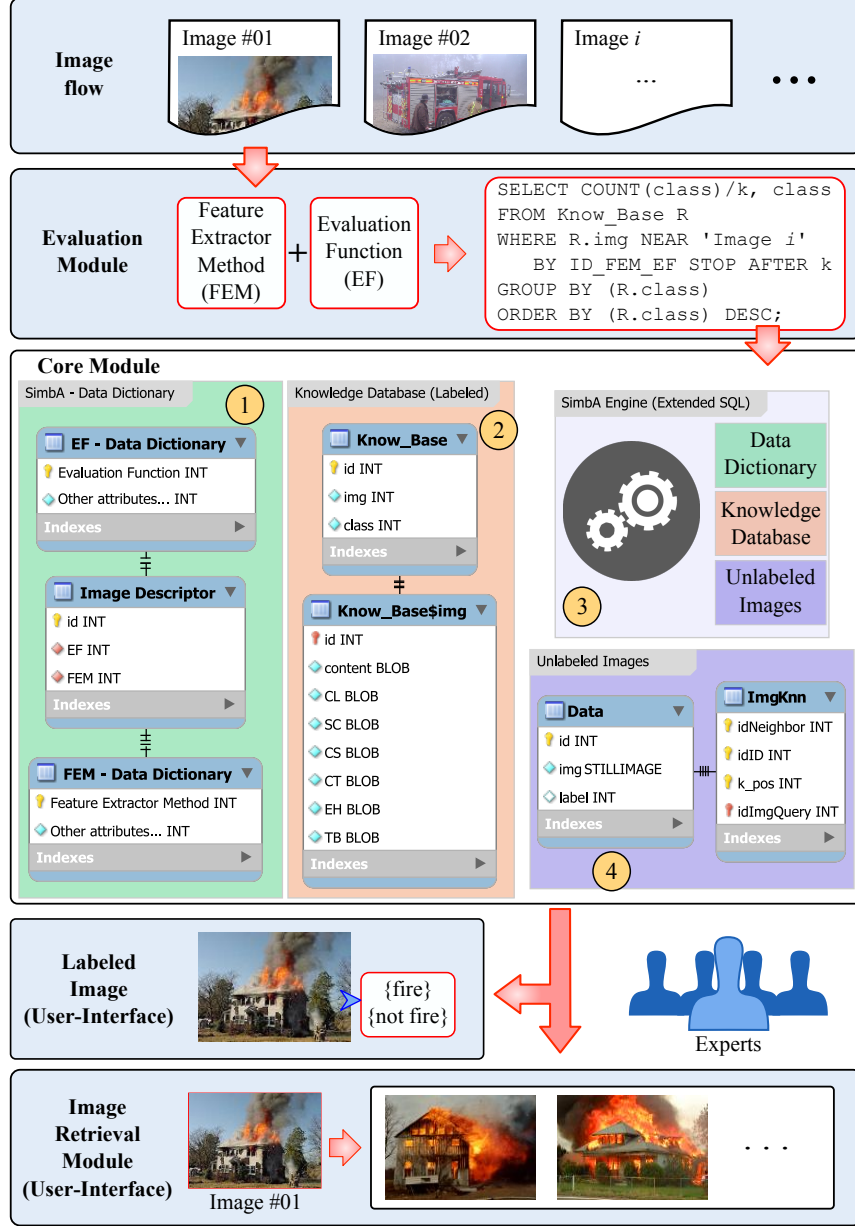
Fig. 2: *FFireDt* physical Design. A flow of images feeds the system, which generates an extended-SQL statement to perform a kNN query. The Core Module uses the SimbA Engine (3) to solve the query. It relies on the image descriptors stored in the Data-Dictionary (1) and in a ground-truth dataset to perform the classification (2). The SimbA Engine output for each unlabeled image is stored in two separated tables (4).

5 EXPERIMENTS

In this section, we analyze the combination of classifiers and image descriptors to identify the most suitable ones for the social media domain. We evaluate the impact of the image descriptors creating a candidate set of 36 descriptors, given by the combination of the 6 feature extractors with the 6 evaluation functions – as introduced in Section 4, and executing the IB1 classifier over the `Flickr-Fire` dataset.

We also measured the average time to perform the $FFireDt$ image classification over SimbA to evaluate the performance improvement achieved with a relational database. The experiments were performed using the following procedure:

1. Calculate the F-measure metric to evaluate the efficacy of each image descriptor (FEM-EF combination);
2. Select the top-six image descriptors, according to the F-measure; proceed by generating Precision-Recall plots, bringing more details about the behavior of the techniques;
3. Validate our partial findings using Principal Component Analysis to plot the feature vectors of the extractors;
4. Employ the top-three image descriptors, according to the previous results, to perform a ROC curve evaluation and to determine the most accurate $FFireDt$ setting;
5. Evaluate the efficiency of the $FFireDt$ architecture, measuring the wall-clock time, considering the multiple configurations of the descriptors;
6. Measure the average time to perform the fire detection strategy to evaluate the performance improvement regarding the use of $framework$ SimbA.

5.1 Obtaining the F-measure

We calculated the F-measure for the 36 image descriptors using 10-fold cross validation, in order to detect fire over the $Flickr - Fire$ dataset. That is, for each round of evaluation, we used one tenth of the dataset to test the IB1 classifier and the remaining data to train. It was performed 10 times and then the average F-measure was calculated. Table 3 presents the F-measure values for all the 36 combinations of feature extractor/evaluation function. The highest values obtained for each row are highlighted in bold. The experiment revealed that distinct descriptor combinations impact on fire detection. More specifically, we observed that the accuracy of extractors based on color is better than that of the extractors based on texture (Edge Histogram, and Texture Browsing).

Moreover, the extractors Color Layout and Color Structure have shown the best efficacy for fire detection, in combination respectively with the evaluation functions Euclidean and Jeffrey Divergence. The highlighted values are pointed out as the best settings for tuning the $FFireDt$ system. We considered the results adequate, since the best descriptor achieved an F-Measure up to 0.866, while the human labeling process accuracy was of 92.8%.

Table 3: F-Measure for each pair of feature extractor method (rows) versus evaluation function (columns). For each feature extractor, the evaluation function with the highest F-Measure is highlighted.

FEM	Evaluation Functions					
	CB	EU	CH	CA	KU	JF
CL	0.834	**0.847**	0.807	0.828	0.803	0.844
SC	**0.843**	0.827	0.811	0.835	0.671	0.798
CS	0.853	0.849	0.821	0.848	0.746	**0.866**
CT	0.799	0.798	0.798	**0.800**	0.734	0.799
EH	0.808	0.806	0.795	0.806	0.462	**0.815**
TB	**0.766**	0.762	0.745	0.751	0.571	0.755

We also compared the best combination achieved by the IB1 classifier, as reported in Table 4, with other classifiers. Those experimental evidences suggest that the instance-based learning (the $FFireDt$ approach) is the most adequate classification strategy for fire detection. To compare the $FFireDt$ classification with other classification paradigms, we experimented system $FFireDt$ with the best EF of each FEM, as reported in Table 3. The comparison results were grouped according to the employed FEM. Table 4 shows the $FFireDt$ results compared to Naïve-Bayes, J48, and RandomForest classifiers. The results show that $FFireDt$ achieved the best F-Measure among the competitors in every setting, but for FEM Scalable Color. In this case, the Random Forest strategy beats the $FFireDt$ classification, although by a narrow F-Measure margin.

Table 4: System $FFireDt$ obtained the highest F-Measure for all but one FEM when compared to other classifiers. For each feature extractor, we highlighted strategy with the highest F-Measure.

FEM	Classifiers			
	FFireDt	Naïve-Bayes	J48	Random Forest
CL	**0.847**	0.787	0.751	0.829
SC	0.843	0.808	0.845	**0.864**
CS	**0.866**	0.406	0.842	0.866
CT	**0.800**	0.341	0.800	0.774
EH	**0.815**	0.522	0.711	0.787
TB	**0.766**	0.476	0.706	0.723

5.2 Precision-Recall

In order to detail the analysis of Section 5.1, we also measured the Precision and Recall of the *image descriptor*s employed as the settings of system $FFireDt$.

This type of graphical analysis enables to better understand the behavior of the image descriptors and, more specifically, the behavior of the feature extractors.

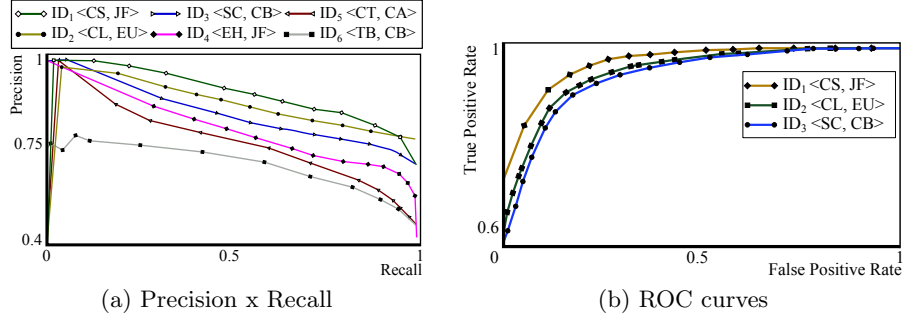(a) Precision x Recall   (b) ROC curves

Fig. 3: Evaluation of the descriptors: (a) *Precision vs. Recall* graphs for each of the most precise image descriptor combinations according to F-measure; and (b) ROC curves for the top three image descriptors in the task of fire detection.

A Precision vs. Recall (P×R) curve is suitable to measure the number of relevant images regarding the number of retrieved elements. We used Precision vs. Recall as a complementary measure to determine the potential of each image descriptor in the $FFireDt$ setting. A rule of thumb on reading P×R curves is: the closer to the top the better the result is. Accordingly, we consider only the most efficient combination of each feature extractor, as highlighted in Table 3: $ID_1\langle$CS, JF$\rangle$, $ID_2\langle$CL, EU$\rangle$, $ID_3\langle$SC, CB$\rangle$, $ID_4\langle$EH, JF$\rangle$, $ID_5\langle$CT, CA$\rangle$ and $ID_6\langle$TB, CB$\rangle$. Figure 3a confirms that the image descriptors ID_1, ID_2, and ID_3 are in fact the most effective combinations for fire detection. It also shows that, for those three descriptors, the precision is at least 0.8 for a recall of up to 0.5, dropping almost linearly with a small slope, which can be considered acceptable. This observation reinforces the findings of the F-measure metric, indicating that the behavior of the descriptors is homogeneous and well-suited for the task of retrieval and, consequently, for classification purposes.

5.3 Visualization of feature extractors

Based on the results observed so far, we hypothesize that the most adequate extractors to be used in system $FFireDt$ are Color Structure, Color Layout, and Scalable Color. In this section, we look for further evidence, using visualization techniques to better understand the feature space of the extractors using Principal Component Analysis (PCA). The PCA analysis takes as input the extracted features, which may have several dimensions according to the FEM domain, and reduces them to two features. Such reduction allows us to visualize the data as a scatter-plot. Our hypothesis shall gain more credibility if the corresponding visualizations allow seeing a better separation of the classes *fire* and
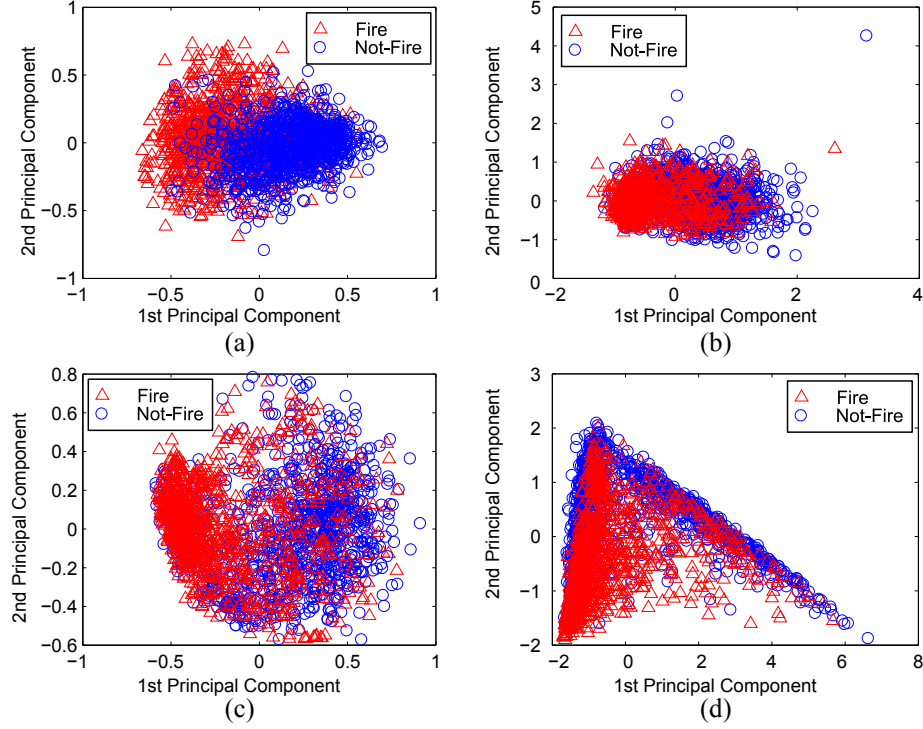
Fig. 4: PCA projection of fire and not-fire images: (a) Color Layout, (b) Color Structure, (c) Scalable Color and (d) Edge Histogram. The Color Layout visually separates the dataset into two clusters.

not-fire, in comparison to the other three extractors. Figure 4 presents the two-dimensional projection of the data, plotting the two principal components of the space generated by each extractor. Figure 4(a) depicts the representation of the data space generated by CL, the extractor that presented the better separability in the classification process. The two clusters can be seen as two well-formed clouds with a reasonably small overlapping, splitting the images as containing fire or not. Figure 4(b) shows the data visualization of the space generated by CS, which was the FEM that obtained the highest F-measure on previous experiments. The data projection shows that each cluster forms a cloud clearly identifiable, having the centers of the clouds distinctly separated. However, this figure reveals that there is a large overlap between the two classes. Figure 4(c) presents the projection of the space generated by the SC extractor. Again, it can be seen that there are two clusters, but with an even larger overlap between them. Visually, the CL outperformed the other color FEMS: the CS and SC, when drawing the border between the two classes. Figure 4(d) depicts the visualization generated by the EH extractor. It can be seen that indeed it has two clouds: one almost vertical to the left and another along the "diagonal" of the

figure. However, the two clouds are not related to the existence of fire, as the elements of both clusters are distributed over both clouds. Figures 4(a), 4(b) and 4 (c) show the visualization of the extractors based on color. The four visualizations show that the corresponding CL, SC and SC indeed generate clusters. However, there are increasing larger overlaps between fire and not-fire instances. Regarding the TB and CT features, the PCA projection was not able to separate the fire and not-fire classes. Concluding, the visualization of the feature spaces shows that extractors based on color are able to separate the data into visual clouds related to the expected clusters. Particularly, Color Layout has shown the best visualization, followed by Color Structure, and Scalable Color, which also have shown to significantly separate the classes. However, the extractors based on texture identify characteristics that are not related to fire, thus presenting the worst separability, as expected.

5.4 ROC curves

In order to determine the best setting for the $FFireDt$ system, we employed one last accuracy measure: the ROC curve. It allows us to determine the experiments overall effectiveness, using measures of sensitivity (recall) and specificity. Figure 3b presents the detailed ROC curves for image descriptors $ID_1 = \langle CS, JF\rangle$, $ID_2 = \langle CL, EU\rangle$, and $ID_3 = \langle SC, CB\rangle$, the top three best combinations according to the F-Measure, Precision-Recall and Visualization experiments. For fire-detection, the area under the ROC curve was up to 0.93 for ID_1; up to 0.87 for ID_2; and up to 0.85 for ID_3. These results indicate that the top three image descriptors have similar and satisfactory accuracy. Therefore, the choice of which descriptor to use becomes a matter of performance.

5.5 Processing Time and Scalability

When monitoring images originated from social media, the time constraint is important because of the high rate at which new images arrive. Thus, we also evaluate the efficiency, given in wall-clock time, of the candidate image descriptors. We ran the experiments in a personal computer equipped with a processor Intel Core i7 R 2.67 GHz with 4GB memory using the Ubuntu 14.04 LTS operating system.

Feature Extractors. Figure 5a shows the average time required to perform the feature extraction on system $FFireDt$ regarding `Flickr-Fire` dataset. Color Structure, the most precise extractor, was the second fastest. The second and third most precise extractors were Color Layout and Scalable Color: the former was three times slower than Color Structure, and the latter was the fastest extractor. Thus, we are now able to state the that extractors Color Structure and Scalable Color are the best choices for fire detection in image streams. Meanwhile, the texture-based extractors Edge Histogram, and Texture Browsing presented low performances, so they are definitely dismissed as

possible choices.

Evaluation Functions. Figure 5b shows the time required to perform 2 trillion evaluation calculations for each evaluation function on feature vectors of 256 dimensions. The plot *average precision vs. wall-clock time* shows that, although the Jeffrey Divergence demonstrated the highest precision, it was the least efficient. In their turn, City-Block and Euclidean distances presented excellent performance and a precision only slightly below the Jeffrey Divergence. Therefore, we can say that they are the most adequate evaluation functions when performance is on concern, such as is the case in our problem domain. Finally, we conclude that the image descriptors given by the combinations $\langle CS, SC \rangle \times \langle CB, EU \rangle$ are the best options in terms of both efficacy (precision) and efficiency (wall-clock time). In Table 5 we reproduce the F-measure results highlighting the most adequate combinations according to our findings.
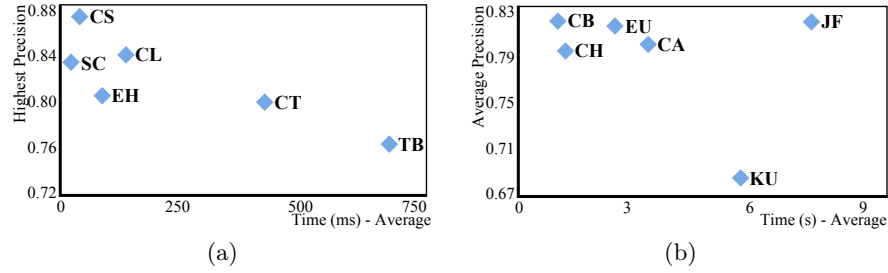

(a)   (b)

Fig. 5: Evaluation of the descriptors: (a) Plot *Highest precision when classifying dataset `Flickr-Fire` vs. Average time to extract the features of one image* for the six feature extractors; and (b) Plot *Average precision when classifying dataset `Flickr-Fire` vs. Time to perform 2 trillion calculations* for the six evaluation functions.

Therefore, according to all the experiments, we point out that the best image descriptor for fire detection, considering our social media dataset `Flickr-Fire`, is given by the combinations of extractors MPEG-7 Color Structure and Scalable Color with distance function City-Block. This combination provides not only more efficacy, but also more efficiency. In general, we noticed that feature extractors based on color were more effective than extractors based on texture. We also identified that the Jeffrey divergence was the most accurate, however, it was also the most expensive evaluation function. The next section presents the performance improvements achieved by the incorporation of similarity queries on system $FFireDt$.

5.6 Performance improvement

We used the most suitable *image descriptor* defined by the previous experiments to run the performance evaluation experiments. According to this, we measured

Table 5: F-Measure for each pair of feature extractor method (rows) and evaluation function (columns), now highlighting the best combinations according to our experiments.

FEM	Evaluation Functions					
	CB	**EU**	CH	JF	KU	CA
CL	0.834	0.847	0.807	0.844	0.803	0.828
CS	**0.853**	**0.849**	0.821	0.866	0.746	0.848
SC	**0.843**	**0.827**	0.811	0.798	0.671	0.835
EH	0.808	0.806	0.795	0.815	0.462	0.806
CT	0.799	0.798	0.798	0.799	0.734	0.800
TB	0.766	0.762	0.745	0.755	0.571	0.751

the average time of the $FFireDt$ system with the setting $ID = \langle SC, CB \rangle$ with and without the use of similarity queries to solve the Instance-Based Learning classification. We used the $Flickr - Fire$ dataset of Section 3.6 as the knowledge database of system $FFireDt$. Next, we performed a set of k-Nearest Neighbor queries for an increasing number of neighbors (k) over this knowledge database to label new 350 images obtained with the Flickr API. Figure 6 presents the average time demanded to perform fire detection using the $FFireDt$ system with and without the similarity queries, for k varying from 15 to 53.
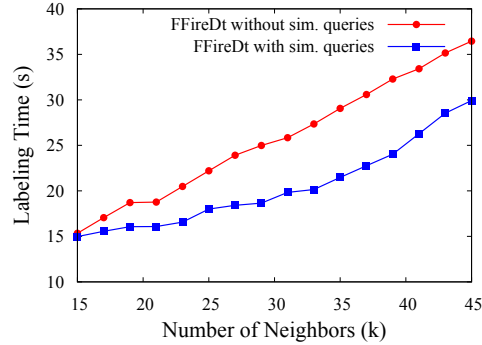

Fig. 6: The $FFireDt$ performance improvement due to the use of indexed similarity queries to solve the IB1 classification. The similarity queries are implemented by *framework* SimbA.

As it can be observed, the adoption of a similarity query engine (SimbA) that employs indexing techniques leads to a performance gain of 30%, on average, if compared to using a bare brute-force scan strategy. Besides, the integration of this methodology to an extended-SQL syntax allows for a rapid experimentation environment, which allowed the wide-scope evaluation (36 image descriptor combinations) reported in this paper. The indexing techniques internal to SimbA lead to performance improvements because they employ metric space properties

to prune the search space when solving a kNN query, reducing the number of distance computations.

6 CONCLUSIONS

In this work we studied the problem of quickly identifying fire in social-media images, in order to assist rescue services during emergencies. The approach was based on an architecture for content-based image storage, retrieval, and classification, relying on database relational technology. Using this architecture, we compared the accuracy and performance (processing time) of 36 image descriptors (feature extractor and evaluation function) in the task of identifying fire in the images. Since we used supervised machine learning (instance-based learning), we built a ground-truth dataset with 2,000 human-annotated images obtained from website Flickr. The results allowed us not only to achieve a high-efficacy classification system, but also to elucidate the adequacy of a vast set of image descriptor possibilities in relation to the social media domain. Our contributions are summarized as follows:

1. **Fast-Fire Detection (FFireDt) Architecture:** we designed and implemented a database-driven, scalable, and accurate architecture for content-based image retrieval and classification that was able to achieve a precision for fire detection that is comparable to that of human annotators;
2. **Evaluation of existing techniques:** we soundly compared 36 combinations of MEPG-7 image descriptors (feature extractor + evaluation function) considering their potential for accurate retrieval as indicated by metrics F-measure and Precision-Recall. As a result we achieved an accuracy of 85%, precise classification (as shown by ROC curves) and efficient performance (wall-clock time);
3. **Improved instance-based learning:** we described and evaluated the design of a classification architecture ($FFireDt$) for fire detection, whose performance suggests the mandatory use of indexed similarity querying in the construction of instance-based classification systems;
4. **Dataset Flickr-Fire:** we built a human-annotated dataset of real images suitable as ground-truth to foster the development of more precise techniques for automatic identification of fire – available online at `http://icmc.usp.br/pessoas/junio/DatasetFlicker/DatasetFlickr.htm`.

We conclude by stressing that detecting fire in social media images could help decision making during emergency situations. To this end, it is required to have automated tools that can flag relevant images from the social media as soon as possible. This work is a significant step towards this goal.

ACKNOWLEDGEMENTS

References

[Aha et al., 1991] Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66.

[Barioni et al., 2009] Barioni, M. C. N., Razente, H. L., Traina, Jr, A. J. M., and Traina, C. (2009). Seamlessly integrating similarity queries in sql. *Softw. Pract. Exper.*, 39(4):355–384.

[Bedo et al., 2015] Bedo, M. V. N., Blanco, G., Oliveira, W. D., Cazzolato, M., Costa, A. F., Rodrigues, J., Traina, A. J. M., and Traina Jr., C. (2015). Techniques for effective and efficient fire detection from social media images. In *International Conference on Enterprise Information Systems*, pages 34–46, Barcelona, Spain.

[Bedo et al., 2014] Bedo, M. V. N., Traina, A. J. M., and Traina Jr., C. (2014). Seamless integration of distance functions and feature vectors for similarity-queries processing. *Journal of Information and Data Management*, 5(3):308–320.

[Celik et al., 2007] Celik, T., Demirel, H., Ozkaramanli, H., and Uyguroglu, M. (2007). Fire detection using statistical color model in video sequences. *Journal of Visual Communication and Image Representation*, 18(2):176 – 185.

[Chunyu et al., 2010] Chunyu, Y., Jun, F., Jinjun, W., and Yongming, Z. (2010). Video fire smoke detection using motion and color features. *Fire Technology*, 46(3):651–663.

[Dimitropoulos et al., 2014] Dimitropoulos, K., Barmpoutis, P., and G., N. (2014). Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *Cirs. and Sys. for Video Technology*, PP(99):7–14.

[Doeller and Kosch, 2008] Doeller, M. and Kosch, H. (2008). The mpeg-7 multimedia database system (mpeg-7 mmdb). *Journal of Systems and Software*, 81(9):1559 – 1580.

[Guyon et al., 2006] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[IEEE MultiMedia, 2002] IEEE MultiMedia (2002). Mpeg-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87.

[Kaster et al., 2010] Kaster, D. S., Bugatti, P. H., Traina, A. J. M., and Traina Jr., C. (2010). FMI-SiR: A flexible and efficient module for similarity searching on Oracle database. *Journal of Information and Data Management*, 1(2):229–244.

[Kasutani and Yamada, 2001] Kasutani, E. and Yamada, A. (2001). The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *Int. Conf. on Image Processing*, pages 674–677 vol.1.

[Ko et al., 2009] Ko, B. C., Cheong, K.-H., and Nam, J.-Y. (2009). Fire detection based on vision sensor and support vector machines. *Fire Safety Journal*, 44(3):322 – 329.

[Kudyba, 2014] Kudyba, S. (2014). *Big Data, Mining, and Analytics: Components of Strategic Decision Making*. Taylor & Francis Group.

[Lee and Chen, 2005] Lee, K.-L. and Chen, L.-H. (2005). An efficient computation method for the texture browsing descriptor of mpeg-7. *Image Vision Comput.*, 23(5):479–489.

[Liu and Ahuja, 2004] Liu, C.-B. and Ahuja, N. (2004). Vision based fire detection. In *Int. Conf. on Pattern Recognition.*, volume 4, pages 134–137 Vol.4.

[Manjunath et al., 2001] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *IEEE Cir. and Sys. for Video Technol.*, 11(6):703–715.

[Ojala et al., 2002] Ojala, T., Aittola, M., and Matinmikko, E. (2002). Empirical evaluation of mpeg-7 xm color descriptors in content-based retrieval of semantic image categories. In *Int. Conf. on Pattern Recognition*, volume 2, pages 1021–1024 vol.2.

[Park et al., 2000] Park, D. K., Jeon, Y. S., and Won, C. S. (2000). Efficient use of local edge histogram descriptor. In *ACM Workshops on Multimedia*, pages 51–54. ACM.

[Russo, 2013] Russo, M. R. (2013). Emergency management professional development: Linking information communication technology and social communication skills to enhance a sense of community and social justice in the 21st century. In *Crisis Management: Concepts, Methodologies, Tools and Applications*, pages 651–663. IGI Global.

[Sato et al., 2010] Sato, M., Gutu, D., and Horita, Y. (2010). A new image quality assessment model based on the MPEG-7 descriptor. In *Advances in Multimedia Information Processing*, pages 159–170. Springer.

[Sikora, 2001] Sikora, T. (2001). The mpeg-7 visual standard for content description-an overview. *IEEE Cir. and Sys. for Video Technol.*, 11(6):696–702.

[Silva et al., 2010] Silva, Y. N., Aly, A. M., Aref, W. G., and Larson, P.-Å. (2010). SimDB: a similarity-aware database system. In *ACM International Conference on Management of Data*, pages 1243–1246, Indianapolis, Indiana, USA.

[Silva et al., 2013] Silva, Y. N., Aref, W. G., Larson, P.-Å., Pearson, S., and Ali, M. H. (2013). Similarity Queries: their conceptual evaluation, transformations, and processing. *The VLDB Journal*, 22(3):395–420.

[Sumathi and Esakkirajan, 2007] Sumathi, S. and Esakkirajan, S. (2007). *Fundamentals of relational database management systems*, volume 47. Springer.

[Tamura et al., 2012] Tamura, S., Tamura, K., Kitakami, H., and Hirahara, K. (2012). Clustering-based burst-detection algorithm for web-image document stream on social media. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 703–708. IEEE.

[Tjondronegoro and Chen, 2002] Tjondronegoro, D. and Chen, Y.-P. (2002). Content-based indexing and retrieval using mpeg-7 and x-query in video data management systems. *World Wide Web*, 5(3):207–227.

[Villela et al., 2014] Villela, K., Breiner, K., Nass, C., Mendonća, M., and Vieira, V. (2014). A smart and reliable crowdsourcing solution for emergency and crisis management. In *Interdisciplinary Information Management Talks*, pages 213–220, Podebrady, Czech Republic.

[Wnukowicz and Skarbek, 2003] Wnukowicz, K. and Skarbek, W. (2003). Colour temperature estimation algorithm for digital images - properties and convergence. In *Opto Eletronics Review*, volume 11, pages 193–196.

[Zezula et al., 2006] Zezula, P., Amato, G., Dohnal, V., and Batko, M. (2006). *Similarity Search - The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer Publishing Company, Berlin, Heidelberg.