Unveiling smoke in social images with the SmokeBlock approach[*]

Mirela T. Cazzolato, Marcos V. N. Bedo, Alceu F. Costa, Jessica Andressa de Souza,
Caetano Traina Jr., Jose F Rodrigues-Jr and Agma J. M. Traina
University of Sao Paulo - USP
P. O. Box 668 - Sao Carlos, Brazil
{mirelac, jessicasouza}@usp.br, {bedo, alceu, junio, caetano, agma}@icmc.usp.br

ABSTRACT

Can we use information from social media and crowdsourced images to detect smoke and assist rescue forces? While there are computer vision methods for detecting smoke, they require movement information extracted from video data. In this paper we propose `SmokeBlock`: a method that is able to segment and detect smoke in still images. `SmokeBlock` uses superpixel segmentation and extracts local color and texture features from images to spot smoke. We used real data from Flickr and compared `SmokeBlock` against state-of-the-art methods for feature extraction. Our method achieved performance superior than the competitors, for the task of smoke detection. Our findings shall support further investigations in the field of image analysis, in particular, concerning images captured with mobile devices.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Miscellaneous; H.2.8 [**Database Applications**]: Image databases

Keywords

Smoke Detection, Segmentation, Social Media, Feature Extraction

1. INTRODUCTION

Public authorities are expected to provide fast and accurate responses to emergency situations; a challenging decision-making duty that must deal with intense flows of information in little time. Computer systems have great potential to support this decision making process by providing timely information and data analysis [12]. The goal is the early detection of fire, smoke, and explosions to assist the rescue forces in preventing further risks to human life and financial losses. The current technology of fire and smoke detection systems is based on hardwired solutions that employ infrared or ultraviolet local sensors to determine the chemical presence of fire and/or of smoke particles. However, such solutions can be unsuitable to crowded urban areas, or large open spaces, due to the high cost of installation, communication, and maintenance of local sensors [8].

On the other hand, cameras embedded in mobile devices can provide visual digital information that can be an alternative to local fire/smoke sensors [9]. This approach is in the core of the RESCUER Project, which aims at developing a system that uses crowdsourcing[1] images and videos sent by mobile devices to support decision making during emergency situations. The fundamental functionality of the project's data analysis module is to identify the pieces of information that present useful data about fire, smoke, and/or explosion. Accordingly, this work introduces a new technique, named `SmokeBlock`, to detect smoke in still images.

Challenges and approach

One of the challenges to perform smoke detection in still images is the absence of movement. In fact, most of the techniques in the literature depend on a set of sequential images (video) to identify fire and/or smoke [8]. Unlike fire detection that can be solved by global analysis using rule-based color models [6], the smoke identification problem does not present well-defined color patterns. For instance, the color of the smoke may change as its temperature increases. Besides, the smoke color is heavily dependent on weather conditions and on the burned material [8]. Although the video-based strategies present good results for fire detection, the results of smoke identification in images are rather modest [8]. To overcome the challenges of smoke detection on still images, we employ superpixel segmentation [2]. A superpixel corresponds to a sub region of an image in which the pixels are coherent according to some visual feature. In this work we use color and texture. Superpixels capture redundancy and greatly reduce the complexity of subsequent image processing, having proved useful for depth estimation, image segmentation, skeletonization, body model estimation, and object localization [1].

Overview of the contribution

`SmokeBlock` is a method that detects smoke on still images without any information about time. To avoid the aforementioned limitations of rule-based color approaches – which are adequate for video analysis, `SmokeBlock` employs superpixel local segmentation along with multiple features (color, texture and shape) simultaneously. Our approach contrasts

[*] This research is supported by FAPESP, CAPES, and the RESCUER Project under grants EU 614154 and CNPq/MCTI 490084/2013-3.

SAC 2016, April 04-08, 2016, Pisa, Italy
© 2016 ACM. ISBN 978-1-4503-3739-7/16/04…$15.00
DOI: http://dx.doi.org/10.1145/2851613.2851634

[1] We assume that the crowdsourced data is available in a dedicated repository or in social media. For the sake of experimentation, we use data gathered from popular social media websites.

with previous works based on global segmentation and on one single feature. Given an image, SmokeBlock segments it into superpixels which are labeled either as smoke or non-smoke. Then, based on the labels of every superpixel, we classify the entire image using supervised machine learning. The result is an automatic image classification technique that identifies visual segments characterized by smoke. In summary, the main contributions of this paper are:

- SmokeBlock: an accurate method to find smoke regions in still images using superpixels with precision rates higher than existing competitors;
- Evaluation: in order to define the optimal setting for SmokeBlock, we compared the performance of several feature extractors and machine learning classifiers. Our comparison charts and conclusions serve as basis for other researchers on the field;
- Flickr-Smoke dataset: we curated a human-annotated dataset of real images from Flickr to work as ground-truth.

According to our experiments, SmokeBlock has surpassed its competitors by 12% in the task of smoke detection regarding F-Measure. Moreover, the SmokeBlock segmentation has reduced the false-positive rate up to 7% compared to previous works.

2. PROBLEM AND RELATED WORK

In this paper we are interested in solving the following problem: *given a set of images obtained from social media or crowdsourcing, find the subset of images that depict smoke while minimizing the rate of false-positives.*

We refer to the images that depict smoke as *positive examples* while images without visual traces of smoke are *negative examples.* In this study our goal is to find positive examples so that humans or resource intensive computer vision methods can analyze the images. Therefore, our secondary goal is to minimize the number of false-positives: images that do not depict smoke but are classified as smoke.

The previous approaches for smoke detection were designed for video data relying on motion analysis, which is not suitable for accurate smoke detection in still images [7, 16]. In comparison to works from the literature, SmokeBlock improves the state-of-the-art by using texture, in addition to color, to reduce false-positives. Our SmokeBlock relies on a small set of parameters that do not depend on the user to tune. To the best of our knowledge, the majority of reported techniques for smoke detection is related to video analysis [4, 6, 16]

In Çelik *et al.* [8] and Chen *et al.* [9] the authors propose a rule based model to classify the pixels as smoke based on respectively the RGB and HSI color spaces. Such models are limited by the coarse correspondence between the rule-based color models and the smoke characterization. Therefore, in such approaches, Motion Analysis [17] is required to produce acceptable results.

2.1 Background and Notation

An image I is a set of pixels, such that $I = \{P_i \mid 0 \leq i < n\}$, where n is the total number of pixels of I. A pixel P_i is a tuple $P_i = (R_i, G_i, B_i)$, where R_i, G_i, and B_i represent the pixel intensity in the RGB color space. In turn, a superpixel S, as explained in Section 1, corresponds to a subset $S \subseteq I$ defined as $S = \{P_j \mid 0 < j < m\}$, where m ($m \leq n$) is the number of pixels inside a superpixel. If $m = n$, then $S = I$. For our methodology, we consider sets of images denoted as $\mathcal{I}$ and sets of superpixels denoted as $\mathcal{S}$. A feature extractor method ε, or just extractor, is a non-bijective function such that $\varepsilon : \mathcal{S} \rightarrow \mathbb{R}^d$; i.e., given a superpixel or an image, ε maps it to a d-dimensional feature vector.

Also, let $\mathcal{L}$ be a set of labels, and $\mathcal{T} \subset \mathbb{R}^d$ be a set of vectors that represent a particular set of images $\mathcal{I}$ by using the extractor ε – we refer to $\mathcal{T}$ as training set. In the training set $\mathcal{T}$, every vector $t_i \in \mathcal{T}$ has a label $l_i \in \mathcal{L}$ assigned by an expert. For the remaining of this work, we consider that a supervised classifier C uses the set $\mathcal{T}$ to construct a model capable of predicting the label $l_j \in \mathcal{L}$ for any element $t_j \in \mathbb{R}^d$. Therefore, given a set of images (or superpixels) $\mathcal{S}$, and its corresponding set of feature vectors $\mathcal{T}$ computed with an extractor ε, a classifier is a function $C : \mathcal{T} \rightarrow \mathcal{L}$. That is, a classifier corresponds to a function C that maps a label $l_i \in \mathcal{L}$ to one given image $I \in \mathcal{I}$ based on its feature vector $t_i = \varepsilon(I)$. In this study $\mathcal{L} = \{$smoke, non-smoke$\}$ and the classifiers that we employ include Naive-Bayes [13], IBL [3], and RandomForest [5].

Superpixel segmentation

The pixels that will define a superpixel depend on given criteria, which are provided by a superpixel generation algorithm. A superpixel generation algorithm is described as the process $G(I, K_p) = \{S_j \mid 0 \leq j < K_p\}$ that takes an image I and generates K_p superpixels S_j, $S_j \subset I$. A comparative *survey* for the state-of-the art of superpixel algorithms can be found in the work of Achanta *et al.* [2]. The superpixel generation algorithms are expected to provide regions with good adherence to image boundaries in such way that the superpixels can be used to support segmentation strategies [14].

In this study, we used this principle to segment images into regions with pixels that could indicate the presence of smoke. To do so, we chose the SLIC (*Simple Linear Iterative Clustering*) strategy to generate the superpixels. According to the analysis of Achanta *et al.* [2], SLIC is a fast, memory efficient algorithm that uses an improved version of k-means for superpixel generation.

Feature extraction

The features extracted from a superpixel are numerical values that represent its visual features. Such features capture characteristics of the entire image (global extractors) or for image sub-regions (local extractors). Both global and local extractors can represent low-level features based on color, texture and shape. For instance, Color-based extractors usually have low computational cost when compared to texture and shape extractors. On the other hand, the MPEG-7 Color Layout extractor captures spatial color distribution, while the Scalable Color extractor represents color distribution into an HSV space followed by a Haar compression. Another MPEG-7 color-based extractor is the Color Structure, which enables the representation of both spatial and color distributions [15].

On the other hand, texture-based extractors describe the surface of natural elements, what is suitable for the representation of smoke. Examples of texture extractors are the Haralick features [10], the Local-Binary Pattern and its variant, the Texture-Spectrum [11]. Finally, shape-based extractors usually depend on a previous edge detection/segmentation step. A representation of shape and moments is generated for each object that was identified.

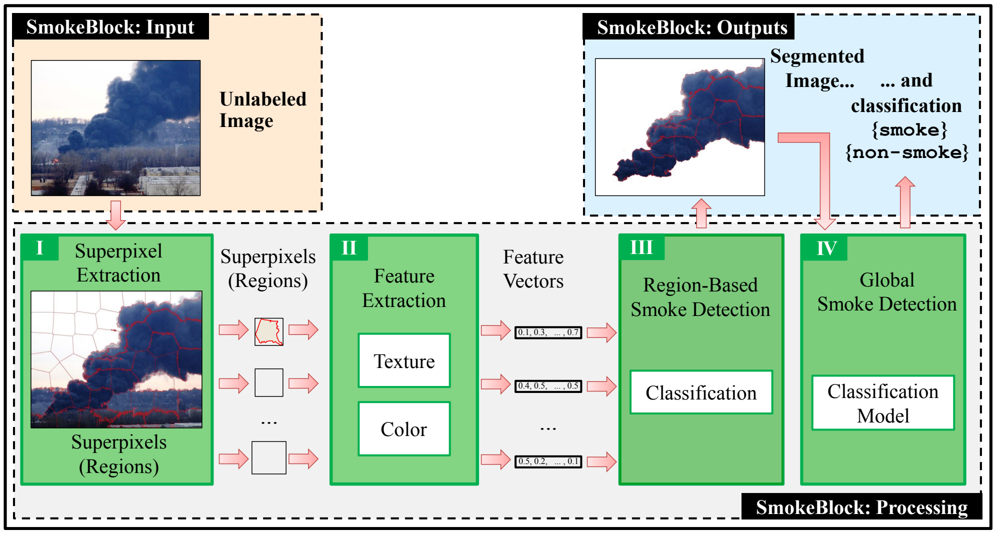


Figure 1: The execution pipeline of `SmokeBlock` method. For a given image, it provides superpixel-based segmentation and smoke detection. The `(I) Superpixel Extraction` module extracts superpixels from the input image. Each superpixel is represented as a feature vector by the `(II) Feature Extraction` module. The feature vectors are then classified by the `(III) Region-Based Smoke Detection` module, which generates the automatic segmented image with smoke regions. Finally, the `(IV) Global Smoke Detection` module performs global smoke classification.

3. PROPOSED METHOD: SmokeBlock

In this section, we introduce our proposal for smoke detection, the `SmokeBlock` method. The pipeline of the `SmokeBlock` execution is presented in Figure 1, which is divided into four modules: (I) Superpixel Extraction, (II) Feature Extraction, (III) Region-Based Smoke Detection, and (IV) Global Smoke Detection. Given an image I, `SmokeBlock` starts at the Superpixel Extraction (I) module by creating a set $\mathcal{S}$ of K_p superpixels $S_j \in \mathcal{S}$ that potentially contain traces of smoke. In the Feature Extraction (II) module, we extract a set of low-level features from each generated superpixel using the Haralick and Color Layout feature extractors. The advantages of extracting features from superpixels instead of individual pixels are the following:

- Reduction of redundancy: pixels with similar visual properties that are spatially close are processed together (as a single super-pixel) improving performance;
- By analyzing a group of pixels we extract texture features, which is not possible for individual pixels.

Smoke Segmentation: For the Region-Based Smoke Detection Module (III), we constructed a training set $\mathcal{T}$ of superpixels manually annotated according to the labels of set $\mathcal{L}$. That is, with or without smoke[2]. We use the training set to train a binary Naive-Bayes classifier $C : \mathcal{T} \rightarrow \mathcal{L}$ that decides whether a superpixel depicts smoke or not, based on its feature vector. The output of the module III is a segmented image as shown in Figure 1.

Smoke Detection: In the Global Smoke Detection Module (IV), `SmokeBlock` can also be used to decide whether an image presents smoke or not. This is useful, for instance, in a scenario in which we have a large set of images from a social media service, and we want to find suspicious images (that are likely to contain smoke). A naive approach would be to use the output of module III and classify an image as positive (suspicious) if at least one superpixel was classified as depicting smoke. The drawback of this approach is that if a single superpixel is wrongly classified, the entire image will also be wrongly classified as containing smoke.

To tackle this problem, `SmokeBlock` extracts a new set of feature vectors now regarding only the regions of I that were segmented by module III. Then, a final representation of the segmented image is built using the Principal Component Analysis dimensionality reduction method over the feature vectors. Finally, our Classification Model (IV) relies on a supervised classifier to label the segmented image representation as `{smoke, non-smoke}`. This approach is superior when compared to global feature extraction (see experimental results in Section 4), because `SmokeBlock` only extracts features from regions of the image that already is a candidate of having smoke.

Algorithm 1 describes the procedure used by the `SmokeBlock` Global Smoke Detection Module. `SmokeBlock` takes as input an image I and the trained Naive-Bayes classifiers. After finding the superpixels, `SmokeBlock` generates a segmented image I_{seg} where all superpixel that were not classified as smoke candidates are not considered for extraction. The color and texture feature vectors v_{color} and v_{texture} are concatenated into a single feature vector v that is classified either as smoke or non-smoke.

Algorithm 1: SmokeBlock

```
Input  : Unlabeled image $I$
         Superpixel classifier $C_S$, Image classifier $C_I$.
Output : Image label $l \in \{\text{smoke}, \text{not-smoke}\}$.
$I_{\text{seg}} \leftarrow I$, $\mathcal{S} \leftarrow$ SLIC($I$, $K_p$);
for $S_i \in \mathcal{S}$ do
    if $C_S(S_i) \neq smoke$ then
        $I_{seg}[S_i] \leftarrow \emptyset$;
$v_{\text{color}} \leftarrow \varepsilon_{\text{color}}(I_{\text{seg}})$, $v_{\text{texture}} \leftarrow \varepsilon_{\text{texture}}(I_{\text{seg}})$;
$v \leftarrow v_{\text{color}} \oplus v_{\text{texture}}$;
return $l \leftarrow C_I(v)$;
```

[2] The details of this manually annotated dataset are reported in Section 4.2.

SmokeBlock was developed in a modularized scheme that allows the addition of new feature extractors and classifiers. Moreover, it was designed to be integrated with existing fire detection methods, what enables the development of a *framework* for fire and smoke detection.

4. EXPERIMENTAL ANALYSIS

In this section, we describe experiments to evaluate the performance and to define the best settings for SmokeBlock. We evaluate the impact of color, texture, and shape extractors for smoke segmentation when compared to rule-based color models, as employed in related works. We also evaluate the precision and the number of false-positives generated by SmokeBlock regarding the global smoke detection procedure. The goals of our experiments are to determine:

1. *The most suitable low-level features for SmokeBlock:* in the first experiment, we evaluate the impact of color and texture feature extractor methods to represent the segments that are candidates for smoke detection. The extractors with best performance are used in the remaining experiments;
2. *The accuracy of SmokeBlock for smoke segmentation:* in this experiment, we compare our proposal against state-of-the-art competitors, discussing the improvements and limitations of SmokeBlock for smoke segmentation;
3. *The SmokeBlock performance for smoke detection:* in the last experiment, we evaluate the superpixel-based classification of SmokeBlock in comparison to global classification using the following metrics: F-Measure, precision and recall.

We used the same setup for all experiments: Intel Core i7-4770(3.40GHz), 16GB RAM, Ubuntu 14.04 (64-bit) OS. The SLIC parameter k – the number of desired superpixels – was set to 100, according to empirical tests.

4.1 Dataset

To simulate the environment proposed by the RESCUER project, we used the Flickr API[3] to download images from the Flickr social network under the Creative Commons license. The images where retrieved using textual queries such as: “smoke fire”, or “smoke forest”. In order to build the ground-truth for evaluation each image was manually annotated. To perform this annotation, we asked seven subjects, all of them aging between 20 and 30 years, familiar with the issue, and non-color-blinded. Two subjects labeled each image. For images with divergent annotations, we asked a third subject to untie the classification. The average disagreement was 7.2%. The dataset includes 832 images labeled as smoke and 834 as non-smoke. We made the dataset available online[4] to allow the reproduction of our experiments

4.2 Visual Features for Smoke Detection

The content representation of a super pixel depends on the employed feature extraction method. Therefore, for the SmokeBlock method, we must take into account the question:

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

To provide a proper answer to this question, we experiment with a representative set of ten widely employed feature extraction methods regarding color, texture and shape. The color set includes the following extractors Color Layout, Color Structure, Color Temperature, Scalable Color, and Normalized Histogram; the texture set includes the extractors Edge Histogram, Haralick, LBP, and Texture Spectrum; and the shape set includes the Zernike extractor. The details and implementation of every extractor can be obtained by using the Artemis image processing library[5].

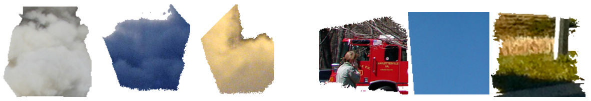

(a) Smoke (b) Non-smoke

Figure 2: Samples of superpixels manually classified for the purpose of training the machine learning classifiers used in our methodology. (a) Superpixels with smoke, and (b) Superpixels without smoke.

To evaluate the representativeness of each extractor, we selected 1,202 superpixels from the Flickr-Smoke dataset, where half of them were manually labeled as depicting smoke and the other half as non-smoke. Figure 2a presents six examples of superpixels manually labeled as smoke, whereas Figure 2b depicts six examples of non-smoke labeled superpixels. In the evaluation of the feature extractors, we used the Naive-Bayes classifier. With this classifier, we computed the F-Measure (harmonic mean between the precision and recall) obtained by each feature extractor using ten-fold cross validation. Figure 3 presents the results: the vertical axis presents the F-Measure and the horizontal axis the dimensionality of each extractor. The best extractors (yellow region) are those with high performance and low dimensionality (high compactness). This is because SmokeBlock combines texture and color features. Therefore, the low-level features with compact representation are those suitable to deal with the “dimensionality curse”. The results illustrated by Figure 3 provide the answer for question Q1. Hence, based on this plot, we selected the extractors Color Layout and Haralick which have the lowest dimensionality and highest F-Measure, respectively.

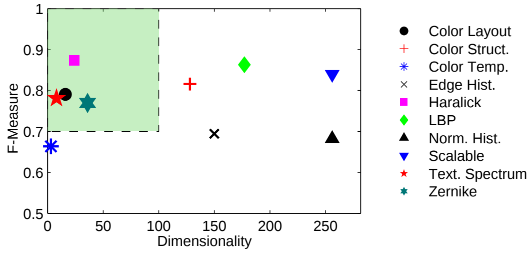


Figure 3: Comparison of ten feature extractors regarding performance – as given by the F-Measure and dimensionality. The best extractors refer to those with high performance and low dimensionality (inside green square).

4.3 Smoke Segmentation

This part of the experiments presents a qualitative answer to the following question: *Q2. Is our approach for smoke*

[3] The Flickr API: www.flickr.com/services/api/.

[4] Available at: github.com/alceufc/SmokeBlockDataset

[5] Artemis Library: gbdi.icmc.usp.br

segmentation more suitable than existing ones? To answer this second question, we compare our method against the related works of Çelik *et al.* [7] and Chen *et al.* [9].

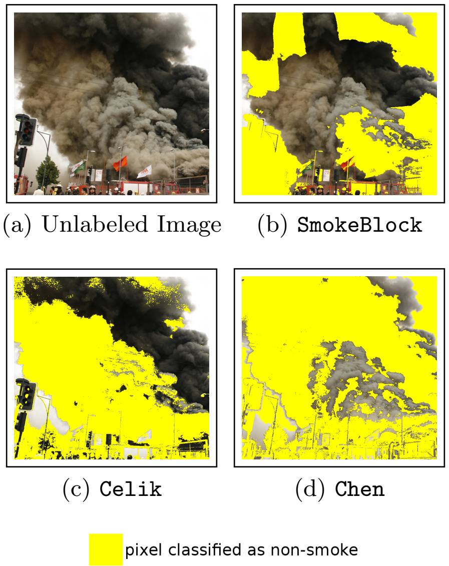


Figure 4: SmokeBlock provided the most accurate smoke segmentation.

For a fair comparison, we tuned the methods of Celik and Chen with the best parameters as achieved with empirical tests. Regarding the Celik method, its single global threshold parameter was set to 25. For the Chen method, the alpha value was set to 18 and the remaining parameters set to their default values. Figures 4 and 5 present the differences between the strategies for two representative situations. Figure 4 shows an image containing regions with smoke. SmokeBlock presented the highest accuracy for the detection of candidate smoke regions while the competitors (Celik and Chen) failed to identify most of the smoke pixels. On the other hand, Figure 5 presents an image without any traces of smoke. Even though, the Celik and Chen methods detected smoke pixels (false-positives), the SmokeBlock strategy correctly discarded those. SmokeBlock results reinforce our hypothesis that the combination of local color and texture improves smoke detection.

4.4 Smoke Detection

Our last experiment provides an analysis to answer the following question: *Q3. Can SmokeBlock spot smoke on images better than global-feature extractor methods?* To answer this question we perform a ten-fold cross validation over the Flickr-Smoke dataset comparing the results generated by SmokeBlock features extracted from the entire image. We refer to the competitors as Global Color Layout, Global Haralick, Global Texture Spectrum and Global Zernike, according to the visual feature employed. As discussed in Section 3, SmokeBlock is different from these approaches because it only extracts visual features from regions that have a high probability of having smoke.

We performed the smoke detection using three different classifiers for the SmokeBlock's Global Smoke Detection module (IV), namely Naive-Bayes (NB), IBL and Random Forest (RF). Table 1 compares SmokeBlock F-Measure to its competitors using the three classifiers. Notice that, SmokeBlock beats all competitors regardless of the classifier strategy.

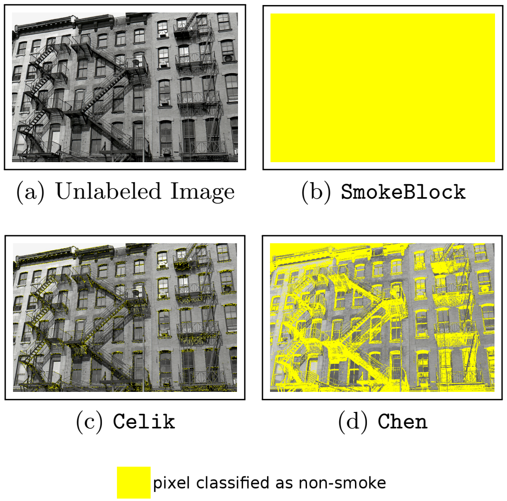


Figure 5: SmokeBlock correctly discarded regions without smoke (b). Celik (c) and Chen (d) misclassified a large number of pixels.

Table 1: F-measure comparison considering classifiers and features. We compare SmokeBlock with the global features Color Layout (CL), Texture Spectrum (TS), Zernike (ZK) and Haralick (HR). The highest values of each row are presented in bold.

	F-Measure				
Classifier	Global CL	Global TS	Global ZK	Global HR	SmokeBlock
NB	0.59	0.46	0.51	0.50	**0.64**
IBL	0.61	0.61	0.51	0.61	**0.68**
RF	0.60	0.61	0.51	0.62	**0.71**

We also highlight that neither the color-only nor the texture-only methods were capable of detecting smoke for a F-Measure higher than 0.64. Figure 6 compares the precision, recall, and F-measure for the entire Flickr-Smoke dataset using the RandomForest classifier. In absolute numbers, SmokeBlock achieved an improvement over the global strategies of Color Layout by 14%, Texture-Spectrum by 12%, Zernike by 23% and Haralick by 12% with respect to F-Measure.

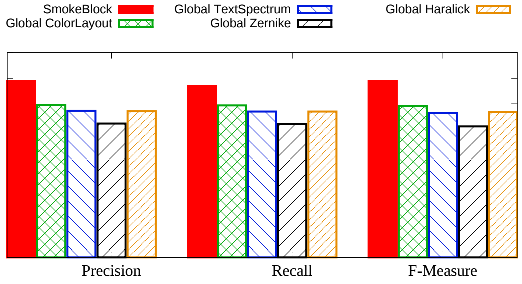
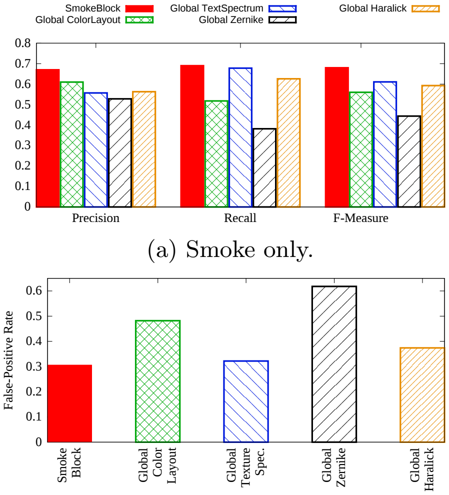


Figure 6: SmokeBlock achieved the highest precision, recall, and F-Measure. The results were computed using the entire Flickr-Smoke dataset.

Figure 7a presents the complete analysis for just the smoke images of the Flickr-Smoke dataset, including precision, recall, and F-Measure. As it can be seen, SmokeBlock presented the best overall performance by a margin of 9% for precision. Although Color Layout presented the second higher value for precision, its recall was the second lowest. Also, to illustrate the behavior of the experimented strategies, we verify the number of false-positive classification for just the portion of the Flickr-Smoke dataset that contain images without smoke. Figure 7b presents the false-positive rate for the subset of Flickr-Smoke that does not contain smoke images. In this case, we verify that SmokeBlock obtained the lowest value for false-positive rate, which is a desirable result. The precision achieved by the Color Layout method (as shown in Figure 6) can be justified by means of its highest false-positive rate for non-smoke images, which is not a desirable behavior. The same rationale can be employed to the Haralick strategy. Therefore, both strategies – Color Layout and Haralick – are unsuitable for smoke detection in the perspective of avoiding false-positives. Moreover, by using previous segmented regions represented by color and texture, SmokeBlock reduced the false-positive rate for non-smoke images up to 7% compared to Texture-Spectrum.



(a) Smoke only.

(b) Non-smoke only. Lower FPR is better.

Figure 7: Comparison regarding the FPR for the smoke-only and non-smoke-only subsets of Flickr-Smoke.

5. CONCLUSIONS

In this work we dealt with the problem of smoke detection in images produced from mobile devices with the goal of assisting rescue services during emergency situations. For experimentation, we used social-media images. Our hypothesis was that using only global or individual pixel color-based features, as reported in former works, is not enough to perform smoke identification with reduced number of false-positives. Therefore, we proposed the SmokeBlock method, which is based on superpixel (local) segmentation and employs color and texture visual features. With this approach, we obtained two outputs: the classification of images as with (or without) smoke, and the segmentation of images regarding their regions containing smoke. The experiments validated our hypothesis leading to the conclusion: *In the task of smoke detection, image classification based on global color-based features is outperformed by classification based on local multiple features.* Furthermore, this paper introduced the following contributions: *(i)* SmokeBlock: we designed and implemented a flexible, scalable, and accurate method for smoke detection to be used as a model for future developments in the field, *(ii)* Evaluation: we extensively compared SmokeBlock against state-of-the-art techniques on feature extraction and image classification, providing relevant conclusions that may support further works on smoke detection, and *(iii)* Flickr-Smoke dataset: we built a human-annotated dataset of real images suitable as ground-truth for similar problems concerning image classification.

6. REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk. SLIC Superpixels. Technical report, EPFL, 2010.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk. SLIC Superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012.

[3] D. W. Aha, D. Kibler, and M. K. Albert. Instance based learning algorithms. *ML*, 6(1):37–66, 1991.

[4] R. Bohush and N. Brouka. Smoke and flame detection in video sequences based on static and dynamic features. In *Signal Processing*, pages 20–25, 2013.

[5] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[6] S. Calderara, P. Piccinini, and R. Cucchiara. Vision based smoke detection system using image energy and color information. *Mach. Vis. and App.*, 22(4):705–719, 2011.

[7] T. Çelik and H. Demirel. Fire detection in video sequences using a generic color model. *Fire Safety J.*, 44(2):147 – 158, 2009.

[8] T. Çelik, H. Özkaramanh, and H. Demirel. Fire and smoke detection without sensors: Image processing based approach. In *EUSIPCO*, pages 1794–1798, 2007.

[9] T. Chen, Y.-H. Yin, S. Huang, and Y. Ye. The smoke detection for early fire-alarming system base on video processing. In *IIH-MSP*, pages 427–430, 2006.

[10] R. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *TSMC*, 3(6), 1973.

[11] D.-C. He and L. Wang. Unsupervised textural classification of images using the texture spectrum. *Patt. Recogn.*, 25(3):247–255, 1992.

[12] S. Huang, F. Cheng, and Y. Chiu. Efficient contrast enhancement using adaptive gamma correction with weighting distribution. *Trans. on Image Process.*, 22(3):1032–1041, 2013.

[13] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *UAI*, pages 338–345, 1995.

[14] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *CVPR*, pages 2097–2104, 2011.

[15] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *TCSVT*, 11(6):703–715, 2001.

[16] H. Tian, W. Li, L. Wang, and P. Ogunbona. Smoke detection in video: An image separation approach. *Int. J. of Comp. Vis.*, 106(2):192–209, 2014.

[17] S. Varadarajan, P. Miller, and H. Zhou. Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. *Patt. Recogn.*, 48(11):3488–3503, 2015.