

# Wildfire Smoke Detection Using Computational Intelligence Techniques

Angelo Genovese, Ruggero Donida Labati, *IEEE, Member*,  
Vincenzo Piuri, *IEEE, Fellow*, Fabio Scotti, *IEEE, Member*

Department of Information Technology  
Università degli Studi di Milano

Milano, 20122, Italy.

{angelo.genovese, ruggero.donida, vincenzo.piuri, fabio.scotti}@unimi.it

**Abstract**—In this paper, we propose an image processing system for the detection of wildfire smoke based on computational intelligence techniques and capable of adapting to different applicative environments. The proposed system is designed for processing with limited computational complexity. The detection process focuses on the extraction of specific features of wildfire smoke. A computational intelligence classifier is adopted to identify the presence of smoke. In order to test its effectiveness, the proposed system has been tested with low quality frame sequences, providing the capability to deal also with low cost cameras. The results indicate that the proposed approach is accurate and can be effectively applied in different environmental conditions.

**Index Terms**—computer vision, neural networks, smoke detection, forest fires.

## I. INTRODUCTION

Forest fires are one of the main problems in regions with hot climate and extensive vegetation. The work in [1] reports that, each year, about 0.1% of the world forest surface is destroyed by fires. In most cases, manned surveillance towers are adopted to watch forest areas which present the greatest risk of fire. Lookout towers equipped with cameras are a more feasible approach, since many cameras from different locations and points of view can be monitored by a single control station. Recently, many research projects have studied the possibility to develop automatic fire and smoke detection systems based on sensor networks or machine vision techniques, in order to achieve a better efficiency and a shorter alarm response time. These approaches have the advantages of a great distance vision, absence of latency, and the possibility to extract more information (such as position, size, growth, and kind of fire and smoke) [2].

In this paper, we propose a new approach for real-time detection of forest fire smoke, which is designed for low power and low cost platforms. Since the smoke is visible before the forest fires [3], our approach focuses on smoke detection. In order to keep the approach feasible for low cost hardware, we consider only images captured under visible light. The approach takes into account bad quality frame sequences and employs computational intelligence techniques, in particular describing the adoption of neural networks.

The paper is structured as follows: in Section II a literature review is presented, in Section III the proposed approach is

described, and Section IV describes the experimental results.

## II. PREVIOUS WORK

Machine vision systems for fire and smoke detection can be divided in two main categories: systems based on dedicated IR/UV cameras, and systems operating under visible light. In the first case, simpler image processing techniques are needed, but the cost of the camera hardware is higher. In the second case, however, image processing and detection algorithms are less straightforward to design.

Machine vision algorithms for smoke detection can also be divided in two different classes: methods based on single images or multiple images. The algorithms of the first class can be considered as a special case of image segmentation techniques. Methods pertaining to the second class can also use dynamic characteristics [4]. The vast majority of smoke detection algorithms operating in visible light conditions belongs to the second class and uses approaches based on color, texture, movement, shape, frequency, and energy.

A method that uses color features is presented in [5], and it is based on searching the pixels with similar values in the RGB color space. Texture features are used by the methods presented in [6 – 8]. The method proposed in [7] performs the smoke detection by first extracting high variability regions from the frame sequences and then comparing the texture features with characteristics previously extracted from each smoke image. The approach described in [8] is based on a predetermined color model for the smoke, and considers the gradual blending of the model in the frame. Texture features such as Tamara features, gray intergrowth matrix, gray level co-occurrence matrix (GCLM) are also used in combination with wavelet features [9] by the system presented in [1].

Movement and shape features such as circumference, area, direction and growth are used by the works described in [10 – 12]. Frequency and energy features are usually exploited with the use of wavelet transform or Fourier coefficients in order to detect the random behavior typical of smoke clouds regions. Such features are often used in combination with other features, as in the methods presented in [13 – 16].

The combined use of visual features, shape descriptions, wavelet features and statistical modeling is described in [17, 18], together with an approach especially directed to the

detection of long-distance wildfire smokes. This approach describes the differences in visual, shape and frequency features of smoke seen from long distances (some kilometers) and introduces the problem of false alarms caused by clouds. The used features are based on the color of smoke and on the detection of regions with rising intensity in subsequent video frames.

Another approach for the forest fire detection is described in [12], and it is based on motion detection, analysis of moving edges and shape properties, such as growing and rising of moving pixels.

Machine vision systems for fire and smoke detection can be based on different classifiers. The methods for the smoke detection in medium-range and open areas depicted in [19, 20] are based on moving region detection and texture analysis, and use probabilistic and Support Vector Machines (SVM) classifiers. SVM are also used in [10, 15, 16]. Neural Networks are used in [9, 14]. Other methods use probabilistic techniques, such as Hidden Markov Models [18] and Bayesian Classifiers [21].

A summary of machine vision techniques for wildfire, medium and long range smoke detection is depicted in Table I.

Commercial machine vision systems are also available and used in the process of wildfire monitoring [23 – 25]. However, they are often wired and expensive. In many situations commercial systems are used in combination with meteorological data and geographical information systems, such as the systems described in [22, 26]. These systems employ a multi-agent architecture with an image processing stage for smoke detection. This stage is based on methods for motion detection, image segmentation, dynamic pattern analysis, color-space analysis and texture analysis.

### III. THE PROPOSED APPROACH

The proposed approach is designed for low quality and low resolution video acquisitions (eg.  $320 \times 240$  pixel at 7 FPS), in order to use off-the-shelf hardware and keep the image processing techniques as fast as possible, while keeping the memory consumption at a minimum. In order to achieve this result, the proposed method is designed to keep track only of a small number of frames, varying from 5 to 10.

The focus on long-distance wildfire makes frequency analysis less discriminant because whirls and spirals typical of its fluid behavior are not detectable and much details are lost. Due to the low contrast of the frame sequences and different atmospheric conditions, color-space analysis is difficult to perform. For these reasons, the extracted features focus on a general behavior, described by movement characteristics.

The proposed approach extracts different features from the input frame sequences and then classifies each pixel into two different categories: “fire” and “not fire”. The feature extraction method is divisible in the sequent steps:

- 1) moving region detection;
- 2) smoke-color analysis;
- 3) sharp edge detection;

- 4) growing region detection;
- 5) rising region detection;
- 6) perimeter disorder analysis.

The extracted features are then used by computational intelligence classifiers.

#### A. Feature extraction

1) *Moving region detection*: The first step consists in the moving region detection, in which the candidate smoke regions are extracted. The moving region detection step is based on the methods proposed in [27] and [17]. The proposed method uses an adaptive system to update the background estimation and the thresholds. The background estimation is computed as follows:

$$B(x, y, t + 1) = \begin{cases} aB(x, y, t) + (1 - a)I(x, y, t) & \text{if } (x, y) \text{ is stationary} \\ B(x, y, t) & \text{if } (x, y) \text{ is a moving pixel} \end{cases}, \quad (1)$$

where  $I(x, y, t)$  represents the intensity of the pixel at location  $(x, y)$  in the  $t$ -th frame of the frame sequence  $I$ ,  $B(x, y, t)$  is the previous estimated background intensity at the same pixel position,  $a$  is a positive real constant close to one. Initially,  $B(x, y, 0)$  is set equal to the first frame  $I(x, y, 0)$ . A pixel positioned at  $(x, y)$  is assumed to be moving if it satisfies the disequations:

$$|I(x, y, t) - I(x, y, t - 1)| > T_I(x, y, t), \quad (2)$$

$$|I(x, y, t) - I(x, y, t - 2)| > T_I(x, y, t), \quad (3)$$

where  $I(x, y, t - 1)$  is the intensity of the pixel at the location  $(x, y)$  in the  $(t - 1)$ -th frame  $I$  and  $T_I(x, y, t)$  is a threshold updated at each frame, according to the equation:

$$T_I(x, y, t + 1) = \begin{cases} bT_I(t) + (1 - b)(c|I(x, y, t) - B(x, y, t)|) & \text{if } (x, y) \text{ is stationary} \\ T_I(t) & \text{if } (x, y) \text{ is a moving pixel} \end{cases}, \quad (4)$$

where  $c$  is a real constant greater than one and  $b$  is a positive constant close to one. Initial threshold values are set to a pre-determined non-zero value.

In order to detect even slow moving regions, as described in [17], it is possible to use two different background estimations,  $B^{fast}(x, y, t)$  and  $B^{slow}(x, y, t)$ .  $B^{fast}(x, y, t)$  is updated at every frame and  $B^{slow}(x, y, t)$  is updated every second. For every pixel  $(x, y)$ , the value  $D_M(x, y, t)$  representing its motion is computed as follows:

$$D_M = \begin{cases} 0 & \text{if } |B^{fast} - B^{slow}| \leq T_{low} \\ \frac{|B^{fast} - B^{slow}| - T_{low}(t)}{T_{high} - T_{low}} & \text{if } \leq T_{low} |B^{fast} - B^{slow}| \leq T_{high} \\ 1 & \text{if } \leq T_{high} |B^{fast} - B^{slow}| \end{cases}, \quad (5)$$

where  $0 < T_{low} < T_{high}$  are threshold values. The result is a matrix  $D_M(x, y)$  with values in the range  $[0, 1]$ . The resulting matrix  $D_M$  is thresholded in order to reduce the computational time and by considering regions with low values of movement.

TABLE I  
SUMMARY OF MACHINE VISION TECHNIQUES FOR MEDIUM AND LONG RANGE SMOKE DETECTION

Reference	Methodology	Database size	Performance
[18]	Background estimation, smoke colour detection using YUV thresholds, rising region detection with HMM, shadow detection with RGB rules, classification with LMS active learning (compared with WMA and fixed weights).	5 Videos 500 to 1000 frames long, public database at: <a href="http://signal.ee.bilkent.edu.tr/VisiFire/">http://signal.ee.bilkent.edu.tr/VisiFire/</a>	0.4% false alarms. <sup>1</sup>
[12]	Motion detection, spectrum analysis of moving edges, moving shape properties (growing, rising, expanding).	Videos taken from database at: <a href="http://signal.ee.bilkent.edu.tr/VisiFire/">http://signal.ee.bilkent.edu.tr/VisiFire/</a>	0% false alarms. <sup>2</sup>
[19]	Moving region detection, computation of a probabilistic model based on a time series of speed, shape and texture properties.	Ad-hoc database with multiple videos of same place in different times of day.	Distance metric of probability distribution from ideal smoke probability distribution of features, approach performs well on test on few videos. <sup>3</sup>
[20]	Moving region detection, feature extraction from texture analysis, SVM classification.	Ad-hoc database with multiple videos of same place in different times of day.	Approach performs well on test on few videos. <sup>4</sup>
[22]	Motion detection, image segmentation, smoke dynamic pattern analysis, color-space analysis and texture analysis.	Videos taken from field test during wildfire.	3.7% false alarms

Notes:

<sup>1</sup> Number of frames and subset of videos used for testing not available.

<sup>2</sup> Low number of frames tested.

<sup>3</sup> Classification metrics not available.

<sup>4</sup> Classification metrics not available.

2) *Smoke color analysis*: This step checks for high differences in the intensity channel of adjacent frames. Smoke regions possess a gray-ish color, related to low chrominance values. First, the frame  $I(x, y, t)$  is converted from the RGB to the YUV color space by using the following formulas:

$$\begin{aligned} Y &= 0.299R + 0.567G + 0.114B, \\ U &= -0.147R - 0.289G + 0.436B, \\ V &= 0.615R - 0.515G - 0.100B. \end{aligned} \quad (6)$$

Then, for each pixel at the location  $(x, y)$  and time  $t$ , a smoke color feature  $D_C(x, y, t)$  is computed according to the following equation:

$$D_C = \begin{cases} 1 - \frac{|U - 128| + |V - 128|}{128} & \text{if } Y - U > T_1 \text{ and } Y - V > T_1 \\ & \text{and } T_3 \leq Y \leq T_4 \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where  $T_1, T_2, T_3, T_4$  are predetermined thresholds. The resulting values are then thresholded and normalized in the range  $[0.8, 1]$ , which corresponds to the most significant regions.

3) *Sharp edge detection*: The sharp edge detection checks the intensity channel of adjacent frames. The frames  $I(x, y, t)$  and  $I(x, y, t - 1)$  are converted in the corresponding frames in the Y channel of the YUV color space  $Y(x, y, t)$  and  $Y(x, y, t - 1)$ . The difference frame is computed according to:

$$D_Y(x, y, t) = |Y(x, y, t) - Y(x, y, t - 1)|. \quad (8)$$

The values of  $D_Y$  inferior to a fixed threshold  $T_5$  are discarded in order to match only high values of  $D_Y$ , which are related to sharp edge transitions.

4) *Growing region detection*: The analysis of the growing areas of moving regions is performed by computing the frame difference between the moving region matrix at the time  $t$  and the time  $t - 1$ :

$$D_{M-diff}(x, y, t) = D_M(x, y, t) - D_M(x, y, t - 1). \quad (9)$$

The obtained values are then summed, considering only the positive contributes in order to avoid excessive data fluctuations:

$$D_G(t) = \sum_{x,y} (D_{M-diff}(x, y, t)) \quad \forall D_{M-diff}(x, y, t) \neq 0, \quad (10)$$

where  $D_G(t)$  is the growing value at time instant  $t$ . The values of  $D_G$  are then normalized between two threshold values.

5) *Rising region detection*: The first task in the detection of rising regions is to save the position of the highest moving pixel for a sequence of time instants:

$$H_{t-n} = \min_y D_M(x, y, t - n) \quad \text{if } D_M(x, y, i) > 0 \quad \forall 0 < n < N, \quad (11)$$

where  $H_{t-n}$  is the location of the highest moving pixel at time instant  $t - n$  and  $N$  is the maximum number of considered previous instants of time. The rising value  $D_R(t)$  at the time instant  $t$  is then computed as the difference between the first and the last of the locations:  $D_R(t) = H_{t-0} - H_{t-N}$ . The values of  $D_R(t)$  are then normalized between two empirically estimated thresholds.

6) *Perimeter disorder analysis*: This step analyzes the ratio of perimeter to area in order to compute a measure of the perimeter disorder. The moving region description matrix  $D_M(x, y, t)$  is first binarized by using the same threshold value used in Subsection III-A1. For each distinct 8-connected region of the binary image, the area and perimeter are computed. The perimeter disorder value for each region is computed as:

$$D_P(i, t) = \frac{P_i}{A_i} \quad \forall 1 < i < N_b, \quad (12)$$

where  $D_P(i, t)$  is the perimeter disorder value of the  $i$ -th blob (8-connected region) at time instant  $t$ ,  $P_i$  and  $A_i$  are the perimeter and area of the  $i$ -th blob, and  $N_b$  is the number of the blobs.

### B. Feature classification

The features are extracted pixelwise for  $N$  frames. Global features, such as growing value  $D_G$  and rising value  $D_R$ ,



Fig. 1. Example of smoke frame with pixelwise segmentation: (a) original frame; (b) segmented image.

TABLE II  
SUMMARY OF DATASETS

$N$	Dataset Name	Frame Sequence	Features $\times$ Samples	Non smoke samples / Smoke samples
5	DST1-5	DST1	$14 \times 20318$	19923/395
	DST2-5	DST2	$14 \times 24327$	23485/482
	DST3-5	DST3	$14 \times 28207$	27756/451
	DST4-5	DST1+DST2+DST3	$14 \times 72852$	71524/1328
7	DST1-7	DST1	$18 \times 18977$	18605/372
	DST2-7	DST2	$18 \times 24929$	24437/492
	DST3-7	DST3	$18 \times 18880$	18504/376
	DST4-7	DST1+DST2+DST3	$18 \times 62786$	61546/1240
10	DST1-10	DST1	$24 \times 9523$	9336/187
	DST2-10	DST2	$24 \times 24291$	24436/485
	DST3-10	DST3	$24 \times 18825$	18503/322
	DST4-10	DST1+DST2+DST3	$24 \times 53269$	52275/994

are replicated on every moving pixel. The growing and rising values are considered in a  $N$ -length timeseries. Every pixel corresponds to  $4 + (N \times 2)$  features:

- Moving value  $D_M$ ;
- Smoke color value  $D_C$ ;
- Sharp edge transition value  $D_Y$ ;
- $N \times$  Growing value  $D_G$ ;
- $N \times$  Rising value  $D_R$ ;
- Perimeter disorder value  $D_P$ .

In this work, we compare different classifiers, according to the number of frames used in the feature extraction process. In particular, we use  $N = 5, 7, 10$ .

#### IV. EXPERIMENTAL RESULTS

In this paper we refer to the True Positive as the percentage of smoke samples correctly identified as such, and True Negative as the percentage of non-smoke samples correctly identified as non smoke. We tested our approach by using three different sets of frame sequences, each with pixelwise segmentation (Fig. 1):

- 1) *DST1*: low-quality frame sequences, both smoke and non-smoke;
- 2) *DST2*: medium-quality frame sequences, both smoke and non-smoke;
- 3) *DST3*: synthetic frame sequences, both smoke and non-smoke.

The set of frame sequences DST3 was obtained by using the algorithms proposed in [28].

We used these sets of frame sequences in order to compute different feature datasets. The features are related to different numbers of consecutive frames ( $N = 5, 7, 10$ ). A summary of the datasets is shown in Table II.

TABLE III  
SUMMARY OF PROPOSED APPROACHES RESULTS

Reference Method	Dataset Name	Hidden L. Size	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
5 frames NN pixelwise approach	DST1-5	10	1.94	98.06	0.00	0.00	100.00	100.00	0.00
	DST2-5	10	1.71	97.84	0.18	0.27	86.31	99.82	0.45
	DST3-5	20	1.59	98.37	0.04	0.01	99.33	99.96	0.05
	DST4-5	15	1.73	98.08	0.10	0.10	94.73	99.90	0.19
7 frames NN pixelwise approach	DST1-7	10	1.96	98.04	0.00	0.00	100.00	100.00	0.00
	DST2-7	10	1.50	97.95	0.08	0.48	75.81	99.92	0.55
	DST3-7	15	1.98	98.00	0.01	0.02	99.20	99.99	0.02
	DST4-7	15	1.76	97.96	0.06	0.22	88.89	99.94	0.28
10 frames NN pixelwise approach	DST1-10	10	1.96	98.04	0.00	0.00	100.00	100.00	0.00
	DST2-10	15	1.54	98.01	0.05	0.41	78.97	99.95	0.46
	DST3-10	15	1.71	98.28	0.01	0.01	99.69	99.99	0.02
	DST4-10	25	1.63	98.09	0.05	0.23	87.63	99.95	0.28

Notes: Hidden L. Size = number of hidden layer nodes of the feed-forward neural networks; Sens. = Sensitivity; Spec. = Specificity.

For each frame, we evaluated the correctness of the segmented smoke areas estimated using the proposed method by comparing the obtained results with the ground-truth binary images. Fig. 2 shows some examples of segmented smoke regions obtained by using different values of  $N$ .

We used a  $N$ -fold cross-validation scheme with  $N = 10$  in order to compute all the figures of merits presented in the paper [29]. We used a two-layer Feed Forward Neural Network with different number of nodes in the hidden layer: 10, 15, 20, 25. The output layer is composed by a linear node and the topology of the hidden layer nodes is tan-sigmoidal. The method used to train the neural networks is the back-propagation algorithm.

A summary of the results obtained by the proposed approach is depicted in Table III. It is possible to observe that the method obtained few misclassification errors. In particular, the number of false positives is always  $< 0.2\%$ .

In order to test the performances of the proposed approach under adverse conditions, we introduced artificial effects in the DST1 frame sequences. Table V summarizes the result of the classification under simulated fog, increased and decreased luminance, and additive Poisson noise. These effects are obtained by using the method described in [28] (Fig. 3). It is possible to observe that adverse conditions have a negligible impact on the performances of the proposed approach, resulting only in a small increasing of the total classification error (less than 0.15%). Fig. 4 shows an example of results related to frame sequences with simulated fog.

We also compared the obtained results with the ones achieved by using kNN classifiers. In particular, we used different values of  $k$ : 1, 3, 5, and 10. The results obtained by using kNN classifiers are shown in IV. It is possible to observe that, in most of the cases, neural classifiers obtained a better accuracy. This fact is particularly evident by observing the results obtained by evaluating the feature datasets related to the set of frame sequences DST-2, which is noisier than DST-1 and DST-3. Another advantage related to the neural classifiers is that they require less computational time than the kNN classifiers. For example, the time needed by a feed-forward neural network with 10 neurons in the hidden layer is about  $1/25$  of the time required by a kNN classifier with  $k$

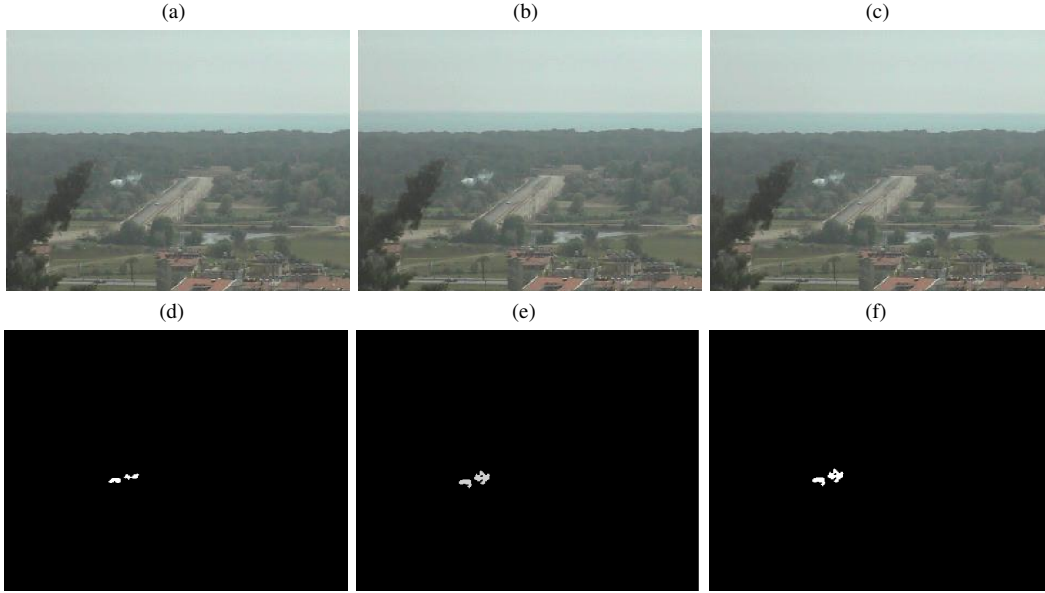


Fig. 2. Examples of input frames and outputs for different classifiers: (a),(d) 5 frames NN pixelwise approach; (b),(e) 7 frames NN pixelwise approach; (c),(f) 10 frames NN pixelwise approach.

TABLE IV  
RESULTS USING KNN CLASSIFIERS

Reference Method	Dataset Name	k	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
5 frames kNN pixelwise approach	DST1-5	1	1.94	98.04	0.01	0.00	99.75	99.98	0.02
	DST2-5	1	1.78	97.79	0.23	0.20	89.83	99.77	0.43
	DST3-5	10	1.58	98.29	0.11	0.01	99.11	99.89	0.12
	DST4-5	1	1.71	98.07	0.11	0.11	93.96	99.89	0.22
7 frames kNN pixelwise approach	DST1-7	3	1.96	98.03	0.01	0.00	100.00	99.99	0.01
	DST2-7	10	1.40	94.54	3.49	0.57	71.14	96.44	4.06
	DST3-7	5	1.98	97.85	0.15	0.01	99.47	99.84	0.16
	DST4-7	10	1.75	96.52	1.51	0.22	88.83	98.46	1.73
10 frames kNN pixelwise approach	DST1-10	3	1.96	97.99	0.04	0.00	100.00	99.96	0.04
	DST2-10	10	1.40	94.55	3.50	0.55	71.96	96.43	4.04
	DST3-10	5	1.70	98.13	0.16	0.01	99.38	99.84	0.17
	DST4-10	10	1.62	96.43	1.71	0.25	86.63	98.26	1.95

Notes: Sens. = Sensitivity; Spec. = Specificity.

TABLE V  
SUMMARY OF RESULTS UNDER ADVERSE CONDITIONS

Reference Method	Dataset Name	Hidden L. Size	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
5 frames NN pixelwise approach	DST1-5	10	1.94	98.06	0.00	0.00	100.00	100.00	0.00
	DST1-5 Fog	10	31.70	1595.90	0.40	0.30	99.06	99.97	0.04
	DST1-5 - Lum	10	46.20	2309.00	0.00	0.00	100.00	100.00	0.00
	DST1-5 + Lum	10	21.90	1095.80	0.00	0.00	100.00	100.00	0.00
	DST1-5 + Noise	15	20.80	1048.30	0.30	0.40	98.11	99.97	0.07
7 frames NN pixelwise approach	DST1-7	10	1.96	98.04	0.00	0.00	100.00	100.00	0.00
	DST1-7 Fog	20	31.50	1595.80	0.50	0.50	98.44	99.97	0.06
	DST1-7 - Lum	10	28.40	1421.00	0.00	0.00	100.00	100.00	0.00
	DST1-7 + Lum	25	26.60	1360.10	1.00	0.60	97.79	99.93	0.12
	DST1-7 + Noise	15	43.10	2155.00	0.20	0.50	98.85	99.99	0.03
10 frames NN pixelwise approach	DST1-10	10	1.96	98.04	0.00	0.00	100.00	100.00	0.00
	DST1-10 Fog	20	32.00	1596.20	0.10	0.00	100.00	99.99	0.01
	DST1-10 - Lum	25	28.10	1397.80	0.10	0.20	99.29	99.99	0.02
	DST1-10 + Lum	15	26.60	1359.60	1.50	0.40	98.52	99.89	0.14
	DST1-10 + Noise	20	42.30	2155.10	0.10	0.60	98.60	100.00	0.03

Notes: Sens. = Sensitivity; Spec. = Specificity; Fog = addition of simulated fog; -Lum: decreased luminance (Y) channel; +Lum = increased luminance (Y) channel; Noise = additive poisson noise.

= 10.

In further studies, other kernel-based classifiers, such as support vector machines and radial basis function networks, can be considered.

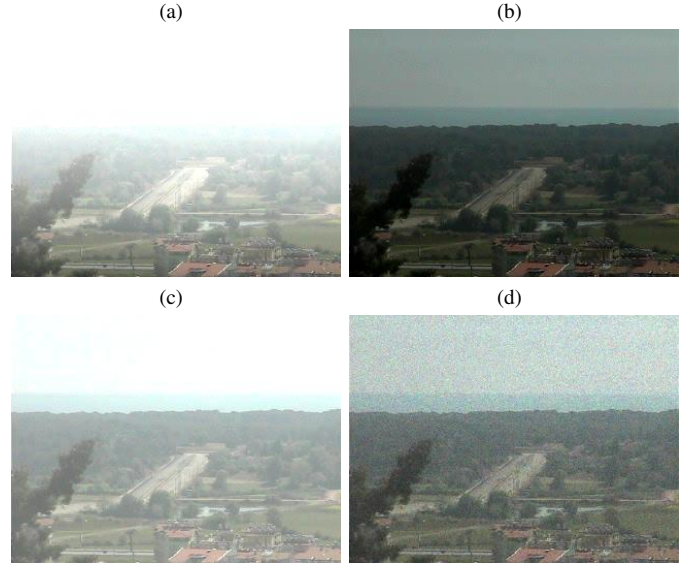


Fig. 3. Examples of simulated adverse conditions: (a) fog; (b) decreased luminance; (c) increased luminance; (d) additive noise.

## V. CONCLUSIONS

The paper proposed a new method to detect wildfire smoke. This method is designed in order to achieve a fast, low cost and reliable system to monitor fire-sensitive areas.

The proposed approach uses feature extraction algorithms based on image processing techniques, and classifiers based on computational intelligence strategies. The first step is a preliminary down-sampling of the frame sequences (both in space and time), followed by a moving region detection, which extracts the candidate smoke regions. Then, the candidate regions are processed by using a smoke color analysis algorithm, a sharp edge detector, and growing and rising region detection



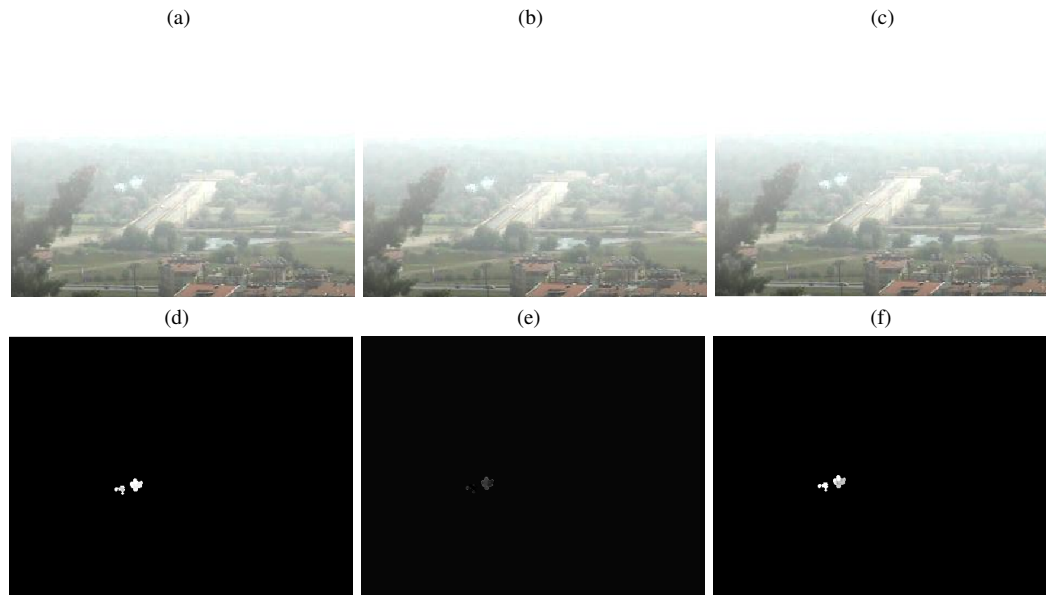


Fig. 4. Examples of input frames and outputs for different classifiers with presence of fog: (a),(d) 5 frames NN pixelwise approach; (b),(e) 7 frames NN pixelwise approach; (c),(f) 10 frames NN pixelwise approach.

techniques. Neural classifiers are then used in order to estimate the areas that describe smoke regions in the different frames.

We tested the performances of the proposed method in different scenarios, comparing the obtained results with the ones obtained by kNN classifiers. Experiments showed that the method is feasible and accurate. The approach performed well even in the presence of non-ideal conditions. Conditions of poor visibility can cause missed detections, but do not raise the number of false alarms.

#### REFERENCES

- [1] F. Luming, X. Aijun, and T. Lihua, "A study of the key technology of forest fire prevention based on a cooperation of video monitor and gis," *Fourth International Conference on Natural Computation*, October 2008.
- [2] Z. Liu and A. Kim, "Review of recent developments in fire detection technologies," *Journal of Fire Protection Engineering*, vol. 13, no. 2, pp. 129–149, May 2003.
- [3] B. Arrue, A. Ollero, and J. Matinez de Dios, "An intelligent system for false alarm reduction in infrared forest-fire detection," *IEEE Intelligent Systems and their Applications*, vol. 15, no. 3, pp. 64–73, May/June 2000.
- [4] T. Jakovcevic, L. Seric, D. Stipanicev, and D. Krstinic, "Wildfire smoke-detection algorithms evaluation," *VI International Conference on Forest Fire Research, D. X. Viegas*, 2010.
- [5] B.-H. Cho, J.-W. Bae, and S.-H. Jung, "Image processing-based fire detection system using statistic color model," *International Conference on Advanced Language Processing and Web Information Technology*, 2008.
- [6] F. Luming, X. Aijun, and T. Lihua, "A study of the key technology of forest fire prevention based on a cooperation of video monitor and GIS," *ICNC'08*, pp. 391–396, October 2008.
- [7] H. Maruta, Y. Kato, A. Nakamura, and F. Kurokawa, "Smoke detection in open areas using its texture features and time series properties," *ISIE 2009*, July 2009.
- [8] P. Piccinini, S. Calderara, and R. Cucchiara, "Reliable smoke detection in the domains of image energy and color," *ICIP 2008*, no. 1376-1379, October 2008.
- [9] Y. Cui, H. Dong, and E. Zhou, "An early fire detection method based on smoke texture analysis and discrimination," *CISP'08*, no. 95, May 2008.
- [10] J. Yang, F. Chen, and W. Zhang, "Visual-based smoke detection using support vector machine," *ICNC'08*, no. 301, October 2008.
- [11] D. K. Kim and Y.-F. Wang, "Smoke detection in video," *2009 World Congress on Computer Science and Information Engineering*, 2009.
- [12] G. Ivan, K. Darko, and S. Karolj, "Intelligent algorithm for smoke extraction in autonomous forest fire detection," *Microelectronics, Electronics and Electronic Technology. GRID and Visualization System. Opatija, Croatia*, p. 243, 2009.
- [13] J. Gubbi, S. Marusica, and M. Palaniswami, "Smoke detection in video using wavelets and support vector machines," *Fire Safety Journal*, vol. 44, no. 6, November 2009.
- [14] Z. Xu and J. Xu, "Automatic fire smoke detection based on image visual features," *CISW 2007*, p. 316, December 2007.
- [15] C.-Y. Lee, C.-T. Lin, and C.-T. Hong, "Spatio-temporal analysis in smoke detection," *ICSIPA 2009*, p. 90, November 2009.
- [16] C.-B. Liu and N. Ahuja, "Vision based fire detection," *ICPR 2004*, p. 134, August 2004.
- [17] B. U. Toreyin, "Fire detection algorithms using multimodal signal and image analysis," Ph.D. dissertation, Bilkent University, 2009.
- [18] B. U. Toreyin and A. E. Cetin, "Wildfire detection using LMS based active learning," *ICASSP'09*, 2009.
- [19] H. Maruta, Y. Kato, A. Nakamura, and F. Kurokawa, "Smoke detection in open areas using its texture features and time series properties," *ISIE 2009*, pp. 1904–1908, July 2009.
- [20] H. Maruta, A. Nakamura, and F. Kurokawa, "A new approach for smoke detection with texture analysis and support vector machine," *ISIE 2010*, pp. 1550–1555, July 2010.
- [21] S. Calderara, P. Piccinini, , and R. Cucchiara, "Smoke detection in video surveillance: A mog model in the wavelet domain," *Lecture Notes in Computer Science*, vol. 5008, pp. 119–128, 2008.
- [22] D. Stipanicev, M. Stula, D. Krstiniæ, L. Seriaæ, T. Jakovëeviaæ, and M. Bugariaæ, "Advanced automatic wildfire surveillance and monitoring network," *VI International Conference on Forest Fire Research D. X. Viegas (Ed.)*, 2010.
- [23] "DTec ForestVu," <http://www.videosmokedetection.com/Forestvu.htm>.
- [24] "AlarmEye AE6000," <http://www.innosys-ind.com>.
- [25] "FireWatch," <http://www.fire-watch.de>.
- [26] "iForestFire," <http://ipnas.fesb.hr/>.
- [27] R. Collins, A. Lipton, and T. Kanaden, "A system for video surveillance and monitor," *Proceedings of the 8-th International Topical Meeting on Robotics and Remote Systems. American Nuclear Society*, April 1999.
- [28] A. Genovese, R. Donida Labati, V. Piuri, and F. Scotti, "Virtual environment for synthetic smoke clouds generation," *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, September 2011.
- [29] R. Duda, P. Hart, and D. Stork, "Pattern classification (2nd edition), 2nd ed." *Wiley-Interscience*, November 2000.