




# Dual Deep Learning Model for Image Based Smoke Detection

Arun Singh Pundir\*  and Balasubramanian Raman, Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India

**Received:** 19 December 2017/**Accepted:** 10 May 2019

**Abstract.** Image-based smoke detection could help in faster and robust detection and monitoring of wildfires. It is becoming the best alternate of sensor based detectors for early detection of wildfire. The limitations of sensor based detector is that, they need close vicinity to fire for raising the alarm which make them vulnerable in case of detecting far-distant wild fire. Hence, vision based detection system which utilizes the surveillance cameras which shows more fastness and robustness as compared to sensor based detectors. These cameras when installed on hill top or mobile tower can raise the early alarm for any possibility of smoke present in the frames of videos whether near-by or far-away smoke. The proposed work presents a robust method for smoke detection which, utilizes a dual deep learning framework. The proposed architecture makes use of framework based on Deep Convolutional Neural Networks, which has proven their supremacy in object recognition tasks. The first deep learning framework is employed for extracting the image-based features from smoke patches, which are being extracted using superpixel algorithm. We have employed total of 20,000 frames with equally distribution of non smoke and smoke classes, out of which 6000 frames are utilized for testing purpose and 14,000 are used for fine tuning purpose. These features are comprised of smoke-color, smoke-texture, sharp edge detection and perimeter disorder analysis. The second deep learning framework is used for extracting motion-based features such as moving region of smoke, growing region and rising region detection. Optical flow method is employed, in order to capture the random motion of smoke. These extracted optical flow are then feed into Deep CNN for extracting motion based features. Features from both the framework are combined to train the Support Vector Machine and end to end classification which is CNN classifier. Accuracy on the nearby smoke and faraway smoke is 98.29% and 91.96% respectively. Testing on different varieties of non-smoke videos such as clouds, fog, sandstorm and images of cloud on water, method proves its precision and robustness. The average accuracy in all the scenarios is 97.49% which outperforms the state of the art method for these scenarios. Contribution of this work lies in the fact that we have given 20,000 frames based smoke and non-smoke dataset and secondly our method outperform the existing method on challenging imaging conditions.

**Keywords:** Smoke detection, Deep learning, Deep CNN, Superpixel segmentation, Optical flow

---

\* Correspondence should be addressed to: Arun Singh Pundir, E-mail: [asp24.dcs2014@iitr.ac.in](mailto:asp24.dcs2014@iitr.ac.in)



## 1. Introduction

The aggrandizement in capabilities of digital cameras acts as a boon in the proliferation of video surveillance applications such as real time object tracking, object detection or activity recognition. In the last few decades, early fire detection using computer vision techniques have gained a notable importance. As opposed to optical fire detectors, whose vicinity is generally restricted to a small room or limited indoor area and require close propinquity to the fire (as needed carbon particles to trigger the alarm). Computer vision techniques using non-optical sensors such as video cameras can detect smoke in an ample range of area including forest, hill tops etc. with a comparable low installation cost. Optical fire detectors such as sensor based fire alarms contain Light Emitting Diode (LED), a photocell and a base opening for carbon particles to be sensed by. In its functionality, a light beam is constantly shot out between the LED and photocell, when carbon particle enters the sensor, it interrupts the light beam, which is considered as a condition for alarm to get triggered. Hence, sensor based alarms take comparatively more response time than video based fire detection which make latter a next generation detection approach for automatic fire detection. Vision based fire detection can be considered as a promising solution as it can detect fire at a very elevated distance, more information such as fire's size and its direction can be measured without any latency.

Detecting smoke using vision based technique is considered more pertinent as smoke can be seen as an earlier indication of fire, early diminutive fire can be strenuous to detect but enormous amount of smoke produce by it, can be effortlessly detectable. Hence, point of convergence of method proposed in this paper is towards vision based smoke detection. However, authentic detection of smoke is an open challenge to many researchers since there are many elements in the surrounding which, resembles characteristics of smoke. Presence of fog, cloud, sand storms or reflection of these in water are some environmental scenarios that makes the vision based smoke detection a more arduous task.

The dataset used in this work having different variations of smoke from the very blackish smoke to light grayish to completely whitish smoke. The videos are chosen in such a way that they contain all the varieties from the low quantity indoor smoke to very high quantity forest smoke. Different varieties of background are chosen such as inside home, hilly areas, forest and hill-base smoke videos. The work is also tested on various challenging imaging conditions such as clouds, fog, sandstorm and images of cloud on water, in which the texture and color of smoke is very similar to given imaging conditions. Our work is based on proposing a smoke detector which works on image based detection hence we have considered all the nature based situations which are very similar to smoke.

Antecedently, researchers mainly focused on eminent features of smoke such as color [9, 10, 18, 38], texture [12–14, 45], motion [6, 7, 41, 44] and geometrical shapes [41, 44]. Despite the contrary, the pursuit for feature extraction by researchers were sketchy due to feature engineering. To fulfill this gap, we have adopted method which is competent enough for representation learning and have

the exceptional capability of spatial invariance. To eliminate the limitations of feature engineering, deep learning models can be adopted which are superlative in typify two-dimensional signals. Deep CNN [28] can robustly extract features to eradicate the reliance on human subjectivity or experience. Hence, for computing eminent features of smoke such as smoke-color analysis, smoke-texture analysis, sharp edge detections and perimeter disorder analysis, we have employed CNN. Other significant features of smoke that uniquely analyze smoke are based on its state of motion. Some of the features are moving region of smoke, growing region and rising region detection. Another deep model based on optical flow [19] that has been used for extracting the motion based features of smoke. One of the requirements for smoke detection using deep learning framework is the unavailability of large dataset. In our method, we have manually created a diverse dataset of around 11,000 images from different types of smoke.

Towards this end, we proposed a deep learning framework for extracting all the prominent features of smoke and also present a novel method for video based smoke detection. Our main contribution, in this work, is as follows:

- All eminent features of smoke such as smoke-color, smoke-texture, sharp edge detection, perimeter disorder analysis, smoke moving and growing regions are extracted in the given work.
- About 11,000 image-based smoke containing dataset is created in the given method which can be useful to other researchers for future work.
- The Dual Deep CNN is employed in our method for extracting all smoke features.
- Proposed method robustness is tested against very challenging non-smoke dataset such as cloud, fog, sandstorm and reflection of cloud in water.
- To the best of our knowledge, this is the first ever paper on dual deep learning framework for smoke detection.

The structure of this paper is organized as follows. Section 2 presents the related work. The overview of the methodology along with the proposed method is described in Sect. 3. Section 4 contains the testing and analysis for the given method, while summarization of the work done is drawn in Sect. 5.

## 2. Related Work

Majority of the researcher exploits only prominent features of smoke such as color, texture, motion, flicker analysis [41, 44] in video based detection. Segmentation techniques [35] were practiced in smoke detection in smoke-containing images. Some are targeted the spatio-temporal feature of smoke while other targeted the dynamic characteristics of smoke. However, most of the work, focused on color, motion and texture based features of smoke.

### **2.1. Color Based Smoke Detection**

Chen et al. [10] extracted fire and smoke pixels using red, green, blue (RGB) color space. They employed decision function based on intensity and saturation value of red color component. Further, disorder and growth dynamic of smoke verified the occurrence of fire and smoke. Temporal and spatial color variations were analyzed by [38], they proposed the new color space using pigmentation values in RGB color space, and intensity and saturation values in Hue, Saturation and Value (HSV) color space. Spatial and temporal features of flames were examined by [9] to observe movement, color and flickering characteristics of flames. Flame color filtering algorithm was incorporated to categorize the non-candidate and candidate flame regions. Spatial wavelet transform was computed by [18], of the moving fire colored region to examine the color variations in the fire. Hidden Markov Model (HMM) was then incorporated for determining the characteristics of fire boundaries by computing the temporal wavelet analysis.

Philips et al. [34] computed Gaussian-smoothed color histogram to identify the presence of flame-colored pixels and then incorporated temporal variations to ascertain whether computed pixels are fire pixels or not. Three color spaces that is RGB, HSV and Luminance; Chroma:Blue; Chroma:Red (YCbCr) were tried by [8]. They performed the statistical analysis of various video frames of smoke and fire for extracting different color spaces which uniquely identify smoke and fire. Calderara et. al [5] computed the temporal behavior of smoke in the wavelet domain by computing the Mixture of Gaussians (MoG) of variation in the energy to distinguish between smoke and other moving objects. They proposed a blending function to analyze the smoke-color along with textural analysis of smoke.

### **2.2. Motion Based Smoke Detection**

Hybrid background estimation algorithm was employed by [41] for computing the moving regions and pixels in the video sequences. Temporal wavelet transform was computed for tracking the quasi periodic nature of fire boundaries. Background subtraction method with adaptive background update was employed by [44] to determine the moving target region, subsequently disorder, growth, self-similarity and frequent flicker in boundaries of moving target region were examined to exploit the dynamic and static features of smoke. A real time fire detection system was given by [7] that used Gaussian distribution method to compute the adaptive background model, the foreground information was later proved by the statistical color model to verify that moving region belongs to fire regions or not. Continual dynamic envelops of substantial pixels were extracted by [42] using segmentation technique. To distinguish the envelops created by smoke from other sources envelops, they computed complex and transitory motion for creating minor pre-processed envelops.

Piccinini et al. [35] employed a background suppression approach precisely Statistical And Knowledge-Based Object Tracker (SAKBOT) to carried out the segmentation of moving objects. Other methods specifically the ghost suppression, object validation and background bootstrapping were employed to improve the accuracy of segmentation. An accumulative motion model was given by [47]. They

used integral images to quickly determine the orientation in the motion of smoke. The accumulation was repeatedly estimated to improve the accuracy of the orientation. Basic background initialization and regularly updated model was employed by [25] to extract the candidate fire regions, probabilistic fire-models were then applied to hierarchical Bayesian Network to verify the presence of fire regions. In [27], Kopilovic et. al considered the irregular motion property of smoke, hence computed the optical flow of two adjacent smoke regions and entropy of motion directions to differentiate among smoke and non-smoke. Similarly, optimal mass transport optical flow was computed by [26] and single hidden layer neural network was employed to classify smoke and similar objects like smoke.

### 2.3. Texture Based Smoke Detection

Three level smoke based image pyramid was proposed by [48]. Local Binary Patterns (LBPs) were computed at all the three levels of pyramid to create an LBP pyramid. Similarly, Variance based Local Binary Patterns (LBPV) were computed at every level to build an LBPV pyramid. Histograms of these pyramids were used as a feature vector, a neural network was then used to classify smoke and non-smoke regions. Texture analysis tools specifically Gray Level Co-occurrence Matrices (GLCM) and wavelet analysis were employed by [13] to analyze the texture of image containing fire smoke. Artificial Neural Network (ANN) was used as a classifier to distinguish between smoke and non-smoke texture. A dynamic texture descriptor with Hidden Markov Tree (HMT) and surfacelet transform was proposed by [45]. Surfacelet transform with scale continuity model and Gaussian mixture model were computed to give a 3D HMT model. SVM was learn to classify smoke and non-smoke videos.

Dynamic texture analysis of flame regions was given by [14] in which they employed a bag-of-system technique and linear dynamical system. Along with dynamic texture analysis, they also extracted various spatio-temporal features such as flickering, spatio-temporal energy and color probability. A 2-class SVM was employed to classify the flame and non-flame regions. In [15], they proposed a higher order Linear Dynamical System (*h*-LDS) descriptor and exploit dynamic texture analysis in multidimension for smoke detection. Histogram of *h*-LDS descriptor were computed as feature and particle swarm optimization method was used to discriminate between smoke and non-smoke regions. Apart from above mentioned techniques, miscellaneous dynamic texture based approaches [2, 11, 16, 17, 40] had been employed for smoke detection.

Substantial techniques were practiced within a single proposed method and later aggregated to classify smoke and non-smoke regions, but none of the researcher employed a method which can extract multi-features itself. In our work, deep learning framework specifically, CNN is employed which alone can extract all the prominent features of smoke and SVM is then used as a classifier in the last connected layer to classify the smoke and non-smoke regions.

## **2.4. Flicker and Motion Analysis Using Various Transforms**

Many early fire and smoke detection algorithms considered fire-color and smoke-color moving objects as fire and smoke respectively. This may lead to the raising of false alarms due to the movement created by falling of fire-color leaves, or moving person wearing smoke colored clothes. Hence, further analysis was done to remove the false alarms and achieve more robust systems.

Flickering of frames is present in wild-fires, hence detection based on flickering in smoke and fire video [25, 36] and energy analysis of signals in wavelet domain [5, 33] was employed to distinguish non-fire objects from smoke and fire. These algorithms analyzed the behavior of smoke and fire based on temporal information. As can be observed in many potential fire videos, fire pixels disappear and again appear at the perimeter of wild-fires. The work presented in [43] proves that the flicker frequency of wild-fire is around 10 Hz. Hence, the researchers utilize the analysis of frequency to classify fire from non-fire moving things.

## **2.5. Deep Learning for Smoke Detection**

Zhang et al. [49] employed faster R-CNN to detect wild land smoke. To increase the number of images, various synthetic images with real time background were used. Edwin et al. [3] used the fusion method for smoke detection. Inertial Measurement Unit (IMU) sensor and Recurrent Neural Network (RNN) was employed to classify the smoking activity and non-smoking activity. Kaabi et al. [24] employed the Deep Belief Network (DBN) to classifying the smoke and non-smoke patterns. Hu et al. [20] employed a spatial-temporal based CNN for video smoke detection. Yin et al. [46] used deep convolutional motion-space networks for smoke detection.

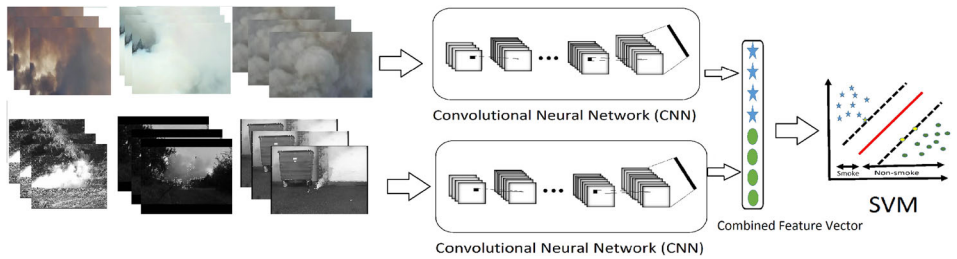
The limitations of the above mentioned systems is that, they have not tested against the various challenging imaging conditions like clouds, fog, sandstorm and images of cloud on water, which are very much similar in color and texture as compared to smoke. Proposed system have outperformed the system which was tested before on such imaging conditions.

# **3. Methodology**

The proposed method aims to extract the eminent features of smoke using two deep learning framework and tries to classify smoke and non-smoke frames using different classifiers. Our training procedure is summarized in Algorithm 1 and Fig. 1.

## **3.1. Preprocess Videos for Creation of Dataset**

In this step, we extracted the frames from smoke based videos. To train a CNN, it requires a lot of computation and a large amount of dataset. Hence, superpixel segmentation algorithm is applied on extracted frames to extract the smoke based regions. Manually, we collected those smoke-based regions and write them into a separate file. About eleven thousand smoke containing images were employed to



**Figure 1. Block diagram of the proposed method with two deep learning framework.**

train the CNN, which can compute color-based, texture-based, sharp-edges and perimeter disorder based features of smoke. This complete procedure generates a image-based deep learning framework. For initiating motion-based deep learning framework, we extract velocities from three consecutive frames in smoke-based videos using Horn-Schunck optical flow method. These velocities are fed into CNN for extracting motion-based features of smoke.

**3.2. Extracting Features From CNN**

Our next approach is to utilize the CNN, which is trained on our created dataset, for the extraction of features. The challenge with CNN is that, it require a lot of computation and training dataset to train it. Hence, we have employed a technique named transfer learning, which addresses above mentioned challenges. In this approach, it employs a trained model instead of training the model from very beginning. Hence, in our work, we have employed the weights from Covnet Alex-Net [28], which was already trained on Imagenet database.

We finetune the framework on our dataset, after initializing the weights from AlexNet. This is done by passing the frame based dataset and optical flow output as input to the training process. Once the training is over, fc-7 layer was used to give the feature vector. The extracted feature vector is used as an input to learn the classifiers.

**3.3. Learning the Classifier**

The final stage of our work involves training an SVM and CNN classifiers by utilizing the features extracted from the previous step. For this, training and testing dataset is created, in which training data is used to train the classifiers and test data is used to compute the accuracy for our model. Training and testing is explained more elaborately in the Sect. 4.



---

**Algorithm1**    Proposed training algorithm
 

---

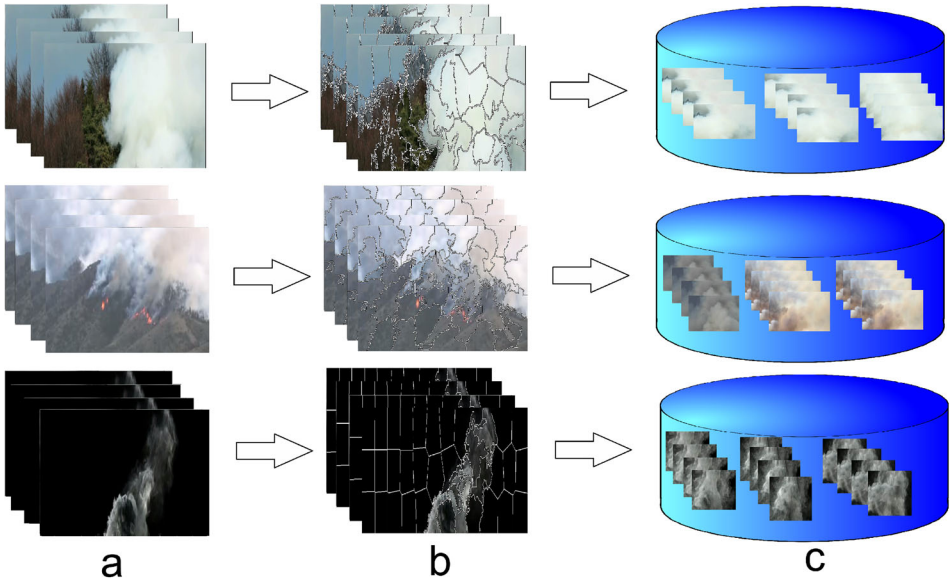
- 1: **procedure** TRAINING DUAL DEEP LEARNING FRAMEWORK
  - 2:     **Step1:** Image-based Deep learning framework
  - 3:     **Step1.1:** Superpixel Algorithm for generating Image-based dataset (*Section3.4*)
  - 4:     **Step1.2:** Using Step 1.1 output to train CNN (*Section3.6*) for extracing image based features
  - 5:     **Step2:** Motion-based Deep learning framework
  - 6:     **Step2.1:** Extract velocities from each video using optical flow method (*Section3.5*)
  - 7:     **Step2.2:** Using Step 2.1 output to train CNN (*Section3.6*) for extracting motion based features
  - 8:     **Step3:** Combined features from Step1 and Step2
  - 9:     **Step4:** Used features from Step 3 to train SVM and CNN classifiers
- 

### 3.4. Superpixel Segmentation

One of the best method to assemble intuitively relevant atomic regions is Superpixel algorithm [1], which can be considered as an alternate to substitute the pixel grid rigid structure. Superpixel proves as an acceptable essential algorithm for computing features in an image, abduct redundancy in an image and proved useful in diminish the complexity of the consequent image processing tasks. Various computer vision based problems such as object localization, depth estimation multi-class object segmentation and basic segmentation were solved by using super pixel algorithm.

In our work, we have employed Simple Linear Iterative Clustering (SLIC) [1] super pixel algorithm to segment the smoke part available in the video frames. SLIC is better in terms of memory efficiency, faster computation than existing super pixel methods and proves its efficiency in upto-date boundary adherence. Using CIELAB color space  $[l \ a \ b]^T$ , clustering was done by initializing  $m$  initial center of clusters  $C_m = [l_m \ a_m \ b_m \ x_m \ t_m]^T$  which are sampled at regular grid  $M$  pixels apart. To avoid centering on an edge,  $3 \times 3$  neighborhood with minimum gradient was used to move the cluster center. For better computational speed, distance measure was computed which extracts the closest cluster center for every pixel value. For each pixel value, nearest cluster center was computed to enumerate the mean  $[l \ a \ b \ x \ y]^T$  vector of every pixel present in the cluster. Figure 2 shows the superpixel segmentation method in three different scenarios for creation of image based dataset.





**Figure 2. Superpixel segmentation: (a) shows smoke frames, (b) shows the superpixel segmentation on corresponding frames and (c) shows the stored dataset created using superpixel segmentation.**

### 3.5. Optical Flow

For computing motion based features of smoke such as moving region, growing and rising region of smoke, we have employed Horn-Schunck optical flow [19]. Horn-Schunck algorithm works on the constraint that flow over the complete frame is smooth. Let  $B(a, b, t)$  denotes the image brightness at any point  $P(a, b)$  in the extracted frame at time  $t$ . Consider the smoothness constraint, brightness can be considered as:

$$B(a + da, b + db, t + dt) = B(a, b, t). \quad (1)$$

Equation for optical flow constraint can be derived by utilizing Taylor's expansion of equation 1 as:

$$\frac{\partial B}{\partial a}u + \frac{\partial B}{\partial b}v + \frac{\partial B}{\partial t} = 0, \quad (2)$$

where  $B_x = \frac{\partial B}{\partial a}$ ,  $B_y = \frac{\partial B}{\partial b}$ ,  $B_t = \frac{\partial B}{\partial t}$  represents the partial derivatives of brightness respectively, with respect to  $a$ ,  $b$  and  $t$ . Another smoothness constraint is used for computing the unknown variable  $[u, v]$ . Normally, an object is having similar velocities for its neighboring points and brightness pattern, contain velocity field in a frame smoothly varies all over the frame. Hence, smoothness in term of

velocities, is computed as a sum of square of Laplacian of  $a$ - and  $b$ - components of the flow as:

$$\nabla^2 u + \nabla^2 v = \frac{\partial^2 u}{\partial a^2} + \frac{\partial^2 u}{\partial b^2} + \frac{\partial^2 v}{\partial a^2} + \frac{\partial^2 v}{\partial b^2}. \quad (3)$$

The basic concept behind the computation of optical flow is to acquire better smoothness by reducing distortions in the flow. The flow is represented in the form of global energy function, the aim is to minimize this energy function. Mathematically, global energy function is represented as:

$$E_{u,v} = \int \int [(B_a u + B_b v + B_t)^2 + \beta^2 (\nabla^2 u + \nabla^2 v)] da db, \quad (4)$$

where  $\beta^2$  is used to scale the global smoothness. Hence, by calculus of variation, we have:

$$B_a^2 u + B_a B_b v + B_a B_t - \beta^2 \nabla^2 u = 0, \quad (5)$$

$$B_b^2 v + B_a B_b u + B_b B_t - \beta^2 \nabla^2 v = 0. \quad (6)$$

Then, Laplacian is numerically approximated as  $\nabla u = \bar{u} - u$ , where  $\bar{u}$  represents the average value of  $u$ , computed around the neighborhood of pixel  $P(a, b)$ . The velocities  $u$  and  $v$ , at every pixel in the frame are given by:

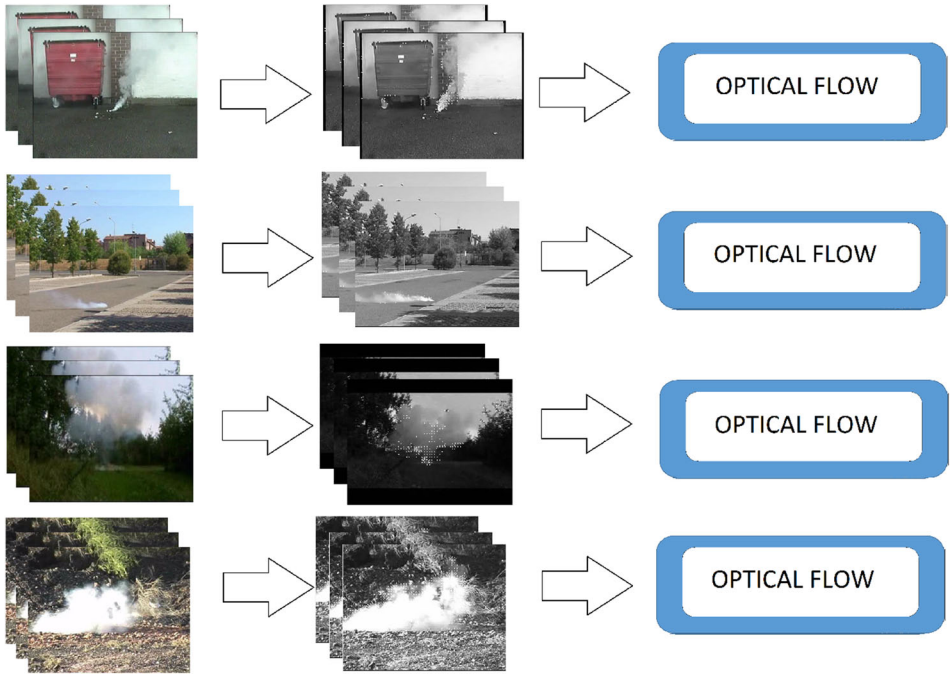
$$u^{i+1} = \frac{u^{-i} - B_a [B_a u^{-i} + B_b v^{-i} + B_t]}{\beta^2 + B_a^2 + B_b^2}, \text{ and} \quad (7)$$

$$v^{i+1} = \frac{v^{-i} - B_b [B_a u^{-i} + B_b v^{-i} + B_t]}{\beta^2 + B_a^2 + B_b^2}. \quad (8)$$

These velocities are computed iteratively, where  $i + 1$  denotes the next iteration and  $i$  is the last computed result. In our method, we have calculated velocities at  $i - 1$ ,  $i$  and  $i + 1$  iterations, and feed these velocities as an input to the CNN, to form another motion based deep learning framework. Figure 3 shows the optical flow method in four different scenarios for creation of motion based dataset.

### 3.6. Convolutional Neural Network

Deep CNN has shown its dominance triumph in many classification problems for extraction of features [4, 29, 30, 32]. CNN structure is made up of numerous layers, starting with convolutional layer which computes the convolutional transforms, pursued by computation of non-linearity and pooling operations. Mostly researchers exploits the CNN to perform the classification tasks only [4, 29–31]



**Figure 3. Optical flow: first column in figure shows smoke-based frames, second column shows the optical flow on corresponding smoke frames and third column shows motion based dataset in the form of velocities.**

but depends upon output of the last network layers, CNN can be employed as the stand-alone feature extractor [21–23, 39], with the features being extracted can be used to train another classifier.

At Convolutional layer, the backpropagation updates can be derived by getting learnable kernels convolved with previous layer feature maps. The output feature map is computed by placing the previous convolved output into activation function. Various input maps now convolve with different output maps. If map  $l$  and output map  $k$  both integrated over input map  $j$ , the applied kernels are different for map  $j$  with map  $k$  and  $l$ . Mathematically it is represented as:

$$y_k^m = g \left( \sum_{j \in N_k} y_j^{m-1} \times l_{jk}^m + b_k^m \right), \quad (9)$$

where  $b$  represents the additive bias and  $N_k$  gives a collection of input maps.

For computing the gradients in CNN, each convolutional layer have consequent  $m + 1$  downsampling layers, let  $v$  be the current layer pre-activation input, which is being computed using derivative of activation function. For evaluating the sensitivities  $\delta$  at current layer  $m$ , the downsampling layer's sensitivity map is upsam-

pled so that both current layer map and downsampling layer map must have same size. Then sensitivity map to be sampled from layer  $m + 1$  are getting element-wise multiplied with current layer  $m$  derivative map. Let us consider a constant  $\gamma$ , so that weight considered at downsampling layer must be equal to that constant  $\gamma$ . The constant  $\gamma$  was chosen to scale up the previous layer output by  $\gamma$ , to complete the evaluation of  $\delta^m$ . Hence,  $\delta^m$  is computed as:

$$\delta_k^m = \gamma_k^{m+1} (g'(v_k^m) \times \text{upsamp}(\delta_k^{m+1})), \quad (10)$$

where  $\text{upsamp}(\cdot)$  defines the upsampling operation that binds each pixel in the output both vertically and horizontally  $n$  times with input if the subsampling is done by factor  $n$ . Upsampling can be defined more efficiently using Kronecker product as:

$$\text{upsamp}(y) = y \otimes \mathbf{1}_{n \times n}. \quad (11)$$

Bias gradient can be computed now using sensitivities for the current map as:

$$\frac{\partial E}{\partial b_k} = \sum_{v,w} (\delta_k^m)_{v,w}. \quad (12)$$

Backpropagation can finally be used for computing the gradients for kernel weights. Hence, summation of the gradients was done to share the same weights across various connections as:

$$\frac{\partial E}{\partial l_{jk}^m} = \sum_{v,w} (\delta_k^m)_{v,w} (p a_j^{m-1})_{vw}, \quad (13)$$

where patch in the  $y_j^{m-1}$  is represented by  $(p a_j^{m-1})_{vw}$ . For computing the pixel value at  $(v, w)$  in the subsequent convolutional map  $y_k^m$ , the patch  $(p a_j^{m-1})_{vw}$  was element-wise multiplied to  $l_{jk}^m$  at the time of convolution.

A subsampling layer used input maps to generate the downsampled version. For  $M$  input maps there must be exactly  $M$  output maps. Mathematically,

$$y_k^m = g(\gamma_k^m \text{downsamp}(y_k^{m-1}) + b_k^m), \quad (14)$$

where  $\text{downsamp}(\cdot)$  indicates the down sub-sampling function. Every output map is provided with its own additive bias  $b$  and multiplicative bias  $\gamma$ .

To learn the combinations of feature maps, an output map that is computed using summation of various convolutions of several input maps. Such input maps can be made to learn combinations to generate output maps, during training. Let  $\omega_{jk}$  represent the weight assigned to input map  $j$  when generating output map  $k$ . Output map here is given by:

$$y_k^m = g \left( \sum_{j=1}^{N_m} \omega_{jk} (y_j^{m-1} \times L_j^m) + b_k^m \right), \quad (15)$$

with constraints  $\sum_j (\omega_{jk} = 1)$ , and  $0 \leq \omega_{jk} \leq 1$ .

These constraints can be implemented by initializing the  $\omega_{jk}$  equal to softmax function having underlying, unconstrained weight  $\alpha_{jk}$  as:

$$\omega_{jk} = \frac{e^{\alpha_{jk}}}{\sum_l e^{\alpha_{lk}}}. \quad (16)$$

Since, underlying weight  $\alpha_{jk}$  for final  $k$  are not dependent of all other similar sets, update for single map can be considered by dropping the  $k$ . Hence, derivative of softmax function is computed as:

$$\frac{\partial \omega_l}{\partial \alpha_j} = \delta_{lj} \omega_j - \omega_j \omega_l. \quad (17)$$

While derivative of square-error loss function with respect to  $\omega_j$  on layer  $m$  is given by:

$$\frac{\partial E}{\partial \omega_j} = \frac{\partial E}{\partial v^m} \frac{\partial v^m}{\partial \omega_j} = \sum_{v,w} (\delta^m (y_j^{m-1} \times L_j^m))_{vw}. \quad (18)$$

here,  $\delta^m$  represents the sensitivity map of an output map with input  $v$ . Chain rule was implemented for computing the gradients of the square error loss function with respect to  $\alpha_j$  as:

$$\frac{\partial E}{\partial \alpha_j} = \sum_l \frac{\partial E}{\partial \omega_l} \frac{\partial \omega_l}{\partial \alpha_j} = \omega_j \left( \frac{\partial E}{\partial \omega_j} - \sum_l \frac{\partial E}{\partial \omega_l} \omega_l \right). \quad (19)$$

The CNN structure, which is employed in our method, is same as that of ALEXNET. The only enhancement in ALEXNET for extracting features is the deletion of final layer that is ‘fc-8’ layer and extracting the output from the ‘fc-7’ layer.

## 4. Testing and Analysis

It requires a lot of computation and huge amount of dataset to train a CNN. Above mentioned techniques that is superpixel segmentation (Sect. 3.4) and optical flow (Sect. 3.5) have been applied on the publicly available dataset which is in the form of smoke containing videos for the creation of frame-based and motion-based dataset. This publicly available dataset has been gathered from various

**Table 1**  
**Different Smoke-Based Videos**

Sr. No.	Description
1	Blackish smoke spreading slowly within a dark room
2	Dispersal of very light smoke in presence of sunlight
3	Smoke in presence of vehicles on open road
4	Spreading of smoke with similar color wall in background
5	Dispersal of smoke near railing with men in motion
6	Wild fire smoke on hill top
7	Explosion based-smoke nearby trees
8	Spreading of smoke within a forest
9	Huge wild fire smoke
10	Dispersal of smoke at hill side

internet sources<sup>1,2,3,4</sup> and for further research in future, smoke videos are available on google's webpage.<sup>5</sup>

For extracting the image-based features of smoke, we have utilized our stored dataset (Sect. 3.1) of 10,000 smoke-based images and about 10,000 non-smoke images were extricated from non smoke videos. We have employed total of 20,000 frames with equally distribution of non smoke and smoke classes. Hence, for fine tuning our system, we have utilized 70% of the frames whereas remaining frames being used for testing purpose. More specifically, total 20,000 frames are employed in our work, out of which 6000 frames are utilized for testing purpose and 14,000 are used for fine tuning purpose.

Similarly, for extracting the motion-based features of smoke, optical flow method has been applied on 10 smoke-based videos and 7 non-smoke based videos. We have applied the method on 10,368 smoke-containing frames and 10,300 non-smoke frames, extracted from smoke-based and non-smoke based videos respectively. The frames are extracted such that they are continuous frames so that the temporal information with in the frame is preserved and later extricated as features. These retrieved features are utilized as motion-based dataset to be feed into CNN. After that, we follow the above mentioned selection criteria for training and testing purpose.

Smoke-containing videos that have been employed in our method have different types of smoke varying from soupy wild fire smoke to a very thin indoor smoke. Non-smoke videos are chosen in a way that they have similar background as in smoke containing videos. The size of each frame and frame rate of the videos are set to  $320 \times 240$  pixels and 30 Hz respectively. The smoke containing videos are described in Table 1 while Figs. 4, 5 and 6 represent all the videos that are availed in the given method.

<sup>1</sup> <http://cvpr.kmu.ac.kr/>.

<sup>2</sup> <http://www.openvisor.org>.

<sup>3</sup> <http://signal.ee.bilkent.edu.tr/VisiFire/Demo>.

<sup>4</sup> <https://www.shutterstock.com/video/search/smoke>.

<sup>5</sup> <https://sites.google.com/site/smokedataset/smokedataset>.





**Figure 4. 10 Smoke containing videos that are employed in our method out of which above row shows 5 non-wild fire smoke videos and below row shows 5 wildfire videos.**



**Figure 5. 7 Non-smoke videos that are employed in our method.**

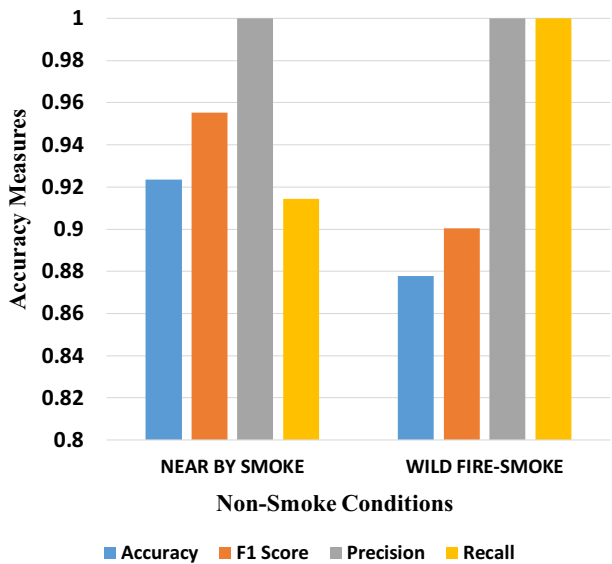


**Figure 6. 7 Non-smoke videos that are utilized as a challenging dataset in our work which comprises of clouds, fog, sandstorm and shadow of clouds on water.**



**Table 2**  
**Analysis of Nearby Distance Smoke Detection**

Classifier	Accuracy (in %)	F1 score	Precision	Recall
CNN	92.3404	0.9552	1	0.9143
SVM	98.2979	0.9904	1	0.9810



**Figure 7. Accuracy measure of CNN Classifier on near by smoke conditions.**

For showing the robustness of our proposed method, we have considered the following three scenarios for experimental results:

**4.1. Performance on Nearby Smoke**

In this scenario, we have employed 7 non-smoke videos and 5 are near distance smoke videos. For training and testing, we have followed the above mentioned criteria like extraction of 20,000 frames from both types of videos with equal distribution of both classes. For performing fine tuning and testing, we have avail 70% and 30% of 20,000 frames. The results for this section are shown in Table 2 and Fig. 7. Since, there is no standard available dataset for smoke detection, we have computed and compared our results with SVM and CNN classifiers, after computing features from CNN.

**Table 3**  
**Evaluation of Wild-Fire Smoke Detection**

Classifier	Accuracy (in %)	F1 score	Precision	Recall
CNN	87.7814	0.9005	1	0.8190
SVM	91.9614	0.9367	1	0.8810

**4.2. Performance on Wild Fire Smoke**

In this scenario, we have followed the same procedure as in case of near distance smoke. Table 3 and Fig. 8 have shown the accuracy for classifiers on wildfire smoke.

**4.3. Analysis of Proposed Work on Very Challenging Non-smoke Dataset**

To show the robustness of given method, the proposed work is analyzed on very challenging non-smoke dataset. The challenging non-smoke videos have environmental conditions which are similar to smoke containing videos. These non-smoke videos are comprising of cloud, fog, sandstorm and shadow of clouds on water or river. In this scenario, we have chosen 5 videos for each challenging non-smoke condition and follow the same process for testing and training as done in previous sections. Results for this section are shown in Tables 4, 5, 6, 7, Figs. 9 and 10.

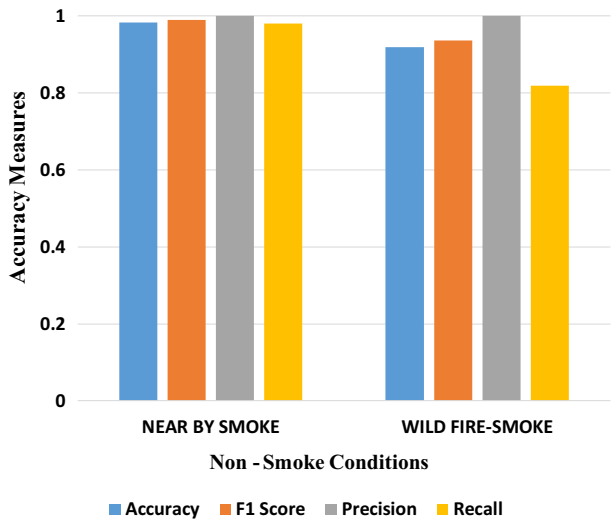
The above results show that our proposed method works better in case of near by smoke as compared to wildfire smoke. In case of challenging non-smoke dataset, for fog condition, CNN works better while in case of sandstorm and clouds, SVM outperform the CNN. In case of ‘shadows of clouds on water’ both classifiers shown at par performance.

**4.4. Comparison of Proposed Method with State of the Art Paper**

Since, there is not any standard dataset available for smoke detection. Hence, to prove the superiority of given method over other method, we have implemented state-of-the-art method on smoke detection, given by [37], on our challenging non-smoke dataset. This state-of-the-art method is based on classification by Deep Belief Networks (DBNs). Results for this section are shown in Tables 8 and 9, which shows the values for different average accuracy matrices.

**4.5. Verification of Proposed Method**

To verify the robustness of proposed system, we have created our own dataset using different materials such as thermocol, wood, dry leaves, dry grass, green grass, paper (cardboard), dry clothes, plastic, rubber (tyres) to generate the different types of smoke. Three different locations were selected for recording wild, outdoor and indoor simulations of smoke. Other probable conditions were also taken care off, such as wind/breeze, static and dynamic background. Total 447 smoke containing videos were created (as shown in Fig. 11), out of which 180 videos are



**Figure 8. Accuracy measure of SVM Classifier on wild-fire smoke conditions.**

**Table 4**  
**Analysis of Proposed Method in Fog Condition**

Classifier	Accuracy (in %)	F1 score	Precision	Recall
CNN	99.0610	0.9910	0.9821	1
SVM	96.7136	0.9677	0.9375	1

**Table 5**  
**Evaluation of Proposed Method in Cloudy Condition**

Classifier	Accuracy (in %)	F1 score	Precision	Recall
CNN	96.5686	0.9671	1	0.9364
SVM	98.5294	0.9852	0.9709	1

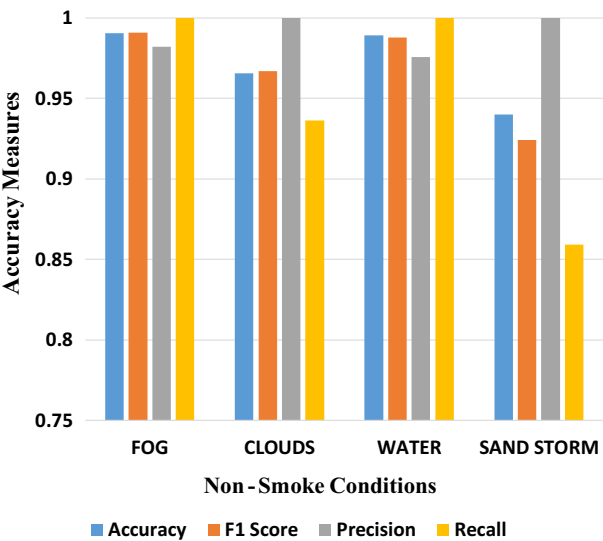
recorded at wild locations (like river bed and farmhouse), 157 are out door videos (outside home locality) and 110 are indoor videos (inside home). 410 video clips are those which contains non-smoke things (clips recorded while making dataset having similar background but are non-smoke videos). Out of which around 300,000 frames were extracted which contains the smoke and similar number of non-smoke frames is also extracted. We performed our experiments on these videos. And follow the same procedure like 70% of the feature vector is used for training purpose and 30% for testing purpose. For calculating accuracy we define

**Table 6**  
**Evaluation of Proposed Method in ‘Shadow of Clouds in Water’ Condition**

Classifier	Accuracy (in %)	F1 score	Precision	Recall
CNN	98.9130	0.9878	0.9759	1
SVM	98.9130	0.9878	0.9759	1

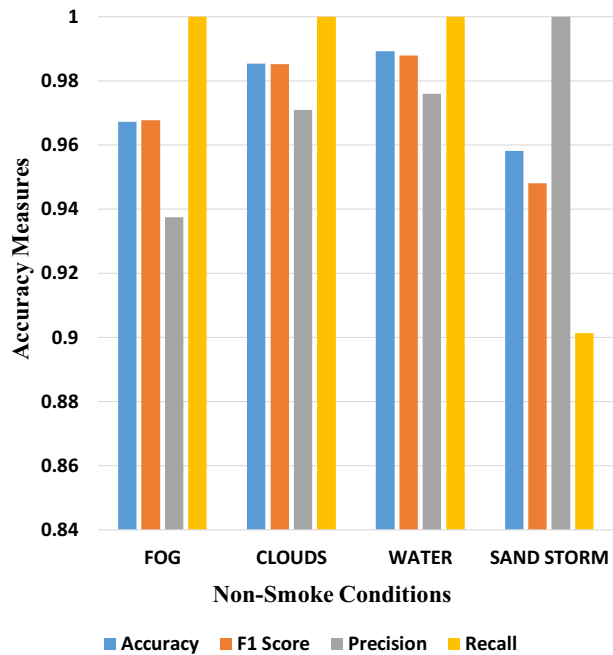
**Table 7**  
**Evaluation of Proposed Method in ‘Sand-Storm’ Condition**

Classifier	Accuracy (in %)	F1 Score	Precision	Recall
CNN	94.0120	0.9242	1	0.8592
SVM	95.8084	0.9481	1	0.9014



**Figure 9. Accuracy measure of CNN Classifier on challenging non-smoke conditions.**

manually define the smoke containing frames and non-smoke frames. That is if small quantity of smoke is present than that frame is considered as smoke otherwise non-smoke. Total time taken by system to classify the smoke and non-smoke frames is 0.5714 s. The accuracy given by system is 94.10% and by CNN is 94.75%, which is very similar as compared to the proposed system.



**Figure 10. Accuracy measure of SVM Classifier on challenging non-smoke conditions.**

**Table 8**  
**Comparison of Proposed Method with State-of-the-Art Method in Different Non-smoke Conditions**

Non-smoke condition	Classifier	Accuracy (in %)	F1 score	Precision	Recall
Fog	CNN	99.0610	0.9910	0.9821	1
	DBN based method	95.6111	0.9533	0.9108	1
	SVM	96.7136	0.9677	0.9375	1
Cloudy	CNN	96.5686	0.9671	1	0.9364
	DBN based method	91.6533	0.9169	0.9092	0.9248
	SVM	98.5294	0.9852	0.9709	1
Shadow of clouds in water	CNN	98.9130	0.9878	0.9759	1
	DBN based method	99.5072	0.9929	0.9891	0.9968
	SVM	98.9130	0.9878	0.9759	1
Sand-storm	CNN	94.0120	0.9242	1	0.8592
	DBN based method	90.2699	0.9055	0.8941	0.9172
	SVM	95.8084	0.9481	1	0.9014

**Table 9**  
**Comparison of Classifiers Based on Average Accuracy Matrices**

Classifiers	Average accuracy (in %)	Average F1 score	Average precision	Average recall
CNN	97.1386	0.9675	0.9895	0.9489
DBN based method	94.2603	0.9421	0.9258	0.9597
SVM	97.4911	0.9722	0.9710	0.9753



**Figure 11. Videos recorded at wild locations, indoor locations and outdoor locations.**

**5. Conclusion**

This paper give the method for frame based smoke detection. One of the finding of our work is that all the essential characteristics of smoke that is texture, color, sharp edge detection, perimeter disorder analysis, moving and the growing regions of smoke has been extracted using two Deep CNNs, which were never being extracted by any method before as per the best of our knowledge. About 11,000 image-based smoke containing dataset is created in the given method (using super pixel technique), which can be useful to other researchers for future work. We have tested our method on near distance smoke and far distant smoke videos. Accuracy on near-by smoke and far-away smoke is 98.29% and 91.96% respectively. The proposed work also works well in the challenging imaging conditions like clouds, fog, sandstorm and reflection of cloud in water, which are very similar to smoke. Our proposed system outperform the method given by [37], which have

given the challenging imaging dataset. The previous accuracy on challenging dataset was 95% which was outperformed by the given method in which accuracy on given method is 97%. For validating the proposed method, we have created our own dataset in which accuracy given by proposed system is 94.10% and by CNN is 94.75%, which is very similar to accuracies computed on the dataset created by superpixel algorithm. The proposed method can also be used to detect video based fire detection and can also be extended for real time smoke detection.

## References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slc superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 34(11):2274–2282
2. Amiaz T, Fazekas S, Chetverikov D, Kiryati N (2007) Detecting regions of dynamic texture. In: International conference on scale space and variational methods in computer vision, SSMCV, pp 848–859. Springer
3. Añazco EV, Lopez PR, Lee S, Byun K, Kim TS (2018) Smoking activity recognition using a single wrist IMU and deep learning light. In: Proceedings of the 2nd international conference on digital signal processing, pp 48–51. ACM
4. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 35(8):1798–1828
5. Calderara S, Piccinini P, Cucchiara R (2008) Smoke detection in video surveillance: a MoG model in the wavelet domain. In: Computer vision systems, pp 119–128
6. Celik T (2010) Fast and efficient method for fire detection using image processing. *ETRI J* 32(6):881–890
7. Celik T, Demirel H, Ozkaramanli H, Uyguroglu M (2007) Fire detection using statistical color model in video sequences. *J Vis Commun Image Represent* 18(2):176–185
8. Çelik T, Özkaramanli H, Demirel H (2007) Fire and smoke detection without sensors: image processing based approach. In: 15th European signal processing conference, pp 1794–1798. IEEE
9. Chen J, He Y, Wang J (2010) Multi-feature fusion based fast video flame detection. *Build Environ* 45(5):1113–1122
10. Chen TH, Wu PH, Chiou YC (2004) An early fire-detection method based on image processing. In: International conference on image processing, ICIP, vol 3, pp 1707–1710. IEEE
11. Chetverikov D, Péteri R (2005) A brief survey of dynamic texture description and recognition. In: Computer recognition systems, pp 17–26. Springer, Berlin, Heidelberg
12. Chino DY, Avalhais LP, Rodrigues JF, Traina AJ (2015) Bowfire: detection of fire in still images by integrating pixel color and texture analysis. In: 28th SIBGRAPI conference on graphics, patterns and images, SIBGRAPI, pp 95–102. IEEE
13. Cui Y, Dong H, Zhou E (2008) An early fire detection method based on smoke texture analysis and discrimination. In: Congress on image and signal processing, CISP, vol 3, pp 95–99. IEEE
14. Dimitropoulos K, Barmpoutis P, Grammalidis N (2015) Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Trans Circuits Syst Video Technol* 25(2):339–351



15. Dimitropoulos K, Barmpoutis P, Grammalidis N (2017) Higher order linear dynamical systems for smoke detection in video surveillance applications. *IEEE Trans Circuits Syst Video Technol* 27(5):1143–1154
16. Doretto G, Chiuso A, Wu YN, Soatto S (2003) Dynamic textures. *Int J Comput Vis* 51(2):91–109
17. Enis Cetin A, Porikli F (2011) Special issue on dynamic textures in video. *Mach Vis Appl* 22(5):739–740
18. Günay O, Taşdemir K, Töreyn BU, Çetin AE (2010) Fire detection in video using IMS based active learning. *Fire Technol* 46(3):551–577
19. Horn BK, Schunck BG (1981) Determining optical flow. *Artif Intell* 17(1–3):185–203
20. Hu Y, Lu X (2018) Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features. *Multimed Tools Appl* 77:1–19
21. Huang FJ, Boureau YL, LeCun Y et al (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: *IEEE conference on computer vision and pattern recognition, CVPR*, pp 1–8. IEEE
22. Huang FJ, LeCun Y (2006) Large-scale learning with SVM and convolutional for generic object categorization. In: *Computer Society conference on computer vision and pattern recognition, CVPR*, vol 1, pp 284–291. IEEE
23. Jarrett K, Kavukcuoglu K, LeCun Y et al (2009) What is the best multi-stage architecture for object recognition? In: *12th International conference on computer vision, ICCV*, pp 2146–2153. IEEE
24. Kaabi R, Sayadi M, Bouchouicha M, Fnaiech F, Moreau E, Ginoux JM (2018) Early smoke detection of forest wildfire video using deep belief network. In: *4th International conference on advanced technologies for signal and image processing (ATSIP)*, pp 1–6. IEEE
25. Ko B, Cheong KH, Nam JY (2010) Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian networks. *Fire Saf J* 45(4):262–270
26. Kolesov I, Karasev P, Tannenbaum A, Haber E (2010) Fire and smoke detection in video with optimal mass transport based optical flow and neural networks. In: *17th IEEE international conference on image processing, ICIP*, pp 761–764. IEEE
27. Kopilovic I, Vagvolgyi B, Szirányi T (2000) Application of panoramic annular lens for motion analysis tasks: surveillance and smoke detection. In: *15th International conference on pattern recognition, ICPR*, vol 4, pp 714–717. IEEE
28. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105. Lake Tahoe, Nevada
29. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
30. LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: *Advances in neural information processing systems*, pp 396–404
31. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
32. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: *International symposium on circuits and systems, ISCAS*, pp 253–256. IEEE
33. Owrutsky JC, Steinhurst DA, Minor CP, Rose-Pehrsson SL, Williams FW, Gottuk DT (2006) Long wavelength video detection of fire in ship compartments. *Fire Saf J* 41(4):315–320
34. Phillips Iii W, Shah M, da Vitoria Lobo N (2002) Flame recognition in video. *Pattern Recognit Lett* 23(1):319–327

35. Piccinini P, Calderara S, Cucchiara R (2008) Reliable smoke detection in the domains of image energy and color. In: 15th IEEE international conference on image processing, ICIP, pp 1376–1379. IEEE
36. Prema CE, Vinsley S, Suresh S (2018) Efficient flame detection based on static and dynamic texture analysis in forest fire detection. *Fire Technol* 54(1):255–288
37. Pundir AS, Raman B (2017) Deep belief network for smoke detection. *Fire Technol* 53(6):1943–1960
38. Qi X, Ebert J (2009) A computer vision based method for fire detection in color videos. *Int J Imaging* 2(S09):22–34
39. Serre T, Wolf L, Poggio T (2005) Object recognition with features inspired by visual cortex. In: Computer Society conference on computer vision and pattern recognition, CVPR, vol 2, pp 994–1000. IEEE
40. Toreyin B, Dedeoglu Y, Cetin AE, Fazekas S, Chetverikov D, Amiaz T, Kiryati, N (2007) Dynamic texture detection, segmentation and analysis. In: 6th ACM international conference on image and video retrieval, CIVR, pp 131–134. ACM
41. Töreyn BU, Dedeoğlu Y, Güdükbay U, Cetin AE (2006) Computer vision based method for real-time fire and flame detection. *Pattern Recognit Lett* 27(1):49–58
42. Vicente J, Guillemant P (2002) An image processing technique for automatically detecting forest fire. *Int J Therm Sci* 41(12):1113–1120
43. Xiong Z, Caballero R, Wang H, Finn AM, Lelic MA, Peng PY (2007) Video-based smoke detection: possibilities, techniques, and challenges. In: IFPA, fire suppression and detection research and applications—a technical working conference (SUPDET), Orlando, FL
44. Xu Z, Xu J (2007) Automatic fire smoke detection based on image visual features. In: International conference on computational intelligence and security workshops, CISW, pp 316–319. IEEE
45. Ye W, Zhao J, Wang S, Wang Y, Zhang D, Yuan Z (2015) Dynamic texture based smoke detection using surfacelet transform and HMT model. *Fire Saf J* 73:91–101
46. Yin M, Lang C, Li Z, Feng S, Wang T (2018) Recurrent convolutional network for video-based smoke detection. *Multimed Tools Appl* 78:1–20
47. Yuan F (2008) A fast accumulative motion orientation model based on integral image for video smoke detection. *Pattern Recognit Lett* 29(7):925–932
48. Yuan F (2011) Video-based smoke detection with histogram sequence of LBP and LBPV pyramids. *Fire Saf J* 46(3):132–139
49. Zhang QX, Lin GH, Zhang YM, Xu G, Wang JJ (2018) Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Eng* 211:441–446