

# A Video Smoke Detection Algorithm Based on Cascade Classification and Deep Learning

Nguyen Manh Dung<sup>1</sup>, Dongkeun Kim<sup>2</sup> and Soonghwan Ro<sup>1</sup>

<sup>1</sup>Dept. of Information and Communication, Kongju National University

<sup>2</sup>Division of Computer Science and Engineering, Kongju National University  
1223-24, Cheonandaero, Subuk-gu, Cheonan, Chungnam, 31080

[e-mail: manhdungbk45@gmail.com, {dgkim, rosh@kongju.ac.kr}]

\*Corresponding author: Soonghwan Ro

*Received March 30, 2018; revised June 4, 2018; accepted July 31, 2018;  
published December 31, 2018*

---

## Abstract

Fires are a common cause of catastrophic personal injuries and devastating property damage. Every year, many fires occur and threaten human lives and property around the world. Providing early important sign for early fire detection, and therefore the detection of smoke is always the first step in fire-alarm systems. In this paper we propose an automatic smoke detection system built on camera surveillance and image processing technologies. The key features used in our algorithm are to detect and track smoke as moving objects and distinguish smoke from non-smoke objects using a convolutional neural network (CNN) model for cascade classification. The results of our experiment, in comparison with those of some earlier studies, show that the proposed algorithm is very effective not only in detecting smoke, but also in reducing false positives.

---

**Keywords:** smoke detection, deep learning, convolutional neural network, cascade classification, BAIR reference CaffeNet

---

This work was supported by the research grant of the Kongju National University in 2017.

## 1. Introduction

Every year, fires cause thousands of human deaths and billions of dollars in property damage. In most cases, however, the fire damage can be prevented, or at least reduced, if the fires are detected earlier. It is therefore very important to develop an automatic fire-alarm system. Since smoke is the most important clue in the early stages of a fire, smoke detection should be a first step in the effort to detect the fire early.

Many of the existing smoke detection systems make use of sensors like ionization detectors, photoelectric sensors and carbon dioxide detectors. The accuracy of such systems primarily depends on the reliability and positions of sensors. These sensors should be distributed densely so as to ensure a high precision of smoke detection systems which may be difficult to install especially in large outdoor spaces.

Recently, digital cameras have been evolving rapidly in the field of security surveillance. Compared to sensor-based systems, security cameras are easy to install and can be used to monitor large open areas. A recent trend is that sensor-based systems have been replaced by developing smoke detection systems based on surveillance camera systems and video analysis techniques.

A large number of image processing algorithms have been proposed for smoke detection by video analysis, and some of them have achieved considerable success. In the following paragraphs, we will discuss some of the most popular and successful technologies which are used for smoke detection.

Most of the proposed algorithms consider smoke as moving objects and assume that smoke will change the background appearance when it appears [2, 3, 5, 6]. Detecting the background change is a technique that is frequently used as the first step of such an algorithm to detect candidate smoke regions and eliminate stationary non-smoke objects. The most effective techniques for detecting background changes include background modeling [1], background subtraction and optical flow estimation.

The background change detection just helps locate candidate smoke regions but fails to distinguish smoke from non-stationary objects like humans, vehicles or varying background illuminations. Further analysis steps are required to verify detected smoke objects.

Colors are widely used to classify smoke [2, 3, 5, 6], which can appear gray, light gray, white or dark gray. In reality, however, there are many objects having similar colors, and in some cases, smoke is semitransparent and thus its color is affected by background colors. Therefore, the color is not always a reliable clue to the presence of smoke.

Such methods as the randomness of smoke area size [2], smoke contours roughness [4,8], growth of smoke regions [8] have been also proposed to eliminate false positives during smoke detection; however, none of these features is perfect and all of them are still prone to generate both false positives or false negative in certain situations.

Another interesting approach for smoke classification is wavelet-based analysis [4, 5]. When the whole or part of a background image is blurred by smoke, the high frequency components and sharpness of the image can decrease on the surface of smoke regions.

Calculating a decrease in wavelet energy provides an important clue for smoke detection. However, this feature is not always correct. For example, the presence of smoke could increase the edge energy of the smooth background surface or some non-smoke objects which have smooth and large surface could decrease the sharpness and edge of the background.

Recently, some image classification algorithms based on local image features (i.e., HOG

---

and SIFT) have been developed to construct bag-of-visual-words representations, and statistical classifiers have been used to classify images into a large number of object categories [7]. This approach shows good results and thus can be used as a way to differentiate between smoke and non-smoke objects [6].

However, the classifiers relying on visual words ignore spatial relationships among patches and confuse background information, when the context of an object is described. Those algorithms have not been extensively tested yet for viewpoint invariance and scale invariance.

Deep learning is a new big trend in machine learning, and a deep learning algorithm called convolutional neural networks (CNNs) [9] has had many recent successes in computer vision, including image classification. It has shown extremely good performance ever and produced a significant change in performance. However, this approach is not able to localize the object position in a video image and costs high in terms of computation. These problems can be overcome by using background subtraction to identify the positions of candidate smoke objects or using a high-speed hardware accelerator (e.g., NVIDIA's CUDA) to increase the computing speed.

Further, most of the non-smoke objects can be eliminated using a cascade model with many weak but fast features, and then computing time can be saved by use of the deep learning classifier as the final step of smoke verification.

This paper proposes a fast and reliable smoke detection algorithm. The methodology and algorithm implementation will be described in section 2. Section 3 will present experimental evaluations of the accuracy and performance of our algorithms. Finally, section 4 will provide some conclusion and discuss future work.

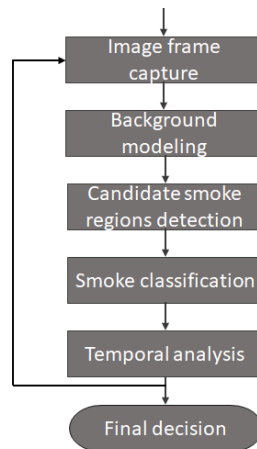
## 2. Video-Based Smoke Detection

**Fig. 1** shows a flowchart of our video-based smoke detection algorithm. Our approach can be divided into three steps: detection of candidate smoke regions, classification of smoke and non-smoke regions, and temporal analysis for making the final classification decision.

**Candidate smoke region detection:** We use the most popular and efficient background modeling algorithm, Mixture of Gaussian Background Modeling (MOG) [1], to detect changes in background pixels. Then we cluster connected pixels into sub regions as candidate smoke regions.

**Smoke classifier:** A set of classifiers (layers) called the “cascade model” is used to classify smoke and non-smoke regions. A candidate smoke region will be classified as true smoke if it passes all layers of the model. In the top layer, we use weak but rapidly processed features such as color, randomness of size variation, and edge energy to eliminate non-smoke regions. However, we have to choose thresholds to make sure that only non-smoke regions are eliminated. A lenient threshold could make for more false positives, but these will be reduced by later classifier layers. The final layer of the cascade model is reliable a deep learning image classifier for verifying candidate smoke regions.

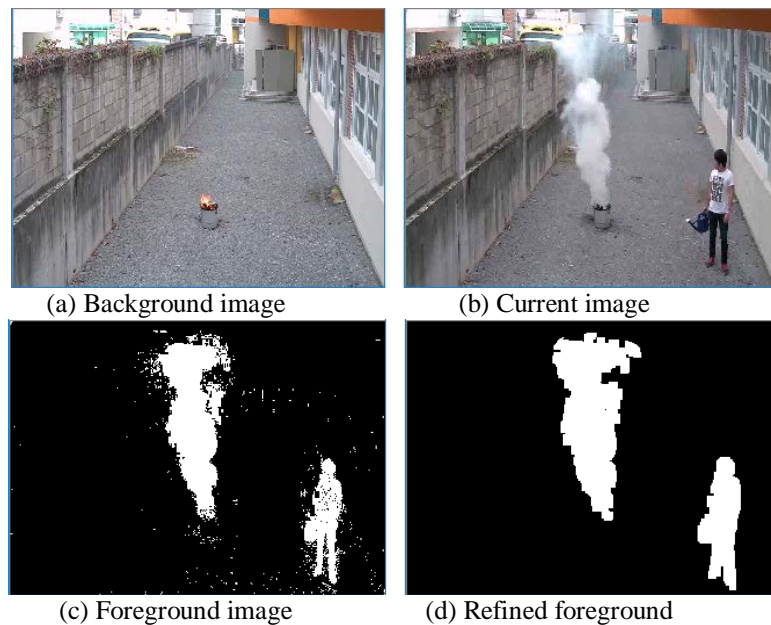
**Temporal Analysis:** The final decision stage, increasing the precision of smoke detection.

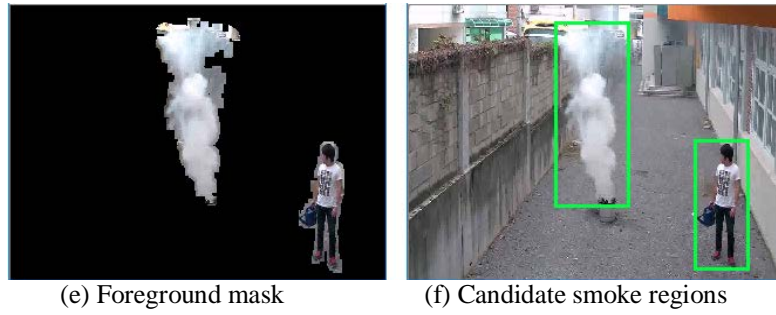


**Fig. 1.** Flow chart of our video- based smoke detection algorithm

## 2.1 Candidate smoke region detection

Once smoke appears, it will change the appearance of the background image scene. Detecting these changes is a good way to identify smoke. Numerous algorithms have been developed for this task; Mixture of Gaussian background modeling (MOG) [1] is the most popular and widely-used. Although MOG still has some problems, such as sensitivity to local or global illumination changes, it is effective for most applications. We applied MOG for the detection of candidate smoke regions, and addressed its problems at later steps in the classification process. Firstly, MOG classifies image pixels into two classes: background pixels and foreground pixels. After that, it clusters connected foreground pixels into blobs, with each blob being one candidate smoke region. **Fig. 2** shows the steps for extracting candidate regions. One or several candidates can be detected concurrently.

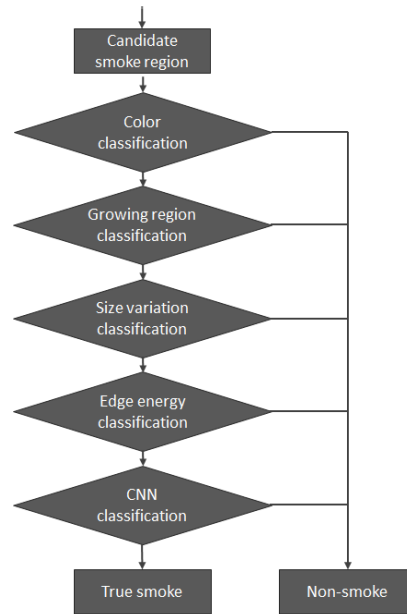




**Fig. 2.** Candidate smoke regions detection

## 2.2 Cascade model for smoke classification

The cascade model consists of multiple smoke classifiers, with each layer being one classifier. Only candidate regions which pass all layers are classified as true smoke. **Fig. 1** shows the overall algorithm architecture, and **Fig. 3** shows the cascade model layer architecture.



**Fig. 3.** Cascade model architecture

### 1) Color classification

Usually smoke is dark gray, gray, light gray, or white in color. Therefore, an image pixel will be classified as a smoke pixel if it meets the following conditions:

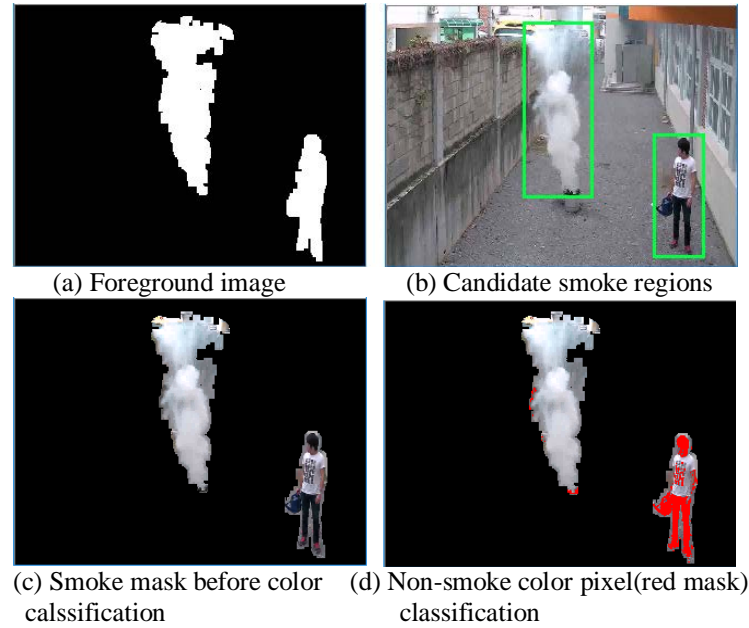
$$|I_R - I_G| < th_c \quad (1)$$

$$|I_R - I_B| < th_c \quad (2)$$

$$|I_G - I_B| < th_c \quad (3)$$

$$I_{min} < I < I_{max} \quad (4)$$

Where  $I_R, I_G, I_B$  are the intensity of red, green and blue color channel of the image pixel,  $th_c$  is a threshold value, and  $I$  is the image pixel intensity. In our experiments,  $th_c$  ranged from 5 to 25 and  $80 < I < 220$  for smoke pixels.



**Fig. 4.** Cascade model architecture

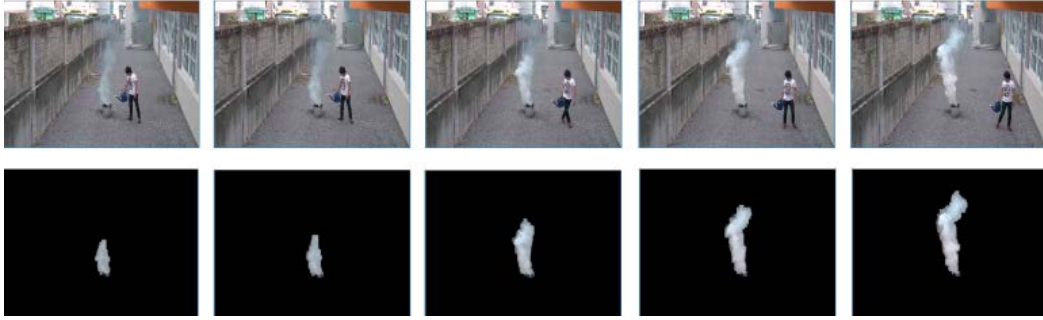
**Fig. 4** show the classification of image pixels inside candidate smoke regions into smoke and non-smoke color pixels. For true smoke region, the number of non-smoke color pixel is very low. So that, to classify a candidate smoke regions into smoke or non-smoke by using color feature we counting all of smoke color pixels  $N_{color}$  and total pixels  $N_{total}$  inside this region. If ratio between  $N_{color}$  to  $N_{total}$  lower than a certain threshold  $th_{rc}$  this region will be classified as non-smoke and will be eliminated for next process.

$$N_{color}/N_{total} < th_{rc} \quad (5)$$

$th_{rc}$  is determined in experiment to make sure that only non-smoke regions are rejected in this step.

## 2) Growing region classification

When smoke first appears, it will quickly spread out into the air; this feature means that smoke regions are constantly growing over a period of time. **Fig. 5** illustrates the growth of one smoke region.



**Fig. 5.** Growth of a smoke region

The analysis of a growing region is as follows: Let's track the size of a candidate smoke region in a period of time. If the size of this region gradually increase it will be determined as a growing region and classified as a possible of smoke for next process.

Assume that  $f_{growth}$  and  $n_{growth}$  are growing factor and growing steps counter of each candidate smoke regions. Initially,  $n_{growth}$  is set to zero and  $n_{growth}$  increments by one if size of the candidate smoke region reached a certain value as described in equation (6).

$$n_{growth} = n_{growth} + 1 \quad \text{if } S_{t+1} > S_t \times f_{growth} \quad (6)$$

Where  $S_t$  and  $S_{t+1}$  are the size of the candidate smoke region at current and next growing step. If  $N_{frame}$  is the number of analysis frame, the candidate smoke region will be classified as a growing region if it meet following condition:

$$\frac{n_{growth}}{N_{frame}} > th_{growth} \quad (7)$$

### 3) Size variation classification

The size of a smoke region changes randomly due to airflow; this can also makes a good feature for classifying smoke and non-smoke areas. Following are the parameters need to be calculated to estimate the size variation feature of candidate smoke regions:

The size change  $d_s^t$  for the smoke region at time t, given by:

$$d_s^t = |S_t - S_{t-1}| \quad (8)$$

The normalized area change  $d_A^t$ , given by:

$$d_A^t = \frac{d_s^t}{S_t} \quad (9)$$

And the standard deviation of size variant  $std_s^t$  over n recent frames at time t, given by:

$$Std_s^t = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (d_s^{t-i} - \overline{d_s})^2} \quad (10)$$

A candidate smoke region will be passed to the next layer if it satisfies the following conditions:

$$d_A^t > th_{dA} \quad (11)$$

$$std_s^t > th_{stds} \quad (12)$$

Where  $th_{dA}$  and  $th_{stds}$  are the decision threshold, which are selected during experiment.

#### 4) Edge energy classification

Once smoke appears, it will blur the background image; the edges of image pixels inside a smoke region can lose their sharpness, leading to a decrease in edge magnitude. To identify decreases in the edge magnitude of a smoke region, we estimate the numbers of pixels that have lost edge magnitude,  $N_{G-}$ , and that have gained edge magnitude,  $N_{G+}$ . For a true smoke region,  $N_{G+}$  is much smaller than  $N_{G-}$ .

In order to estimate  $N_{G+}$  and  $N_{G-}$ , firstly we calculate the gradient magnitude for background image pixels using the following equations:

$$Gbx_{x,y} = B_{x+1,y} - B_{x-1,y} \quad (13)$$

$$Gby_{x,y} = B_{x,y+1} - B_{x,y-1} \quad (14)$$

$$Gb_{x,y} = \sqrt{Gbx_{x,y}^2 + Gby_{x,y}^2} \quad (15)$$

Where  $Gb_{x,y}$   $Gbx_{x,y}$   $Gby_{x,y}$   $B_{x,y}$  are the gradient magnitude, vertical gradient, horizontal gradient, and intensity of the current background image at position (x, y), respectively.

Similarly, we also calculate the gradient magnitude for current image pixels:

$$Gx_{x,y} = I_{x+1,y} - I_{x-1,y} \quad (16)$$

$$Gy_{x,y} = I_{x,y+1} - I_{x,y-1} \quad (17)$$

$$G_{x,y} = \sqrt{Gx_{x,y}^2 + Gy_{x,y}^2} \quad (18)$$

Where  $G_{x,y}$   $Gx_{x,y}$   $Gy_{x,y}$   $I_{x,y}$  are the gradient magnitude, vertical gradient, horizontal gradient, and intensity of the current frame at position (x, y), respectively.

One pixel is considered a lost edge magnitude pixel if:

$$G_{x,y} < Gb_{x,y} - th_{mag} \quad (19)$$

or a gained edge magnitude pixel if:

$$G_{x,y} > Gb_{x,y} + th_{mag} \quad (20)$$

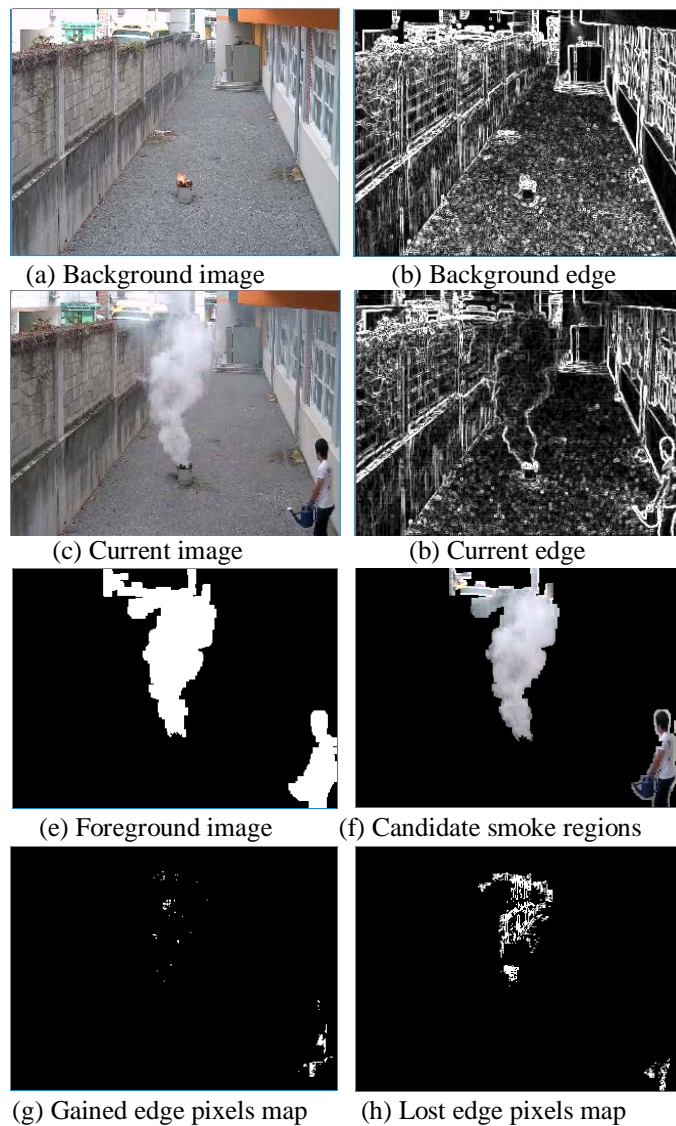
where  $th_{mag}$  is a threshold selected to remove background noise.



**Fig. 7** shows representative maps of lost edge magnitude pixels and gained edge magnitude pixels. For a true smoke region, we can easily see that the number of gained edge magnitude pixels is much smaller than the number of lost edge magnitude pixels. Using these edge magnitude-based features, we identify a candidate as a smoke region if it satisfies the following condition:

$$\frac{N_{G+}}{N_{G-}} < th_e \quad (21)$$

where  $th_e$  is a decision threshold.



**Fig. 6.** Maps of lost edge magnitude pixels and gained edge magnitude pixels for a true smoke region

### 5) *Deep learning image classification using a convolution neural network*

For the final layer of the cascade model, we built a robust and reliable smoke/non-smoke classifier using a deep learning algorithm called a convolutional neural network (CNN). Recent applications of this framework [9] show extremely successful results for the classification of natural images into sub-object categories. Before explaining the application of CNN algorithms to our task, we will briefly describe how a CNN works.

#### **Convolutional Neural Network**

CNNs are a type of machine learning algorithm used for classification tasks. Classifications using machine learning include two phases, training and prediction. In the training phase, we train the algorithm using a known dataset comprised of images and their corresponding class labels. In the prediction phase, we use the trained model to predict the classes of previously unseen images.

CNNs are one kind of Artificial Neural Networks (ANN), which includes a lot of artificial neurons connected together to form neural network. Artificial neuron is a learnable filter, that include a number of inputs and an activation function (usually called as transfer function). Each input will be combined with a weight and the output is the resulting of transfer function applied to weighted sum of inputs. The input weights are adjusted during training network to control the filter output as desired.

A CNN includes a sequence of layers, where output of each layer is the input of next layer. The architecture of CNN starts with input layer, followed by a multi hidden layer and ends with output layer. The hidden layers could be convolutional, pooling, Relu or fully connected layer.

#### **Transfer learning**

Training a CNN requires a large datasets and a lot of computational time. This leads to difficulty when we want to retrain or update a trained model for other categories. Transfer learning [10, 11, 12] aims to overcome these difficulties. Instead of retraining the network from scratch, transfer learning utilizes a trained model on a different dataset and adapts it to train a new classifier.

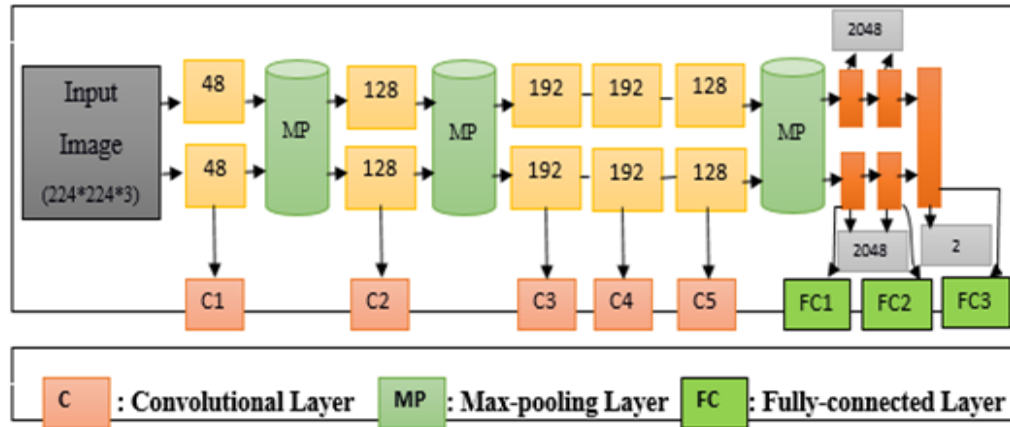
Fine-tune the trained model is one of the transfer learning approaches, this approach fine-tunes the trained model on the new dataset by continuing back propagation. It can either fine-tune all the layers of the network or keep some of its layers.

The advantages of CNN Image classification are robust against distortions, such as change in shape, poses, scale, lighting condition or presence of partial occlusions. Experimental results show that CNN is sufficient for algorithmic use to achieve the state of the art performance in image classification and classify smoke and non-smoke object.

#### **Fine-tune CNN for smoke classification**

We use CaffeNet[14] which was developed by Berkeley AI Research (BAIR). It is a trained CNN model for image classification[9]. The model was trained on the Image Net Database [15], which contains millions of images across 1,000 object categories. In this paper, we use the fine-tune approach in the CNN to retrain our image data for smoke classification.

Fig. 7 shows the original architecture of the trained model, first five layer are convolutional and some of them are followed by max pooling layer. The next are three fully-connected layer, the last fully-connected layer computes class score of 1000 class labels.



**Fig. 7.** Architecture of the CaffeNet CNN model

In order to train the CNN network for smoke classification, we replaced the last layer with a new one trained from scratch using the back-propagation algorithm with our image dataset, which has only two different categories (smoke and non-smoke). The whole layers of cascade model work together as follows.

Candidate smoke region (**Fig. 2f**) is an area containing a moving object, which was detected by using Mixture of Gaussian background Model(Section 2.1). A cascade smoke classification model includes sequence of smoke classifiers, but the input of a cascade model is a single sub-image that contains candidate smoke region. If a candidate smoke region is classified as non-smoke at current layer it will be rejected immediately otherwise it will be passed to next layer for classification process. A candidate smoke region is only classified as true smoke if it pass all of layers of cascade model.

A non-smoke image may be of a human, a vehicle, or just a simple background image. Our dataset has 10,000 smoke images and 10,000 non-smoke images for training and another 2,000 smoke and 2,000 non-smoke images for evaluation.



(a) Smoke image



(b) Non-smoke image

**Fig. 8.** Example of training image

Non-smoke object images were collected from various sources such as the PETA dataset (pedestrian images), the Cars dataset (vehicles), and the PASCAL dataset (backgrounds and other moving objects). We also manually segmented non-smoke objects from surveillance videos which were recorded for the IVS project [16].

Smoke object images were manually segmented from IVS project smoke videos [16], videos from YouTube, and other videos from the internet (<http://signal.ee.bilkent.edu.tr/VisiFire>). Also we uploaded our results on test videos to YouTube.

(<https://www.youtube.com/playlist?list=PLh7GPJJcJC1gFKTBjbC9dJ7p6VSftWg6n>)

We began the fine-tuning process with a learning rate of 0.01, and dropped it by a factor of ten every 2,000 iterations. We used a smaller learning rate for weights being fine-tuned under the assumption that the pre-trained CNN weights were relatively good; we didn't want to distort them too quickly or by too much. The optimization process was run for a maximum of 50,000 iterations. The accuracy of trained smoke classification model is 98%, false negative is 2.3% and false positive is 1.7%

### 2.3. Temporal analysis.

Temporal analysis is final verification stage to increase the reliability of the decision. At this stage we analyze the history of the images sequence over a period of time. For example, if the number of image frames with smoke over a period of time exceeds the threshold for the probability of smoke being included, the system should generate a fire alarm.

## 3. Testing And Evaluation

We implemented our experiments using Visual Studio 2015 and the open source libraries OpenCV 2.4.10, NVIDIA CUDASDK, and Caffe Deep Learning framework. We ran the experiments on a Windows 10 computer with an Intel® Core™ i7-4790 processor and a NVIDIA GeForce GTX 750 graphics card.



We used 15 videos, including 10 smoke videos and 5 non-smoke videos for evaluation. **Fig. 9** shows some example results from our experiments. The upper left image is movie\_01, and the lower right image is movie\_15.



**Fig. 9.** Smoke detection results Red boundaries are detected smoke regions; green boundaries non-smoke moving objects

An algorithm that does not implement a CNN classifier generally requires setting a high decision threshold to reduce false positives, but it does not detect many real smoke regions and still has a high false positive rate. The details of the experiment and evaluation are summarized in **Table 1**.

The algorithm based on motion vector, surface roughness, and area randomness[2] missed detection of smoke in movie\_06 and false detection of smoke in movie\_11 and movie\_15. Follow this algorithm, smoke is classified as true smoke if the variation of the motion vector is large and the size of the smoke region changes randomly and quickly. However, in movie\_06 inside the tunnel, the air flow is low and the smoke seems to gradually spread-out into surrounding, in that case both direction and size of smoke are not varying so much so that smoke might be classified as non-smoke object. This algorithm also has problem in movie\_11 and movie\_15 when a human or human group moving around. In this case, direction, size change, color, and surface roughness are all similar to smoke characteristics and lead to false detection.

The algorithm based on decrease of the background edge [4] has some difficulties if the background edge is poor or the object surface edge is poor. As in movie\_06, if smoke appears when the edges of the background are poor, edge energy of background will increase instead of reducing edge energy, or energy of background edges will decrease on objects of movie\_15 with poor surface edges. The algorithm also uses boundary roughness to classify the smoke and non-smoke. According to this algorithm, the boundary roughness of non-smoke objects looks smoother than smoke objects, however this feature is not always correct. As shown in movie\_06 and movie\_15 of **Fig. 9**, smoke boundaries look smoother than non-smoke objects.

Other algorithms are based on machine learning, which classifies smoke using bag of feature histograms and random forest classifiers[6]. This algorithm has some advantages over the heuristic rule classifier, but it still has disadvantages when it ignores object spatial relationships, confuses background information when computing features, and especially when the smoke is semi-transparent in the background.

In contrast to other techniques, our algorithm tries to select a decision threshold that removes only moving objects with low probability of being smoke. Depending on the robustness and high accuracy of the CNN classifier, our algorithm not only reduces false positives, but also achieves excellent detection rates. As shown in **Table 1**, our algorithm successfully detected smoke on every short video that contained smoke, and false positives were not returned [17].

**Table 1.** Summary of evaluation results

Video No	Description	Number of frame	Smoke Detected/False Positive			
			T. X. Tung [2]	Toreyin [4]	C. Ko [6]	Our
movie_01	Smoke video	2000	Detected	Detected	Detected	Detected
movie_02	Smoke video	2000	Detected	Detected	Misdetected	Detected
movie_03	Smoke video	2000	Detected	Detected	Detected	Detected
movie_04	Smoke video	2000	Detected	Detected	Detected	Detected
movie_05	Smoke video	1000	Detected	Detected	Detected	Detected
movie_06	Smoke video	1000	Misdetected	Misdetected	Detected	Detected
movie_07	Smoke video	500	Detected	Detected	Detected	Detected
movie_08	Smoke video	1499	Detected	Detected	Detected	Detected
movie_09	Smoke video	898	Detected	Detected	Detected	Detected
movie_10	Smoke video	1398	Detected	Detected	Detected	Detected
movie_11	None Smoke video	296	False detect	No smoke	No smoke	No smoke
movie_12	None Smoke video	1942	No smoke	No smoke	No smoke	No smoke
movie_13	None Smoke video	298	No smoke	No smoke	No smoke	No smoke
movie_14	None Smoke video	340	No smoke	No smoke	No smoke	No smoke
movie_15	None Smoke video	4000	False detect	False detect	No smoke	No smoke
Average						

By using NVIDIA computing accelerated hardware, our algorithm also achieves very good computing performance. The processing time for top classifier layers of cascade model less than one millisecond and processing time of CNN classifier is about eight millisecond per image plus processing time for other parts. Test results for system performance is summarized in **Table 2**. Experimental results show that our system can handle 40 frames per second and can detect smoke between 3 to 10 second, making the system suitable for real-time applications.

**Table 2.** Results of performance test

Performance test			
Video No	Description	Detectime (second)	fps
movie_01	Smoke video	3	40
movie_02	Smoke video	10	40
movie_03	Smoke video	5	40
movie_04	Smoke video	4	40
movie_05	Smoke video	4	40
movie_06	Smoke video	5	40
movie_07	Smoke video	3	40
movie_08	Smoke video	5	40
movie_09	Smoke video	5	40
movie_10	Smoke video	3	40

## 4. Conclusion And Future Work

Automatic fire detection is very important, as a prompt fire warning gives people more chance to escape and reduces property damage.

This paper proposed a video-based smoke detection algorithm for use as an early fire-alarm system. The algorithm is based on detection of changes with background modeling, and cascade classification, and a deep learning convolution neural network. The cascade model helps by exploiting the advantages of multi-feature classification, and the success of CNNs in image classification promises high accuracy for smoke detection. Our algorithms show a big improvement over related methods. Experimental results show that this classification method is fast, the algorithms reliable, and the method fully suited for use in a real-time surveillance system.

In future work, we will try to identify additional good classifier features in order to improve the accuracy of our model. We also plan to extend our proposed model to fire flame detection in order to develop a complete fire-alarm system.

## References

- [1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. of IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognition*, vol. 2, 1999. [Article \(CrossRef Link\)](#)
- [2] T. X. Tung and J.-M. Kim, "An effective four-stage smoke-detection algorithm using video images for early fire-alarm systems," *Fire Safety J.*, vol. 46, no. 5, pp. 276-282, Jul. 2011. [Article \(CrossRef Link\)](#)
- [3] W. Zheng, W. Xingang, A. Wenchuan, and C. Jianfeng, "Target-tracking based early fire smoke detection in video," in *Proc. of ICIG '09*, pp. 172-176, Sept. 2009. [Article \(CrossRef Link\)](#)
- [4] B. U. Toreyin, Y. Dedeoglu, and A. Enis Cetin, "Contour based smoke detection in video using wavelets," in *Proc. of 14th Eur. Signal Process. Conf.*, pp. 1-5, Sept. 2006. [Article \(CrossRef Link\)](#)
- [5] C.-Y. Lee, C.-T. Lin, C.-T. Hong, and M.-T. Su, "Smoke detection using spatial and temporal analysis," *Int. J. Innovative Comput., Inf. and Contr.*, vol. 8, no. 7(A), Jul. 2012. [Article \(CrossRef Link\)](#)
- [6] B. C. Ko, J. Y. Kwak, and J. Y. Nam, "Wildfire smoke detection using temporal-spatial features and random forest classifiers," *Opt. Eng.*, vol. 51, no. 1, Feb. 2012. [Article \(CrossRef Link\)](#)
- [7] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. of ICCV 2007*, pp. 1-8, Oct. 2007. [Article \(CrossRef Link\)](#)
- [8] A. Genovese, R. D. Labati, V. Piuri, and F. Scotti, "Wildfire smoke detection using computational intelligence techniques," *CIMSA*, pp. 1-6, Sept. 2011. [Article \(CrossRef Link\)](#)
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. of NIPS'12*, Dec. 2012. [Article \(CrossRef Link\)](#)
- [10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," *2014 IEEE CVPR*, pp. 1717-1724, Jun. 2014. [Article \(CrossRef Link\)](#)
- [11] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. of UTLW'11 Proc. 2011 Int. Conf. Unsupervised and Transfer Learning workshop*, vol. 27, pp. 17-37, 2012. [Article \(CrossRef Link\)](#)
- [12] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, "Fine-tuning deep convolutional networks for plant recognition," in *Proc. of Working Notes of CLEF 2015-Conf. and Labs of the Evaluation Forum CLEF 2015*, Sept. 2015. [Article \(CrossRef Link\)](#)
- [13] Convolutional Neural Networks, [Article \(CrossRef Link\)](#)
- [14] BAIR/BVLC CaffeNet Model, [Article \(CrossRef Link\)](#)

- [15] ImageNet, [Article \(CrossRef Link\)](#)  
[16] Intelligent video surveillance system(IVS) Project, [Article \(CrossRef Link\)](#)



**Nguyen Manh Dung** received B.S. degrees from the Department of Electronics And Telecommunication Engineer at Hanoi University Of Science And Technology in 2005, and M.S degrees from Department of Information And Communication at Kongju National University in 2009. He was a senior research engineer in the Research And Development Department of IVS Technology. Since 2017 he became a PhD Student in DC lab of Information And Communication Department at Kongju National University. His interests include embedded system, image processing and video analysis algorithms for surveillance camera system.



**Dongkeun Kim** is a professor of computer science and engineering at Kongju National University. He received his B.S, M.S, and Ph.D degrees from Chungnam National University in 1989, 1991, and 1996, respectively. His research interests are Image processing, Computer vision, and Machine learning



**Soonghwan Ro** received B.S., M.S., and Ph.D degrees from the Department of Electronics Engineering at Korea University in 1987, 1989, and 1993, respectively. He was a research engineer of Electronics and Telecommunications Research Institute and University of Birmingham in 1997 and 2003, respectively. Since March 1994 he has been a professor at Kongju National University, Korea. His research interests include 5G communication, mobile network, and embedded systems.