Video Smoke Detection For Surveillance Cameras Based On Deep Learning In Indoor Environment

Viet Thang Nguyen*, Cong Hoang Quach, Minh Trien Pham
* VNU University of Engineering and Technology
Ha Noi, Viet Nam
Email: 16020048@vnu.edu.vn

***Abstract*—An early fire detection in indoor environment is essential for people's safety. During the past few years, many approaches using image processing and computer vision techniques were proposed. However, it is still a challenging task for application of video smoke detection in indoor environment, because the limitations of data for training and lack of efficient algorithms. The purpose of this paper is to present a new smoke detection method by using surveillance cameras. The proposed method is composed of two stages. In the first stage, motion regions between consecutive frames are located by using optical flow. In the second stage, a deep convolutional neural network is used to detect smoke in motion regions. To overcome the problem of lacking data, simulated smoke images are used to enrich the dataset. The proposed method is tested on our data set and real video sequences. Experiments show that the new method is successfully applied to various indoor smoke videos and significant for improving the accuracy of fire smoke detection. Source code and the dataset have been made available online.**

***Index Terms*—Deep convolutional neural networks, Smoke detection, Simulated smoke image**

I. Introduction

Smoke detection is necessary and important for public safety. Among different approaches, the use of visible-range video captured by surveillance cameras are particularly convenient for smoke detection, as they can be deployed and operated in a cost-effective manner. As smoke spreads faster and in most cases will occur much faster than flame in the field of view of the cameras [1], smoke detection provides earlier fire alarms than flame detection. Image smoke recognition is a fundamental problem for visual smoke detection since a video is composed of sequential images.

According to object detection, visual smoke detection can be roughly categorized into traditional and deep learning-based methods. The traditional methods mainly focus on features extraction for smoke recognition. The typical features used for detection are hand-crafted features, including color, texture, motion orientation, etc. The traditional methods detect smoke in an image by judging whether the number of features extracted as smoke surpasses a threshold. Chen [2] proposed a method using wavelet transformation to distinguish smoke. Yu [3] used optical flow computation to calculate the motion feature of smoke. Those proposals tend to be less effective in different images dataset because of the poor robustness of the algorithm. In recent years, deep learning has garnered tremendous success in a variety of application domains. Experimental results show state-of-the-art performance using deep learning on computer vision tasks, including image classification and object detection. Deep learning methods, especially Convolutional neural networks (CNNs) can learn complex characteristics from large amounts of images dataset and avoid hand-crafted design features in contrast to traditional methods. Recently, many approaches use CNNs for smoke detection. Frizzi [4] trained a convolutional neural network for wildfire smoke recognition, which applied a sliding window to select region and used CNNs to detect the fire and smoke in a frame video. Sharma [5] proposed a model consisting of a full image CNNs and local patch classifier for forest fire detection, both of which share the same deep neural networks. In summary, the proposed methods above have sliding windows or region proposals to generate region of interests (ROIs) first and after that focus on image classification.

This paper proposes a two-stage approach for video smoke detection in indoor environment. In indoor environment, the task of detecting small smoke is fundamental in early fire detection systems. It means the smoke located on a few small parts of an image. To avoid the difficulty in using sliding windows to generate ROIs, our method uses optical flow to detect motion and locate regions that can

be smoke. Because an image is in high-dimensional space, coming with various sizes or resolutions, visual information of small objects is less than medium or big objects, so it is hard to exploit these information for detecting small objects. We perform our evaluation on the current state-of-the-art approaches based on Deep Learning as Faster R-CNN [6] and SSD [7] models then show how well-performed the detection models are when applying them to detect smoke in ROIs from large images.

Meanwhile, available smoke images for training are obtained generally from internet and experiments, which are limited in scale and diversity for training model. Because the image gathering of smoke and fire is complex and there are many safety concerns in indoor environment, the lack of data is a difficult problem for training model. We solve this problem by using synthetic smoke images to extend the training set. Similar work is described in [8], which focused on synthesizing wildfire smoke images. In my approaches, the benefit of synthetic smoke in indoor environment is to increase accuracy of the deep learning model trained on them.

This paper is organized as follows. In section II, we present ours proposed for video smoke detection in details. Section III describes the synthesizing method of indoor smoke datasets. Experimental results are given in Section IV. At last, we conclude the paper in Section V.

II. Proposed Video Smoke Detection Method

Smoke objects normally have single color and grow with undefined shapes. Based on smoke definition, the proposed method includes two steps. Fig. 1 presents the flow chart of the proposed method. Firstly, we apply optical flow to detect motion between consecutive frames and locate regions of interest. Secondly, we use deep learning model to detect smoke in the detected regions.

A. Detect motion regions

Optical flow estimation is still one of the key problems in computer vision. Optical flow is a widely used method for measurement of target velocity in video by computing difference of frame sequence. Some researches utilized optical flow for motion estimation in dynamic texture. From the original approaches of Horn and Schunck [9] as well as Lucas and Kanade [10], researchers developed many new approaches for dealing with shortcomings of previous models. In smoke detection problem, the dynamic characteristics of the smoke has significant changes in the optical flow field when smoke breaks out. In general, optical flow algorithms can be roughly classified into the following categories: "gradient" methods, "phase" methods, "region-based matching" methods and "feature-based" methods. Consider a pixel $I(x, y, t)$, which is the intensity of a pixel at location (x, y) and time t , it moves by distance (dx, dy) in next frame taken after dt time. Since those pixels are the same and intensity does not change, it leads to the equation:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1)$$

Then take taylor series approximation of right-hand side, remove common terms and divide by dt to get the following equation:

$$f_x u + f_y v + f_t = 0 \quad (2)$$

where:

$$f_x = \frac{\partial f}{\partial x}; f_y = \frac{\partial f}{\partial y}; u = \frac{dx}{dt}; v = \frac{dy}{dt} \quad (3)$$

Above equation is called optical flow equation. In it, we can find f_x and f_y, they are image gradients. Similarly f_t is the gradient along time. But (u, v) is unknown. We cannot solve (2) with two unknown variables. So several methods are provided to solve this problem and one of them is Lucas-Kanade's method [10] or Gunner Farneback's method [11]. Since smoke flow has many different shapes, densities and is smooth, sparse optical flow is infeasible in this case. To address this problem, we apply dense optical flow to locate region of interest. In fig 2, we illustrate the optical flow result tested on sample videos. In the first row, we show the result from Gunner Farneback's algorithm and in the second is from Lucas-Kanade's method.

To locate regions of interest, the frame is divided into a grid, each sub-region size is 16x16 pixels.

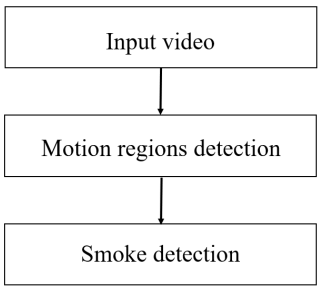


Fig. 1: Flow chart of the proposed method

From the optical flow map, if the number of changed pixels in a sub-region is higher than a threshold, the sub-region is a motion region. After that, we locate the regions on interest by detect all connected components of the motion sub-region. In fig 2, we show the motion regions which were detected form the optical flow map.

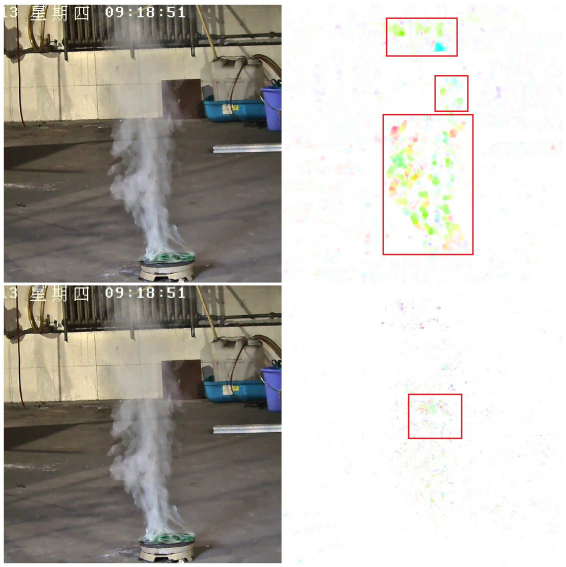


Fig. 2: Optical flow in smoke videos

B. Smoke Detection With Neural Networks

The convolutional neural network was first introduced in 1980 by Fukushima [12]. In object detection, current state-of-the-art object detectors consist of one-stage detectors and two-stage detectors [13]. In two-stage detectors, the first stage generates a set of candidate regions, which may contain objects, while filtering out the majority of negative locations, and in the second stage the classifier determines objects in the proposed regions. Recent two-stage detectors are mainly based on region proposal network, suck as Faster R-CNN [6], R-FCN [14]. In one-stage detectors, after using a feature extractor, the model performs object proposal with multiple convolutional layers instead of region proposal network to ease the inconsistency between the sizes of objects and receptive fields, and run faster. Recently, SSD [7] and YOLO [15] are current state-of-the-art models in one-stage detectors. In this paper, we use two state-of-the-art such as Faster R-CNN and SSD that have achieved one of the lowest errors in object detection tasks.

In Faster R-CNN and SSD, they use a convolutional neural network as a feature extractor. In convolutional neural network, kernels are used to see where particular features are present in an image by convolution with the image. The size of the kernels gives rise to locally connected structure which are each convolved with the image to produce feature maps. In this paper, we use two state-of-the-art convolutional neural network such as VGG16 [16] and Resnet-50 [17] as feature extractors:

- VGG16: The main purpose of the paper was to investigate the effect of depth in CNN models. The 19 layer architecture (VGG-19) won the ImageNet challenge in 2014, but the 16 layer architecture, VGG16 achieved an accuracy which was very close to VGG19. Both the models are simple and sequential. The 3×3 convolution filters are used in the VGG models which is the smallest size and thus captures local features. The 1×1 convolutions can be viewed as linear transformations and can also be used for dimensionality reduction. We choose the VGG16 over the VGG19 because it takes less time to train and the classification task in hand is not as complex as ImageNet challenge.
- Resnet50: This model was created by Microsoft Research, they introduced residual learning. Residual learning involves learning residual functions. If a few stacked layers can approximate a complex function, $F(x)$ where, x is the input to the first layer, then they can also approximate the residual function $F(x) - x$. So, instead the stacked layers approximate the residual function $G(x) = F(x) - x$, where the original function becomes $G(x) + x$. Even though both can capable of approximating the desired function, the ease of training with residual functions is better. These residual functions are forwarded across layers in the network using identity mapping shortcut connections. The Resnet architectures consist of networks of various depths: 18 layers, 34 layers, 50 layers, 101 layers and 152 layers. We choose the architecture with intermediate depth, i.e. 50 layers.

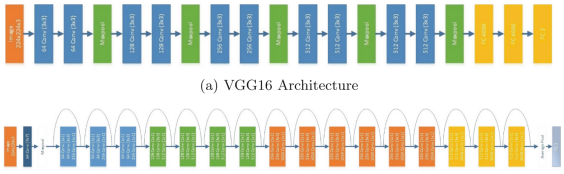
Fig. 3 show the original VGG16 and Resnet-50 architectures respectively. In our approaches, we train those model as features extraction networks with smoke datasets, including real smoke images and simulated smoke images. Faster R-CNN and SSD have different architectures:

- SSD uses a single feed-forward convolutional network to directly predict categories and anchor offsets without requiring a second stage per-proposal classification operation. SSD adds

convolutional feature layers to the end of the base network. These layers decrease in size progressively and allow predictions of detections at multiple scales. Based on the multi-scale feature layers, convolutional predictors produce detection predictions using a set of convolutional filters. All the predictions produced by each detection branch will be integrated together for sampling.

- Faster R-CNN is one of a pioneer which is open for the trend of object detection based on Deep Leaning. In this work, the authors showed the progress to create hypotheses before taking them into classifiers is a crucial step in detection and it takes most of the time of data processing of the entire progress. The authors indicate that this is a bottleneck so they have proposed a new method called Region Proposal Network (RPN) that shares convolutional features of the whole image with the network used for detection, hence it enables mostly cost-free region proposals. By using the RPN, Faster R-CNN is speeded up.

It is very hard to have a fair comparison between SSD and Faster R-CNN. Sample architecture of those networks are showed in Fig. 4.



(a) VGG16 Architecture

(b) Resnet50 Architecture

Fig. 3: The original VGG16 and Resnet-50 architectures

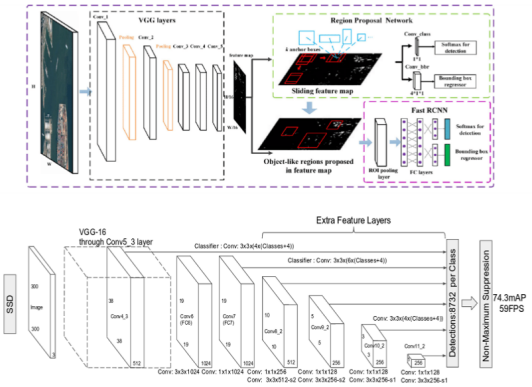


Fig. 4: Faster R-CNN and SSD architectures

III. SYNTHESIZING METHOD OF SMOKE DATA

Since it is difficult to collect many smoke images in indoor environment, using smoke images from the simulator in order to train and validate the smoke detection systems appears as a feasible alternative. In general, the two major contexts related to the simulation of smoke frame sequences are the computational fluid dynamics and computer graphics. All of the methods appertaining to these areas are based on the equations of the fluid flow. The Navier–Stokes equations [18] describe the physical model used in fluid dynamics by considering the flow of a compressible and viscous fluid in terms of a velocity vector field:

$$\frac{\partial}{\partial t}(\rho \boldsymbol{v}) + \nabla \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v}) = -\nabla \cdot p\boldsymbol{I} + \nabla \cdot \boldsymbol{\tau} + \rho \boldsymbol{g} \quad (4)$$

where ρ is the fluid density, $\boldsymbol{v}$ is the flow velocity, ∇ is the divergence operator, p is the pressure, t is time, $\boldsymbol{I}$ is an identity matrix, $\boldsymbol{\tau}$ is Cauchy stress tensor, $\boldsymbol{g}$ represents body accelerations acting on the continuum, and $\otimes$ is the outer product. There are many discrete methods to solve the Navier-Stokes equation. We can use a classic method to solve the Navier-Stokes equation and generate a huge number of pure smoke images with RGBA channels by adopt volume rendering methods. Each pure smoke image has four channels, the RGB channels for a smoke color and an alpha channel for smoke density α.

To overcome the problem of generating a variety of smoke with different shapes, densities and colors, we use a third-party free 3D modeling software, Blender [19], to simulate and visualize smoke. Blender allows users to freely add wind, motion and gravity to greatly vary smoke appearance. We use high-resolution 3D grids to generate high-quality smoke images. To speed up the process, we use GPU computing to accelerate rendering process. Since each simulated smoke image contains RGB channels (s) and an alpha channel (α), we can use flowing equation to blend a pure smoke image (s and α) and a background image (b):

$$I(x) = b(x)(1 - \alpha(x)) + s(x)\alpha(x) \quad (5)$$

The above equation is just the linear color composition formula. To blending with background images, we apply (5) to red, green and blue channels, respectively.

IV. EXPERIMENTAL RESULTS

A. The Dataset

We created our own dataset by collecting images from the internet and rendering images. There are two parts in my dataset. The first part is a dataset for training backbone networks, such as VGG16 and Resnet-50. This part consists of more than 3000 image in total: 1700 smoke images and 1600 non-smoke images and is divided into training and testing sets. There are 1200 smoke images and 1200 non smoke images in training set. To avoid overfitting, we also use data augment techniques, such as affine transformation and gamma correction. The second part is a dataset for training smoke detection model. We use the method in Section III to synthesize about 1000 smoke images with RGBA channels, and the background images were randomly collected from the internet to suit indoor conditions. In total, the number of images in the second part is about 5000 images. We also used data augment techniques to avoid overfitting. Fig. 5 shows the simulated smoke patterns and the blended images. To test the solution with surveillance cameras, we recorded 5 videos in indoor environment in different conditions, including natural light, artificial light, dense smoke, sparse smoke.

B. Result

In this section, we present results that we achieved through experiments. We perform all experiments on a personal computer with CPU Intel Xeon (R) CPU E5-2620 v4 @ 2.10GHz, GPU GEFORCE GTX 2080ti, 8Gb of RAM and a embedded system which is named NVIDIA Jetson TX2 with CPU Dual-core Denver 2 64-bit CPU and quad-core ARM A57 complex, GPU 256-core NVIDIA Pascal architecture, 8Gb of RAM.

Table. I gives the video processing speed of the proposed methods, where subscripts 1 and 2 correspond to Gunner Farneback's optical flow algorithm and Lucas Kanade's optical flow algorithm respectively, S and F correspond to deep Convolution Neural Network models, SSD and Faster R-CNN respectively. When tested on PC, the highest speed is 95 frames per second and the lowest speed is 16 frames per second. SSD is faster than Faster R-CNN in same test cases. From Table. I, we find the method are able to reach real-time performance when tested on PC. When tested on the embedded system, the method get lower performance, especially the highest speed is 16 frames per second. Table II shows the accuracy of different models, including Faster R-CNN and SSD with different backbone networks. The point is that the the deep models achieve testing accuracy greater than 90%. Examples of the testing result are shown in Fig. 6. In conclusion, Faster R-CNN is more accurate than SSD when tested on the same dataset.

Fig. 5: Example of simulated images and blended images.

To quantitatively evaluate the experimental results of our method when tested on videos, we used three evaluation metrics: DR (detection rate), FAR (false alarm rate) and ER (error rate). The three metrics are defined separately as follows:

$$DR = \frac{TP}{P} \times 100\% \tag{6}$$

$$FAR = \frac{FP}{N} \times 100\% \tag{7}$$

$$ER = \frac{FN + FP}{N + P} \times 100\% \tag{8}$$

where TP, FP, FN, P and N are the numbers of positive frames detected correctly, negative frames detected incorrectly, positive frames detected incorrectly, total positive frames and total negative frames , respectively. Table. III gives the result of the proposed method when tested on the 5 recorded videos. For all smoke videos, the detection rate is more than 97%. In all test, the false alarm rate and error rate is less than 2%. False alarm may occur in complicated environment, such as low light, complex motion. Experiments show that the proposed method reach high accuracy when tested in indoor environment.

Overall, the proposed method performs well on our dataset. Results show that the proposed method achieves low false alarm rates while keeping the detection rate high. Experiments show that the proposed method has good discriminative ability for smoke detection in indoor environment.

TABLE I: Video processing speed of the proposed method (frames/s)

Video Resolution	PC				Embedded system			
	S, O_1	S, O_2	F, O_1	F, O_2	S, O_1	S, O_2	F, O_1	F, O_2
1280x720	82	48	19	16	5	3	1	0.8
640x480	90	65	20	18	7	5	2	0.5
320x240	95	82	24	22	8	6	2	0.5

TABLE II: Comparison between deep CNN models

Model	Training Accuracy(%)	Testing Accuracy(%)
Faster R-CNN VGG16	97.20	95.40
Faster R-CNN Resnet50	98.50	95.50
SSD VGG16	94.50	91.50
SSD Resnet50	95.30	92.50

TABLE III: Smoke detection result in video

Video sequences	DR(%)	FAR(%)	ER(%)
1	97.7	1.55	1.71
2	97.3	1.40	1.68
3	97.8	1.27	1.50
4	98.3	0.98	1.33
5	97.6	1.43	1.63

V. Conclusions And Future Works

In this work, we have proposed a new approach to detect smoke in indoor environment by using surveillance cameras. We test the proposed method on our dataset which is made specifically to replicate real world environment. The results prove the feasibility of this solution. Our future work will focus on finding the rationale in false-positive images to further improve the detection performance and optimize the algorithm for real-time performance within a low computational hardware platform.

Acknowledgment

This work is partly supported by the Ministry of Science and Technology (MoST) of Vietnam under grant number 01/2019/VSCCN-DTCB.

References

[1] A. Cetin, K. Dimitropoulos, B. Gouverneur, G. Nikos, O. Günay, Y. Habiboğlu, B. Töreyin, and S. Verstockt, "Video fire detection – review," *Digital Signal Processing*, vol. 23, p. 1827–1843, 12 2013.

[2] J. Chen, Y. Wang, Y. Tian, and T. Huang, "Wavelet based smoke detection method with rgb contrast-image and shape constrain," in *2013 Visual Communications and Image Processing (VCIP)*, 2013, pp. 1–6.

[3] Y. Chunyu, F. Jun, W. Jinjun, and Z. Yongming, "Video fire smoke detection using motion and color features," *Fire Technology*, vol. 46, pp. 651–663, 07 2010.

[4] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," 10 2016, pp. 877–882.

[5] J. Sharma, O.-C. Granmo, M. Goodwin, and J. Fidje, "Deep convolutional neural networks for fire detection in images," 08 2017, pp. 183–193.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 06 2015.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, "Ssd: Single shot multibox detector," vol. 9905, 10 2016, pp. 21–37.

[8] R. Donida Labati, A. Genovese, V. Piuri, and F. Scotti, "Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 1003–1012, 2013.

[9] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 08 1981.

[10] E. Memin and P. Perez, "Hierarchical estimation and segmentation of dense motion fields," *International Journal of Computer Vision*, vol. 46, pp. 129–155, 02 2002.

[11] G. Farneback, "Two-frame motion estimation based on polynomial expansion," vol. 2749, 06 2003, pp. 363–370.

[12] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 07 2018.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 12 2016.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 06 2016, pp. 779–788.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 06 2016, pp. 770–778.

[18] J. Stam, "Stable fluids," *ACM SIGGRAPH 99*, vol. 1999, 11 2001.

[19] "https://www.blender.org/."

Fig. 6: Example of result images.