Real-Time Video-Based Fire And Smoke Detection Using Machine Learning

Deepak Chaudhary
Dept. of Computer Science And Engineering (Artificial Intelligence & Machine Learning)

Greater Noida Institute Of Technology,Gautam Buddh Nagar, Uttar Pradesh, India
deepakchaudhary8303@gmail.com

Ravi Kumar Ray
Dept. of Computer Science And Engineering (Artificial Intelligence & Machine Learning)

Greater Noida Institute Of Technology,Gautam Buddh Nagar, Uttar Pradesh, India

Tanmay Raj
Dept. of Computer Science And Engineering (Artificial Intelligence & Machine Learning)

Greater Noida Institute Of Technology,Gautam Buddh Nagar, Uttar Pradesh, India

***Abstract*— Due to the wide range of smoke colour, brightness, and shape, video-based smoke detection is a difficult task. Handcrafting discriminative features for smoke detection is a complicated and expensive task, which cannot represent smoke characteristics accurately. There are some Convolutional Neural Network (CNN)-based smoke detection algorithms, but the majority of them are computationally expensive and difficult to achieve real-time detection. We present a novel data processing pipeline based on a deep learning algorithm in this research to increase the accuracy and efficiency of the smoke detection task. In the process, the Smoke Region Proposal is proposed to extract the suspected smoke regions, and the convolutional neural network model is pruned and reconstructed to fulfil the goal of real-time detection. The reconstructed CNN model is named SCCNN.**

Keywords*— *Smoke Detection, Convolutional Neural Networks, Fire Detection

I. Introduction

Consider the possibility of more effective early fire detection techniques, Would the Mandi Dabwali Fire Tragedy have been prevented? Consider how many more How many lives and properties could be saved if wildfires were detected and stopped early? Could families be saved? In fact, 5 early fire detection has been used for decades. Human society's urgent need has already been identified. Video-based smoke detection has recently gained popularity due to its advantages of immediate response, non-contact detection, and large detection range coverage at a low cost, when compared to traditional methods such as ionised sensor, which is typically restricted to indoor scenes and necessitates close proximity to a fire or smoke. Nonetheless, video-based smoke Due to the wide range of smoke colour, detection is a difficult task texture as well as shapes. Moreover, there are lots of disturbances from nature including sky, haze, gray clothes, white moving objects, etc.

According to research from the last decade, smoke detection algorithms based on video sequences can be divided into two categories. The traditional machine learning method falls into the first category. These methods are primarily concerned with developing a feature extractor that converts raw data into a suitable internal representation or feature vector. These are known as traditional features or handcrafted features. Handcrafting discriminative features is a difficult and costly task that cannot accurately represent smoke characteristics and is susceptible to various disturbances. Handcrafted features are frequently effective for only one scene and have poor robustness. The key factor influencing the accuracy of these methods is how to extract robust smoke features. The second category involves the use of deep learning algorithms such as Convolutional Neural Networks (CNNs), which can automatically extract robust features from raw data and jointly optimise the feature representation and classifier to achieve the best model results.

In recent years, there has been a surge in interest in the detection of fire smoke using surveillance cameras equipped with machine vision. The image processing method entails the Using frame difference technologies, the smoke plume was extracted from the background. Color processing outperforms gray-scale processing in the segmentation of fire features. Color processing can avoid false alarms caused by variations in lighting conditions, such as natural background illumination, better than gray-scale processing. In addition, a video camera is a volume sensor that can potentially monitor a larger area. The traditional point sensor looks at a point in space. Since the point sensor may not be affected by smoke, fire would be undetected. However, vision-based smoke detection still has great technical challenges, since fire smoke are non-rigid objects, none of the primitive image features and variability in density, lighting, etc. Smoke is a hydrogen, carbon, and oxygen compound. The chemical components of the combustible material, the burning temperature, the supply of oxygen, and other factors influence the constituents and quantity of smoke. The visual pattern of smoke is difficult to model, and the density of smoke varies with the environment. Smoke from an uncontrolled fire, on the other hand, can be easily observed even if the flames are not visible, and the fire can be detected early on before it spreads.

The proposed real-time fire smoke detection technique employs spectral features (i.e., saturation and chromatic features), spatial features (i.e., disordered features), and temporal features (i.e., flicking feature) for extracting fire smoke pixels.

The chromatic feature of smoke is used to detect fires first. In terms of the general colour of fire smoke, the proposed system recognises that the chromatic feature of fire smoke can dominate the decision function of fire smoke pixels.

Regarding the spatial characteristics of fire smoke, the fire smoke will flicker abruptly, implying that the fire smoke shape is dynamically changeable in visual images. As a result, the extracted fire smoke pixels will be validated further by examining the disordered characteristics of the burning fire. The fuzzy reasoning system weights the statistical distribution of the chromatic and disordered measurement value to give the potential fire smoke candidate region. The temporal probability density is applied by extracting the flickering area with level crossing to separate the alias objects with the fire smoke region. Then, the continuously adaptive mean shift (CAMSHIFT) vision-tracking algorithm is employed to provide feedback of the fire smoke real-time position.

The main differences and advantages of the proposed approach are as follows:

It combines the spectral, spatial, and temporal fire smoke characteristics to perform the machine vision-based fire early detection and tracks the fire smoke region based on the CAMSHIFT algorithm.

The detection and tracking algorithm is used to process the image sequences and recognise the fire smokes at 20[ms] or lower video rates. After detecting fire smoke, only the regions surrounding the current tracking window must be processed, resulting in a significant reduction in computational costs.

Not only does the proposed fire smoke detection algorithm have a high correct decision rate and a better ability to avoid false alarms caused by environmental illumination, but it can also track the fire smoke regions from the image much more precisely than the other systems.

Not only does the proposed fire smoke detection algorithm have a high correct decision rate and a better ability to avoid false alarms caused by environmental illumination, but it can also track the fire smoke regions from the image much more precisely than the other systems.

This paper is organized as follows. The overall fire smoke detecting algorithm structure is introduced and analyzed in Section II. Then some experimental results are given in Section III. Finally, the conclusions are given in Section IV.

II. Fire Smoke Detection Method

A. Convolutional Neural Networks

CNN is a feed forward neural network that is generally used to analyze visual images by processing data with grid like topology. A CNN is also known as "Convonet".

The convolutional neural network, or CNN for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and threedimensional data. Central to the convolutional neural network is the convolutional layer that gives the network its name. This layer performs an operation called a "convolution". In the context of a convolutional neural network, a convolution is a linear operation that involves the multiplication of a set of weights with the input, much like a traditional neural network. Given that the technique was designed for two-dimensional input, the multiplication is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel. The filter is smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a dot product. A dot product is the element-wise multiplication between the filtersized patch of the input and filter, which is then summed, always resulting in a single value. Because it results in a single value, the operation is often referred to as the "scalar product".

Using a filter smaller than the input is intentional as it allows the same filter (set of weights) to be multiplied by the input array multiple times at different points on the input. Specifically, the filter is applied systematically to each overlapping part or filter-sized patch of the input data, left to right, top to bottom. This systematic application of the same filter across an image is a powerful idea. If the filter is designed to detect a specific type of feature in the input, then the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image. This capability is commonly referred to as translation invariance, e.g. the general interest in whether the feature is present rather than where it was present.

The output from multiplying the filter with the input array one time is a single value. As the filter is applied multiple times to the input array, the result is a two-dimensional array of output values that represent a filtering of the input. As such, the two-dimensional output array from this operation is called a "feature map". Once a feature map is created, we can pass each value in the feature map through a nonlinearity, such as a ReLU, much like we do for the outputs of a fully connected layer.

If you come from a digital signal processing field or related area of mathematics, you may understand the convolution operation on a matrix as something different. Specifically, the filter (kernel) is flipped prior to being applied to the input. Technically, the convolution as described in the use of convolutional neural networks is actually a "crosscorrelation". Nevertheless, in deep learning, it is referred to as a "convolution" operation. In summary, we have a input, such as an image of pixel values, and we have a filter, which is a set of weights, and the filter is systematically applied to the input data to create a feature map.

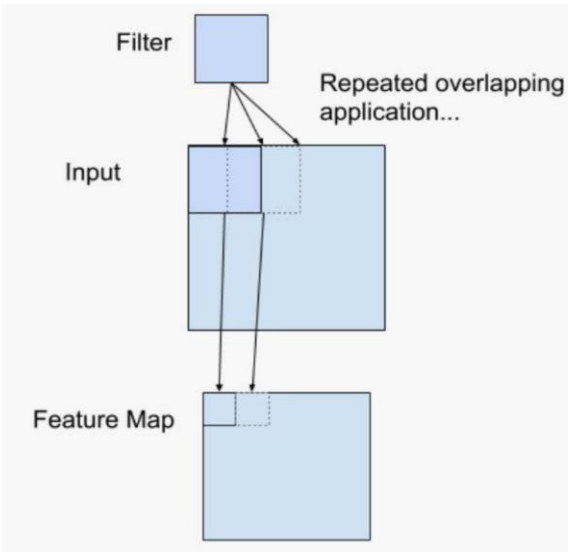
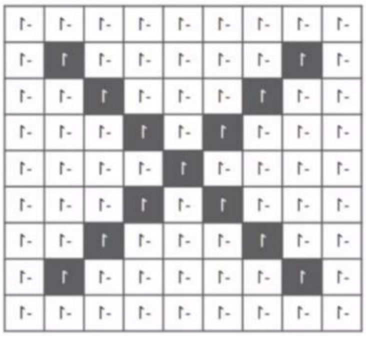


Fig. 1. Example of a Filter Applied to a Two-Dimensional Input to Create a Feature Map

Convolutional Neural Networks have following layers:

- Convolution
- ReLU Layer
- Pooling
- Fully Connected

A computer understands an image using numbers at each pixels. In our example, we have considered that a black pixel will have value 1 and a white pixel will have -1 value.



Input

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

Output

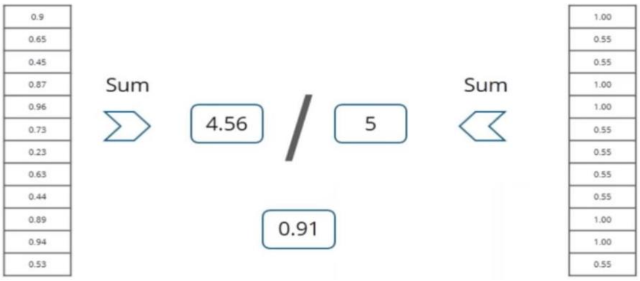
- Fully Connected Layer

This is the final layer where the actual classification happens Here we take our filtered and shrinked images and put them into a single list.

Comparing the input vector with existing vector:

When we feed in, input vector and existing vector. Then there will be some element in the vector that will be high.

Consider the image below, as you can see for input vector there are different element that are high and similarly, for existing vector we have different elements that are high.



Input vector Existing vector

So, the input vector is classified as existing vector.

Table 1

Structure hyper-parameters of SCCNN.

Layer	Layer Type	Hyper-parameters
Input		Image size:32 × 32
C1	Convolution	Filter size:3 × 3 Filter number:32 Stride:1 × 1 Activation function: ReLU
NC2	Normalization and convolution	Filter size:3 × 3 Filter number:64 Stride:1 × 1 Activation function: ReLU
P3	Pooling	Pooling region size:2 × 2 Stride:2 × 2 Pooling method: max-pooling
NC4	Normalization and convolution	Filter size:3 × 3 Filter number:128 Stride:1 × 1 Activation function: ReLU
NC5	Normalization and convolution	Filter size:3 × 3 Filter number:256 Stride:1 × 1 Activation function: ReLU
P6	Pooling	Pooling region size:2 × 2 Stride:2 × 2 Pooling method: max-pooling
NC7	Normalization and convolution	Filter size:3 × 3 Filter number:128 Stride:1 × 1 Activation function: ReLU
P8	Pooling	Pooling region size:2 × 2 Stride:2 × 2 Pooling method: max-pooling
FL9	Fully-connected	Neurons number:512 Dropout probability:0.5 Activation function: ReLU
FL10	Linear layer	Neurons number:2 Dropout probability:0.5 Activation function: None

B. CAMSHIFT Tracking Algorithm

The CAMSHIFT method is a non-parametric technique for efficiently tracking a target's 2D position through a sequence of photos. The CAMSHIFT tracking engine is based on the histogram projection algorithm, which is an effective technique for colour object recognition, particularly in complex background environments. Histogram back-projection is a simple procedure that finds and determines the relationship between pixel values in a captured image and values in a specific histogram bin. Histogram and back-projection operations performed on any consecutive frame would result in a probability image in which the value of each pixel represents the probability of the exact same pixel from the input belonging to the target histogram that was employed. Given that m histogram bins are used, we can define n image pixel locations. Thus, histograms {ŷu},u=1,…,m and pixel locations {xi},i=1,…n can be calculated. Let us also define a function c:R 2→{1,…,m} that associates a pixel at location xi * with a histogram bin index c(xi *). Then, the histograms can be computed in the equation.

$$\sum_{i=1}^{n} \delta\left[c\left(x_i^*\right)-u\right],$$

Where δ is the Kronecker delta function. In all cases, values in the histogram bin are rescaled to fit within the discrete pixel range of the possible output 2D probability distribution image with the function

$$\left\{\widehat{p}_u = \min\left(\frac{UPPER}{\max\left(\hat{y}\right)}\hat{y}_u, UPPER\right)\right\}_{u=1\ldots m}.$$

That is, values in the histogram bin, originally in the range [m, max(ŷu)], now lie in the new range [0, UPPER]. In the end, the input pixels with the highest probability of being in the sample histogram will be mapped onto a 2D histogram back-projection image with the highest visible intensities.

The CAMSHIFT method is a non-parametric technique for efficiently tracking a target's 2D position through a sequence of photos. The CAMSHIFT uses a colour probability distribution image obtained from colour histograms to monitor the 2D position of a coloured item, in this case a fire smoke coloured region. The centre and size of the targeted object region are calculated and utilised to define the search window on the next frame of the video sequence. The images of the tracked object in the digital photos that have been detected and processed are shown in Fig. 2, and the yellow bounding ellipse represents the smoke region tracked by the CAMSHIFT method.

C. Real-Time Detection

Using the CAMSHIFT algorithm, the tracking procedures may be completed in 20[ms] video rates. The colour probability distribution is calculated on a small region surrounding the current CAMSHIFT window, which includes photos of the specified item that have been turned into a discrete probability image. This usually results in a significant reduction in computational expenses. Multimedia timer functionality and PC-based real-time control are used, as well as particular C++ software modules.

III. ANALYSIS AND COMPARISON

This part presents an analysis of experiments applying the proposed process derived in previous sections. A standard digital colour video camera is employed to collect various fire smoke sample image sequences with pixel resolutions of 320×240 to achieve the suggested fire smoke detection system. On an Intel Pentium computer running at 2.0 GHz, the system performs at a rate of 11-60 frames per second. Fuels were burned in various settings and the previously defined decision rules on spectral, spatial, and temporal variables were used for the full evaluation.

Fig. 2. Example frames of experimental test data.

Fig 3 and Table II illustrate the results of applying the suggested fire smoke detection system in various lighting settings (Movies 1-6). Table III contains tabulated smoke detection findings, where nt is the number of frames in a video clip. The parameter f - represents the number of false negative fire smokes, which means that the system does not detect fire smokes in an image frame even when it contains fire smoke frames. Similarly, the field f+ is the number of false positive fire smoke, which means that the system recognizes fire smoke in an image frame while there is no fire smoke at all. The detection rate, rd , of a video is defined as the ratio

TABLE II
PROPERTIES OF THE TEST VIDEOS

Video Sequences	*Video description*
Movie 1	Burning the paper.
Movie 2	Smoking in front of a grayish concrete wall.
Movie 3	Fire in a garden and lighting a lamp to confuse.
Movie 4	A moving hand and walking in a room to confuse.
Movie 5	Burning paper at a barbecue site.
Movie 6	Smoking at a barbecue site.

where nd is the number of true positives and is the rate of accurately detecting a real fire smoke in a video clip as a fire smoke. It is apparent that the fire smoke can be separated appropriately and an appropriate alarm can be issued. Figure 4 depicts the results of fire smoke detection using the suggested approach in various test movies. The average true positive detection rate of the proposed fire smoke detection is shown in Table III.

TABLE III
RESULTS OF PROPOSED FIRE SMOKE DETECTING ALGORITHM

Video Sequences	n_t	f_+	f_-	r_d (%)
Movie 1	1844	85	460	70.4
Movie 2	1295	0	929	28.3
Movie 3	688	4	83	87.4
Movie 4	2956	0	0	100
Movie 5	484	0	39	91.9
Movie 6	1699	0	446	73.7
Total	8966	89	1957	77.2

In Table III, the average true positive detection rate of the proposed fire smoke detection is 77.2%. The fire smoke detection rate falls short in the case of movie 2 due to the fact that airflow caused by wind causes the fire smoke to move randomly.

IV. CONCLUSIONS

In conclusion, our real-time video-based fire smoke detection system demonstrated excellent performance in detecting the presence of fire and smoke in real-time from video feeds. This system has the potential to provide more timely and accurate detection of fires and smoke, as well as the ability to identify the location and severity of the fire. Further research is needed to evaluate the system's performance in real-world settings, and to optimize its performance for different types of buildings and environments.

This paper presents a method for detecting fire smoke in real-time alarm systems. Fuzzy logic and spectral, spatial, and temporal features are used to extract genuine fire smoke and are used to aid in the validation of that fire smoke. The experimental results show that fire smoke can be detected successfully under a variety of environmental conditions, such as indoor, outdoor, day, simple, or complex background image, and so on. The proposed algorithm can be integrated into existing surveillance systems to detect fire smoke in video databases as well as detect fire in real time.

Fig. 3. Test videos which contain real fire smoke detection results using the proposed method.

REFERENCES

[1] J. Fang, J. Jie, Y. Hong-Yong, and Z. Yong-Ming, "Early fire smoke movements and detection in high large volume spaces," Building and Environment, vol. 41, pp. 1482–1493, 2006.

[2] J. Davis, "Recognizing movement using motion histograms," Technical Report 487, MIT Media Lab, 1999.

[3] Fire detection - Wikipedia