Case Studies in Thermal Engineering 19 (2020) 100625

Contents lists available at ScienceDirect

Case Studies in Thermal Engineering

journal homepage: http://www.elsevier.com/locate/csite

# Image fire detection algorithms based on convolutional neural networks

Pu Li [a,b,*], Wangda Zhao [a]

[a] School of Civil Engineering, Central South University, Changsha, 410075, Hunan, China
[b] Zhengzhou Airport Economy Zone Fire Brigade, Zhengzhou, 450000, China

ARTICLE INFO

*Keywords:*
Image
Fire detection
Deep learning
Convolutional neural network

ABSTRACT

As a new fire detection technology, image fire detection has recently played a crucial role in reducing fire losses by alarming users early through early fire detection. Image fire detection is based on an algorithmic analysis of images. However, there is a lower accuracy, delayed detection, and a large amount of computation in common detection algorithms, including manually and machine automatically extracting image features. Therefore, novel image fire detection algorithms based on the advanced object detection CNN models of Faster-RCNN, R–FCN, SSD, and YOLO v3 are proposed in this paper. A comparison of the proposed and current algorithms reveals that the accuracy of fire detection algorithms based on object detection CNNs is higher than other algorithms. Especially, the average precision of the algorithm based on YOLO v3 reaches to 83.7%, which is higher than the other proposed algorithms. Besides, the YOLO v3 also has stronger robustness of detection performance, and its detection speed reaches 28 FPS, thereby satisfying the requirements of real-time detection.

## 1. Introduction

With rapid economic development, the increasing scale and complexity of constructions has introduced great challenges in fire control. Therefore, early fire detection and alarm with high sensitivity and accuracy is essential to reduce fire losses. However, traditional fire detection technologies, like smoke and heat detectors, are not suitable for large spaces, complex buildings, or spaces with many disturbances. Due to the limitations of above detection technologies, missed detections, false alarms, detection delays and other problems often occur, making it even more difficult to achieve early fire warnings.

Recently, image fire detection has become a hot topic of research. The technique has many advantages such as early fire detection, high accuracy, flexible system installation, and the capability to effectively detect fires in large spaces and complex building structures [1]. It processes image data from a camera by algorithms to determine the presence of a fire or fire risk in images. Therefore, the detection algorithm is the core of this technology, directly determining the performance of the image fire detector.

There are three main stages in the process of image fire detection algorithms, including image preprocessing, feature extraction, and fire detection. Among, feature extraction is the core part in algorithms. Traditional algorithm depends on the manual selection of fire feature and machine learning classification. The shortcoming of algorithms is that manual feature selection should depend on professional knowledge. Though the researchers develop many studies in image features of smoke and flame, only simple image features, such as color, edges and simple textures, are discovered. However, because of complex fire types and scenes as well as many

* Corresponding author. School of Civil Engineering, Central South University, Changsha, 410075, Hunan, China.
*E-mail address:* lipu_861202@sina.com (P. Li).

https://doi.org/10.1016/j.csite.2020.100625
Received 23 February 2020; Received in revised form 5 March 2020; Accepted 9 March 2020
Available online 10 March 2020
2214-157X/

interference events in practical application, the algorithms extracting low and middle complex image features are difficult to distinguish fire and fire-like, thereby causing lower accuracy and weak generalization ability.

Image recognition algorithms based on convolutional neural networks (CNNs) can automatically learn and extract complex image features effectively. This kind of algorithms has attracted great concerns and achieved excellent performance on visual search, automatic driving, medical diagnosis, etc. Therefore, some scholars introduce CNNs into the field of image fire detection, thereby developing the self-learned algorithm in collection of fire image features [1–9]. modified the state-of-the-art models of AlexNet, VGG, Inception, ResNet, etc., and developed smoke and flame detection algorithms [7,9]. Introduced the time-series information into algorithms [10]. Mentioned that VGG-Net was reformed to develop an algorithm in detecting smoke and flame, simultaneously [11]. Revealed the feasibility of five networks, including GoogleNet, Modified GoogleNet, etc., used in the fire detection of forest via unmanned aerial vehicle (UAV) [12]. Proposed a moving object detection method to generate proposal regions based on background dynamic update and dark channel priori. Subsequently, the algorithm based on CNNs was used to detect smoke in proposal regions [13]. Reported that the cluster of Mixture of Gaussian Background Modeling (MOG) was used to discriminate background and foreground, subsequently used a cascade model to identify the smoke regions. Finally, the network of CaffeNet was carried out to detect the smoke in proposal regions [14]. Mentioned that color features were used to region proposals, subsequently used the modified AlexNet to detect the flame in proposal regions. Table 1 lists the full name of the CNNs mentioned in this paper.

Though fire detection algorithms based on CNNs have more promotion in the detection accuracy in complex scenes than traditional algorithms, some problems still exist. First, current algorithms based on machine learning mostly considered image fire detection as a classification task, and the region proposal stage was ignored. The algorithms classify the entire image to one class. However, in the early stage of fire, smoke and flame only covered a small area of the image. If the feature of smoke and flame is not obvious, use of the entire image feature without region proposals would decrease the accuracy of detection and delay fire detection and alarm activation. Therefore, proposal regions should be determined before the image classification to improve the ability of algorithm in detecting early fire. Second, some scholars designed the algorithms generating proposal regions by manually selecting features and classifying proposal regions by CNNs. This kind of algorithm, generating the proposal regions through computing individually, does not use CNNs to the global process of detection, thus leading to a large amount of computation and slow detection speed.

This study reveals four types of advanced convolutional neural networks applied in image fire detection. The image fire detection algorithms are developed and trained by the self-built fire image dataset. Finally, optimum detection performance in the four proposed algorithms is determined. The results of the study can provide useful information for modification of detection algorithms for preventing fire accidents.

## 2. The proposed framework

### 2.1. Convolutional neural network architectures

According to the principle of object detection algorithms, the flow of image fire detection algorithms based on convolutional neural networks is designed in Fig. 1. The detection CNN has functions of region proposals, feature extraction and classification. Firstly, The CNN takes an image as input and outputs region proposals by convolution, pooling, etc. Secondly, the region-based object detection CNN decides the presence or absence of fire in proposal regions through convolutional layers, pooling layers, fully-connected layers, etc.

The convolutional layer is the core part of CNNs. Different from other neural networks using connection weights and weighted sums, the convolutional layer uses image transform filters called convolution kernel to generate feature maps of original images. The convolutional layer is a set of convolution kernels. The convolution kernel slides on the images and computes a new pixel by a weighted sum of the pixels it floats over to generate a feature map. The feature map reflects the features of an aspect for the original image. Equation (1) is the calculation formula of the convolution layer.

$$y=\sum_{j=0}^{J-1}\sum_{i=0}^{I-1}w_{ij}x_{m+1,n+j}+b,(0\leq m\leq M,0\leq n\leq N) \tag{1}$$

**Table 1**
Full name of convolutional neural networks.

| Name | Full name of convolutional neural network |
|---|---|
| AlexNet | AlexNet |
| VGG | Very Deep Convolutional Networks for Large-Scale Image Recognition |
| Inception | Inception |
| ResNet | Residual Network |
| Inception Resnet V2 | Inception Residual Version 2 |
| Darknet-53 | Darknet-53 |
| Faster-RCNN | Faster Regions with CNN features |
| R–FCN | Region-based Fully Convolutional Network |
| SSD | Single Shot MultiBox Detector |
| YOLO v3 | You Only Look Once Version 3 |

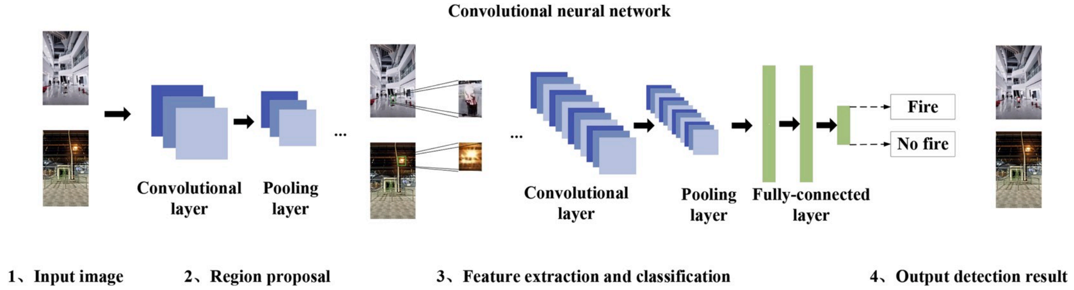


**Fig. 1.** Flow chart of image fire detection algorithms based on detection CNNs.

where $x$ denotes an input image of a size $W \times H$ and $w$ denotes a convolution kernel of a size $J \times I$ and $b$ denotes bias and $y$ denotes output feature maps. In practice, the value of $w$ and $b$ is determined through training.

Fig. 2 shows the 32 kernels of the first convolutional layer in Inception Resnet V2 (a state-of-the-art CNN) and the 32 feature maps of a fire image generated by these kernels. The number of feature maps and convolution kernels is equal. For example, there are three convolution kernels in this layer, so three feature maps are generated. The color of pixels represents the degree of activations. White pixels at some location in the feature map indicate the pixels are strongly positively activated at the same position in the original image. Black pixels indicate strongly negative activations. Gray pixels represent not strong activations. Comparing to the original image, the feature map generated by the convolutional kernel 14 of this layer is activated on edges. The dark/light areas are activated positively on upper/lower edges, and light/dark areas are activated negatively on upper/lower edges. The feature map generated by the convolutional kernel 26 is activated on orange pixels, because the whiter pixels in the map correspond to orange areas in the original image. It indicates that the kernels in earlier layers mainly learn and extract simple features like color, edges, etc. Through analyzing these feature maps, it is discovered that simple features cannot distinguish fire and disturbance, when scenes are complex as well as many interference events. Therefore, it is necessary to develop image fire detection algorithms that can extract complex image features for fire detection in real scenarios. Deep convolutional neural networks are better at this aspect. Fig. 3 shows samples of kernels in the first, third and sixth convolutional layer of Inception Resnet V2. It indicates that the networks extract more complex features in later layers. Therefore, extracting complex image features need a deep network. This paper selects Inception Resnet V2 [15] and Darknet-53 [16] as feature extraction networks, having 235 and 53 convolutional layers.

Four image object detection networks (Faster-RCNN, R–FCN, SSD and YOLO v3) with excellent performance in detection accuracy and speed [16,17] are selected to develop image fire detection algorithms. Fig. 4 shows the architectures of the four algorithms.

### 2.1.1. *Faster-RCNN*

There are two stages in Faster-RCNN setting (Fig. 4. (a)). In the first stage, feature maps of the original image are generated by feature extraction networks (VGG , ResNet , Inception , Inception Resnet -v2, etc.). And the feature map from some selected intermediate convolutional layer is used to predict proposal regions with objectness scores and locations by Region Proposal Network (RPN). This stage just output scores that estimate the probability of object or not object and box regression for each proposal by a two-class softmax layer and the robust loss function (Smooth L1). In the second stage, the locations of the proposal regions are used to crop features from the same intermediate feature map by ROI pooling. And the regional feature map for each proposal region is fed to the remainder of the network to predict class-specific scores and refine box locations. This network achieves sharing part of the computation by cropping proposals from the feature map generated by the same intermediate convolutional layer in the first stage. This method avoids inputting each proposal region into the front-end CNN to compute the regional feature map. However, per proposal region must be input into the remainder of the network to compute separately. Thus, the detection speed depends on the number of proposal regions from the RPN.

### 2.1.2. *R–FCN*

It is proved that CNNs should be deep enough to extract more complex features, which can increase the accuracy of the networks. However, the small-scale feature maps generated by deeper convolutional layers is less sensitive to translation, leading to inaccurate localization for the object detection task. Therefore, Faster-RCNN adds a region proposal network (RPN) in parallel to some intermediate convolutional layer to save location information of objects. But the regional feature maps generated by standard ROI pooling

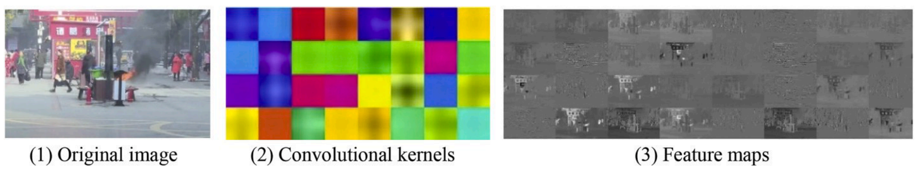


**Fig. 2.** Features extracted by the first convolutional layer.

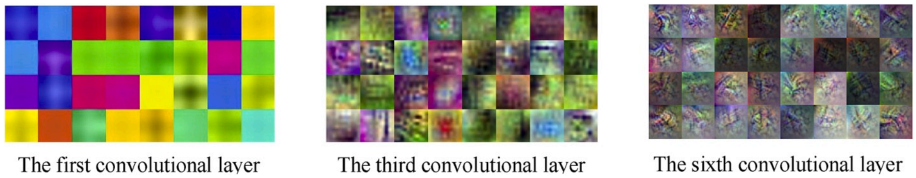


**Fig. 3.** Samples of kernels in some convolutional layers.

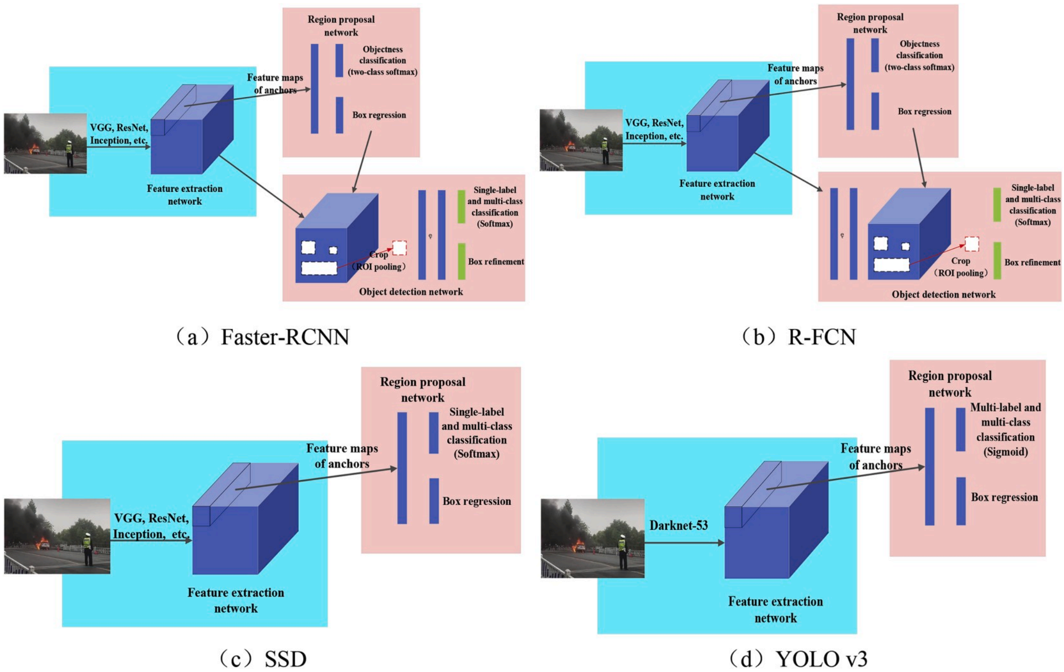


**Fig. 4.** Diagrams of fire detection algorithms based on the four CNNs.

operation don't have translation variance. Thus, per regional feature map should be input into the object detection network to predict. This amount of unshared per ROI computation lead to a lower speed. Therefore, R–FCN (Fig. 4. (b)) does not ROI pooling the feature maps from the same intermediate convolutional layer where region proposals are predicted. It selects the last layer of feature maps before prediction to ROI pool, which minimizes the amount of computation. To increase the sensitivity to translation of small-scale feature maps from deep layers, R–FCN proposes a Position-sensitive ROI pooling operation to encode position information into per ROI.

### *2.1.3. SSD*

Faster-RCNN and R–FCN, including a region proposal network, belong to the two-stage object detection networks. The detection speed of this kind of network is slower. Therefore, the one-stage object detection network, like SDD and YOLO v3, is proposed. It predicts the object class and location by a single forward CNN. SSD (Fig. 4. (c)) is mainly divided into three parts: (1) The basic convolutional layers consist of the feature extraction network like VGG, ResNet, Inception, Inception Resnet -v2, etc. The intermediate convolutional layer of this part generates a large-scale feature map, which could be divided into more cells and has smaller size of receptive fields, to detect smaller objects. (2) The additional convolutional layers connect to the last layer of the basic convolutional network. This part of layers generates multi-scale feature maps having larger size of receptive fields for larger object detection. (3) The prediction convolutional layers using a small convolutional kernel predict bounding box locations and confidences for multiple categories.

### *2.1.4. YOLO v3*

To keep translation variance, SSD selects the earlier layers to generate large-scale feature maps, which is used to detect small objects. However, the features in these maps from the earlier layers are not complex enough, which leads to worse performance on smaller objects.

To solve aforementioned problems, YOLO v3 (Fig. 4. (d)) improves the accuracy of detection objects by referring to the idea of residual network. And its one-stage strategy performs excellently on detection speed. Fig. 5 shows the details of the architecture of

YOLO v3. It uses Darknet-53 without the last three layers to generate a small-scale feature map, which is 32 times down sampled from the original image. For example, if the size of the original image is 416 × 416, the size of the feature map is 13 × 13. The small-scale feature map is used to detect large objects. Different from SDD selecting the earlier layers to generate large-scale feature maps, YOLO v3 generates a large-scale feature map by up sampling the small-scale feature map and concatenating with a feature map from an earlier layer. This large-scale feature map having the location information of the earlier layers and complex features of deeper layers is used to detect small objects. The three scales of feature maps are 8, 16, and 32 times down sampled from the original image.

In Fig. 5., N in ResN denotes there are N units of Res Unit connecting in series. Concat denotes concatenation operation, which is different from the Add operation in residual layers. Concatenation expands the dimensions of feature maps, while Add operation just adds them without changing the dimensions.

Instead of using a softmax to predict single-label classification, YOLO v3 uses independent sigmoid functions to predict multilabel classification for per bounding box. That is, per bounding box could belong to multiple categories like smoke and fire. This design is useful for detecting the regions where smoke and fire appear simultaneously.

To meet the needs of fire detection, the end of networks is modified by changing the number of detection object categories to two (fire and smoke). Faster-RCNN、R–FCN and SSD use the Inception Resnet -v2 as the feature extraction network. YOLO v3 uses its own DarkNet-53.

### 2.2. *Training algorithm*

#### 2.2.1. *Fire image dataset*

Training the algorithms based on CNNs needs a large number of data. However, current small-scale image/video fire databases cannot meet the needs. Some small-scale fire image/video databases are listed in Table 2. Therefore, in this paper, 29,180 images are collected from small public fire image/video databases, large public images/video data sets, previous experimental data from research institutions, and the Internet. And the structure of the fire image dataset is shown in Fig. 6, using PASCAL visual object classes dataset [18] as the template. Each annotation file of the image provides three types of information (general information, fire information, and background information). Details of annotations are listed in Table 3.

Images in the data set obtained from different sources vary in size. The horizontal images have a size of approximately 500 × 375 pixels, and the longitudinal images have a size of approximately 375 × 500 pixels. Each image is annotated with a bounding box for two object classes (“fire” and “smoke”) and two disturbance classes (“fire-like” and “smoke-like”). Fig. 7 shows some samples of images in the dataset.

Table 4 summarizes the statistics of fire images in the dataset. The number of fire images in the data set is 13,400. These images were captured in outdoor and indoor conditions. The data set contains 9695 “fire” and 7442 “smoke” objects.

Moreover, there are 15,780 images containing no fire in the dataset. These images were collected from sixteen types of application scenarios and contain 49,614 disturbances. Each image contains at least one disturbance.

For this study, 50% of images in the dataset are selected as the training/validation set, and the other 50% images are used as the test set.

#### 2.2.2. *Transfer learning*

Considering the four CNNs have been trained on the large-scale image dataset (COCO) and proved to be excellent in image object detection. This paper transfers the pre-trained networks on COCO. The transfer learning strategy is freezing the front end of the feature extraction network and just fine-tuning the reminder of the networks on the training/validation set in 2.2.1.

The machine with an Intel Core I7-7700 CPU @ 3.6 GHZ, 16 GB DDR4 RAM 2400 MHz，NVIDIA Titan X Pascal GPU with 3840 CUDA is used to train all the networks. The operating system is Ubuntu 16.0.4.

The asynchronous SGD with momentum is set to 0.9. IOU threshold is 0.6. The initial learning rate is 0.001. The learning rate is reduced by 10x after 120 K iterations and another 10x after 160 K iterations. The total number of iterations is 200 K. Other parameters

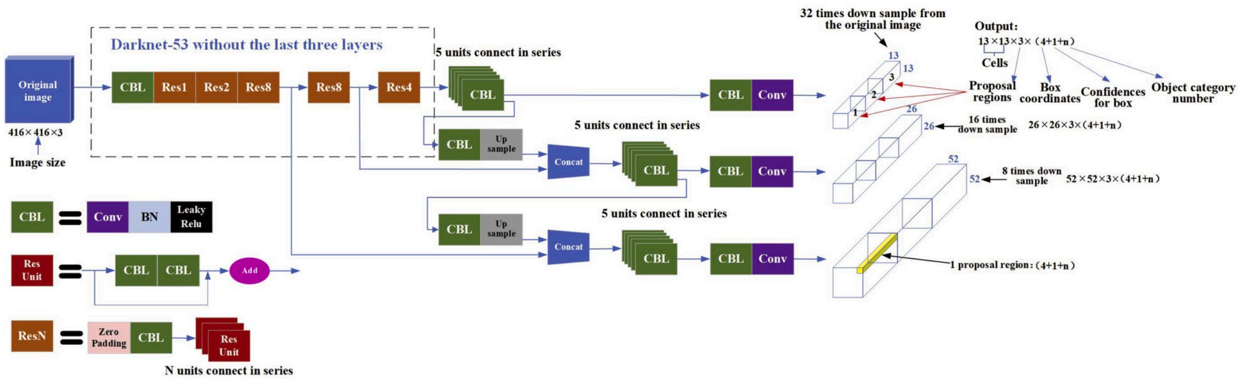


**Fig. 5.** Architecture of YOLO v3.

**Table 2**
Small-scale fire image/video databases.

| Institutions | Format | Object | Web site |
|---|---|---|---|
| Bilkent University | Video | fire, smoke, disturbance | http://signal.ee.bilkent.edu.tr/VisiFire/index.html |
| CVPR Lab. at Keimyung University | Video | fire, smoke, disturbance | https://cvpr.kmu.ac.kr |
| UMR CNRS 6134 SPE， Corsica University | Dataset | fire | http://cfdb.univ-corse.fr/index.php?menu=1 |
| Faculty of Electrical Engineering， Split University | Image, Video | smoke | http://wildfire.fesb.hr/ |
| Institute of microelectronics, Seville, Spain | Image, Video | smoke | http://www2.imse-cnm.csic.es/vmote/english_version/ |
| National Fire Research Laboratory, NIST | Video | fire | https://www.nist.gov/topics/fire |
| State Key Laboratory of Fire Science, University of Science and Technology of China | Image, Video | smoke | http://smoke.ustc.edu.cn/datasets.htm |

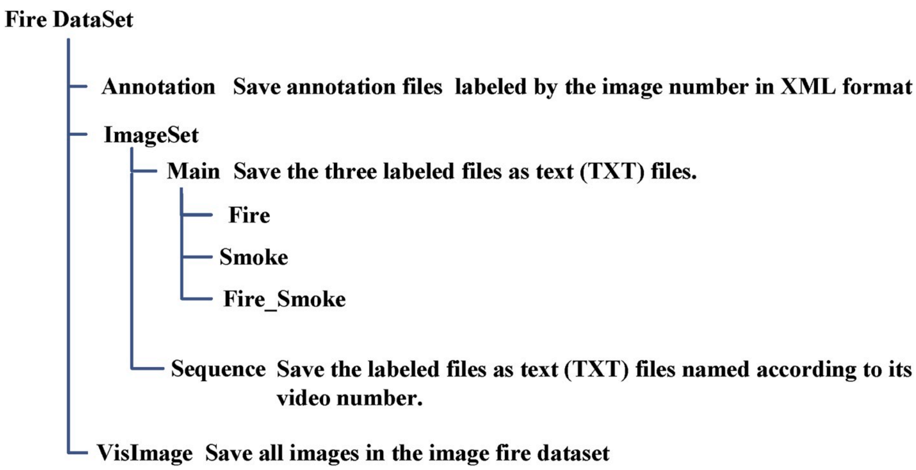


**Fig. 6.** Structure of the fire image dataset.

**Table 3**
Details of annotations.

| Field | Format/Possible value | Description |
|---|---|---|
| General information | | |
| <filename> | TXT | Image number. |
| <source> | TXT | Consists of three fields (<database>, <annotation>, and <image>) denoting the data set name, name of the research institution that provided the annotation, and source of the image, respectively. |
| <size> | number | The <width>, <height>, and <depth> field denote the width, height, and channel number of the image, respectively. |
| <sequence> | 0/1 | Indicates whether it is one of image in videos; “0” indicates it is not, and “1” indicates it is one of image in videos, which is labeled “1" in one of label files in sequence folder |
| Fire information | | |
| <name> | fire/smoke | The class of the object. |
| <truncated> | 0/1 | Indicates whether the object is truncated; “0” indicates no truncation, and “1” indicates truncation. |
| <bndbox> | number | Indicates the position of the object in the image according to an axis-aligned bounding box reflecting the extent of the object’s visibility in the image. |
| <occupation> | [0,100] | The percentage of the object bounding box in the image. |
| Background information | | |
| <environment> | TXT | Describes the environment in terms of presence of ambient light and disturbances, etc. In the background. |
| <luminosity> | [0,100] | Provides information regarding the amount of ambient light present in the image, which is computed using the average of channel V of HSV color space in the background region. |
| <difficult> | [0,1] | Indicates the image complexity, which is computed using the comprehensive image complexity metric based on the Inception Resnet -v2 predictor (in Session 3). |
| <disturbance> | text | Comprises three fields: (1) The <class > field indicates the class of the disturbance (fire-like or smoke-like). (2) The < description > field describes the disturbance in TXT format. (3) The < bndbox> field indicates the position of the disturbance in the image according to an axis-aligned bounding box. |

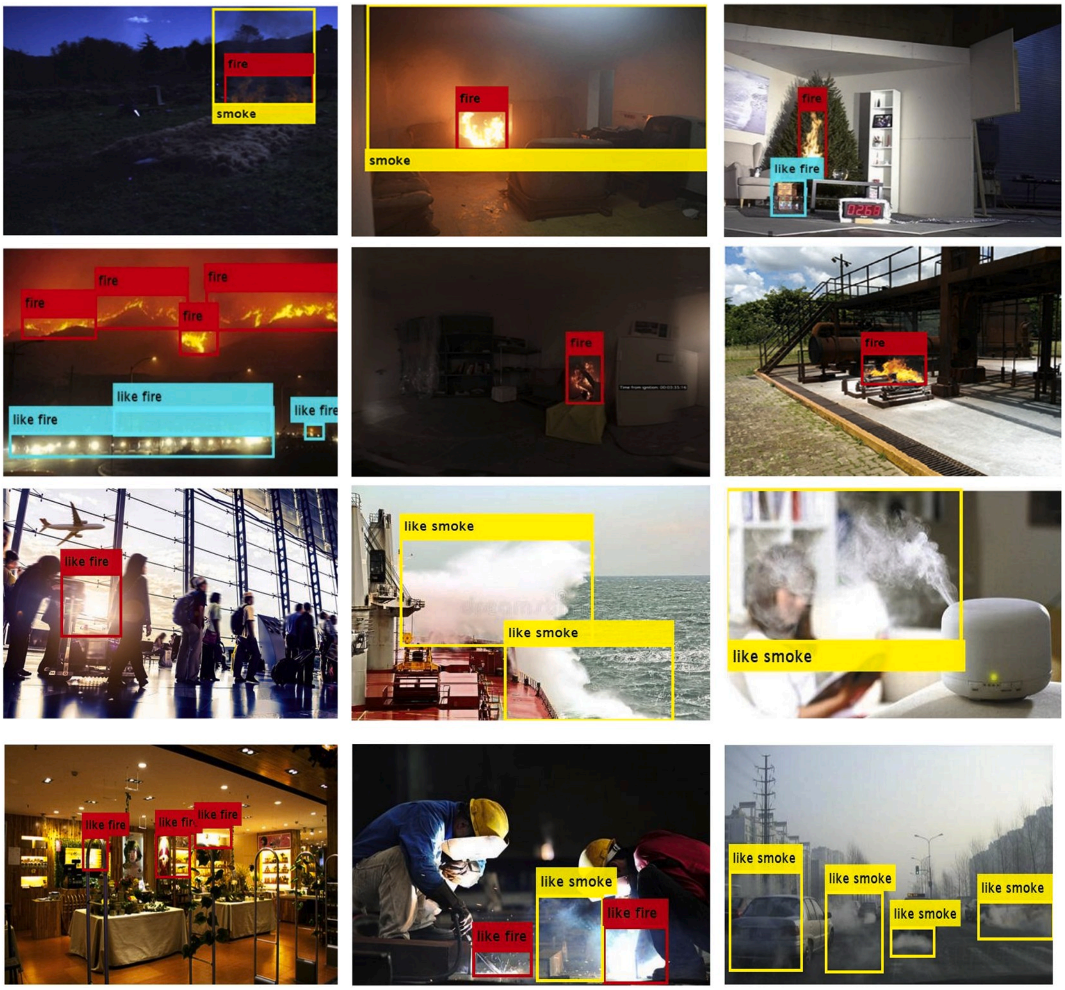


**Fig. 7.** Samples of images in the dataset.

**Table 4**
Statistics of fire images in the dataset.

| Scenario | Objects | | Disturbances | | Images |
|---|---|---|---|---|---|
| | smoke | fire | smoke-like | fire-like | |
| **Indoor** | 4376 | 5639 | 1241 | 3276 | 7634 |
| **Outdoor** | 3066 | 4056 | 2758 | 3948 | 5766 |
| **Total** | 7442 | 9695 | 3999 | 7224 | 13,400 |

use the sets in Ref. [17]. The training time varies with different network architectures.

## 3. Results and discussion

This section conducts experiments on Testset1 [19] and Testset2 developed in Session 2 to evaluate the performance of the proposed algorithms.

### 3.1. Performance on Testset1

The Testset1 is a benchmark fire video database consisting of 31 videos [19]. This database has 14 fire videos and 17 videos containing no fire collected from different scenarios. The evaluation results (in Table 5) of algorithms 1–8 are taken from Ref. [1]. The algorithms for comparison are selected considering the publication year, fire features used to detect and database used to evaluate. The algorithms 1–6 extract features by manually selection, and the algorithms 7 and 8 extract features by machine automatically learning

**Table 5**
Comparison with different fire detection algorithms.

| No. | Class | Algorithm | Missed detection rate (%) | False alarm rate (%) | Accuracy (%) |
|---|---|---|---|---|---|
| 1 | Manually extraction features | Chen [20] | 11.76 | 14.29 | 87.10 |
| 2 | | Celik [21] | 29.41 | 0 | 83.87 |
| 3 | | Raffiee [22] | 17.65 | 7.14 | 87.10 |
| 4 | | Habibuglu [23] | 5.88 | 14.29 | 90.32 |
| 5 | | De Lascio [24] | 13.33 | 0 | 92.86 |
| 6 | | Foggia [19] | 11.67 | 0 | 93.55 |
| 7 | CNN | Muhammad [8] (Alexnet) | 9.07 | 2.13 | 94.39 |
| 8 | | Muhammad [1] (GoogleNet) | 0.054 | 1.5 | 94.43 |
| 9 | | Faster-RCNN | 0.018 | 0.69 | 99.43 |
| 10 | | R–FCN | 0.018 | 0.97 | 99.20 |
| 11 | | SDD | 0.036 | 1.29 | 98.93 |
| 12 | | YOLO v3 | 0 | 0.46 | 99.62 |

based on image classification CNNs. The algorithms 9–12 are proposed in this paper. The results indicate that the algorithms based on CNNs are obviously better than the traditional algorithm extracting fire features manually. The proposed algorithms based on object detection CNNs can easily detect fire in this kind of lower complex scene like Testset1. The missed detection and false alarm rates are lower than the other algorithms. However, the Testset1 is not challenging for the four proposed algorithms, which cannot satisfactorily discern the advantages and disadvantages of these algorithms. Therefore, a more detailed evaluation is conducted through the self-built Testset2.

### 3.2. Performance on Testset2

Testset2 is the image fire test set developed in this paper, which has 14,590 images, including 3890 smoke samples and 5322 fire samples. Testset2 is very challenging as it collects images from more scenarios containing a large number of smoke-like and fire-like disturbances. Therefore, it is more suitable for evaluating the performance of the proposed algorithms.

#### 3.2.1. Average precision and detection time

Table 6 shows the average precision and detection time of four proposed algorithms. The four all achieve high average precision, which indicates that using the object detection CNNs to detect fire in images is entirely feasible. One-stage algorithms detect more quickly, which could detect more than 15 frames/s. It is able to achieve real-time performances. And the highest accurate algorithm based on YOLO v3, with 83.7% accuracy, detects fire the most quickly (28 FPS).

#### 3.2.2. Bootstrapping AP and rank

According to statistics, the average precision estimated from a single set of test samples is the point estimation. And the point estimate fluctuates with using different test sets to evaluate. It falls into a confidence interval with a certain probability. And there is an overlap between confidence intervals of different algorithms. Therefore, the point estimate from a single test set cannot compare the relative strength of two algorithms. Therefore, referring to the idea of Pascal VOC Challenge [18], the bootstrap method is used to produce 1000 bootstrap replicates, where the data points are sampled with replacement from the original $n$ test point. Then the bootstrap replicates are used to generate point estimates $\widehat{\theta}_1^*, \widehat{\theta}_2^*, \cdots, \widehat{\theta}_{1000}^*$ by evaluating an algorithm 1000 times. Then, the confidence interval of AP for each algorithm is $(\widehat{\theta}_{\alpha/2}^*, \widehat{\theta}_{1-\alpha/2}^*)$ (at the $1-\alpha$ level). And the confidence intervals are used to compare whether there is significant difference between two algorithms. Finally, the rank ranges are estimated. Table 7 shows bootstrapping AP and the rank of smoke detection. It indicates that the smoke detection average precision of the algorithm based on YOLO v3 is not statistically significantly different from the algorithm based on Faster-RCNN. Table 8 shows bootstrapping AP and rank of fire detection. It indicates that the fire detection average precision of the algorithm based on YOLO v3 is statistically significantly different from the other three. But differences between the other three are not significant.

#### 3.2.3. Robustness

According to statistics, the distribution of test results via bootstrapping, like detection rate and average precision, should follow a normal distribution for a robust algorithm. Otherwise, it can be considered that the algorithm has some shortcomings that lead to unstable detection results. The stability of the algorithm is studied through analyzing the distribution of bootstrapping mAP using skewness and kurtosis test.

The distributions are shown in Fig. 8. $DIS>1$ indicates the distribution is not follow the normal distribution. That is, the performance on detection average precision of the algorithm is not stable enough. The results reveal that the algorithms based on Faster-RCNN and R–FCN are not robust, which should be modified further for the needs of fire detection. The algorithms based on SSD and YOLO v3, especially YOLO v3, are robust enough.

**Table 6**
Average precision and detection time of the proposed algorithms.

| Algorithm | AP (%) | | mAP (%) | Detection speed (FPS) |
|---|---|---|---|---|
| | smoke | fire | | |
| Faster-RCNN | 79.7 | 84.9 | 82.3 | 3 |
| R–FCN | 78.5 | 83.3 | 80.9 | 5 |
| SSD | 72.8 | 82.8 | 77.8 | 16 |
| YOLO v3 | 81.2 | 87.8 | 84.5 | 28 |

IOU = 0.5. mAP denotes the mean AP for smoke and fire.

**Table 7**
Bootstrapping AP and rank of smoke detection.

| Algorithm | AP range | | | Rank | Rank range |
|---|---|---|---|---|---|
| | 0.025 | 0.5 | 0.975 | | |
| Faster-RCNN | **77.5** | **79.8** | **82.2** | **2** | **1–2** |
| R–FCN | 76.1 | 78.6 | 81.2 | 3 | 2–3 |
| SSD | 70.1 | 72.8 | 75.5 | 4 | 4 |
| YOLO v3 | **79.1** | **81.3** | **83.6** | **1** | **1–2** |

$\alpha$ = 0.05. The leading methods that are not statistically significantly different from each other are highlighted in bold.

**Table 8**
Bootstrapping AP and rank of fire detection.

| Algorithm | AP range | | | Rank | Rank range |
|---|---|---|---|---|---|
| | 0.025 | 0.5 | 0.975 | | |
| Faster-RCNN | 83.0 | 84.9 | 86.8 | 2 | 2–3 |
| R–FCN | 81.2 | 83.4 | 85.6 | 3 | 2–4 |
| SSD | 80.7 | 82.9 | 85.2 | 4 | 3–4 |
| YOLO v3 | 86.0 | 87.8 | 89.7 | 1 | 1 |

$\alpha$ = 0.05. The leading methods that are not statistically significantly different from each other are highlighted in bold.

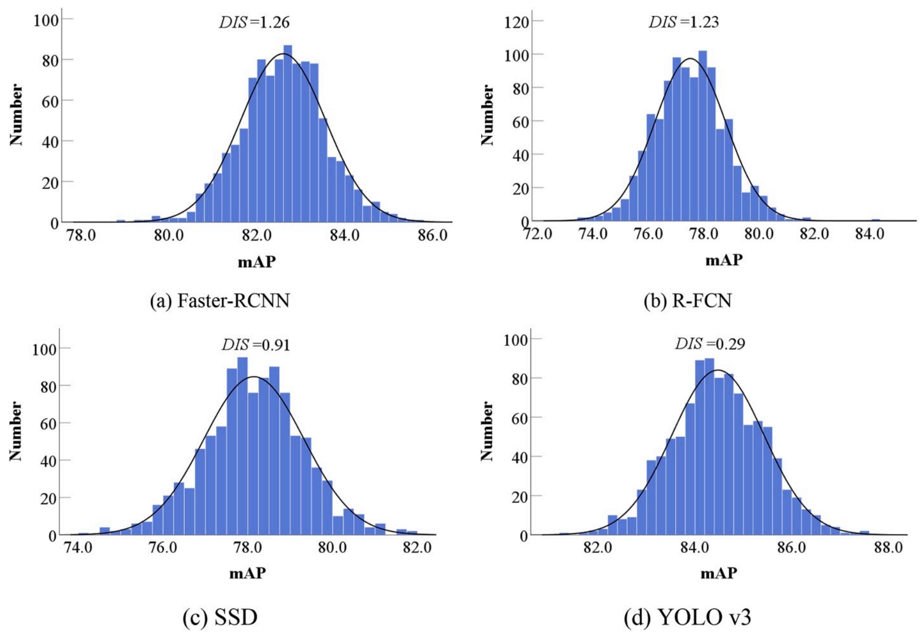


**Fig. 8.** Histogram of the distribution of bootstrapping AP ($\alpha$ = 0.05. *DIS* denotes the index of normal distribution.).

## 4. Conclusion

To improve the performance of image fire detection technology, the advanced object detection CNNs of Faster-RCNN, R–FCN, SSD, and YOLO v3 are used to develop algorithms of image fire detection. The proposed algorithms can automatically extract complex image fire features and successfully detect fire in different scenes. The evaluation experiments results are given as follows:

(1) Experiments on Testset1 reveal that the missed detection and false alarm rates of the proposed algorithms are lower than current algorithms. It is proved that the algorithms based on CNNs are obviously better than the traditional algorithm, which can easily detect fire in simple scenes like Testset1.
(2) Experiments on Testset2 reveal that the four proposed algorithms all achieve high average precision, which indicates that using the object detection CNNs to detect fire in images is entirely feasible. One-stage algorithms are able to achieve real-time performances, which could detect more than 15 frames/s.
(3) The fire detection average precision of the algorithm based on YOLO v3 is statistically significantly different from the other three. But the differences between smoke detection average precision of the algorithms based on YOLO v3 and Faster-RCNN are not significant.
(4) The highest accurate algorithm based on YOLO v3, with 83.7% accuracy, detects fire the most quickly (28 FPS) and is the strongest robust.

## Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Pu Li:** Conceptualization, Methodology, Investigation, Software, Writing - original draft. **Wangda Zhao:** Methodology, Writing - review & editing.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos.51904229, 51676210 and 51906261). The authors appreciate the supports deeply.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csite.2020.100625.

## References

[1] K. Muhammad, J. Ahmad, I. Mehmood, et al., Convolutional neural networks based fire detection in surveillance videos, IEEE Access 6 (2018) 18174–18183.
[2] C. Tao, J. Zhang, P. Wang, Smoke detection based on deep convolutional neural networks, in: 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016, pp. 150–153.
[3] A. Filonenko, L. Kurnianggoro, K. Jo, Comparative study of modern convolutional neural networks for smoke detection on image data, in: 2017 10th International Conference on Human System Interactions (HSI), 2017, pp. 64–68.
[4] Z. Yin, B. Wan, F. Yuan, et al., A deep normalization and convolutional neural network for image smoke detection, IEEE ACCESS 5 (2017) 18429–18438.
[5] A.J. Dunnings, T.P. Breckon, Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 1558–1562.
[6] A. Namozov, Y. Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, Adv. Electr. Comput. Eng. 18 (2018) 121–128.
[7] W. Mao, W. Wang, Z. Dou, Y. Li, Fire recognition based on multi-channel convolutional neural network, Fire Technol. 54 (2018) 531–554.
[8] K. Muhammad, J. Ahmad, S.W. Baik, Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing 288 (2018) 30–42.
[9] Y. Hu, X. Lu, Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features, Multimed. Tool. Appl. 77 (2018) 29283–29301.
[10] A. Namozov, Y. Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, Adv. Electr. Comput. Eng. 18 (2018) 121–128.
[11] L. Wonjae, K. Seonghyun, L. Yong-Tae, L. Hyun-Woo, C. Min, Deep neural networks for wild fire detection with unmanned aerial vehicle, in: 2017 IEEE International Conference on Consumer Electronics (ICCE), 2017, pp. 252–253.
[12] Y. Luo, L. Zhao, P. Liu, D. Huang, Fire smoke detection algorithm based on motion characteristic and convolutional neural networks, Multimed. Tool. Appl. 77 (2018) 15075–15092.
[13] N.M. Dung, D. Kim, S. Ro, A video smoke detection algorithm based on cascade classification and deep learning, KSII Internet Inf. 12 (2018) 6018–6033.
[14] Z. Zhong, M. Wang, Y. Shi, W. Gao, A convolutional neural network-based flame detection method in video sequence, Signal Image Video Process. 12 (2018) 1619–1627.
[15] S. Bianco, R. Cadène, L. Celona, P. Napoletano, Benchmark analysis of representative deep neural network architectures, IEEE Access 6 (2018) 64270–64277.
[16] J. Redmon, A. Farhadi, YOLOv3: an Incremental Improvement, 2018.
[17] J. Huang, V. Rathod, C. Sun, et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3296–3297.

[18] M. Everingham, S.M.A. Eslami, L. Van Gool, et al., The pascal visual object classes challenge: a retrospective, Int. J. Comput. Vis. 111 (2015) 98–136.
[19] P. Foggia, A. Saggese, M. Vento, Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion, IEEE T CIRC SYST VID 25 (2015) 1545–1556.
[20] C. Thou-Ho, W. Ping-Hsueh, C. Yung-Chuen, An Early Fire-Detection Method Based on Image Processing, in: 2004 International Conference on Image Processing, vol. 3, 2004, pp. 1707–1710.
[21] T. Çelik, H. Demirel, Fire detection in video sequences using a generic color model, Fire Saf. J. 44 (2009) 147–158.
[22] A. Rafiee, R. Dianat, M. Jamshidi, et al., Fire and smoke detection using wavelet analysis and disorder characteristics, in: 2011 3rd International Conference on Computer Research and Development, 2011, pp. 262–265.
[23] Y.H. Habiboğlu, O. Günay, A.E. Çetin, Covariance matrix-based fire and flame detection method in video, Mach. Vis. Appl. 23 (2012) 1103–1113.
[24] R. Di Lascio, A. Greco, A. Saggese, M. Vento, in: A. Campilho, M. Kamel (Eds.), Improving Fire Detection Reliability by a Combination of Videoanalytics, Springer International Publishing, Cham, 2014, pp. 477–484.