



Fire detection in video surveillance using superpixel-based region proposal and ESE-ShuffleNet

Pengyu Wang¹ · Jianmei Zhang¹ · Hongqing Zhu¹

Received: 16 August 2020 / Revised: 10 May 2021 / Accepted: 8 July 2021 /

Published online: 8 September 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

This paper proposes a forest fire detection framework using superpixel-based suspicious fire region proposal and light-weight convolutional neural network. The proposed methodology contains two main steps. In suspicious fire region proposal, we introduce a novel superpixel algorithm (SCMM) driven by Cauchy mixture model. Then, the negative Under-segmentation Error (UE) of each superpixel is applied to inter-frame comparison for predicting varying superpixels. After that, by computing the features of motion superpixels using Local Difference Binary (LDB) descriptor for two adjacent frames, the suspicious fire regions are localized. In following fire identification, to improve network performance while reducing computational complexity, this study presents a light-weight network architecture, called Expanded Squeeze-and-Excitation ShuffleNet (ESE-ShuffleNet). All suspicious fire regions are sent into this network to identify as either fire or non-fire included. Experiments show that our framework performs well on fire detection tasks. Code is available at <http://www.imagetechnopolynomials.com/ESE-ShuffleNet.html>.

Keywords Forest fire detection · Cauchy mixture model · Superpixel · Light-weight network · ShuffleNet

1 Introduction

Video based forest fire detection [34] is one of the most cost-effective way, which has a broad prospect in practical application. Fire detection design usually faces many challenging factors, such as complicated background, varying fire property, misleading weather that lead to rising false alarm. Over the past decade, a large number of computer vision methods for fire detection performed strongly depend on description of fire color, texture, geometric and motion characteristics. Although conventional methods [6,

✉ Hongqing Zhu
hqzhu@ecust.edu.cn

¹ School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

[10] usually have lower running time, it is less capable at distinguishing fire regions and fire-like moving targets. Recent years, Convolutional Neural Networks (CNNs) showing excellent performance in various computer vision tasks have also been applied to forest fire detection [4, 27–29]. Different from above methods using manual setting features, data-driven networks can perform automatic fire features extraction more effectively. By comparison, CNN-based methods have stronger anti-interference ability to complex weather, background, and fire changes. However, relatively complex network architecture is established, and extensive training data is collected that place the high requirement on computer hardware. For examples, recent fire detection networks such as [28] and [4] based on Inception-V4 [38] and Faster R-CNN [32] have large network complexity.

In general, light-weight networks need to be designed instead of deep CNN-based methods for fire detection in real-time surveillance purpose. Although running time is one crucial aspect for network-based fire detection approaches, promoting accuracy for fire region proposal is a main concern for later studies. Since no limitation on CNN selection for fire detection is placed in general, structural advantages could be exploit as much as possible to enhance detection accuracy. In addition, color and motion characteristics of fire could be well engaged in design of the proposed method.

In this paper, we integrate superpixel segmentation, moving object detection and light-weight network for efficient and accurate implementation aims at resolving above issues. In specific, this paper presents a forest fire detection framework divided into suspicious region localization and fire identification. Firstly, this paper introduces a preprocessing step by using superpixel segmentation strategy that transforms images from pixel-level to superpixel-level. It can significantly reduce subsequent method computations. A new superpixel algorithm (SCMM) driven by Cauchy Mixture Model (CMM) is proposed. This segmentation approach adopts five-dimensional Cauchy distributions to synthetically consider color and spatial information of each pixel. Then, we introduce a robust and efficient Local Difference Binary (LDB) [43] operator to detect motion area. Together with negative Under-segmentation Error (UE) comparing between every frame with the started frame, and updating frame background by LDB descriptors, a motion prediction on suspicious superpixels is operated. This motion analysis approach can remove obvious irrelevant regions as well as help to capture local fire regions based on reasonably segmented superpixels. Finally, considering that light-weight network is more efficient than deep CNNs in real-time video surveillance, representative ShuffleNet series [23, 45] is taken as basic architecture with several structural modifications. Specifically, the proposed Expanded Squeeze-and-Excitation ShuffleNet (ESE-ShuffleNet) improves current light-wight networks by two aspects: (i) group convolution realizing channel expansion of features is employed to construct attention-based inverted residual network unit with depthwise convolution; (ii) the Squeeze-and-Excitation (SE) module [16] is inserted within each network unit, which provides channel-wise attention to expanded intermediate features for improving classification accuracy. By experiments, average classification accuracy achieves 99.61% on Foggia dataset and 94.56% on Dunning's dataset while only 1.636M parameters are required by this new network.

The main contributions of this paper are as follows:

- A novel and widely applicable superpixel segmentation method is proposed by implementing Cauchy mixture model in five-dimensional feature space.
- Superpixel segmentation is firstly conducted to achieve superpixel-level motion object detection for accurate fire detection.

- We develop a motion detection ideology for fire suspicious region localization by describing the LDB features difference between superpixels for adjacent frames.
- ESE-ShuffleNet is proposed as a widely applicable efficient light-weight network. Placing the inserted SE module within units after the depthwise convolution, and combining it with an inverted residual structure, features are learned to their highest probability with as least computation as possible.

2 Related works

In this section, we will review the related researches of superpixel, wildfire detection and light-weight networks.

Superpixel segmentation is a popular pre-segmentation approach in computer vision. N-cuts [33] is the first developed superpixel segmentation method. Lazy Random Walk (LRW) [37] is another classical superpixel model that adopts image texture to generate superpixels and has considerable regularity. After that, Linear Spectral Clustering (LSC) [8] superpixel is proposed, which uses the estimation scheme of k -means clustering to optimize N-cuts algorithm. Then, Content-Adaptive Superpixel (CAS) is proposed by Xiao et al. [40] where image color, contour and texture are comprehensively considered. And Ban et al. [3] introduced Gaussian Mixture Model Superpixel (GMMS). GMMS outperforms above models in segmentation accuracy but still remains some irregular superpixel boundaries unsolved. Lately, an Adaptive High-Precision (AHP) [41] method was reported to generate superpixels by the iterative calculation of distance-measurement in CLELAB color space. Another well-known superpixel approach Simple Linear Iterative Clustering (SLIC) [1] performs based on k -means clustering and has relatively good boundary adherence. Dunnings et al. [9] introduced SLIC for fire images preprocessing and finally realized scene-independent fire detection by combining with a simplified CNN. CA et al. [7] discussed the performance of SLIC followed by Inception-V4 [38] for fire detection. Although applying superpixel segmentation to fire detection has achieved good accuracy, the above literatures haven't aimed at developing better segmentation methods instead of using traditional approaches where stronger pre-segmentation approach would benefit consequence network significantly.

Early fire detection methods usually describe fire regions based on manual designed features. Concretely, Borges et al. [6] integrated image color information and Bayesian classifier for fire recognition. Another approach proposed by Foggia et al. [10] considers fire color, shape, and motion features comprehensively, but tiny fire could hardly be identified. With the development of deep learning, CNN has brought notable improvements in wildfire detection. Barmpoutis et al. [4] identified fire regions by modifying Faster R-CNN [32], and wildfire are detected based on the locality criterion on manifold. Recently, a branch of latest studies extensively employs YOLO series to fire detection. Initially, Li et al. [22] compared fire detection results by Faster R-CNN, RFCN, SSD and YOLO v3, and concluded that YOLO v3 achieves the most satisfactory performance. Moreover, Zhang et al. [44] combined multi-scale output mechanism and channel-wise attention with YOLO v3 [31] to increase network generalization, Huang et al. [18] improved regression box loss in YOLO v4 [5] to enhance small-scale fire detection results. However, computational cost of YOLO series is generally higher than other advanced light-weight networks (MobileNet series, ShuffleNet series, etc.) for classification due to challenging object detection task.

Recently, many scholars have paid attention to fire detection in video surveillance. Matukhina et al. [25] firstly utilized Gaussian mixture model to detect moving regions, and identify fire through RGB color analysis. Muhammad et al. [26] developed a CNN based fire detection framework that applies dynamic channel selection to fire detection on house, forest and vehicle. Then, considering of reducing computational cost for real-world video surveillance. Saponara et al. [36] adopted classical edge boxes algorithm to achieve region proposal and classification. Muhammad et al. [27] introduce a SqueezeNet [19] based architecture for fire video detection, which can better balance the efficiency and accuracy. Benefit from the fire model using squeeze-expand manner, high effectiveness is shown. In [28] and [29], two video-based fire detection networks are proposed by modifying Inception-V4 [38] and MobileNet [35]. In general, although light-weight network is considered efficient in real-world video surveillance system, it will cause false alarm to a certain extent. To resolve above problem, structural improvement based on light-weight network is targeted in the proposed method in which superpixel-based moving object detection is also taken.

Towards real-time performance, light-weight network is the best choice in wildfire detection. Specifically, SqueezeNet is proposed by Iandola et al. [19], which combined 3×3 and 1×1 convolutions to reduce computational cost, and achieves the AlexNet-level [20] level accuracy with only 1/50 parameters of it. Then, Howard et al. proposed the networks of MobileNet family [14, 15, 35]. In MobileNet V1 [15], the depthwise separable convolution that compress network parameters within acceptable error range is addressed. In MobileNet V2 [35], a residual learning scheme different with original ResNet [13] is introduced in network. This network adopts inverted residuals and linear bottlenecks that achieve good performance. In MobileNet V3 [14], the SE model [16] is added in each network unit, hardware-aware network architecture search and Netadapt algorithm [42] for network architecture construction are also combined. Another network family ShuffleNet V1 [45] replaces pointwise convolution with channel shuffle, which achieves information communication between different feature-map groups. In its upgraded version, ShuffleNet V2 [23] designs a new network structure based on channel split and branch concatenation. Although the number of parameters are somewhat increased, further improved performance has been seen. Moreover, some researchers study on automated and learned network architectures, such as MnasNet [39]. This model has excellent accuracy even though relatively large amount of computation is needed in network optimization. To comprehensively employ these advance structures, we integrate channel shuffle and the channel-wise attention into inverted residual unit and propose an effective light-weight classification network ESE-ShuffleNet.

3 Proposed scheme

This section presents a detailed description of our scheme consisted of two main steps: (i) suspicious fire region proposal using SCMM superpixel segmentation and LDB based motion object localization; (ii) suspicious fire region identification using ESE-ShuffleNet. Figure 1 shows a block diagram of our fire detection scheme.

3.1 Suspicious fire region proposal

The framework of suspicious fire region localization is introduced as shown in Fig. 2.

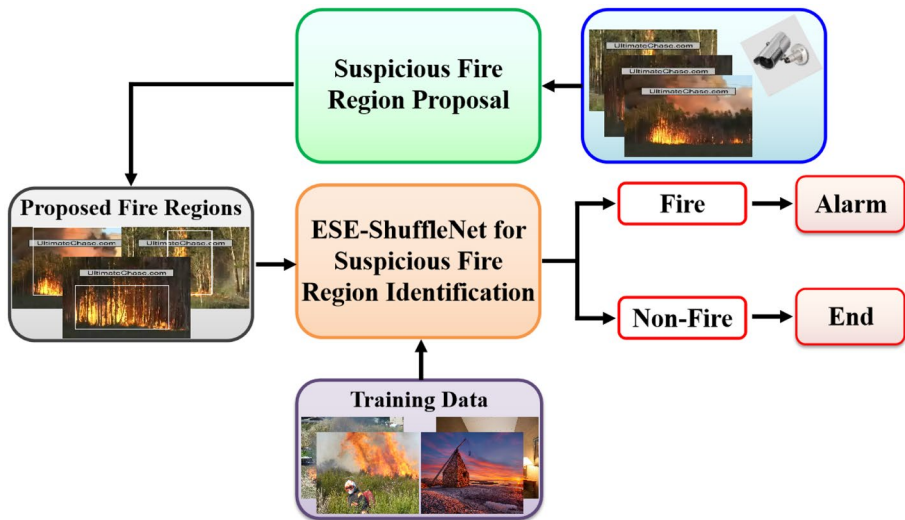


Fig. 1 Overview of the proposed scheme

- 1) **SCMM superpixel segmentation model:** Superpixel segmentation is widely used in image preprocessing, which can divide an image into the expected number of connected and uniform pixel groups according to color, texture and spatial information. Here, we propose a novel superpixel segmentation approach driven by CMM. Cauchy distribution is a special case of Student's t -distribution when the degree of freedom equals one, so that the parameter estimation is simpler than that of Student's t -distribution. In superpixel segmentation, each superpixel (image local region) is represented by one probability distribution. Therefore, a heavy tailing probability distribution selected could well adapt variance of data distribution and thus fit the outliers of each superpixel. Figure 3 visualizes the intensity fitting results of superpixels. Figure 3(b)–(e) show the pixel intensity distributions of superpixels and their fitting results using Gaussian and

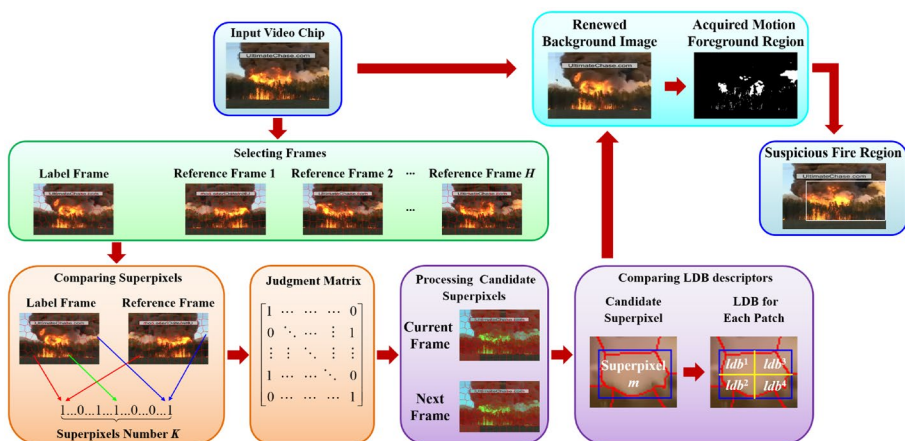


Fig. 2 The framework of the suspicious fire region proposal

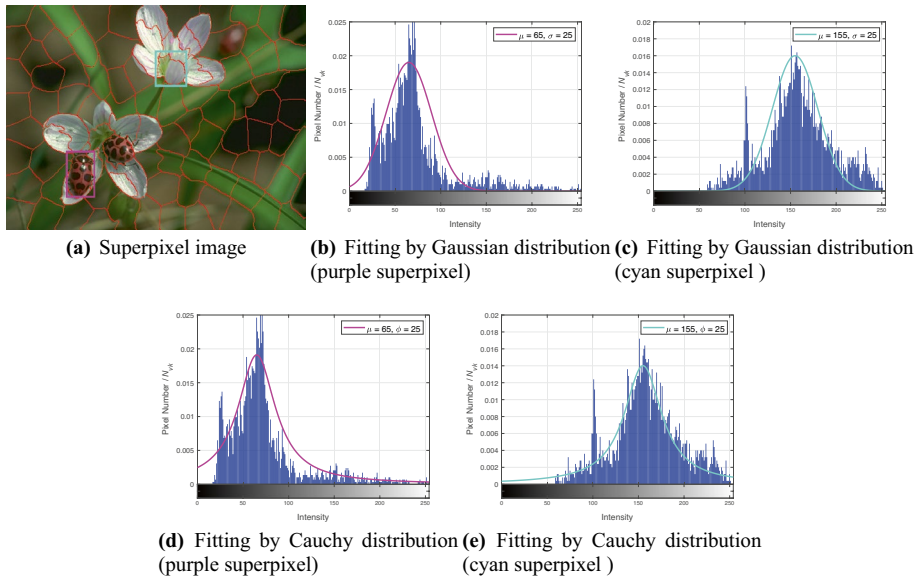


Fig. 3 The intensity fitting results of superpixels using Gaussian and Cauchy distributions, where N_{v_k} denotes the total pixel numbers in the k -th superpixel

Cauchy distributions. It is noticeable that the Cauchy distribution with heavy tail and narrow peak presents better approximation of the intensity distribution.

For an input image I with N pixels, we denoted n ($n = 1, 2, \dots, N$) as pixel index, while comprehensively consider pixel color and spatial information, the five-dimensional observation z_n of each pixel can be defined

$$z_n = (l_n, a_n, b_n, x_n, y_n)^T, \quad (1)$$

where l_n , a_n , b_n represent L, a and b components of the n -th pixel in CLELAB color space, and x_n , y_n represent its vertical and horizontal coordinates.

In our SCMM, giving K as the expected superpixel number and k as superpixel index, each superpixel is associate with a multi-dimensional Cauchy distribution $C_d(z_n | \Theta_k)$ that has the following form

$$C_d(z_n | \Theta_k) = \frac{\Gamma[(d+1)/2] |\Phi_k|^{-\frac{1}{2}}}{\pi^{\frac{1+d}{2}} [1 + (z_n - \mu_k)^T \Phi_k^{-1} (z_n - \mu_k)]^{\frac{1+d}{2}}}, \quad (2)$$

where d is the dimension of distribution (here $d = 5$), $|\cdot|$ is the determinant operation, $\Gamma(\cdot)$ is the Gamma function, and $\Theta_k = \{\mu_k, \Phi_k\}$ represents model parameters, μ_k is a five-dimensional mean vector

$$\mu_k = (l_k, a_k, b_k, x_k, y_k)^T, \quad (3)$$

and Φ_k is a 5×5 covariance matrix with color component $\Phi_{k,c}$ and spatial component $\Phi_{k,s}$

$$\Phi_k = \begin{bmatrix} \Phi_{k,c} & 0 \\ 0 & \Phi_{k,s} \end{bmatrix}, \quad (4)$$

According to above definitions, we stipulate that each pixel can only own an independent superpixel label $\Omega_n \in [1, K]$. Then, the density function is written as

$$f(z_n | \Pi, \Theta) = \sum_{k=1}^K \pi_{nk} C_d(z_n | \Theta_k), \quad (5)$$

where the prior probability set is denoted as $\Pi = \{\pi_{nk}\}$, and its constraints are displayed by

$$0 \leq \pi_{nk} \leq 1 \text{ and } \sum_{k=1}^K \pi_{nk} = 1. \quad (6)$$

By multiplying the density functions, the joint conditional density function of the whole observation set $Z = (z_1, z_2, \dots, z_N)$ is

$$P(Z | \Pi, \Theta) = \prod_{n=1}^N f(z_n | \Pi, \Theta) = \prod_{n=1}^N \sum_{k=1}^K \pi_{nk} C_d(z_n | \Theta_k). \quad (7)$$

To optimize model parameters, the log-likelihood function of proposed superpixel model is described as

$$\mathcal{L}(\Pi, \Theta | Z) = \sum_{n=1}^N \log \left\{ \sum_{k=1}^K \pi_{nk} C_d(z_n | \Theta_k) \right\}. \quad (8)$$

Based on Jason's inequality [21], the posterior probability G_{nk} is introduced.

$$G_{nk} = \frac{\pi_{nk} C_d(z_n | \Theta_k)}{\sum_{i=1}^K \pi_{ni} C_d(z_n | \Theta_i)}, \quad (9)$$

and the final objective function is obtained

$$J(\Pi, \Theta | Z) = \sum_{n=1}^N \sum_{k=1}^K G_{nk} \{ \log \pi_{nk} + \log C_d(z_n | \Theta_k) \}. \quad (10)$$

Here, the Expectation-Maximization (EM) [30] algorithm is used to estimate the maximum likelihood. Using (2), the objective function (10) is rewritten as

$$J(\Pi, \Theta | Z) = \sum_{n=1}^N \sum_{k=1}^K G_{nk} \left\{ \log \pi_{nk} + \log \frac{2}{\pi^3} - \frac{1}{2} \log |\Phi_k| + 3 \log [1 + (z_n - \mu_k)^T \Phi_k^{-1} (z_n - \mu_k)] \right\}. \quad (11)$$

Before estimating model parameters, the five-dimensional mean vector μ_k is split into color component $\mu_{k,c} = (l_k, a_k, b_k)^T$ and spatial component $\mu_{k,s} = (x_k, y_k)^T$ for initialization

$$\mu_{k,c}(l_k, a_k, b_k) = \frac{1}{N_{\partial_k}} \sum_{j \in \partial_k} z_{j,c}(l_j, a_j, b_j), \quad (12)$$

$$\mu_{k,s}(x_k, y_k) = z_{\delta_{k,s}}(x_{\delta_k}, y_{\delta_k}), \quad (13)$$

where ∂_k is the pixel set of the k -th initialized superpixel, N_{∂_k} is the pixel number of ∂_k . Giving W as input image width, the index δ_k of center observation z_{δ_k} is defined as

$$\delta_k = \sqrt{\frac{N}{K}}(k \bmod \sqrt{\frac{W^2 K}{N}}) + k \frac{N}{K} + \sqrt{\frac{(W+1)^2 N}{4K}}, \quad (14)$$

To simplify computation, we define the initial prior probability π_{nk} as $1/K$. The color component $\Phi_{k,c}$ of covariance matrix Φ_k is initialized as a diagonal matrix

$$\Phi_{k,c} = \begin{bmatrix} \sigma_{kl} & 0 & 0 \\ 0 & \sigma_{ka} & 0 \\ 0 & 0 & \sigma_{kb} \end{bmatrix}, \quad (15)$$

where σ_{kl} , σ_{ka} , and σ_{kb} represents the standard deviation of color component in the k -th initialized superpixel. And spatial component $\Phi_{k,s}$ is initialized as a 2×2 identity matrix.

After model parameters initialization, we calculate the derivative of objective function (11) with respect to μ_k , Φ_k and π_{nk} . The solution of $\partial J(\Pi, \Theta|Z)/\partial \mu_{k,c} = 0$ and $\partial J(\Pi, \Theta|Z)/\partial \mu_{k,s} = 0$ yields the estimation of $\mu_{k,c}$ and $\mu_{k,s}$

$$\mu_{k,c}^{(t+1)} = \frac{\sum_{n=1}^N G_{nk}^{(t)} w_{nk,c}^{(t)} z_{n,c}}{\sum_{n=1}^N G_{nk}^{(t)} w_{nk,c}^{(t)}}, \quad \mu_{k,s}^{(t+1)} = \frac{\sum_{n=1}^N G_{nk}^{(t)} w_{nk,s}^{(t)} z_{n,s}}{\sum_{n=1}^N G_{nk}^{(t)} w_{nk,s}^{(t)}}, \quad (16)$$

where $w_{nk,s}$ and $w_{nk,c}$ represent the expected spatial and color weights of n -th observation about k -th superpixel obtained by

$$w_{nk,c}^{(t)} = \frac{6}{1 + (z_{n,c} - \mu_{k,c}^{(t)})^T (\Phi_{k,c}^{(t)})^{-1} (z_{n,c} - \mu_{k,c}^{(t)})}, \quad (17)$$

$$w_{nk,s}^{(t)} = \frac{6}{1 + (z_{n,s} - \mu_{k,s}^{(t)})^T (\Phi_{k,s}^{(t)})^{-1} (z_{n,s} - \mu_{k,s}^{(t)})}. \quad (18)$$

Similarly, to estimate covariance matrix Φ_k , we set the derivative of $J(\Pi, \Theta|Z)$ with respect to $\Phi_{k,c}$ and $\Phi_{k,s}$, respectively, then we have

$$\Phi_{k,c}^{(t+1)} = \frac{\sum_{n=1}^N G_{nk}^{(t)} w_{nk,c}^{(t)} (z_{n,c} - \mu_{k,c}^{(t+1)}) (z_{n,c} - \mu_{k,c}^{(t+1)})^T}{\sum_{n=1}^N G_{nk}^{(t)}}, \quad (19)$$

$$\Phi_{k,s}^{(t+1)} = \frac{\sum_{n=1}^N G_{nk}^{(t)} w_{nk,s}^{(t)} (z_{n,s} - \mu_{k,s}^{(t+1)}) (z_{n,s} - \mu_{k,s}^{(t+1)})^T}{\sum_{n=1}^N G_{nk}^{(t)}}. \quad (20)$$

Applying the constraints in (6) and Lagrange multiplier scheme, we obtain the estimation of prior probability by solution $\partial J(\Pi, \Theta|Z)/\partial \pi_{nk} = 0$ as follows

$$\pi_{nk}^{(t+1)} = \frac{G_{nk}^{(t)}}{\sum_{i=1}^K G_{ni}^{(t)}}. \quad (21)$$

Thus far, the estimation about parameters completed, the superpixel label of each pixel could be obtained by

$$\Omega_n = \operatorname{argmax}_{k \in K} \frac{\pi_{nk} C_d(z_n | \Theta_k)}{\sum_{i=1}^K \pi_{ni} C_d(z_n | \Theta_i)}, \quad (22)$$

The implementation steps of our SCMM is summarized as **Algorithm 1**.

Algorithm 1 SCMM superpixel segmentation

Input: Image I , superpixels number K .

Output: The superpixel label $\Omega_n \in [1, K]$ of each pixel.

1: Initialize the mode parameters μ_k , Φ_k and π_{nk} .

2: For $t = 1$ to iterations **do**

3: E step

4: Evaluate the posterior probability G_{nk} using (9).

5: M step

6: Update mode parameters μ_k , w_{nk} , Φ_k , and π_{nk} using (16), (17), (18), and (19), respectively.

7: End for

8: Calculate the superpixel label of pixels using (20).

- 2) **LDB based motion object localization:** Since the object varies slowly in video, successive frames are highly similar. We select one frame per ten frames for further reduced computation complexity. Every selected frame would be segmented into superpixel image with the same superpixel number using proposed SCMM method. The first selected frame in the video would be regarded as label frame and the other H frames as reference. Then, a UE metric that is originally used to evaluate the extent superpixel overlaps with ground truth is adopted. Here, we introduce a negative UE to obtain the proportion every reference frame differentiate with label frame as follows.

$$\text{negative UE} = 1 - \varphi(r_k, u_k) \frac{|r_k|}{N_{r_k}}, \quad (23)$$

where r_k and u_k represent pixel sets of reference frame and label frame respectively in the k -th superpixel, N_{r_k} is the number of pixels in r_k and

$$\varphi(r_k, u_k) = \begin{cases} 1, & \text{if } |r_k \cap u_k| > \omega |r_k| \\ 0, & \text{if } |r_k \cap u_k| \leq \omega |r_k| \end{cases}, \quad (24)$$

here, we default the parameter ω as 0.05 for acceptable effectiveness and stability [3]. A vector generated by calculating the difference of every superpixel between that reference frame and the label could be expressed as below

$$[1, \dots, 0, \dots, 1, \dots, 1, \dots, 0, \dots, 0]_{1 \times K}. \quad (25)$$

By obtaining this vector for the other H reference frames with the label frame, we can obtain an $H \times K$ judgment matrix Q with element either 1 or 0 as

$$Q = \begin{bmatrix} 1 & \dots & \dots & \dots & 0 \\ 0 & \ddots & \dots & \vdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \dots & \dots & \ddots & 0 \\ 0 & \dots & \dots & \dots & 1 \end{bmatrix}_{H \times K}, \quad (26)$$

where element 1 indicates that a superpixel at this reference frame is significantly different from the corresponding superpixel in label frame. The column index represents superpixel index, and row index denotes reference frame index. Not if the k -th column of this matrix have all its value showed 0, otherwise, this superpixel would be consider within candidate fire region. The steps of the judgment matrix Q can be summarized **Algorithm 2**.

Algorithm 2 Judgment matrix Q for candidate superpixel selection

Input: Label frame and H reference frames, superpixel number K .

Output: The judgment matrix Q .

1: **Initialize** $Q \in R^{H \times K}$ matrix

2: **For** $h = 1$ to H **do**

3: **Obtain** superpixel image of the h -th reference frame using **Algorithm 1**.

4: **For** $k = 1$ to K **do**

5: **Compute** negative UE between the k -th superpixels of the h -th reference frame and label frame.

6: **Record** $Q_{h,k}$ = value of negative UE.

7: **End for**

8: **End for**

The next step is to obtain accurate motion position by enhancing predicted background and significantly changed superpixel information. Based on Q , those superpixels that have at least one 1 in all H frames would be selected as candidate superpixels in this step. Here, we adopt LDB descriptor to process features of every candidate superpixel, and the implementation details are presented in **Algorithm 3**. While applying this feature descriptor, superpixel features could be described by testing average intensity and first-order gradient difference.

Algorithm 3 Candidate superpixel feature description using LDB descriptor

Input: Superpixel image with M candidate superpixels.

Output: LDB descriptor $LDB \in R^{4 \times 108}$ of each candidate superpixel.

1: **For** $m = 1$ to M **do**

2: **Calculate** smallest rectangle that could contain all pixels of the m -th candidate superpixel

3: **Divide** the rectangle into quarters, each patch is defined as $q_p (p = 1, 2, 3, 4)$

4: **For** $p = 1$ to 4 **do**

5: **Calculate** a 108-dimensional LDB sub-descriptor ldb for q_p using the 3×3 strategy [10], which comprehensively considers the local average intensity, horizontal and vertical gradients of patch

6: **Record** $LDB(p, :) = ldb$

7: **End for**

8: **End for**

After that, for each frame, only pixels located in the position at label frame of the candidate superpixel rectangles would engage in this method. The smallest rectangle that could contain all these pixels in one candidate superpixel would be prepared for dividing four equal patches according to LDB descriptor [43]. Then another 3×3 parti-

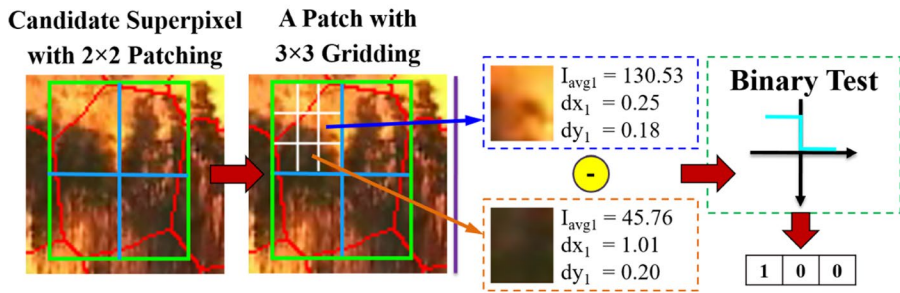


Fig. 4 The LDB descriptor for superpixel

tion would be implemented on each patch (see Fig. 4). LDB describes color and gradient features, and its variances between superpixels are judged to determine suspicious motion area. Due to the intensity and two orientation gradients, a four encoded binary 108-dimensional features could be used to describe the contents inside a superpixel. In Fig. 5, we report the conditions for candidate superpixels updating in a pair of adjacent frames. Technically, if the LDB features of a candidate superpixel in adjacent frames is similar (XOR result between LDB features higher than the threshold), this superpixel can be considered as unchanged and would be replaced by the same position contents in previous frame (see cyan boxes in Fig. 5). In contrast, if XOR result is smaller than threshold (see yellow boxes in Fig. 5), this candidate superpixel would be considered with considerable motion change and suspected as fire region. Therefore, a weighted fusion using relevant frames would be implemented according to (25) to renew candidate superpixel at this frame. Processes of candidate superpixels updating is formulated as:

Fig. 5 Update conditions of candidate superpixels in adjacent frames, (a) previous frame; (b) current frame; (c) update conditions

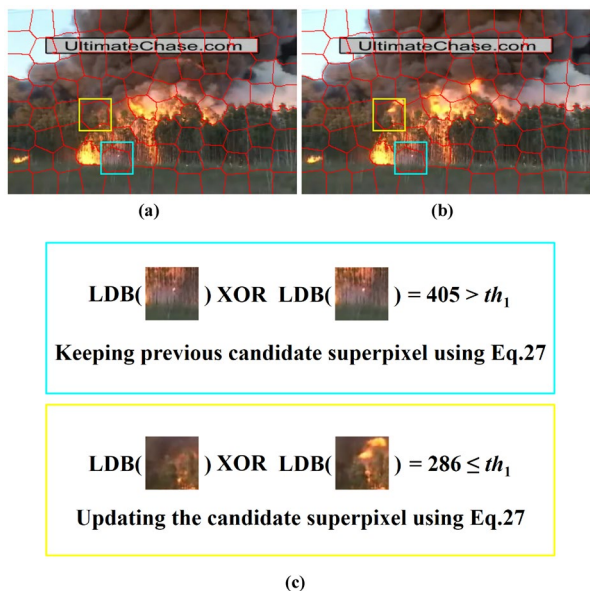




Fig. 6 Detection results of motion foreground (up) and suspicious fire region (down)

$$B_{h+1,m} = \begin{cases} B_{h,m}, & \text{if } \text{sum}(\text{XOR}(LDB_{h,m}, LDB_{h+1,m})) \leq th_1, \\ \alpha B_{h,m} + \beta D_{h+1,m} + (1 - \alpha - \beta) B_{1,m}, & \text{else} \end{cases}, \quad (27)$$

where h and $h + 1$ represent current frame and the next. XOR represents the exclusive or operation, and m is the index of candidate superpixels. $B_{h+1,m}$ is the m -th renewed candidate superpixel of the $(h + 1)$ -th frame and its LDB descriptor is $LDB_{h+1,m}$. $D_{h+1,m}$ denotes the corresponding region of $B_{h+1,m}$ in original frame, and th_1 is a decision threshold. The trade-off parameters α and β are set to control contributions of the m -th candidate superpixel $B_{h,m}$ in the h -th frame, and the original frame $D_{h+1,m}$ in the $(h + 1)$ -th frame. Here, we set $\alpha = 0.5$ and $\beta = 0.3$ for better performance in motion object localization.

Finally, by comparing with the original frame, we acquire a binary motion foreground according to the extent each image is renewed by the identification of motion as (26). Giving the least rectangle that could contain all white areas, final localization of suspicious fire region could be seen (see Fig. 6).

$$A_{h+1}(x, y) = \begin{cases} 255, & \text{if } |D_{h+1}(x, y) - B_{h+1}(x, y)| > th_2 \\ 0, & \text{if } |D_{h+1}(x, y) - B_{h+1}(x, y)| \leq th_2 \end{cases} \quad (28)$$

3.2 Suspicious fire region identification

Although many deep CNN based methods achieve qualified performance on fire images classification, the high computational cost is a main weakness. Thus, the paper proposes a light-weight network ESE-ShuffleNet to recognize fire regions. Our network is designed based on ShuffleNet V2 [23], and SE module [16] is firstly inserted in each unit of ShuffleNet. Adopting SE within each convolution unit would selectively emphasis contributing features by interrelation of convolution feature between channels [16]. Besides, the unit also established an inverted residual structure between two 1×1 group convolution layers instead of the original bottleneck architecture [13], so that the dimension of intermediate feature maps could be expanded. Applying this inverted residual architecture with an inner

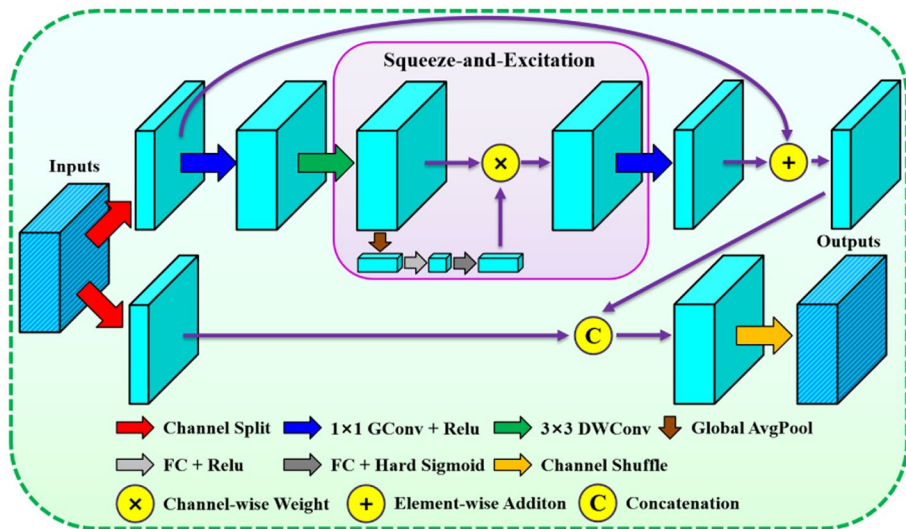


Fig. 7 The basic unit of ESE-ShuffleNet

SE model simultaneously, feature description would be strengthened at the most contributing portion. Combining 3×3 depthwise convolution with this inverted residual architecture, smaller computational cost is needed compared to those current design. In general, this newly formed architecture achieves higher classification accuracy and fewer computation.

- 1) **ESE-ShuffleNet architecture:** ShuffleNet V2 is a classical light-weight network adopted here as the basic frame of our network. Mainly due to the depthwise convolution and half-split channel scheme, computational cost of ShuffleNet V2 already stay at a low level. Besides, channel shuffle operation also enhances information communication between different groups of feature maps. These characteristics could benefit fire detection that only limited number of training data could be collected while a high-level of accuracy is required. Based on these advantages, we would like to redefine ShuffleNet V2 by two modifications to further conform our network in fire detection tasks.
- 2) **Inner SE module:** Since fire usually has obvious color and texture features that is greatly observable in images, adding an attention mechanism to network can emphasize features with fire characteristics. Therefore, we introduced an SE module to increase the attention of our network on fire features by channel-wise feature recalibration as shown in Fig. 7. Different from traditional post-positively manner [16], we insert this SE module between 3×3 depthwise convolution and the second group convolution to have the intermediate feature channels weighted. This design enables SE module to apply more feature maps in network while maintaining fewer parameters. It could not only expand the application range of attention, but also well enhance cross-channel communication in each unit, so that informative features could be strengthen to its most.
- 3) **Group convolution based inverted residual:** The original convolution design of one unit in ShuffleNet V2 has same input, middle and output channels, including two 1×1 pointwise convolutions and a depthwise convolution on one split of the channels. This is already an efficient network as the depthwise separable convolution and branch-split scheme achieves a greatly decrease of parameters. However, based on the concerns of

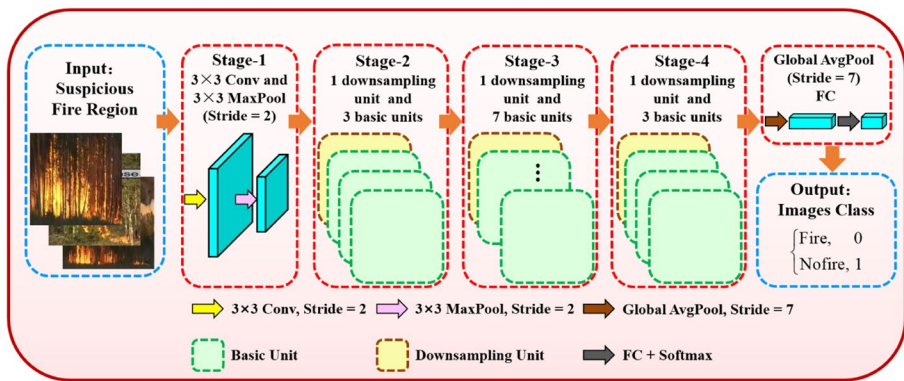


Fig. 8 The architecture of ESE-ShuffleNet

mostly used depthwise separable convolution by only limited accuracy, we would reform this existed network by two aspects: (i) inverted bottleneck residual structure [35] is firstly constructed on a branch of the ShuffleNet V2 unit. The first 1×1 layer is used to expand input channels for accuracy improvement, and the expanding feature maps would be reduced to original channel number by the second 1×1 layer for branch concatenation; (ii) original 1×1 convolution layers are replaced by 1×1 group convolution, so that parameters in our network could be reduced by groups. Meanwhile, benefiting from the inverted residual learning and our inner SE modification, the decreased accuracy by group scheme could be well recovered. In this way, the intermediate SE module could gain representatives between channels to its highest probability. Even though feature channels within the inverted residual increase compared to classical ShuffleNet V2 unit, the group convolutions that we inserted together with original depthwise design that higher number of channels devoted to fewer computations still have this new structure perform within a superior range. The increased performance by these widen feature maps for SE module also have this modification proved worthwhile.

Figure 8 exhibits the whole architecture of ESE-ShuffleNet consisting of four main stages. Stage 1 is a common implementation consisting of a 3×3 standard convolution and a 3×3 max pooling with stride 2. Stage 2 to Stage 4 is a fundamental part of the network that is utilized to realize features extraction and discrimination. In the downsampling unit (see Fig. 9), feature map size at every stage is halved while corresponding dimension doubling, and the different number of units follow by all maintain the same input-output channels. Based on the architecture of classical ShuffleNet V2 [23], Stage 3 in our network is designed to have the largest number of basic units. Finally, we obtain probability score for classification through a global average pooling and a fully connected layer.

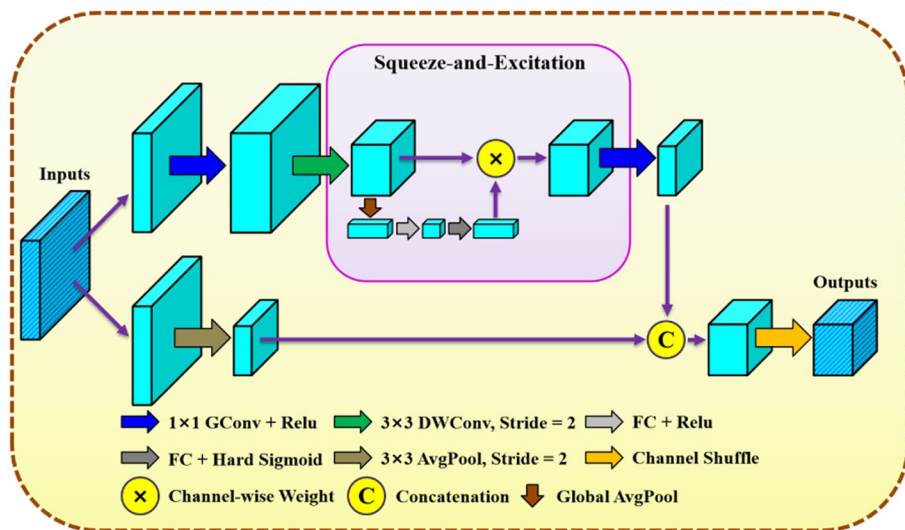


Fig. 9 The downsampling unit of ESE-ShuffleNet

4 Experiments

4.1 Benchmark datasets

In the experiments, we adopt six widely used datasets BSDS500 [2], Microsoft research Cambridge Version 2 (MSRC V2)¹, Corsican², Cair³, Foggia [10] and Dunning⁴ for validation. The former two datasets are used for superpixel segmentation, while the others are applied for fire recognition.

- **BSDS500:** This dataset contains three subsets of train, test and validation, with a total of 500 natural images of size 481×321 or 321×481 pixels. Each natural image has at least four manually labeled ground truth.
- **MSRC V2:** This dataset consists of 591 natural images with the resolution of 320×213 or 213×320 for object segmentation.
- **Corsican:** This fire dataset contains 1135 images captured in different environments where all images have been segmented manually as ground truth. This fire dataset can be downloaded for research purposes via a customized interface.
- **Cair:** This dataset contains 541 normal images and 110 images with fire from a variety of scenarios and different fire situations, such as intensity, luminosity, size, environment, etc.
- **Foggia:** This dataset contains a total of 31 video clips including 14 fire videos and 17 non-fire videos. We adopt this dataset mainly considering of its great challenge to fire

¹ <https://www.microsoft.com/en-us/research/project/image-understanding/>

² <http://cfdb.univ-corse.fr/>

³ <https://github.com/UIA-CAIR/Fire-Detection-Image-Dataset>

⁴ <https://collections.durham.ac.uk/files/r2d217qp536#.Xwl8g0UzaUn>

video tasks due to the interfering non-fire videos engaging fire-like objects and scene, and some other moving conditions such as cloud and fog.

- **Dunnings:** This dataset consists of three types of fire data, image, superpixel and video. Specifically, there are 36 fire videos and 12 non-fire videos, of which 4 videos are about forest fire.

4.2 Evaluation metrics

We adopt 9 metrics and divide them into two groups. The first group is for superpixel segmentation including boundary recall (BR) [8], achievable segmentation accuracy (ASA) [40] and weighted isoperimetric quotient (WIPQ) [3]. The second group consisting of Sensitivity, Specificity, Accuracy (ACC), Precision, Recall and F-measure are applied to evaluate the whole fire detection.

- 1) **Superpixel segmentation quality metrics:** BR is a standard measuring the extent to which segmented superpixel boundaries fall on ground truth boundaries [8], which is

$$BR = \frac{\sum_{i \in Gb} [(\min_{j \in Sb} |x_i - x_j| \leq 2) \wedge (\min_{j \in Sb} |y_i - y_j| \leq 2)]}{\text{num}(Gb)}, \quad (29)$$

here, Sb and Gb are the boundary results of superpixel segmentation and ground truth, $\text{num}(\cdot)$ is the pixels number.

ASA quantifies the greatest segmentation accuracy by assigning each segmented superpixel to ground truth to its maximum overlap.

$$ASA = \frac{1}{N_{v_k}} \sum_{k=1}^K \max \{|v_k \cap g_k|\}, \quad (30)$$

where v_k and g_k denote pixel sets of the k -th segmented superpixel and its corresponding ground truth.

As mentioned above, UE can evaluate the incorrect segmentation ratios of superpixel image. Different from BR and ASA, smaller UE indicates the accurate segmentation result.

$$UE = \frac{1}{N} \left(\sum_{|v_k \cap g_k| > \omega |v_k|} |v_k| \right) - 1, \quad (31)$$

where N is the total number of pixels, the parameter ω is set as 0.05 in terms of [3].

As for WIPQ, the regularity of superpixels could be evaluated by:

$$WIPQ = \frac{1}{N_{v_k}} \sum_{k=1}^K \frac{4\pi |v_k|^2}{S_k^2}, \quad (32)$$

where S_k indicates boundary pixels number of the k -th superpixel.

In general, UE, BR and ASA are used to evaluate segmentation accuracy, where BR tends to quantify boundary adherence. While higher BR and ASA value indicates better segmentation performance, WIPQ provides an observation of whether the superpixels are moderately segmented to shapes.

- 2) **Wildfire recognition quality metrics:** For the whole fire detection model, a result of fire or non-fire frame would be identified. Sensitivity [12] is adopted here to reflect the

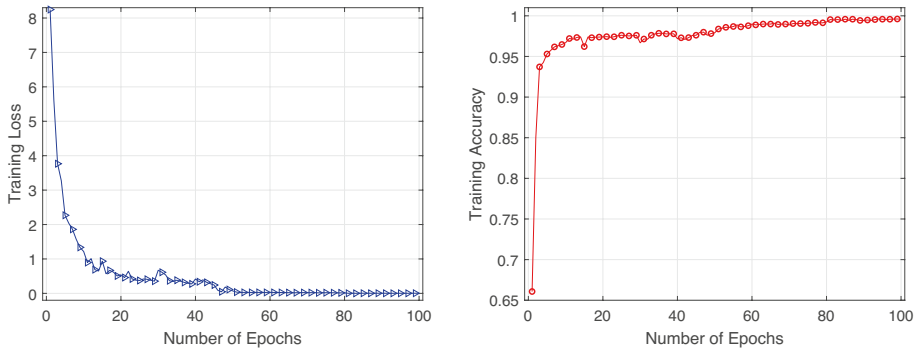


Fig. 10 The training loss and accuracy verse the number of epochs

ability that a method could identify fire video. Specificity [12] here indicates the ability a model could exclude non-fire video. ACC [23] shows the ratio of correctly identified frames of a video. F-measure [27] denotes the weighted harmonic average of Precision [27] and Recall [12], and can give a more objective assessment results when Precision and Recall conflict. By varying the discrimination thresholds, a receiver operating characteristic (ROC) curve by Sensitivity versus 1-Specificity and a PR curve by Precision versus Recall can be plotted. Area under the curve (AUC) calculates the area under PR or ROC curves, where higher value denotes better performance of this approach.

4.3 Implementation details

This subsection evaluates the proposed framework on both accuracy and efficiency, all experiments are implemented by hardware platform: Intel (R) Core (TM) i7-10700k CPU (8 cores 16 threads and 5.0 GHz), 32GB RAM (DDR4 3600MHz) and NVIDIA RTX 2080 Ti GPU (11GB video memory). And our softwares include: Python 3.6.10, PyTorch 1.5.0, OpenCV-Python 3.4.2, NumPy 1.18.1 and Windows 10 OS.

We conduct the training and testing of ESE-ShuffleNet on three publicly available datasets, Corsican, Cair and Foggia. All 1135 fire images of Corsican dataset and the 110 fire images and 541 non-fire images from Cair dataset are all used in network training. Whereas in Foggia, the 14 fire videos are divided into 4 fire videos capturing 90 frames for network training, and 10 fire videos for network testing. The other 17 non-fire videos are divided into 10 videos for testing and 7 videos that generating 774 non-fire images to balance the types of training set. Besides, four fire videos and four non-fire videos from Dunnings dataset are employed for network testing. Finally, 2650 images are selected for network training, and 28 video clips are used to test.

- 1) **Network training:** During training phase, all training images are re-sampled using linear interpolation with resolution 224×224. The proposed ESE-ShuffleNet was trained using Adam optimizer with initial learning rate at 0.001 and batchsize of 64. There are 100 epochs in network training and learning rate is multiplied by 0.2 at the 50-th and 80-th epochs. Figure 10 presents the training loss and accuracy curve of ESE-ShuffleNet, which shows the stability and rapid convergence with the increase of epochs.

- 2) **Network testing:** In testing phase, all testing videos are conveyed into the whole proposed framework, SCMM based superpixel segmentation and LDB based motion object localization followed by a suspicious region identification using ESE-ShuffleNet, to see the final accuracy of fire detection. The results of our framework would be evaluated later together with corresponding experimental details and quantitative analysis as shown in Sect. 4.4.

4.4 Results and discussion

- 1) **Superpixel segmentation evaluation:** We evaluate the performance of SCMM by comparing its accuracy and regularity with six well-known approaches: The comparison methods in Fig. 11 can be cited as SLIC⁶ [1], LRW⁵ [37], Waterpixels⁸ [24], LSC⁷ [8], GMMS¹⁰ [3] CAS⁹ [40]. Our visual evaluation is based on three test images from BSDS500 and MSRC V2 respectively. Each test image is divided into 200 superpixels by above six models. From Fig. 12, LRW exhibits excellent regularity, while local details are segmented relatively unsatisfied. Compared with LRW, SLIC shows slightly better local details while preserving regular boundaries. Although LSC and GMMS both achieve good accuracy, LSC shows better visual compactness, and GMMS has more appropriate boundary adherence. Waterpixels shows the most inferior accuracy even compared to LRW, while relatively good regularity are obtained. CAS segmented superpixels quite accurately, whereas regularity stays at a low level similar to GMMS. Comparing with above six methods, our superpixel approach presents a competitive result in terms of best local details and visual effect. Even though SCMM shows slightly worse regularity compared to the best approach LRW, the overall trade-off between different metrics still prove the effectiveness of this proposed method.

Next, the quantitative comparison between above six superpixel methods and our approach is illustrated in Fig. 12, and the numbers of segmented superpixels are 100, 200, 300, 400, 500, 600, 700, and 800, respectively. In addition, the BR, UE, ASA and WIPQ are calculated by the 200 images from BSDS500 test set and all 591 images from MSRC V2. From all above results, we can conclude that: (i) GMMS, CAS and SCMM all obtain satisfactory segmentation accuracy, and the proposed SCMM has the best BR, ASA and UE at 0.945, 0.973 and 0.151 on BSDS500; (ii) the WIPQ of our method is the highest among all methods except LRW, which indicates superior regularity of the segmentation.

- 2) **Layout of ESE-ShuffleNet with SE module:** In this section, we would discuss different manners of using SE module in ESE-ShuffleNet. As shown in Table 1, three models are selected including ESE-ShuffleNet without using SE module, using SE module in traditional post-positive manner and using SE module as the proposed framework. We can see that architecture abandoning SE module has the least network parameters, but all quantitative results are relatively unsatisfied. When post-positively setting SE as in SE-Net [16], network performance has been improved due to the channel-wise attention

⁵ <https://github.com/shenjianbing/lrw14>

⁶ <http://ivrl.epfl.ch/research/superpixels>

⁷ <http://jschenth.wweebly.com/projects.html>

⁸ <http://cmm.enscm.fr/~machairas/waterpixels.html>

⁹ <https://github.com/YuejiaoGong/CAS>

¹⁰ <https://github.com/ahban/GMMSP>

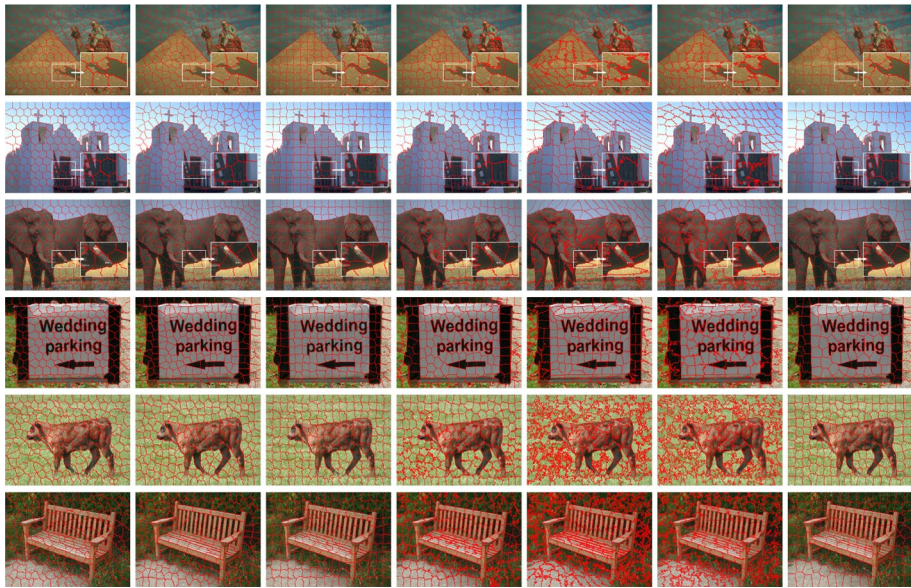


Fig. 11 Visual comparison of superpixels produced by various methods, the upper three lines belong to BSDS500 datasets, whereas others below are from MSRC V2 dataset, from left to right: SLIC [1], LRW [37], Waterpixels [24], LSC [8], GMMS [3], CAS [40] and Ours

by the SE module. However, adopting inserted SE module as in ESE-ShuffleNet presents even higher classification accuracy although network parameters increase to some extent. This is due to the inserted SE module can receive wider feature maps and better infer the importance of each channel. Therefore, a wise selection of ESE-ShuffleNet layout with inserted SE modules is verified largely contributing to network performance.

- 3) **Network parameter evaluation:** In this part, we discuss the influence of varying number of groups in group convolution for ESE-ShuffleNet performance. Table 2 shows the overall configuration of our network by four grouping schemes. It can be seen that the higher numbers of group inserted, the less parameters network calculates. When only one group is inserted, group convolution becomes pointwise convolution that the most parameters appear. Besides, floating-point operations per second (FLOPs) and multiply-adds (MAdds) can describe the computational cost a network required as shown in Fig. 13. While FLOPs represents the theoretical amount of floating point arithmetic, MAdds refers to addition and multiplication. We also compare the classification performance of our network on four grouping schemes by ACC and F-measure, and the details are shown in Fig. 14. It could be observed that classification performance is inversely proportional to group number except for the single group scheme. Thus, experimental results show that the number of groups at 2 is the most efficient scheme in terms of computational cost and classification accuracy.
- 4) **Network performance evaluation:** Some of the detection results of tested videos on two datasets using ESE-ShuffleNet are reported in Table 3, where accuracy with corresponding processing information are illustrated. Since some training images also come from Foggia, the same style of video resulted in a slightly more accurate results compared to Dunning testing videos as shown in Table 3. In spite of this, figures in Table 3 still present a highly-accepted level above 0.91.

Fig. 12 Performance evaluation of superpixel segmentation algorithms using the test images of BSDS500 ► and MSRC V2

In Fig. 15, probability scores are demonstrated to show the assured extent of precision before final classification. Finally, the ROC and PR curves of our ESE-ShuffleNet are presented in Fig. 16 where more ROC and PR approaches top-left and top-right, the better fire videos are classified. It could be seen that both AUC values of our method reach a relatively high level at 0.9574 and 0.9564 respectively.

Figure 17 shows the quantitative results on Foggia and Dunnings between ESE-ShuffleNet and other two classical CNNs, ResNet-34 [13] and DenseNet-121 [17], and eight current light-weight networks, MobileNet V1¹¹ [15], MobileNet V2¹² [35], MobileNet V3¹³ [14], ShuffleNet V1¹⁴ [45], ShuffleNet V2¹⁵ [23], SqueezeNet¹⁶ [19] and MnasNet¹⁷ [39] and GhostNet¹⁸ [11]. Sensitivity, Specificity, ACC, Precision, Recall and F-measure comparisons are illustrated in Fig. 17, while requirements of parameters and memory are demonstrated in Fig. 18 (except for the most complex ResNet-34 and DenseNet-121) and Table 4. Comparing with ShuffleNet V2, the parameters of our proposed ESE-ShuffleNet are reduced by about 1/3, while classification accuracy rises to 99.61%. Compared with our network, although the state-of-the-art MnasNet, MobileNet V3 and GhostNet achieve satisfactory classification performance in terms of ACC and F-measure, their parameters are at about twice of us. SqueezeNet and ShuffleNet V1 have advantages in computation than ESE-ShuffleNet, but ACC on Foggia Dataset only reaches 95.60% and 95.38% relatively inferior compared to 99.61% of ESE-ShuffleNet. In addition, classical deep CNNs ResNet-34 and DenseNet-121 achieve good accuracy in forest fire detection, but network parameters (21.278M and 6.948M) and total memory (1044.48MB and 4823.04MB) stay at a relatively high-level. Overall, benefiting from the effective use of inserted channel-wise attention, our network can obtain higher classification accuracy and lower parameters compared with MobileNet V1 and MobileNet V2.

In general, it could be observed that our network has the highest Specificity, ACC, Precision and F-measure on both Foggia and Dunnings datasets, which are 0.9985, 0.9961, 0.9986, 0.9963, and 0.9605, 0.9456, 0.9612, 0.9459, respectively. Even though our approach achieved slightly moderate Sensitivity and Recall compared with MobileNet V3 in Dunnings dataset, its relative level among all methods still prove to be efficient in fire detection. For model complexity, parameters that we require is only more than ShuffleNet V1 and SqueezeNet. Moreover, total memory of our network is similar to ShuffleNet V1, less than MobileNet V1, MobileNet V3, SqueezeNet and MnasNet. Overall, our network shows comprehensively competitive performance, especially on accuracy and efficient computation that indicate real-time capability of forest fire alarm.

¹¹ <https://github.com/wjc852456/pytorch-mobilenet-v1>

¹² <https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet>

¹³ <https://github.com/kuan-wang/pytorch-mobilenet-v3>

¹⁴ <https://github.com/AlloNighthawk/ShuffleNet-v1>

¹⁵ <https://github.com/ericusun99/Shufflenet-v2-Pytorch>

¹⁶ <https://github.com/DeepScale/SqueezeNet>

¹⁷ <https://github.com/tensorflow/tpu/tree/master/models/official/mnasnet>

¹⁸ <https://github.com/huawei-noah/CV-Backbones>

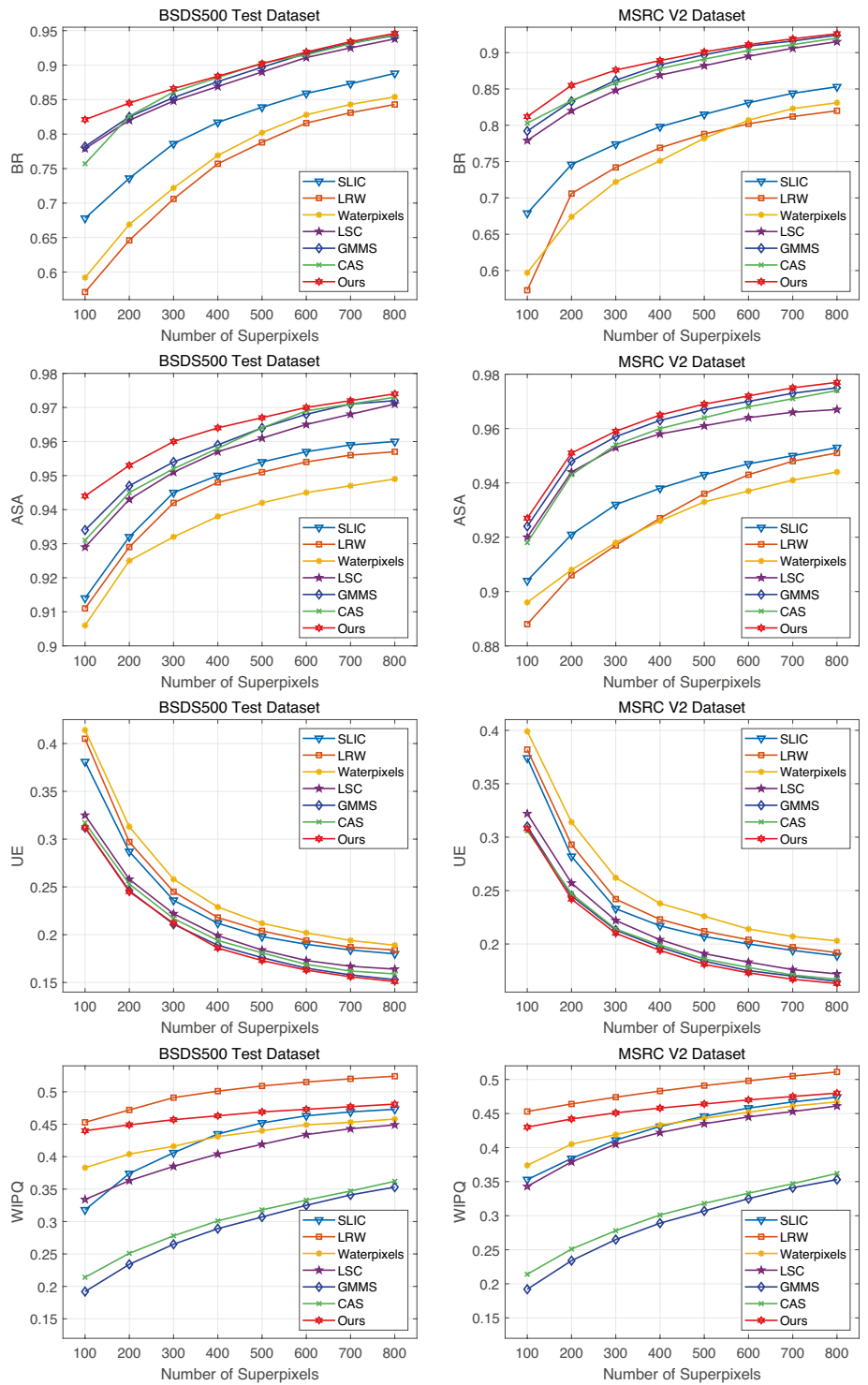


Table 1 Analysis of the use of SE module in the proposed ESE-ShuffleNet

Models	Parameters	FLOPs	Sensitivity	Specificity	ACC	Precision	Recall	F-measure
ESE-ShuffleNet (without SE)	1.427M	200.71M	0.9761	0.9936	0.9846	0.9939	0.9761	0.9849
ESE-ShuffleNet (post-positively SE)	1.480M	200.76M	0.9881	0.9936	0.9908	0.9940	0.9881	0.9910
ESE-ShuffleNet	1.636M	200.91M	0.9940	0.9985	0.9961	0.9986	0.9940	0.9963

Table 2 The configuration of ESE-ShuffleNet

Layers	Output	KSize	Stride	Repeat	Output Channels (Groups)			
					1	2	4	8
Image	224×224				3	3	3	3
Conv	112×112	3×3	2	1	64	64	64	64
MaxPool	56×56	3×3	2	1	64	64	64	64
Stage2	28×28		2	1	128	128	128	128
	28×28		1	3	128	128	128	128
Stage3	14×14		2	1	256	256	256	256
	14×14	1	7	256	256	256	256	256
Stage4	7×7		2	1	512	512	512	512
	7×7		1	3	512	512	512	512
Conv	7×7	1×1	1	1	1024	1024	1024	1024
Global AvgPool	1×1	7×7	1	1	1024	1024	1024	1024
FC					2	2	2	2
Parameters					2.455M	1.636M	1.226M	1.022M

4.5 Limitations and discussions

Although the proposed framework can achieve good efficiency and accuracy on forest fire detection as demonstrated in above experiments, some limitations still exist. The proposed network is designed based on light-weight network ShuffleNet such that basic network complexity keeps at a low-level. However, the proposed method isn't the most light-weight network as discussed since modified architectures for better accuracy increase computation to some extent. In addition, our superpixel segmentation and moving object detection are implemented on CPU. Although the current version is efficient, implementing them on GPU can further reduce the running time. In the future, we will optimize the proposed framework so that it can run completely on GPU.

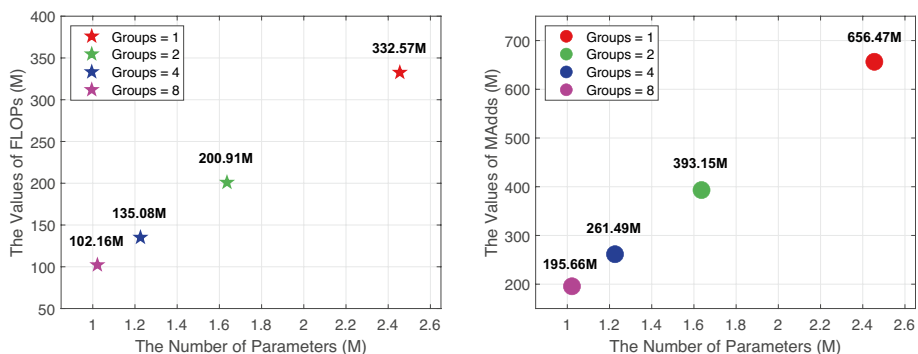
**Fig. 13** The computational cost of ESE-ShuffleNet with four grouping schemes

Fig. 14 Performance evaluation of ESE-ShuffleNet with four grouping schemes

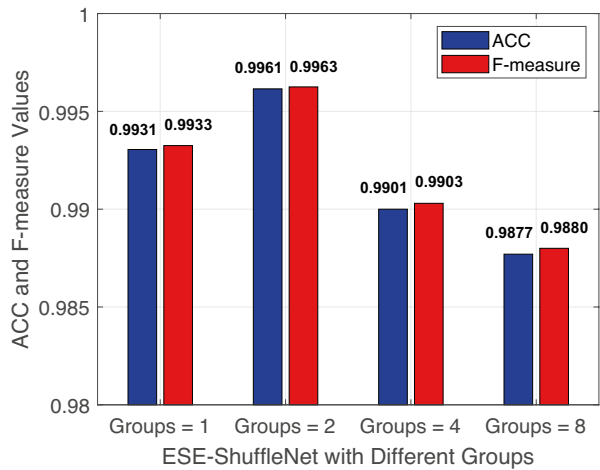
















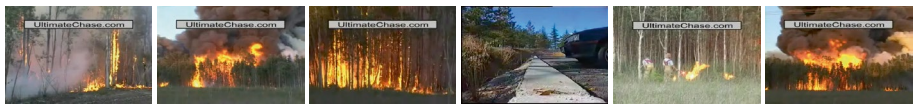
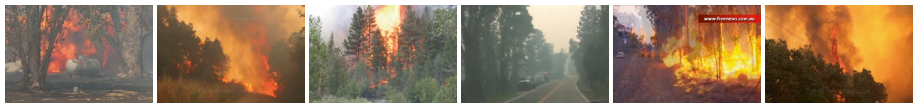


Table 3 The detection results of video clips on different datasets

Index	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6	Video 7	Video 8
Videos (Foggia)								
Selected Frames	26	25	21	20	26	41	9	61
Accuracy	0.9615	1.000	0.9545	1.000	0.9615	1.000	1.000	0.9836
Length/Test Cost(s)	17/7.7	16/ 7.4	14/6.3	13/6	17/7.8	40/12.4	8/2.8	60/18.3
Videos (Dunning)								
Selected Frames	537	932	1147	717	187	395	903	1744
Accuracy	0.9143	0.9163	0.9393	0.9247	0.9679	0.9241	0.9955	0.9948
Length/Test Cost(s)	179/193	311/ 276	383/346	287/220	62/56	132/118	361/272	982/533



(a) Fire: 95.11%, (b) Fire: 96.85%, (c) Fire: 97.49%, (d) Fire: 3.16%, (e) Fire: 97.94%, (f) Fire: 99.87%,
Non-fire: 4.89% Non-fire: 3.15% Non-fire: 2.51% Non-fire: 96.84% Non-fire: 2.06% Non-fire: 0.13%



(g) Fire: 99.85%, (h) Fire: 99.98%, (i) Fire: 95.73%, (j) Fire: 3.86%, (k) Fire: 97.62%, (l) Fire: 98.42%,
Non-fire: 0.15% Non-fire: 0.02% Non-fire: 4.27% Non-fire: 96.14% Non-fire: 2.38% Non-fire: 1.58%

Fig. 15 The predicted probability scores of fire and non-fire frames, up: Foggia dataset, down: Dunning dataset

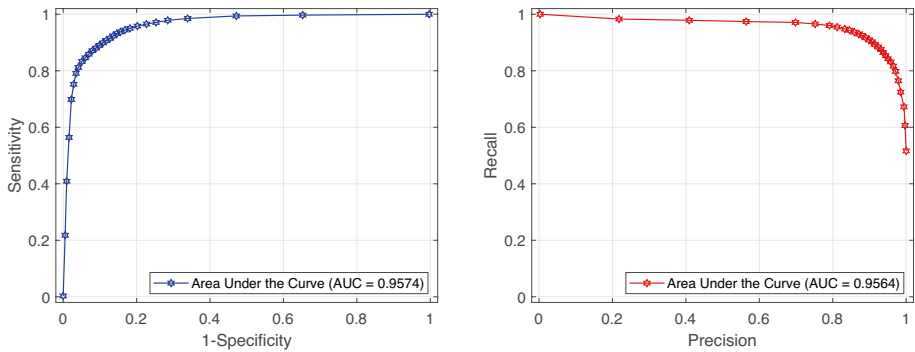


Fig. 16 The ROC and PR curves with corresponding AUC values

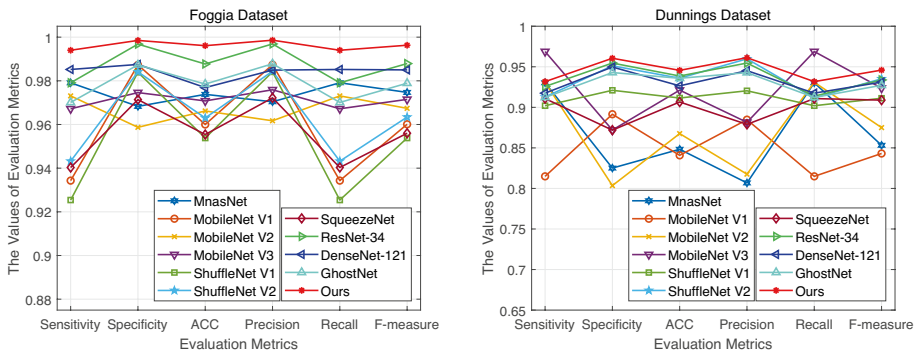


Fig. 17 Overall comparison of several popular light-weight networks and Ours

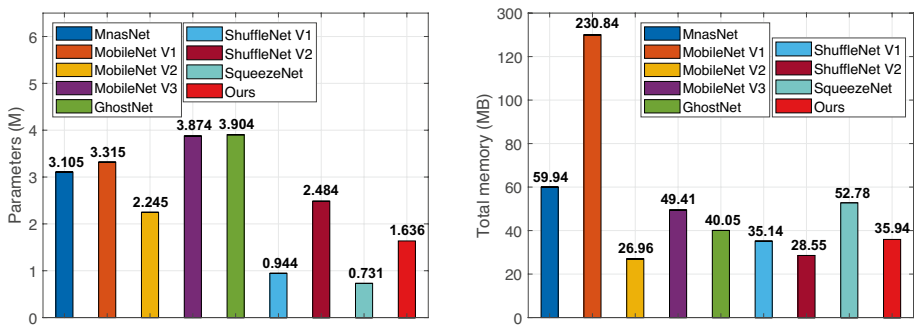


Fig. 18 Memory and parameter comparison of several popular light-weight networks with Ours

Table 4 Memory and parameters comparison between popular CNNs with Ours

Methods	Parameters (M)	Total memory (MB)
MnasNet [39]	3.105M	59.94MB
MobileNet V1 [15]	3.315M	230.84MB
MobileNet V2 [35]	2.245M	26.96MB
MobileNet V3 [14]	3.874M	49.41MB
ShuffleNet V1 [45]	0.944M	35.14MB
ShuffleNet V2 [23]	2.484M	28.55MB
SqueezeNet [19]	0.731M	52.78MB
ResNet-34 [13]	21.278M	1044.48MB
DenseNet-121 [17]	6.948M	4823.04MB
GhostNet [11]	3.904M	40.05MB
Ours	1.636M	35.94MB

5 Conclusions

In this paper, we presented a novel framework to detect the wildfire by suspicious fire region proposal and ESE-ShuffleNet identification. Specifically, a new superpixel segmentation approach driven by CMM that could be applied to a wide-range of superpixel segmentation task was developed. Here, we combined this novel superpixel method with LDB description and motion detection as targeting specifically at fire region localization. Next, the detected suspicious fire regions were input into the proposed light-weight network ESE-ShuffleNet to identify fire or non-fire for real-time alarm. The proposed framework was highly applicable in real scenes based on challenging testing videos. Comparative experiments with six superpixel segmentation approaches and seven competitive networks for fire detection demonstrate high accuracy and low computational cost our proposal achieved that could comprehensively affirm the feasibility of this method. Not only accurate and fast fire detection that we aimed at in this paper, the new designed superpixel segmentation approach and two modification ideologies that we proposed in ESE-ShuffleNet are all believed to be well contributed in relative application areas that share the same methodology.

Acknowledgements This work was supported by the National Nature Science Foundation of China under Grant 61872143.

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 34(11):2274–2282
2. Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 33(5):898–916
3. Ban Z, Liu J, Cao L (2018) Superpixel segmentation using Gaussian mixture model. *IEEE Trans Image Process* 27(8):4105–4117
4. Barmoutis P, Dimitropoulos K, Kaza K, Grammalidis N (2019) Fire detection from images using faster R-CNN and multidimensional texture analysis. In *Proceedings of IEEE Int Conf Acoustics Speech Signal Process (ICASSP)*, pp. 8301–8305
5. Bochkovskiy A, Wang CY, Liao HYM (2020) YOLO v4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*

6. Borges PVK, Izquierdo E (2010) A probabilistic approach for vision-based fire detection in videos. *IEEE Trans Circuits Syst Video Technol* 20(5):721–731
7. CA GS, Bhowmik N, Breckon TP (2019) Experimental exploration of compact convolutional neural network architectures for non-temporal real-time fire detection. In *Proceedings of IEEE Int Conf Mach Learn Appl (ICMLA)*, pp. 653–658
8. Chen J, Li Z, Huang B (2017) Linear spectral clustering superpixel. *IEEE Trans Image Process* 26(7):3317–3330
9. Dunning AJ, Breckon TP (2018) Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection. In *Proceedings of IEEE Int Conf Image Process (ICIP)*, pp. 1558–1562
10. Foggia P, Saggese A, Vento M (2015) Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE Trans Circuits Syst Video Technol* 25(9):1545–1556
11. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) GhostNet: More features from cheap operations. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 1580–1589
12. Hashemzadeh M, Zadamehdi A (2019) Fire detection for video surveillance applications using ICA k-medoids-based color model and efficient spatio-temporal visual features. *Expert Syst Appl* 130:60–78
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 770–778
14. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for MobileNetV3. In *Proceedings of IEEE Int Conf Comput Vis (ICCV)*, pp. 1314–1324
15. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*
16. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 7132–7141
17. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 4700–4708
18. Huang H, Kuang P, Fan L, Shi H (2020) An improved multi-scale fire detection method based on convolutional neural network. In *Proceedings of Int Comput Conf Wavelet Active Med Tech Inf Process (ICCWAMTIP)*, pp. 109–112
19. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint, arXiv:1602.07360*
20. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In *Proceedings of Adv Neural Inf Process Syst (NIPS)*, pp. 1097–1105
21. Kuczma M (2009) An introduction to the theory of functional equations and inequalities: Cauchy's equation and Jensen's inequality. Springer Science & Business Media
22. Li P, Zhao W (2020) Image fire detection algorithms based on convolutional neural networks. *Case Stud Therm Eng* 19
23. Ma N, Zhang X, Zheng H, Sun J (2018) ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *Proceedings of European Conf Comput Vision (ECCV)*, pp. 116–131
24. Machairas V, Faessel M, Cárdenas-Peña D, Chabardes T, Walter T, Decencière E (2015) Waterpixels. *IEEE Trans Image Process* 24(11):3707–3716
25. Matukhina O, Amaeva L, Merzlyakov S (2020) Fire detection system with utilization of industrial video surveillance system. In *Proceedings of Int Multi-Conf Ind Eng Modern Tech (FarEastCon)*, pp. 1–5
26. Muhammad K, Ahmad J, Baik SW (2018) Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* 288:30–42
27. Muhammad K, Ahmad J, Lv Z, Bellavista P, Yang P, Baik SW (2018) Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Trans Syst Man Cybern A Syst Humans* 49(7):1419–1434
28. Muhammad K, Ahmad J, Mehmood I, Rho S, Baik SW (2018) Convolutional neural networks based fire detection in surveillance videos. *IEEE Access* 6:18174–18183
29. Muhammad K, Khan S, Elhoseny M, Ahmed SH, Baik SW (2019) Efficient fire detection for uncertain surveillance environment. *IEEE Trans Ind Inform* 15(5):3113–3122
30. Nguyen TM, Wu QJ (2012) Fast and robust spatially constrained Gaussian mixture model for image segmentation. *IEEE Trans Circuits Syst Video Technol* 23(4):621–635

31. Redmon J, Farhadi A (2018) YOLO v3: An incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
32. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 6:1137–1149
33. Ren X, Malik J (2003) Learning a classification model for segmentation. In *Proceedings of IEEE Int Conf Comput Vis (ICCV)*, pp. 10–17
34. Saeed F, Paul A, Hong WH, Seo H (2020) Machine learning based approach for multimedia surveillance during fire emergencies. *Multimed Tools Appl* 79:16201–16217
35. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 4510–4520
36. Saponara S, Elhanashi A, Gagliardi A (2020) Exploiting R-CNN for video smoke/fire sensing in anti-fire surveillance indoor and outdoor systems for smart cities. In *Proceedings of IEEE Int Conf Smart Comput (SMARTCOMP)*, pp. 392–397
37. Shen J, Du Y, Wang W, Li X (2014) Lazy random walks for superpixel segmentation. *IEEE Trans Image Process* 23(4):1451–1462
38. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of Thirty-first AAAI Conf Artif Intell (AAAI-17)*
39. Tan M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, Le QV (2019) MnasNet: Platform-aware neural architecture search for mobile. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 2820–2828
40. Xiao X, Zhou Y, Gong Y (2018) Content-adaptive superpixel segmentation. *IEEE Trans Image Process* 27(6):2883–2896
41. Xie X, Xie G, Xu X, Cui L (2019) Adaptive high-precision superpixel segmentation. *Multimed Tools Appl* 78:12353–12371
42. Yang TJ, Howard A, Chen B, Zhang X, Go A, Sandler M, Sze V, Adam H (2018) Netadapt: Platform-aware neural network adaptation for mobile applications. In *Proceedings of European Conf Comput Vision (ECCV)*, pp. 285–300
43. Yang X, Cheng KT (2012) LDB: An ultra-fast feature for scalable augmented reality on mobile devices. In *Proceedings of IEEE Int Sym Mixed Augment Reality (ISMAR)*, pp. 49–57
44. Zhang X, Qian K, Jing K, Yang J, Yu H (2020) Fire detection based on convolutional neural networks with channel attention. In *Proceedings of Chinese Autom Congr (CAC)*, pp. 3080–3085
45. Zhang X, Zhou X, Lin M, Sun J (2018) ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of IEEE Conf Comput Vis Pattern Recognit (CVPR)*, pp. 6848–6856

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.