A Deep Learning Framework for Autonomous Flame Detection

Zhenglin Li[a,*], Lyudmila Mihaylova[a], Le Yang[b]

[a] *Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK*
[b] *University of Canterbury, Christchurch, New Zealand*

Abstract

This paper proposes a novel framework of flame region-based convolutional neural network for autonomous flame detection. The task of flame detection is especially challenging since flames have greater diversity in colour, texture, and shape than regular rigid objects. To cope with these difficulties due to the various appearances and unclear edges of flames, a proposal generation approach is developed to effectively select candidate flame regions based on two crucial properties of flames, i.e., their dynamics and colours. The candidate flame regions together with a convolutional feature map are further processed by additional layers to output detected flames. The diversity in flame colours is well represented by approximating the distribution using a Dirichlet Process Gaussian mixture model with variational inference. The proposed framework is evaluated on publicly available videos and achieves an average frame-wise accuracy higher than 88%, which outperforms the state-of-the-art methods.

Keywords: Flame detection, Flame R-CNN, Dirichlet process Gaussian mixture model, Variational inference

[*] Corresponding author
Email address: zhenglin.li@sheffield.ac.uk (Zhenglin Li)

1. Introduction

1.1. Background

Fires cause great property damage and human casualties every year. The huge losses make fire detection techniques essential for the modern society and daily life. Compared with conventional techniques using sensors, frameworks applying machine learning and computer vision methods can detect fires more accurately and efficiently by recognising flames in videos [1] . These methods have several outstanding advantages. First, they are capable of detecting fires accurately in large geographical areas, e.g., wildfires can be detected in videos by trained deep neural networks [2], while the sensor-based techniques cannot provide reliable results in such environments. Next, the video-based frameworks are robust to environmental changes. Third, these approaches can provide faster and more accurate solutions than conventional techniques [3]. For example, the optical flow estimation algorithm [4, 5] and temporal wavelet transform [6] can achieve a high detection rate while keeping the false alarm rate low. Finally, the newly developed frameworks using computer vision methods can be easily incorporated into the existing monitoring systems without high extra costs. Similar to other tasks of computer vision [7, 8], recently developed approaches for flame recognition in videos can be subdivided into two main groups, methods based on conventional computer vision techniques of feature extraction, and frameworks using deep neural networks.

Although the vision-based flame detection techniques have been developing fast, the accurate detection of flames is still a challenging task. Different from most rigid objects, flames usually have unclear edges and irregular shapes because of their non-rigid property. Various burning materials and combustion intensities further result in a large diversity of flame colours. Some weak flames are even semi-transparent and can hardly be detected in a short time. The diversity in shape and colour leads to the various appearances of flames, making their detection more difficult than most rigid targets. Furthermore, flames may occupy small areas in the scene if fires happen at a distant place from cameras or at the beginning of combustion.

To tackle these difficulties, a framework of flame Region based CNN (R-CNN) is proposed to autonomously detect flames in complex environments. The contributions of this paper can be summarised as follows.

- The developed framework processes candidate flame regions to prevent the features of flames from being overwhelmed by those of the background or other objects. It can robustly distinguish flames of various appearances from distracting objects through fusing the crucial features described by a probabilistic flame colour model, convolutional layers and online robust principal component analysis (R-PCA).
- To effectively select candidate regions of flames that are non-rigid and diverse in appearances, a flame proposal generation approach is developed and included within the proposed framework by utilizing the dynamic and colour properties, which is different from the methods in fast R-CNN [9, 10] and faster R-CNN [11, 12]. These two characteristics are effective in selecting flames, especially the weak and distant ones. Specifically, the motion of flames, even the semi-transparent ones, can be detected accurately by an online R-PCA algorithm conducted on the R channels of frames.
- A DPGMM-based flame colour model with variational inference is proposed to model the diverse colours of flames. The distribution of colours is approximated by a Gaussian mixture model whose prior is set to a Dirichlet process. As such, the number of clusters can be learned from training data instead of being set empirically. The DPGMM based flame colour model is trained using variational inference that can scale to a large volume of training data and thus achieve accurate estimation of the distribution.

The paper is organised as follows. The proposed DPGMM flame colour model with variational inference and framework of flame R-CNN are introduced in Section 2. Subsequently, the framework is evaluated and its performance is discussed in Section 3. The main results of this work are summarised in Section 4.

1.2. Related Work

Considerable efforts on autonomous flame detection have been carried out, contributing to the fast development of the associate techniques. Among the approaches using conventional machine learning and computer vision methods, various rules and diverse features have been proposed based on the knowledge of flames. Since the number or dimensionality of features is not very large, few methods for flame detection conduct feature selection before further processing [13]. Instead, fusion is commonly employed to enhance the robustness and effectiveness of the developed methods [14, 15]. These frameworks can provide frame-wise decisions on the existence of flames or detect regions containing flames, based on the designed rules or developed features together with classifiers [14, 15]. To achieve satisfactory detection performance, most existing works rely on features built on the crucial properties of flames, such as colour [4, 16], texture [15, 17], and shape [18], which make flames distinguishable from common distracting objects (e.g., lights and pedestrians in red). All the visual attributes of flames mentioned above vary significantly over time due to winds and the airflow caused by heat, so features describing the dynamic characteristics also play an important role in flame detection in videos [5, 15, 19].

To decrease the computation load and mitigate the disturbance of non-flame objects, a motion detection phase is commonly utilized as a preprocessing step to filter out static regions, such as the sun and steady lights. Many background subtraction methods have been embedded into flame detection systems, including the adaptive background subtraction [19], Gaussian mixture model (GMM) based background subtraction [14] and motion history images [20].

Besides motion detection, colour models of flames are also widely employed and have been proven effective and efficient in the selection of candidate flame pixels. Selective rules in the RGB and YCbCr colour spaces were proposed in [21] and [16], respectively. However, the colours of flames are quite diverse because of the various burning materials, different intensities of combustion, and the presence of smoke, which makes it difficult to model the colours with manually designed rules. Therefore, an increasing number of researchers focus on training models with real flame pixels. In [22], the ratio of the Cb and Cr values of flame pixels is modelled using a univariate Gaussian distribution. Additionally, flame colours are described by a GMM with the number of mixture components being set empirically [19]. However, the number of clusters is not intuitively known to researchers and the training of the model is not dynamic, meaning that it needs to be re-trained when new data are available [23]. To address the former problem, the distribution of flame colours is modelled by a Dirichlet process Gaussian mixture model (DPGMM) of which the cluster number and other parameters can be learned from training data using Gibbs sampling [6].

To obtain reliable results, features that describe the dynamic properties of flames are usually extracted to further verify the existence of flames in candidate regions. The flickering property of flames can be described by analysing the temporally changing patterns of flame pixels in the wavelet domain [19]. Moreover, optical flow estimation methods also work well in describing the dynamics of combustion regions [4, 5].

Final decisions of the existence of flames need to be made based on the extracted features. Different classifiers are typically applied to reduce the false alarm rate and enhance detection performance [24]. Typical methods include support vector machine [25], shallow neural network [5], fuzzy finite automata [26], and AdaBoost [27]. Apart from using classifiers, some work estimates the probabilities of flames in the scene and makes hard decisions using thresholds [22].

The approaches discussed above use pixel-wise features or low-dimensional features of regions. To effectively describe the texture or shapes unique to flames, deep convolutional neural networks (CNNs) are employed on flame detection because of their excellent performance in many tasks of computer vision. A particular architecture of CNN called SqueezeNet [28] which has fewer parameters than other networks is used in [29] for flame detection. To effectively detect flames, [30] used the framework of you only look once (YOLO) [31]. Additionally, candidate regions were selected using a colour model and classified by a CNN in [32]. Since the dynamic properties of flames are crucial, a two-stream CNN is applied to candidate flame regions in [33] to exploit the temporal information.

2. Methodology

Most tasks of object detection work on rigid targets, such as vehicles, pedestrians and animals, which have limited diversity in appearance within each class. Different from rigid objects, flames have a rich variety of shapes and colours, which makes it challenging to generate proposals of flames in the same way as the frameworks of rigid object detection, e.g. using the selective search algorithm [9, 10] and region proposal network [11, 12]. Additionally, some flames, especially weak ones, are of semi-transparent colours, which induces unclear edges and visibility of the objects behind them. Therefore, it is more difficult to generate proposals of flames than rigid objects. In the two-stage frameworks of object detection, region of interest (RoI)s of objects are selected in the first stage, based on which the classification and bounding box regression are conducted in the second stage. Therefore, the generation of proposals has a great influence on the performance of flame detection frameworks. Specifically, failing to generate proposals over regions of flames will lead to accuracy degeneration. To address this problem, a novel framework of flame R-CNN is proposed in this paper for the task of flame detection, in which the proposals are generated based on features describing the characteristics unique to flames. As mentioned in Section 1, the colours and dynamics are two crucial properties of flames. Therefore, a flame proposal generation approach is developed based on these two properties. The dynamic characteristic is described by the online R-PCA algorithm of which the details are provided in Appendix A, while the colour property is modelled by the proposed DPGMM based flame colour model introduced in this section.

2.1. Proposed Flame Colour Model Based on DPGMM with Accelerated Variational Inference

The distribution of flame colours is modelled by a GMM of which the prior is set to a Dirichlet process (DP) [6]. Denote the colour of a flame pixel as a vector $\mathbf{x}_i = [R_i, G_i, B_i]^\mathsf{T}$. The generative model is given by

$$\boldsymbol{\theta}_i \sim DP(\alpha_0, G_0), \tag{1}$$

$$\mathbf{x}_i|\boldsymbol{\theta}_i \sim \mathcal{N}(\mathbf{x}_i; \boldsymbol{\theta}_i), \tag{2}$$

where $\boldsymbol{\theta}_i$ denotes the parameters of the Gaussian component related to $\mathbf{x}_i$. According to the definition of the DP introduced in Appendix B.1, $\{\mathbf{x}_i\}_{i=1}^N$ are generated from a mixture with an unbounded number of clusters, meaning that K does not need to be set empirically before training the model. Combined with the stick-breaking process introduced in Appendix B.1, the generative model can be represented as follows

$$\boldsymbol{\pi} \sim \text{GEM}(\alpha_0), \tag{3}$$

$$\boldsymbol{\theta}_k^* \sim G_0, \tag{4}$$

$$z_i \sim \boldsymbol{\pi}, \tag{5}$$

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{x}_i; \boldsymbol{\theta}_{z_i}^*), \tag{6}$$

where the distribution of $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$ is defined in [34], and $\boldsymbol{\theta}_k^* \triangleq \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ denotes the parameters of the k-th component. According to the mixture model theory [35], an observation $\mathbf{x}_i$ is generated by first specifying a cluster indexed by z_i which is distributed according to $\boldsymbol{\pi}$. Afterwards, $\mathbf{x}_i$ is sampled from the chosen Gaussian component with the parameters $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{z_i}^*$. As such, the generative model can be interpreted as an 'infinite mixture model'. The estimated number of clusters may increase as more training data are given. In reality, the cluster number K would be finite when given a finite number of data.

The clusters of the mixture model can be learned from the training data by variational inference. Denote $\mathbf{z} = \{z_i\}_{i=1}^N$ as the set of indicator variables of the training data, $\boldsymbol{\beta} = \{\beta_k\}_{k=1}^K$ as the set of β_ks drawn independently from a Beta distribution $\text{Beta}(1, \alpha_0)$, and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k^*\}_{k=1}^K$ as distinct components sampled independently from the base distribution $G_0(\boldsymbol{\theta}^*|\lambda)$. Since $\boldsymbol{\pi}$ is defined based on $\boldsymbol{\beta}$ [37], $\boldsymbol{\beta}$ is approximated instead of $\boldsymbol{\pi}$. Let $\mathbf{W} = \{\boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{z}\}$ be the collection of all the latent variables. The probability of a testing sample $\mathbf{x}'$ being a flame pixel based on its colour can be written as

$$
\begin{aligned}
p(\mathbf{x}'|\mathbf{X}) &= \int p(\mathbf{x}'|z', \mathbf{W}, \mathbf{X})p(z'|\mathbf{W}, \mathbf{X})p(\mathbf{W}|\mathbf{X})dz'd\mathbf{W} && (7)\\
&= \int p(\mathbf{x}'|z', \boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{z}, \mathbf{X})p(z'|\boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{z}, \mathbf{X})p(\mathbf{W}|\mathbf{X})dz'd\mathbf{W} && (8)\\
&= \int p(\mathbf{x}'|z', \boldsymbol{\Theta})p(z'|\boldsymbol{\beta})p(\mathbf{W}|\mathbf{X})dz'd\mathbf{W} && (9)\\
&= \int p(\mathbf{x}'|\boldsymbol{\theta}_{z'}^*)p(z'|\boldsymbol{\beta})p(\mathbf{W}|\mathbf{X})dz'd\mathbf{W}, && (10)
\end{aligned}
$$

where z' is the indicator variable of $\mathbf{x}'$. The first term $p(\mathbf{x}'|\boldsymbol{\theta}_{z'}^*)$ in Eq. (10) can be calculated by (6), while $p(z'|\boldsymbol{\beta})$ can be calculated according to Eq. (5) and [37]. The predictive density $p(\mathbf{x}'|\mathbf{X})$ is available if the posterior $p(\mathbf{W}|\mathbf{X})$ is known. However, this posterior is not tractable and thus needs to be approximated rather than being calculated analytically. The inference of the flame colour model can be implemented by the Gibbs sampling [6], of which the high computational complexity limits the quantity of training data in use and affects the performance of trained models. Therefore, the target posterior distribution will be approximated by the variational inference algorithm.

According to the mean-field approximation, a family of factorized distributions are proposed to approximate the target posterior [36, 37]. The distribution within the family that minimizes the Kullback-Leibler divergence (KLD) between itself and the exact posterior is chosen as the optimal variational approximation. The variational distribution $q(\mathbf{W}; \phi)$ is designed as

$$
q(\mathbf{W}; \phi) = \prod_{k=1}^{K} [q(\beta_k; \phi_k^\beta)\ q(\boldsymbol{\theta}_k^*; \phi_k^{\theta^*})] \prod_{i=1}^{N} q_{z_i}(z_i), \tag{11}
$$

where $\{q_{z_i}(z_i)\}_{i=1}^N$ are categorical distributions, and $\phi_k = \{\phi_k^\beta, \phi_k^{\theta^*}\}$ with ϕ_k^β and $\phi_k^{\theta^*}$ denoting the parameters of the distributions $q(\beta_k)$ and $q(\boldsymbol{\theta}_{\boldsymbol{k}}^*)$, respectively.

It is assumed that all the parameters $\phi_k = \{\phi_k^\beta, \phi_k^{\theta^*}\}$ are tied and equivalent to the prior for $k > T^*(T^*$ is a preset parameter and $T^* \ll K)$. Specifically, for all components with $k > T^*$

$$q(\beta_k) = p(\beta_k) = \text{Beta}(1, \alpha_0), \tag{12}$$

$$q(\boldsymbol{\theta}_k^*) = p(\boldsymbol{\theta}_k^*) = G_0(\boldsymbol{\theta}_k^*; \lambda). \tag{13}$$

The prior of β_k is a Beta distribution as in Eq.(12), while $q(\beta_k; \phi_k^\beta)$ (for $k \leq T^*$) is also assumed to be a Beta distribution as

$$q(\beta_k; \phi_k^\beta) = \text{Beta}(\phi_{k,1}^\beta, \phi_{k,2}^\beta), \tag{14}$$

where $\phi_k^\beta = \{\phi_{k,1}^\beta, \phi_{k,2}^\beta\}$ are the variational parameters of $q(\beta_k; \phi_k^\beta)$. The prior of $\boldsymbol{\theta}^*$ is set to a normal inverse Wishart distribution, which is a conjugate prior of the likelihood function in (6), i.e.

$$p(\boldsymbol{\theta}_k^* | \lambda) = \mathcal{N}\left(\boldsymbol{\mu}_k; \boldsymbol{\mu}_0, \frac{1}{\kappa_0}\boldsymbol{\Sigma}_k\right) \cdot \mathcal{W}^{-1}(\boldsymbol{\Sigma}_k; \boldsymbol{\Sigma}_0, \nu_0), \tag{15}$$

where $\lambda = \{\boldsymbol{\mu}_0, \kappa_0, \boldsymbol{\Sigma}_0, \nu_0\}$ and $\mathcal{W}^{-1}(\cdot)$ denotes the inverse Wishart distribution. Similarly, $q(\boldsymbol{\theta}_k^*)$ is assumed to be distributed according to a normal inverse Wishart distribution as

$$q(\boldsymbol{\theta}_k^*; \phi_k^{\theta^*}) = \mathcal{N}\left(\boldsymbol{\mu}_k; \boldsymbol{\mu}_{k,0}, \frac{1}{\kappa_k}\boldsymbol{\Sigma}_k\right) \cdot \mathcal{W}^{-1}(\boldsymbol{\Sigma}_k; \boldsymbol{\Sigma}_{k,0}, \nu_k), \tag{16}$$

where $\phi_k^{\theta^*} = \{\boldsymbol{\mu}_{k,0}, \kappa_k, \boldsymbol{\Sigma}_{k,0}, \nu_k\}$ are the parameters of $q(\boldsymbol{\theta}_k^*)$.

The probability $q_{z_i}(z_i = k)$ can be calculated by

$$q_{z_i}(z_i = k) = \frac{\exp(E_{i,k})}{\sum_{j=1}^{\infty} \exp(E_{i,j})}, \tag{17}$$

where $E_{i,k}$ is defined by

$$E_{i,k} = \mathbb{E}_{q_{\boldsymbol{\beta}}}[\log p(z_i = k | \boldsymbol{\beta})] + \mathbb{E}_{q_{\theta_k^*}}[\log p(\mathbf{x}_i | \boldsymbol{\theta}_k^*)]. \tag{18}$$

Other variational parameters are updated as follows

$$\phi^{\beta}_{k,1} = 1 + \sum_{i=1}^{N} q_{z_i}(z_i = k), \tag{19}$$

$$\phi^{\beta}_{k,2} = \alpha_0 + \sum_{i=1}^{N} \sum_{j=k+1}^{\infty} q_{z_i}(z_i = j), \tag{20}$$

$$\kappa_k = \kappa_0 + \sum_{i=1}^{N} q_{z_i}(z_i = k), \tag{21}$$

$$\nu_k = \nu_0 + \sum_{i=1}^{N} q_{z_i}(z_i = k), \tag{22}$$

$$\boldsymbol{\mu}_{k,0} = \frac{\kappa_0 \boldsymbol{\mu}_0 + \sum_{i=1}^{N} q_{z_i}(z_i = k)\mathbf{x}_i}{\kappa_0 + \sum_{i=1}^{N} q_{z_i}(z_i = k)}, \tag{23}$$

$$\boldsymbol{\Sigma}_{k,0} = \boldsymbol{\Sigma}_0 + \kappa_0 \boldsymbol{\mu}_0 \boldsymbol{\mu}_0^{\mathsf{T}} - \kappa_k \boldsymbol{\mu}_{k,0} \boldsymbol{\mu}_{k,0}^{\mathsf{T}} + \sum_{i=1}^{N} q_{z_i}(z_i = k)\mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}. \tag{24}$$

The variational parameters $\{\phi^{\beta}_{k}, \phi^{\theta^*}_{k}\}_{k=1}^{T^*}$ and distributions $\{q_{z_i}(z_i)\}_{i=1}^{N}$ are updated iteratively by evaluating Eqs. (19) to (24) and Eq. (17) until the free energy $\mathcal{F}$ is minimized [37].

With the trained colour model, each pixel in testing videos is assigned a probability that describes how likely it is part of flames based on its colour. Flame-coloured pixels will obtain high probabilities, while other regions are likely to have low probabilities. Given an appropriately chosen threshold, multiple candidate pixels can be obtained for further processing.

2.2. Framework of Flame R-CNN

The diagram of the framework is illustrated in Figure 1. It can be seen that the framework of flame R-CNN takes each frame of videos as input, and outputs the regions containing flames. A frame-wise decision can be made according to those detected regions of flames. The input frame is processed by several convolutional layers (as well as rectified linear unit (ReLU) layers and max pooling layers) to produce a convolutional feature map. Simultaneously, flame proposals are generated by the proposal generation approach based on the colour and dynamic properties of flames. The generated proposals are subsequently

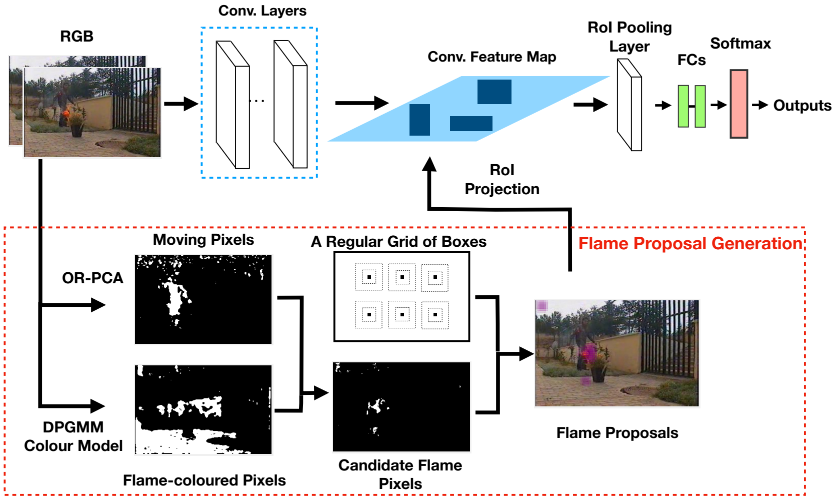


Figure 1: Diagram of the framework of flame R-CNN.

projected to the corresponding locations on the convolutional feature map. Using an RoI pooling layer, the features inside each generated flame proposal over the feature map are transferred into a small map of a fixed size, which can be further processed by fully connected layers and a softmax classification layer. The framework outputs the proposals that are classified as flames.

2.2.1. Flame Proposal Generation

The proposed flame proposal generation approach produces proposals using the dynamic and colour properties of flames, which are effective in selecting candidate flame regions and distinguishing flames from the background and distracting objects. On one hand, the motion of probable flame regions can be detected by the online R-PCA algorithm introduced in Appendix A. On the other hand, flame-coloured pixels are selected by the DPGMM based colour model proposed in Section 2.1. The pixels selected by both the colour model and R-PCA are candidate pixels of flames which will contribute to the generation of proposals. As some weak flames are nearly transparent and the background behind them is visible, the intensities of these flame pixels do not change as significantly as most flames do. In such situations, the dynamic flame regions are difficult to be detected because their intensity values are influenced by the static background behind flames. To solve this problem, the proposed framework performs the online R-PCA algorithm on the R channels of frames instead of the grey-scale images, since the red colour is dominant in flame regions and obvious changes can be observed in the intensities of R channels even if flames are semi-transparent.

Flame proposals are generated using grid boxes, as shown in Figure 1. The boxes, in which the ratios of candidate flame pixels are higher than a threshold τ_f, are treated as flame proposals. It is noteworthy that the boxes can be multi-scaled, but have to be of a fixed aspect ratio to ensure that the geometric layouts of the features within different generated proposals will be changed in the same manner by the RoI layer. Specifically, each flame proposal is projected onto the convolutional feature map to produce a small feature map, which will be reshaped by the RoI layer. Consequently, the geometrical layouts of the features will be changed in significantly different ways if proposals have diverse aspect ratios. This will influence the similarities between reshaped feature maps and thus degrade the performance of the framework. It also explains another reason why the flame proposals are not generated by the selective search approach [10] or region proposal network [11]. The proposals generated in these ways are diverse in aspect ratios due to the various appearances of flames. Additionally, the grid boxes can be set at a fine level and generate multi-scale proposals to improve the detection of flames occupying small regions in the scene. Furthermore, the proposed approach can select candidate flame pixels only using the colour model, which enables it to work with both videos and images.

2.2.2. Loss Function of the Framework of Flame R-CNN

As mentioned above, the flame proposals are generated based on grid boxes. They are not designed to bound each flame region with a bounding box, which is a common setting in rigid object detection. Instead, the proposed framework aims to cover as many flame regions as possible. Therefore, it is unnecessary to perform the regression of boxes and accordingly only the classification loss needs to be considered. It is different from the existing methods for object detection, such as fast R-CNN [38] and YOLO [31]. For each proposal of flames, the loss function of the proposed framework is given by

$$\mathcal{L}(p_c, c^*) = -\log p_c(c^*), \tag{25}$$

where $c \in \{0, 1\}$ indicates the classes of proposals, p_c is the probability distribution output by the softmax layer, and c^* is the ground truth class label of the proposal. Specifically, $c = 0$ and $c = 1$ correspond to the background and flames, respectively.

2.2.3. Training of Flame R-CNN

The ground truth RoIs of flames are given in the format of grid boxes instead of the tightest bounding boxes. The boxes are superimposed on the training images. In the diagram, the boxes are non-overlapping for visualisation purpose, but they overlap in actuality. For each box, it is labelled as 'flame' if the ratio of ground truth flame pixels is higher than 30%. The positive training proposals (containing flames) are those that have intersection over union (IoU) overlap with any ground truth box of flames of at least 50%, while the ones of which the maximum IoU overlap with ground truth boxes is within the range $[0.05, 0.2)$ are treated as negative training proposals (non-flame).

3. Experimental Results and Discussion

3.1. Benchmarking Database and Performance Evaluation Methods

The proposed framework of flame R-CNN is trained on 729 images from the datasets [17, 21], and tested on 16 videos of 3968 frames from [15, 26]. A brief description of testing videos is presented in Table 1. The proposed framework is trained on images instead of videos because the frames from the same video are similar and may induce the problem of over-fitting. Training the network with images which are different from the testing videos can avoid this problem, and

Table 1: Information of the testing videos for experiments

Video	Burning Objects	Distractors	Positive Frames	Negative Frames	Lighting Condition	Smoke Condition	Location
VC1	Hay	A walking person	26	0	Bright	Thick	Outdoor
VC2	Hay	A working man	93	0	Bright	Thick	Outdoor
VC3	Unknown	Moving people	48	0	Bright	Thick	Outdoor
VC4	Hay	Moving people	41	0	Bright	Thick	Outdoor
VC5	Trees	None	214	0	Bright	Thin	Outdoor
VC6	Trees	None	176	0	Dark	Thin	Outdoor
VC7 [a]	Branches	A walking man	687	5	Bright	Medium	Outdoor
VC8 [b]	Assembly line	Moving workers	572	69	Bright	Thin	Indoor
VC9	Grass	None	386	0	Bright	Medium	Outdoor
VC10	Papers	A moving light	395	0	Bright	Thin	Indoor
VC11	Trees	None	186	0	Bright	Thick	Outdoor
VC12	None	Flashing carlights	0	139	Dark	None	Outdoor
VC13	None	Flashing carlights and walking people	0	144	Dark	None	Outdoor
VC14	None	A walking person in red clothes	0	155	Bright	None	Indoor
VC15	None	Crashing cars	0	378	Bright	None	Indoor
VC16	None	Walking people	0	254	Bright	None	Indoor

[a] Frame 531, 532, 533, 658, 660 are negative, other frames contain flames in them

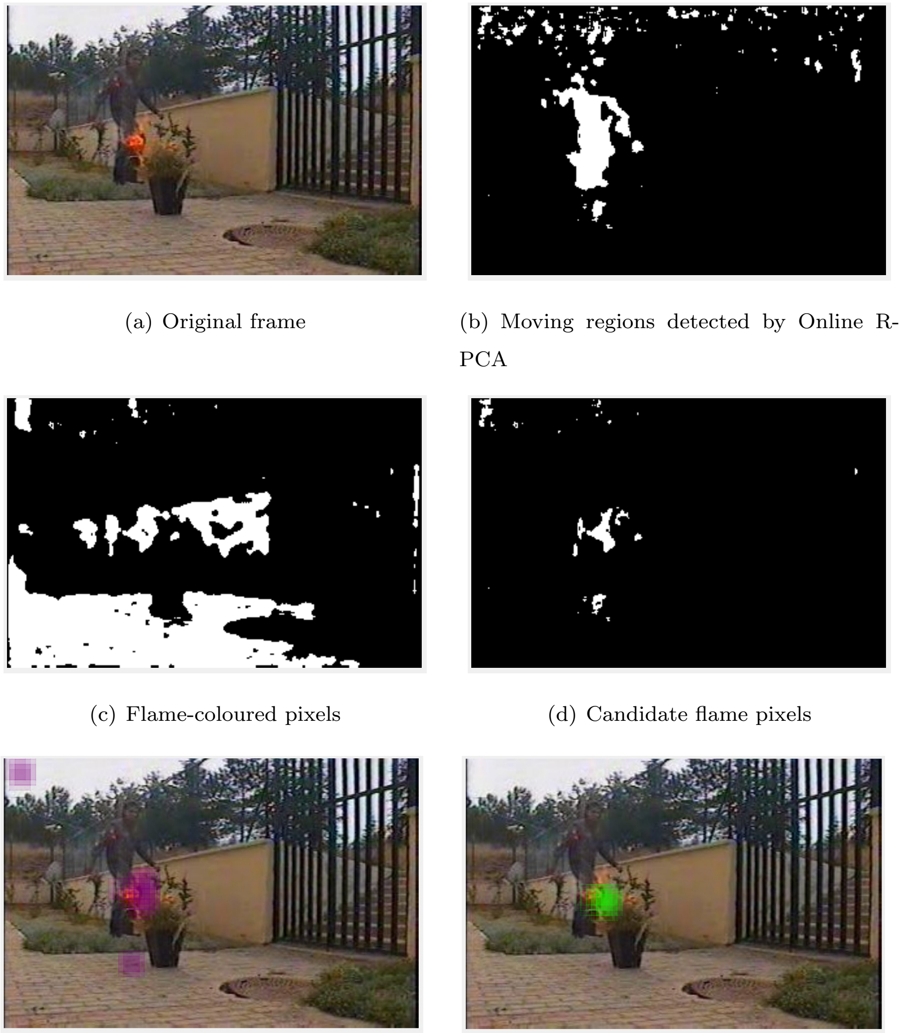
[b] The first 69 frames are negative and others are positive.

prove the robustness of the proposed framework. The framework of flame R-CNN is fine-tuned with ground truth flame proposals using Resnet50 [39] as the backbone, which is pre-trained on the database for the ImageNet Large-Scale Visual Recognition Challenge [40].

The DPGMM based colour model in the framework of flame R-CNN is trained on 100 images randomly selected from [21]. The proposed framework is evaluated by the frame-wise true positive rate (TPR) and true negative rate (TNR), and is compared with two state-of-the-art approaches using the SqueezeNet [29] and faster R-CNN [11]. The size of boxes is set to 16×16 with a stride of 4. The threshold τ_f for the ratio of candidate flame pixels is set to 0.3 in the flame proposal generation approach. The threshold for the logarithmic probability of flame colours is set to -1.2. The threshold is set to a relatively low value to enhance the performance of proposal generation, which will influence the final detection results. In the flame proposal generation approach, a low threshold of flame colour probabilities usually leads to more proposals than a high threshold. The situation that proposals are not generated in video frames containing flames will lead to missed detection. In contrast, a proposal with no flame in it will be further refined by additional layers within the framework of flame R-CNN and can still produce a reliable result. Additionally, the flame R-CNN is trained by the adaptive moment estimation (Adam) algorithm [41], of which the initial learning rate is set to 0.0001.

3.2. Detection Performance Evaluation and Discussion

The intermediate and final detection results of some testing videos are illustrated in this section. From them, it can be seen that the framework of flame R-CNN achieves accurate detection of flame regions. The effectiveness of the proposed flame proposal generation approach can be proven by the results in Figure 2 and Figure 3. The online R-PCA algorithm works effectively in detecting moving foreground objects while ignores the noise in the background. Simultaneously, the flame colour model successfully detects the pixels of flame colours. Thanks to the combination of these two methods, the framework can detect probable



(a) Original frame

(b) Moving regions detected by Online R-PCA

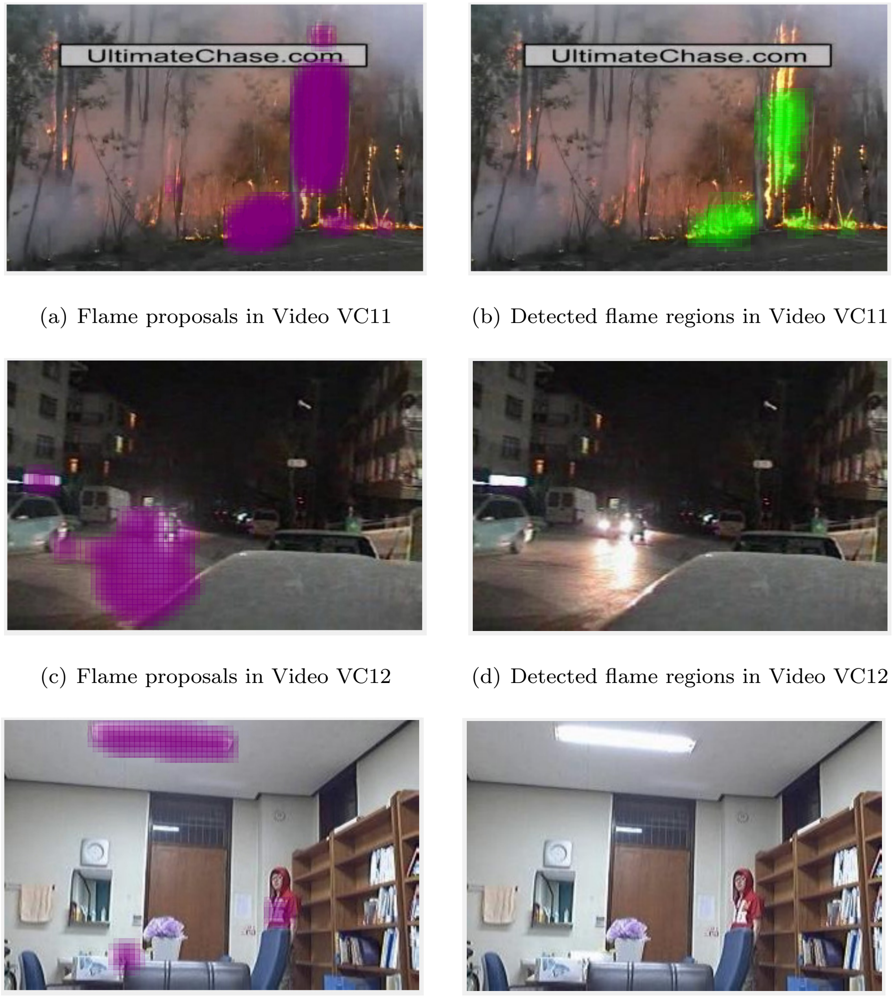
(c) Flame-coloured pixels

(d) Candidate flame pixels

(e) Generated flame proposals

(f) Detected flame regions

Figure 2: Detection results of the framework of flame R-CNN tested on Video VC7, in which a man walking around burning branches.



(a) Flame proposals in Video VC11

(b) Detected flame regions in Video VC11

(c) Flame proposals in Video VC12

(d) Detected flame regions in Video VC12

(e) Flame proposals in Video VC14

(f) Detected flame regions in Video VC14

Figure 3: Generated flame proposals and detected flame regions by the framework of flame R-CNN tested on Video VC11, VC12, and VC14.

regions of flames based on the dynamic and colour properties. The detected candidate flame pixels include the majority (if not all) of flame pixels and a small number of non-flame ones, and thus help to generate proposals which are likely to contain flames in them. For example, the online R-PCA algorithm detects motion caused not only by flames but also by heated airflow and a walking man in Figure 2, but most non-flame regions are discarded by the colour model, which helps to reduce the number of proposals without flames, as shown in Figure 2(e).

The proposals of flames will be further verified by additional layers of the framework for reliable detection. From the results shown in Figures 3(c), 3(d) and 3(e), 3(f), it can be seen that the regions of flashing car lights and the walking person in red clothes are successfully classified as negative by the framework, although they are similar to flames in appearance. Additionally, the proposed framework also works well in detecting flame regions, which can be observed from the results shown in Figure 2 and Figure 3(b). The video in Figure 2 is among the most challenging videos for flame detection, since the semi-transparent colours of the flame regions make the background behind flames visible, resulting in difficulties in motion detection. Besides, the texture of the bushes behind flames plays a dominant role when the flames are weak, which mixes the features of the bushes and flames and confuses the trained network. Despite these difficulties, the proposed framework achieves accurate detection of flames in most frames of this video. The non-flame proposals at the left top corner are discarded while the flame regions in the centre are detected successfully.

The average frame-wise TPR and TNR of the proposed framework tested on the videos in Table 1 are shown in Table 2 together with the performance of the SqueezeNet and faster R-CNN. In the flame proposal generation approach, proposals can also be generated only based on the estimated probabilities of flame colours. Specifically, candidate pixels of flames are detected only by the colour model, based on which flame proposals are generated (a simplified version of flame R-CNN). The results of the proposed framework (both full and

Table 2: Average flame detection performance of the SqueezeNet, faster R-CNN and Flame R-CNN

Method	TPR	TNR	Accuracy	F-score
SqueezeNet	0.3913	0.9913	0.5643	0.5611
faster R-CNN, 3 anchors	0.7040	0.3007	0.5877	0.7085
faster R-CNN, 4 anchors	0.8499	0.7299	0.8153	0.8675
faster R-CNN, 5 anchors	0.4851	0.7823	0.5708	0.6167
Simplified flame R-CNN, th = 0.5	0.9217	0.7719	0.8785	0.9153
Simplified flame R-CNN, th = 2.5	0.9929	0.6154	**0.8841**	**0.9242**
Simplified flame R-CNN, th = 3.5	0.9005	0.8024	0.8722	0.9094
Flame R-CNN, th = -1.5	0.8523	0.8392	0.8485	0.8890
Flame R-CNN, th = -1.2	0.9093	0.8103	0.8808	0.9157
Flame R-CNN, th = 0.5	0.8796	0.8400	0.8682	0.9048

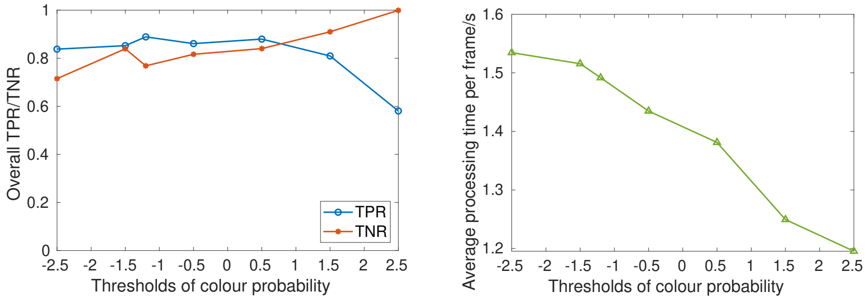
simplified) using different thresholds for flame colour probabilities are provided in Table 2. The faster R-CNN is trained and tested with different number of anchors, i.e. 3, 4 and 5 anchors. A k-means clustering algorithm is conducted with an IoU based distance metric to choose sizes of the anchor boxes from the training images.

The results illustrate that the framework of flame R-CNN achieves higher TPRs than the comparing frameworks. It is because the SqueezeNet based method conducts classification using all the features of input frames, resulting in the situation that the features of flames occupying small regions in the scene are overwhelmed by the features of the background or other salient objects. In contrast, the frameworks of faster R-CNN and flame R-CNN both detect probable regions of flames and provide results only based on the features within those regions, which enhances the accuracy of detection. However, the sizes of anchors in the framework of faster R-CNN have a large influence on the performance of flame detection. Specifically, the anchor boxes that match the shapes and sizes of flames will lead to good performance, while the flames whose shapes are very different from the anchors can hardly be detected. Since the flames are non-rigid and diverse in shapes, it is difficult to set appropriate sizes for anchors. The framework based on flame R-CNN does not have this problem since it generates the proposals based on grid boxes, and the colour and dynamic properties of flames.

On the contrary, the method using SqueezeNet achieves better TNRs than the proposed framework. As the TPR and TNR are usually competing, it is not surprising the SqueezeNet has fewer false positive errors than the frameworks of faster R-CNN and flame R-CNN. However, higher TNRs are achieved by the proposed framework than faster R-CNN on most negative testing videos, showing the effectiveness of the flame proposal generation approach. Considering the great losses due to fires every year, the false negative errors of flame detection cause larger damage than the false positive ones and thus should be avoided at all expense. In a nutshell, the flame R-CNN achieves better performance than the methods based on SqueezeNet and faster R-CNN in reducing the losses caused by fires.

3.3. Threshold of the Flame Colour Probability of Flame R-CNN

In the flame proposal generation approach, the threshold for the flame colour probability will influence the performance of detection. In this section, the proposed framework is trained and tested using different thresholds to explore their influence on the accuracy and processing time of detection. From 4(a), it can be seen that the TPR of the framework fluctuates between 0.8 and 0.95 when the threshold increases from -2.5 to 0.5, and has a significant drop with a threshold of 2.5. In contrast, the TNR rises when the threshold increases from -2.5 to -1.5, and oscillates over the threshold range of $[-1.5, 0.5]$. An upward trend in the TNR can be seen with a threshold larger than 1.5. According to the flame proposal generation approach, a large threshold of colour probability will lead to fewer candidate flame pixels, and thus results in a smaller number of flame proposals compared with a small threshold. However, the relationship between the threshold and TPR/ TNR is not monotonic. When



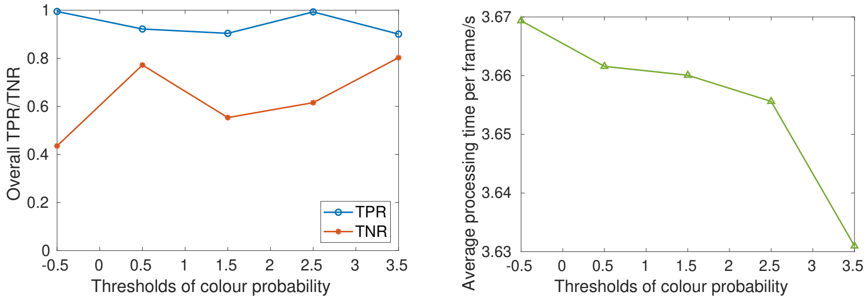
(a) Overall TPRs and TNRs of the flame R-CNN with different thresholds for the estimated colour probability.

(b) Average processing time of one frame of the flame R-CNN with different thresholds for the estimated colour probability.

Figure 4: Detection performance and computational complexity of the framework of flame R-CNN with different thresholds for the colour probability.

the threshold is too large and discards many flame pixels, it will result in a small number of proposals of flames and lead to false negative errors, which explains the sharp decrease in the TPR when the threshold increases to 2.5. In contrast, a small threshold does not decrease the TNR significantly, since the convolutional features within flame proposals are processed and classified accurately by additional layers. Different from the performance, the computational cost of the framework decreases monotonically with the threshold for flame colour probabilities, which can be seen from 4(b). It is because a large threshold leads to a small number of proposals that will decrease the computational cost of the framework.

The experiments of the simplified flame R-CNN are also carried out with different thresholds, of which the TPRs/TNRs and processing time are shown in Figure 5. The TPR and TNR of the simplified flame R-CNN vary with the threshold in a similar way to the full flame R-CNN. Additionally, the average processing time of one frame declines with the threshold as well, similar to the framework of full flame R-CNN. However, its average processing time is much longer than that of the full flame R-CNN. Although the online R-PCA algorithm for motion detection increases the computational cost, the simplified



(a) Overall TPRs and TNRs of the simplified flame R-CNN with different thresholds for the estimated colour probability.

(b) Average processing time of one frame of the simplified flame R-CNN with different thresholds for the estimated colour probability.

Figure 5: Detection performance and computational complexity of the framework of simplified flame R-CNN with different thresholds for the colour probability.

flame R-CNN has a larger number of proposals to process, and thus needs longer time for each frame compared with the framework of full flame R-CNN given the same threshold. Furthermore, the changes of the threshold also cause larger fluctuations in the performance of the simplified flame R-CNN than the full one, which can be explained by the way of generating flame proposals. In the simplified framework, the threshold for colour probability influences the number of candidate flame pixels as well as proposals directly, while the impact is relieved by the online R-PCA algorithm in the full flame R-CNN.

The computational complexity of the detection process is investigated by analysing the complexity of each module within the proposed framework as follows. In the online R-PCA, frames in a video are reshaped into column vectors which later form a matrix to be processed. The dimension of each column (equalling to the number of pixels in one frame) is denoted by d, while r is the intrinsic dimensionality of the subspace underlying the formed matrix. The computational complexity of the online R-PCA algorithm is $\mathcal{O}(d \cdot r^2)$ where usually $d >> r$. For the DPGMM based flame colour model, the inversion and determinant of the covariance matrices of the trained model can be calculated in advance to reduce the computational burden. Therefore, the time complexity of calculating the probabilities of pixels being flames is $\mathcal{O}(d \cdot K)$, where K is the estimated number of clusters. The complexity of convolutional layers is $\mathcal{O}(\sum_l a_{l-1} \cdot a_l \cdot f_l^2 \cdot b_l^2)$ with l being the index of convolutional layers and a_{l-1}, a_l, f_l, b_l denoting the number of input channels, number of filters, spatial size of the filters, and spatial size of the output feature map of the l-th layer, respectively. The time complexity of fully connected layers is $\mathcal{O}(n_p \cdot h_W \cdot h_H \cdot \sum_t g_t)$ where n_p, t, h_W, h_H and g_t represent the number of flame proposals, index of fully connected layers, width and height of the output of RoI pooling layers and the output dimension of the t-th fully connected layer.

3.4. Ablation Study

To validate the effectiveness of each module of the proposed framework, ablation experiments are conducted in this subsection as follows. The results are summarized in Table 3.

1. The proposed flame proposal generation approach is replaced by the selective search method [38]. As such, the framework reduces to the widely used fast R-CNN method [38].
2. Instead of utilising the colour model and R-PCA, flame proposals are generated only using grid boxes. Specifically, all grid boxes are projected to the convolutional feature map and processed by the following layers. The resulting framework is denoted as 'grid R-CNN' in the table.
3. Detection is conducted without deep neural networks, but only based on selected candidate flame pixels by the colour model and R-PCA algorithm. Grid boxes of the same size and steps as the full flame R-CNN are used to generate flame proposals, but the threshold of candidate flame pixel ratio is set to 0.7 instead of 0.3 for better performance. A frame is classified as positive once a candidate flame proposal is detected. It is denoted as 'Colour+Dynamics' in Table 3.
4. We also evaluate the full framework proposed in this paper with all the modules described in Section 2.

Table 3: Average flame detection performance of ablation experiments

	Fast R-CNN	Grid R-CNN	Colour+Dynamics	Full flame R-CNN
TPR	0.3863	0.9968	0.9359	0.9093
TNR	0.5035	0.1128	0.3601	0.8103
Accuracy	0.4201	0.7419	0.7699	0.8808
F-score	0.4867	0.8461	0.8527	0.9157

The results in Table 3 illustrate that the proposed framework of flame R-CNN captures the crucial features of flames and thus achieves the best detection performance. The introduced flame proposal approach significantly enhances the accuracy while achieving better balance between TPR and TNR, which can be seen from the results compared with the fast R-CNN. The colour and dynamic properties of flames together with the convolutional features work effectively in both detecting flames and discarding non-flame objects, contributing to the enhanced overall accuracy.

4. Conclusion

In this paper, a framework of flame R-CNN was proposed for autonomous flame detection using video sequences. A flame proposal generation approach was developed to generate flame proposals based on the colour and dynamic properties. The proposals and a feature map produced by several convolutional layers are subsequently processed by additional layers to produce flame regions and frame-wise results can be obtained accordingly. Since flames have unclear edges and a rich diversity in the appearance, the flame proposal generation approach works effectively in selecting probable regions of flames by utilizing the crucial properties, which contributes to high TPRs of the developed framework. It has achieved frame-wise accuracy of 88.41% and F-score of 92.42%.

Appendix A. Online R-PCA for Motion Detection in Videos

The R-PCA algorithms decompose a data matrix into two parts, a low-rank matrix and a sparse matrix. Then frames in a video can be reshaped into columns and combined to form a matrix which is processed by the R-PCA algorithms. The low-rank component naturally corresponds to the stationary background in a video, while the sparse matrix contains the information of moving objects. Therefore, an algorithm of online R-PCA is embedded into the proposed framework of flame R-CNN to detect moving regions that are likely to be flames.

Denote the matrix of reshaped frames as $\mathbf{M} = \{\mathbf{m}_i\}_{i=1}^{n}$ with $\mathbf{m}_i \in \mathbb{R}^d$ denoting the i-th vectorized frame in a video, where d is the dimension of each column and n is the number of frames. The matrix $\mathbf{M}$ can be decomposed as

$$\mathbf{M} = \mathbf{Y} + \mathbf{S}, \tag{A.1}$$

where $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{n}$, $\mathbf{y}_i \in \mathbb{R}^d$ and $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{n}$, $\mathbf{s}_i \in \mathbb{R}^d$ represent the low-rank and sparse matrices, respectively. Principal component pursuit (PCP) [42], as one of the most widely used R-PCA algorithms, recovers the low-rank matrix $\mathbf{Y}$ from $\mathbf{M}$ by solving a minimization problem given by

$$\min_{\mathbf{Y},\mathbf{S}} \frac{1}{2}\|\mathbf{M} - \mathbf{Y} - \mathbf{S}\|_F^2 + \widetilde{\lambda}_1\|\mathbf{Y}\|_* + \widetilde{\lambda}_2\|\mathbf{S}\|_1, \tag{A.2}$$

where $\widetilde{\lambda}_1, \widetilde{\lambda}_2$ are balancing parameters, and $\|\cdot\|_F^2$, $\|\cdot\|_*$, and $\|\cdot\|_1$ denote the Frobenius norm, nuclear norm and ℓ_1-norm of a matrix, respectively. However, the PCP method works in a batch manner and needs to access all samples in each iteration, which requires large storage for data and results in delay in processing.

To address the aforementioned drawback, Feng and Xu proposed an alternative method by employing an equivalent form of the nuclear norm [42], and rewrote the problem in (A.2) as

$$\min_{\mathbf{L},\mathbf{R},\mathbf{S}} \frac{1}{2}\|\mathbf{M} - \mathbf{L}\mathbf{R}^{\mathsf{T}} - \mathbf{S}\|_F^2 + \frac{\widetilde{\lambda}_1}{2}(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2) + \widetilde{\lambda}_2\|\mathbf{S}\|_1, \tag{A.3}$$

where the low rank matrix is factorized into two parts $\mathbf{L} \in \mathbb{R}^{d\times r}$ and $\mathbf{R} \in \mathbb{R}^{n\times r}$ with r denoting the upper bound of the rank of $\mathbf{Y}$. The matrix $\mathbf{L}$ can be treated as bases of the low dimensional subspace and $\mathbf{R}$ are the coefficients of samples projected to the bases. It can be proven that the local minima of (A.3) are the global minima of (A.2) [42]. Therefore, solving the problem in (A.3) will provide estimations of the low dimensional subspace and sparse component equivalent to those obtained by solving (A.2). The algorithm of online R-PCA processes each new sample once it is obtained without accessing the observations after it. It finds the coefficients $\mathbf{r} \in \mathbb{R}^{r}$ as well as the sparse component $\mathbf{s} \in \mathbb{R}^{d}$ of the new sample, and updates the subspace bases $\mathbf{L}$ alternatively using a stochastic optimization algorithm.

Appendix B. Definitions of Dirichlet Process and Details of Accelerated Variational Inference

Appendix B.1. Dirichlet Process and Stick-breaking Process

Given a measurable space and a probability measure G_0 on the space [43], a DP is defined as a distribution of a probability measure G over the space, satisfying the condition that for any finite measurable partition $(A_1, ..., A_K)$ of the space, $(G(A_1), ..., G(A_K))$ is distributed according to a Dirichlet distribution with parameters of $(\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_K))$, i.e.

$$(G(A_1), ..., G(A_K)) \sim \mathrm{Dir}(\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_K)), \tag{B.1}$$

where α_0 is a positive real number and $\mathrm{Dir}(\cdot)$ denotes the Dirichlet distribution. The model is denoted as $G \sim DP(\alpha_0, G_0)$ with a concentration parameter α_0 and a base distribution $G_0(\cdot\,;\lambda)$, where λ is a hyperparameter of G_0.

The stick-breaking process provides a way to construct a DP [44], defined by

$$\beta_k \sim \mathrm{Beta}(1, \alpha_0), \tag{B.2}$$

$$\boldsymbol{\theta}_k^* \sim G_0, \tag{B.3}$$

$$\pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) = \beta_k \left(1 - \sum_{l=1}^{k-1} \pi_l\right), \tag{B.4}$$

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta}_k^*}, \tag{B.5}$$

where both $\{\beta_k\}_{k=1}^{\infty}$ and $\{\boldsymbol{\theta}_k^*\}_{k=1}^{\infty}$ are independent and identically distributed (i.i.d.) random variables, $\delta_{\boldsymbol{\theta}_{\boldsymbol{k}}^*}$ represents the Dirac measure centred at $\boldsymbol{\theta}_k^*$, and Beta$(\cdot)$ denotes the Beta distribution. The distribution over $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{\infty}$ can also be expressed as $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha_0)$, which comes from the initials of Griffiths, Engen and McCloskey [34].

Appendix B.2. Accelerated Variational Inference

Minimising the KLD is equivalent to minimising the free energy defined by $\mathcal{F} = \mathbb{E}_q[\log q(\mathbf{W}; \phi)] - \mathbb{E}_q[\log p(\mathbf{W}, \mathbf{X})]$ with respect to (w.r.t.) $\mathbf{W}$ [37]. Taking the variational distribution in Eq. (11) into the definition of free energy, we have

$$\mathcal{F} = \sum_{k=1}^{T^*} \left\{ \mathbb{E}_{q_{\beta_k}}\left[\log \frac{q(\beta_k; \phi_k^{\beta})}{p(\beta_k | \alpha_0)}\right] + \mathbb{E}_{q_{\theta_k^*}}\left[\log \frac{q(\boldsymbol{\theta}_k^*; \phi_k^{\theta^*})}{p(\boldsymbol{\theta}_k^* | \lambda)}\right] \right\} + \sum_{i=1}^{N} \mathbb{E}_{q_{z_i}}\left[\log \frac{q_{z_i}(z_i)}{p(z_i | \boldsymbol{\beta}) p(\mathbf{x}_i | \boldsymbol{\theta}_{z_i}^*)}\right].$$

The free energy $\mathcal{F}$ is a function of T^* sets of parameters $\{\phi_k^{\beta}, \phi_k^{\theta^*}\}_{k=1}^{T^*}$ and N distributions $\{q_{z_i}(z_i)\}_{i=1}^{N}$. The first two terms are truncated at level T^* because no parameters need to be optimized beyond T^*. However, the variational distribution still provides $q_{z_i}(z_i)$ with infinite support as it allows components beyond T^* to have non-zero probabilities. Based on the settings in Eqs. (12) and (13), the free energy is nested w.r.t. T^*, which guarantees the existence of optimal parameters when changing T^*. Therefore, the value of T^* is adaptive during optimization and can be initialised to one.

The variational inference can be accelerated using a KD-tree [45]. A KD-tree is a binary tree, where the data stored in each child node are a subset of its parent node. The accelerated variational Dirichlet process constrains that all the data in a node share the same assignment of clusters. The variational parameters are updated in a similar way as in Eqs. (19)-(24), with the value of each data point changed to the average of all the data of a node.

CRediT author statement

Zhenglin Li: Conceptualization, Methodology, Software, validation, Writing - Original Draft, Writing - Review & Editing, Visualization. Lyudmila Mihaylova: Conceptualization, methodology, validation, formal analysis, Writing - Review & Editing. Le Yang: Methodology, Writing - Review & Editing.

Acknowledgment

This work was supported by the UK EPSRC [grant number EP/T013265/1], USA National Science Foundation [grant number NSF ECCS 1903466], and China Scholarship Council [grant number 201506060163].

References

[1] F. Bu, M. S. Gharajeh, Intelligent and vision-based fire detection systems: A survey, Image and Vision Computing 91 (2019) 103803.

[2] M. J. Sousa, A. Moutinho, M. Almeida, Wildfire detection using transfer learning on augmented datasets, Expert Systems with Applications 142 (2020) 112975.

[3] P. Borges, E. Izquierdo, A probabilistic approach for vision-based fire detection in videos, IEEE Transactions on Circuits and Systems for Video Technology 20 (5) (2010) 721–731.

[4] Z. Li, O. Isupova, L. S. Mihaylova, L. Rossi, Autonomous flame detection in video based on saliency analysis and optical flow, in: Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Baden-Baden, Germany, 2016, pp. 218–223.

[5] M. Mueller, P. Karasev, I. Kolesov, A. Tannenbaum, Optical flow estimation for flame detection in videos, IEEE Transactions on Image Processing 22 (7) (2013) 2786–2797.

[6] Z. Li, L. S. Mihaylova, O. Isupova, L. Rossi, Autonomous flame detection in videos with a Dirichlet process Gaussian mixture color model, IEEE Transactions on Industrial Informatics 14 (3) (2017) 1146–1154.

[7] S. Zhang, F. He, DRCDN: learning deep residual convolutional dehazing networks, The Visual Computer 36 (9) (2020) 1797–1808.

[8] S. Zhang, F. He, W. Ren, NLDN: Non-local dehazing network for dense haze removal, Neurocomputing 410 (2020) 363–373.

[9] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.

[10] J. R. Uijlings, K. E. Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, International Journal of Computer Vision 104 (2) (2013) 154–171.

[11] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proceedings of Conference on Advances in Neural Information Processing Systems, Montreal, Canada, 2015, pp. 91–99.

[12] H. Huang, H. Zhou, X. Yang, L. Zhang, L. Qi, A. Zang, Faster R-CNN for marine organisms detection and recognition using data augmentation, Neurocomputing 337 (2019) 372–384.

[13] H. Li, F. He, Y. Liang, Q. Quan, A dividing-based many-objective evolutionary algorithm for large-scale feature selection, Soft Computing 24 (2020) 6851–6870.

[14] J. Chen, Y. He, J. Wang, Multi-feature fusion based fast video flame detection, Building and Environment 45 (5) (2010) 1113–1122.

[15] Y. H. Habiboğlu, O. Günay, A. E. Çetin, Covariance matrix-based fire and flame detection method in video, Machine Vision and Applications 23 (6) (2012) 1103–1113.

[16] T. Celik, H. Demirel, Fire detection in video sequences using a generic color model, Fire Safety Journal 44 (2) (2009) 147–158.

[17] D. Y. Chino, L. P. Avalhais, J. F. Rodrigues, A. J. Traina, Bowfire: detection of fire in still images by integrating pixel color and texture analysis, in: Proceedings of the 28th Conference on Graphics, Patterns and Images, Salvador, Brazil, 2015, pp. 95–102.

[18] 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 Transactions on Circuits and Systems for Video Technology 25 (9) (2015) 1545–1556.

[19] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, A. E. Cetin, Computer vision based method for real-time fire and flame detection, Pattern Recognition Letters 27 (1) (2006) 49–58.

[20] D. Han, B. Lee, Flame and smoke detection method for early real-time detection of a tunnel fire, Fire Safety Journal 44 (7) (2009) 951–961.

[21] T. Toulouse, L. Rossi, M. Akhloufi, T. Celik, X. Maldague, Benchmarking of wildland fire colour segmentation algorithms, Image Processing 9 (12) (2015) 1064–1072.

[22] D. Wang, X. Cui, E. Park, C. Jin, H. Kim, Adaptive flame detection using randomness testing and robust features, Fire Safety Journal 55 (2013) 116–125.

[23] H. Li, F. He, Y. Chen, Learning dynamic simultaneous clustering and classification via automatic differential evolution and firework algorithm, Applied Soft Computing 96 (2020) 106593.

[24] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, et al., Video fire detection–review, Digital Signal Processing 23 (6) (2013) 1827–1843.

[25] B. C. Ko, K. Cheong, J. Nam, Fire detection based on vision sensor and support vector machines, Fire Safety Journal 44 (3) (2009) 322–329.

[26] B. C. Ko, S. J. Ham, J. Y. Nam, Modeling and formalization of fuzzy finite automata for detection of irregular fire flames, IEEE Transactions on Circuits and Systems for Video Technology 21 (12) (2011) 1903–1912.

[27] F. Yuan, A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with AdaBoost for video smoke detection, Pattern Recognition 45 (12) (2012) 4326–4336.

[28] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size, arXiv preprint arXiv:1602.07360 (2016).

[29] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, S. W. Baik, Efficient deep CNN-based fire detection and localization in video surveillance applications, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (7) (2018) 1419–1434.

[30] P. Li, W. Zhao, Image fire detection algorithms based on convolutional neural networks, Case Studies in Thermal Engineering (2020) 100625.

[31] H. Chen, Z. He, B. Shi, T. Zhong, Research on recognition method of electrical components based on YOLO V3, IEEE Access 7 (2019) 157818–157829.

[32] Z. Zhong, M. Wang, Y. Shi, W. Gao, A convolutional neural network-based flame detection method in video sequence, Signal, Image and Video Processing 12 (8) (2018) 1619–1627.

[33] N. Yu, Y. Chen, Video flame detection method based on two-stream convolutional neural network, in: Proceedings of the 8th IEEE Joint International Information Technology and Artificial Intelligent Conference, Chongqing, China, 2019, pp. 482–486.

[34] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical Dirichlet processes, Journal American Statistical Association 101 (476) (2006) 1566–1581.

[35] S. J. Gershman, D. M. Blei, A tutorial on Bayesian nonparametric models, Journal Mathematical Psychology 56 (1) (2012) 1–12.

[36] D. M. Blei, M. I. Jordan, Variational inference for Dirichlet process mixtures, Bayesian Analysis 1 (1) (2006) 121–143.

[37] K. Kurihara, M. Welling, N. Vlassis, Accelerated variational Dirichlet process mixtures, in: Proceedings of Conference on Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 2006, pp. 761–768.

[38] R. Girshick, Fast R-CNN, in: Proceedings of IEEE International Conference on Computer Vision, Las Condes, Chile, 2015, pp. 1440–1448.

[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 770–778.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.

[41] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, IEEE Transactions on Image Processing 26 (7) (2017) 3142–3155.

[42] J. Feng, H. Xu, S. Yan, Online robust PCA via stochastic optimization, in: Proceedings of Conference on Advances in Neural Information Processing Systems, Lake Tahoe, CA, USA, 2013, pp. 404–412.

[43] J. Paisley, C. Wang, D. M. Blei, M. I. Jordan, Nested hierarchical Dirichlet processes, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2) (2014) 256–270.

[44] Y. Lai, Y. Ping, K. Xiao, B. Hao, X. Zhang, Variational Bayesian inference for a Dirichlet process mixture of beta distributions and application, Neurocomputing 278 (2018) 23–33.

[45] J. L. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM 18 (9) (1975) 509–517.