Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications

Khan Muhammad, Jamil Ahmad, Student Member, IEEE, Zhihan Lv, Member, IEEE, Paolo Bellavista, Senior Member, IEEE, Po Yang, Member, IEEE , Sung Wook Baik, Member, IEEE

Abstract—Convolutional neural networks (CNN) have yielded state-of-the-art performance in image classification and other computer vision tasks. Their application in fire detection systems will substantially improve detection accuracy, which will eventually minimize fire disasters and reduce the ecological and social ramifications. However, the major concern with CNN-based fire detection systems is their implementation in real-world surveillance networks, due to their high memory and computational requirements for inference. In this work, we propose an energy-friendly and computationally efficient CNN architecture, inspired by the SqueezeNet architecture for fire detection, localization, and semantic understanding of the scene of the fire. It uses smaller convolutional kernels and contains no dense, fully connected layers, which helps keep the computational requirements to a minimum. Despite its low computational needs, the experimental results demonstrate that our proposed solution achieves accuracies that are comparable to other, more complex models, mainly due to its increased depth. Moreover, the paper shows how a trade-off can be reached between fire detection accuracy and efficiency, by considering the specific characteristics of the problem of interest and the variety of fire data.

Index Terms— Convolutional Neural Networks, Deep Learning, Fire Detection, Fire Localization, Fire Disaster, Image Classification, Surveillance Networks

I. Introduction

Recently, a variety of sensors have been introduced for different applications such as setting off a fire alarm [1], vehicle obstacle detection, visualizing the interior of the human body for diagnosis [2-4], animal and ship monitoring, and surveillance [5]. Of these applications, surveillance has primarily attracted the attention of researchers due to the enhanced embedded processing capabilities of cameras. Using smart surveillance systems, various abnormal events such as road accidents, fires, medical emergencies etc. can be detected at early stages, and the appropriate authority can be autonomously informed [6]. A fire is an abnormal event which can cause significant damage to lives and property within a very short time [7]. The main causes of such disasters include human error or a system failure which results in severe loss of human life and other damage [8]. In Europe, fire disasters affect 10,000 km^2 of vegetation zones each year; in North America and Russia, the damage is about 100,000 km^2. In June 2013, fire disasters killed 19 firefighters and ruined 100 houses in Arizona, USA. Similarly, another forest fire in August 2013 in California ruined an area of land the size of 1042 km^2, causing a loss of $127.35 million [9]. According to an annual disaster report [10], fire disasters alone affected 494,000 people and resulted in a loss of $3.1 billion USD in 2015. In order to avoid such disasters, it is important to detect fires at early stages utilizing smart surveillance cameras.

Two broad categories of approach can be identified for fire detection: traditional fire alarms and vision sensor-assisted fire detection. Traditional fire alarm systems are based on sensors that require close proximity for activation, such as infrared and optical sensors. These sensors are not well suited to critical environments and need human involvement to confirm a fire in the case of an alarm, involving a visit to the location of the fire. Furthermore, such systems cannot usually provide information about the size, location, and burning degree of the fire. To overcome these limitations, numerous vision sensor-based methods have been explored by researchers in this field [11-14]; these have the advantages of less human interference, faster response, affordable cost, and larger surveillance coverage. In addition, such systems can confirm a fire without requiring a visit to the fire's location, and can provide detailed information about the fire including its location, size, and degree, etc. Despite these advantages, there are still some issues with these systems, e.g. the complexity of the scenes under observation, irregular lighting, and low-quality frames; researchers have made several efforts to address these aspects, taking into consideration both color and motion features.

Chen et al. [8] examined the dynamic behavior of fires using RGB and HSI color models and proposed a decision rule-assisted fire detection approach, which uses the irregular properties of fire for detection. Their approach is based on frame-to-frame differences, and hence cannot distinguish between fire and fire-colored moving regions. Marbach et al. [15] investigated the YUV color space using motion information to classify pixels into fire and non-fire components. Toreyin et al. [16] used temporal and spatial wavelet analysis to determine fire and non-fire regions. Their approach uses many heuristic thresholds, which greatly restricts its real-world implementation. Han et al. [17] compared normal frames with their color information for tunnel fire detection; this method is suitable only for static fires, as it is based on numerous parameters. Celik et al. [18] explored the YCbCr color space and presented a pixel classification method for flames. To this end, they proposed novel rules for separating the chrominance and luminance components. However, their method is unable to detect fire from a large distance or at small scales, which are important in the early detection of fires. In addition to these color space-based techniques, Borges et al.

[19] utilized the low-level features including color, skewness, and roughness in combination with a Bayes classifier for fire recognition.

Rafiee et al. [20] investigated a multi-resolution 2D wavelet analysis to improve the thresholding mechanism in the RGB color space. Their method reduced the rate of false alarms by considering variations in energy as well as shape; however, false alarms can be higher in this approach for the case of rigid body movements within the frames, such as the movement of a human arm in the scene. In [21], the authors presented a modified version of [20] based on a YUC color model, which obtained better results than the RGB version. Another similar method based on color information and an SVM classifier is presented in [22]. This method can process 20 frames/sec; however, it cannot detect a fire from a large distance or of small size, which can occur in real-world surveillance footage. Color-based methods typically generate more false alarms due to variations in shadows and brightness, and often mis-classify people wearing red clothes or red vehicles. Mueller et al. [23] attempted to solve this issue by analyzing changes in the shape of a fire and the movement of rigid objects. Their algorithm can distinguish between rigid moving objects and a flame, based on a feature vector extracted from the optical flow and the physical behavior of a fire. De Lascio et al. [24] combined color and motion information for the detection of fire in surveillance videos. Dimitropoulos et al. [25] used spatio-temporal features based on texture analysis followed by an SVM classifier to classify candidate regions of the video frames into fire and non-fire. This method is heavily dependent on the parameters used; for instance, small-sized blocks increase the rate of false alarms, while larger blocks reduce its sensitivity. Similarly, the time window is also crucial to the performance of this system; smaller values reduce the detection accuracy, while larger values increase the computational complexity. These dependencies greatly affect the feasibility of this approach for implementation in real surveillance systems. Recently, the authors of [21] proposed a real-time fire detection algorithm based on color, shape, and motion features, combined in a multi-expert system. The accuracy of this approach is higher than that of other methods; however, the number of false alarms is still high, and the accuracy of fire detection can be further improved. A survey of the existing literature shows that computationally expensive methods have better accuracy, and simpler methods compromise on accuracy and the rate of false positives. Hence, there is a need to find a better trade-off between these metrics for several application scenarios of practical interest, for which existing computationally expensive methods do not fit well.

To address the above issues, we investigate convolutional neural network (CNN)-based deep features for early fire detection in surveillance networks. The key contributions can be summarized as follows:

1. We avoid the time-consuming efforts of conventional hand-crafted features for fire detection, and explore deep learning architectures for early fire detection in closed-circuit television (CCTV) surveillance networks for both indoor and outdoor environments. Our proposed fire detection framework improves fire detection accuracy and reduces false alarms, compared to state-of-the-art methods. Thus, our algorithm can play a vital role in the early detection of fire to minimize damage.
2. We train and fine-tune an AlexNet architecture [26] for fire detection using a transfer learning strategy. Our model outperforms conventional hand-engineered features based fire detection methods. However, the model remains comparatively large in size (238 MB), making its implementation difficult in resource-constrained equipment.
3. To reduce the size of the model, we fine-tune a model with a similar architecture to the SqueezeNet model for fire detection at the early stages. The size of the model was reduced from 238 MB to 3 MB, thus saving an extra space of 235 MB, thus minimizing the cost and making its implementation more feasible in surveillance networks. The proposed model requires 0.72 GFLOPS/image compared to AlexNet, whose computational complexity is 2 GFLOPS/image. This makes our proposed model more efficient in terms of inference, allowing it to process multiple surveillance streams.
4. An intelligent feature map selection algorithm is proposed for choose appropriate feature maps from the convolutional layers of the trained CNN, which are sensitive to fire regions. These feature maps allow a more accurate segmentation of fire compared to other hand-crafted methods. The segmentation information can be further analyzed to assess the essential characteristics of the fire, for instance its growth rate. Using this approach, the severity of the fire and/or its burning degree can also be determined. Another novel characteristic of our system is the ability to identify the object which is on fire, using a pre-trained model trained on 1,000 classes of objects in the ImageNet dataset. This enables our approach to determine whether the fire is in a car, a house, a forest or any other object. Using this semantic information, firefighters can prioritize their targets by primarily focusing on regions with the strongest fire.

The remainder of this paper is organized as follows. We propose our architecture in Section 2. Our experimental results using benchmark datasets and a feasibility analysis of the proposed work are discussed in Section 3. Finally, the manuscript is concluded in Section 4 and possible future research directions are suggested.

II. The Proposed Framework

Fire detection using hand-crafted features is a tedious task, due to the time-consuming method of features engineering. It is particularly challenging to detect a fire at an early stage in scenes with changing lighting conditions, shadows, and fire-like objects; conventional low-level feature-based methods generate a high rate of false alarms and have low detection accuracy. To overcome these issues, we investigate deep learning models for possible fire detection at early stages during surveillance. Taking into consideration the accuracy, the embedded processing capabilities of smart cameras, and the number of false alarms, we examine various deep CNNs for the target problem. A systematic diagram of our framework is given in Fig. 1.

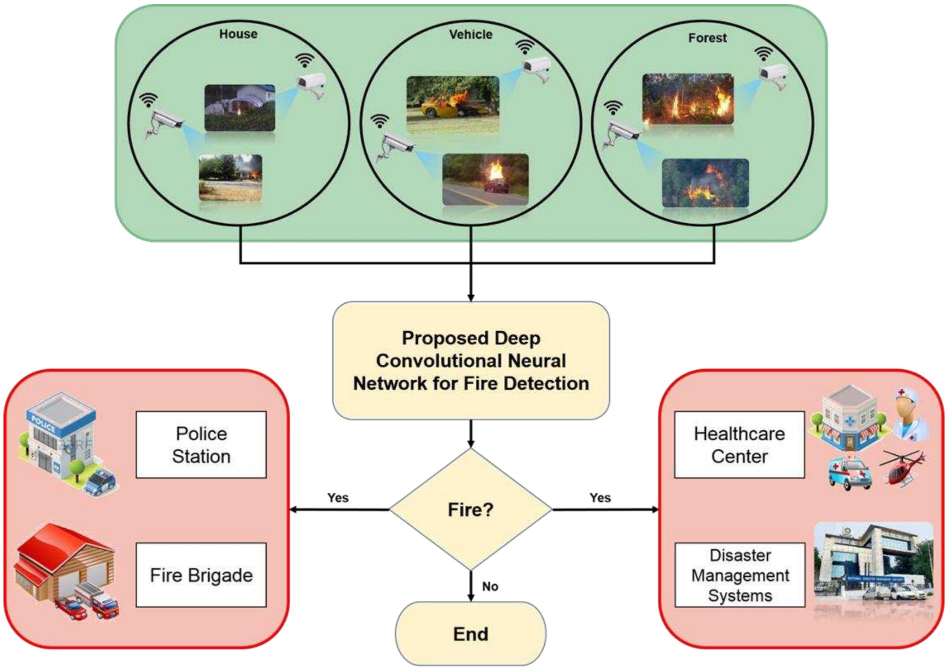

House
Vehicle
Forest
Proposed Deep Convolutional Neural Network for Fire Detection
Fire?
Yes
Yes
No
End
Police Station
Fire Brigade
Healthcare Center
Disaster Management Systems

Fig. 1: Overview of the proposed system for fire detection using a deep CNN

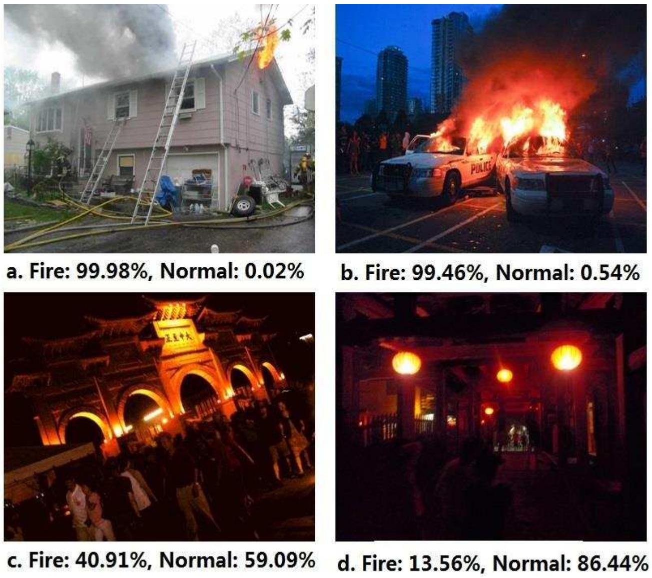
A. Convolutional Neural Network Architecture

CNNs have shown encouraging performance in numerous computer vision problems and applications, such as object detection and localization [27, 28], image segmentation, super-resolution, classification [29-31], and indexing and retrieval [32]. This widespread success is due to their hierarchical structure, which automatically learns very strong features from raw data. A typical CNN architecture consists of three well-known processing layers: 1) a convolution layer, where various feature maps are produced when different kernels are applied to the input data; 2) a pooling layer, which is used for the selection of maximum activation considering a small neighborhood of feature maps received from the previous convolution layer; the goal of this layer is to achieve translation invariance to some extent and dimensionality reduction; and 3) a fully connected layer which models high-level information from the input data and constructs its global representation. This layer follows numerous stacks of convolution and pooling layers, thus resulting in a high-level representation of the input data. These layers are arranged in a hierarchical architecture such that the output of one layer acts as the input of the next layer. During the training phase, the weights of all neurons in convolutional kernels and fully connected layers are adjusted and learnt. These weights model the representative characteristics of the input training data, and in turn can perform the target classification.

We use a model with an architecture similar to that of SqueezeNet [33], modified in accordance with our target problem. The original model was trained on the ImageNet dataset and is capable of classifying 1000 different objects. In our case, however, we used this architecture to detect fire and non-fire images. This was achieved by reducing the number of neurons in the final layer from 1000 to 2. By keeping the rest of the architecture similar to the original, we aimed to reuse the parameters to solve the fire detection problem more effectively.

There are several motivational reasons for this selection, such as a lower communication cost between different servers in the case of distributed training, a higher feasibility of deployment on FPGAs, application-specific integrated circuits, and other hardware architectures with memory constraints and lower bandwidth. The model consists of two regular convolutional layers, three max pooling layers, one average pooling layer, and eight modules called “fire modules”. The input of the model is color images with dimensions of 224×224×3 pixels. In the first convolution layer, 64 filters of size 3×3 are applied to the input image, generating 64 feature maps. The maximum activations of these 64 features maps are selected by the first max pooling layer with a stride of two pixels, using a neighborhood of 3×3 pixels. This reduces the size of the feature maps by factor of two, thus retaining the most useful information while discarding the less important details. Next, we use two fire modules of 128 filters, followed by another fire module of 256 filters. Each fire module involves two further convolutions, squeezing, and expansion. Since each module consists of multiple filter resolutions and there is no native support for such convolution layers in the Caffe framework [34], an expansion layer was introduced, with two separate convolution layers in each fire module. The first convolution layer contains 1 x 1 filters, while the second layer consists of 3×3 filters. The output of these two layers is concatenated in the channel dimension. Following the three fire modules, there is another max pooling layer which operates in the same way as the first max pooling layer. Following the last fire module (Fire9) of 512 filters, we modify the convolution layer according to the problem of interest by reducing the number of classes to two (M=2 (fire and normal)). The output of this layer is passed to the average pooling layer, and result of this layer is fed directly into the Softmax classifier to calculate the probabilities of the two target classes.

A significant number of weights need to be properly adjusted in CNNs, and a huge amount of training data is usually required for this. These parameters can suffer from overfitting if insufficient training data is used. The fully connected layers usually contain the most parameters, and these can cause significant overfitting. These problems can be avoided by introducing regularization layers such as dropout, or by replacing dense fully connected layers with convolution layers. In view of this, a number of models were trained based on the collected training data. Several benchmark datasets were then used to evaluate the classification performance of these models. During the experiments, a transfer learning strategy was also explored in an attempt to further improve the accuracy. Interestingly, we achieved an improvement in classification accuracy of approximately 5% for the test data after fine-tuning. A transfer learning strategy can solve problems more efficiently based on the re-use of previously learned knowledge. This reflects the human strategy of applying existing knowledge to different problems in several domains of interest. Employing this strategy, we used a pre-trained SqueezeNet model and fine-tuned it according to our classification problem with a slower learning rate of 0.001. We also removed the last fully connected layers to make the architecture as efficient as possible in terms of classification accuracy. The process of fine-tuning was executed for 10 epochs; this increased the classification accuracy from 89.8% to 94.50%, thus giving an improvement of 5%.


a. Fire: 99.98%, Normal: 0.02%

b. Fire: 99.46%, Normal: 0.54%

c. Fire: 40.91%, Normal: 59.09%

d. Fire: 13.56%, Normal: 86.44%

Fig. 2: Prediction scores for a set of query images using the proposed deep CNN.

B. Difference with other network models

The key difference of our proposed CNN architecture in Fig.3 with SqueezeNet [28] is that our model simplifies the SqueezeNet model by removing no residual connections, which is more light-weight and balanced computational efficiency.

As shown in Fig.3, looking at the architectural similarity between our CNN's Fire and Inception modules, note that in Inception modules, Fire modules have multiple sizes of filters at the same level of depth in the NN. For example, Inception-v1 modules have multiple instances with 1x1, 3x3, and 5x5 filters alongside each other. This arose the relevant question "how does a CNN architect decide how many of each size of filter to have in each module?" Some versions of the inception modules have 10 or more filter banks per module. Doing careful A/B comparisons of "how many of each type of filter" would easily lead to a combinatorial explosion. But, in the Fire modules, there are just 3 filter banks (1x1_1, 1x1_2, and 3x3_2). With this setup, it can be further asked that: What are the tradeoffs in "many 1x1_2 and few 3x3_2" vs "few 1x1_2 and many 3x3_2" in terms of metrics such as model size and accuracy? From [1], it is evident that 50% 1x1_2 and 50% 3x3_2 filters generate the same accuracy level as 99% 3x3_2. But there is a significant difference in the model size and computational footprint of these models. The lesson learnt is the suitability to adopt, to some extent, a simple step-by-step methodology: look for the point where adding more spatial resolution to the filters stops improving accuracy, and stop there; otherwise computation and model parameters are being wasted.

Also, in comparison to other network models like AlexNet [26] and GoogleNet [27]. Our proposed network is light-weight, requiring a memory of 3 MB which is less than AlexNet and GoogleNet. It also is computationally inexpensive, requiring only 0.72 GFLOPS/image compared to other networks such as AlexNet (which needs 2 GFLOS/image). Thus, our proposed model maintains a better trade-off between the computational complexity, memory requirement, fire detection accuracy and number of false alarms compared to other networks.

Looking at GoogLeNet-v1, some of the Inception-v1 modules are set up such that the early filter banks have 75% the number of filters as the late filter banks. This is like they have a "squeeze ratio" (SR) of 0.75. Another interesting point was to find the tradeoffs that emerge if the number of filters at the beginning of each module are more aggressively cut down. It was experimentally found, again, that there is a saturation point where going from SR=0.75 to SR=1.0; here, the increase in computational footprint and model size does not correspond to a significant improvement, but it does not improve accuracy. Thus, the Fire modules have been very useful in our experience for understanding the tradeoffs that emerge when selecting the number of filters inside of the CNN modules.

C. Deep CNN for Fire Detection and Localization

Although deep CNN architectures learn very strong features automatically from raw data, some effort is required to train the appropriate model considering the quality and quantity of the available data and the nature of the target problem. We trained various models with different parameter settings, and following the fine-tuning process obtained an optimal model which can detect fire from a large distance and at a small scale, under varying conditions, and in both indoor and outdoor scenarios.

Another motivational factor for the proposed deep CNN was the avoidance of pre-processing and features engineering, which are required by traditional fire detection algorithms. To test a given image, it is fed forward through the deep CNN, which assigns a label of ‘fire’ or ‘normal’ to the input image.

This label is assigned based on probability scores computed by the network. The higher probability score is taken to be the final class label of the input image. A set of sample images with their predicted class labels and probability scores is given in Fig. **2**. To localize a fire in a sample image, we employ the framework given in Fig. **3**.

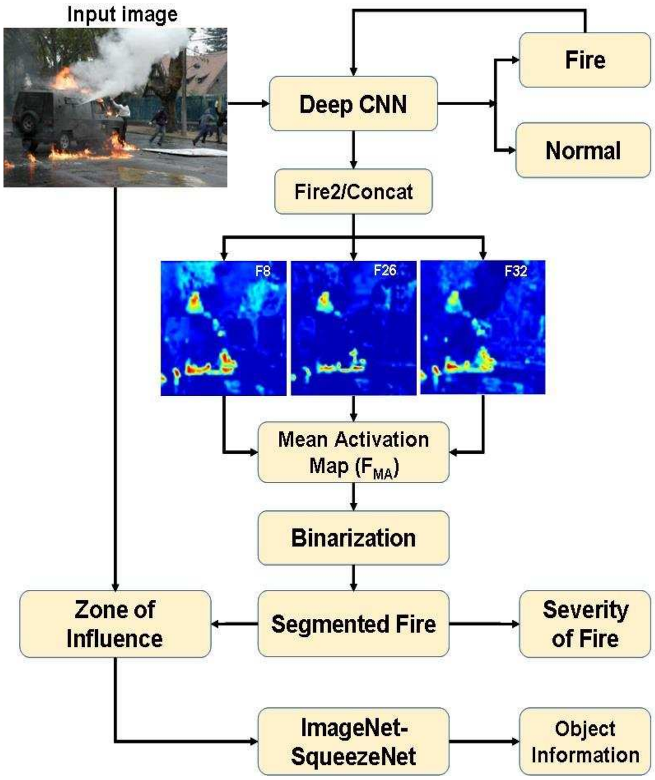


Fig. 3: Fire localization using the proposed deep CNN.

First, a prediction is obtained from our deep CNN. In non-fire cases, no further action is performed; in the case of fire, we perform further processing of its localization, as given in Algorithms **1** and **2**.

After analyzing all the feature maps of the different layers of our proposed CNN using Algorithm 1, feature maps 8, 26, and 32 of the "Fire2/Concat" layer were found to be sensitive to fire regions and to be appropriate for fire localization. We therefore fused these three feature maps and applied binarization to segment the fire. A set of sample fire images with their segmented regions is given in Fig. **4**.

The segmented fire is used for two further purposes: 1) determining the severity level/burning degree of the scene under observation; 2) finding the zone of influence from the input fire image. The burning degree can be determined from the number of pixels in the segmented fire. The zone of influence can be calculated by subtracting the segmented fire regions from the original input image. The resultant zone of influence image is then passed from the original SqueezeNet model [33], which predicts its label from 1000 objects. The object information can be used to determine the situation in the scene, such as a fire in a house, a forest, or a vehicle. This information, along with the severity of the fire, can be reported to the fire brigade to take appropriate action.

```
**Algorithm 1.** Feature Map Selection Algorithm for Localization
**Input:** Training samples (TS), ground truth (GT), and the proposed deep CNN model (CNN-M)
1. Forward propagate TS through CNN-M
2. Select the feature maps $F_N$ from layer L of CNN-M
3. Resize GT and $F_N$ to 256×256 pixels
4. Compute mean activations map $F_{MAi}$ for $F_N$
5. Binarize each feature map $F_i$ as follows:
       $F(x,y)_{bin(i)} = \begin{Bmatrix} 1, & F(x,y)_i > F_{MA(i)} \\ 0, & Otherwise \end{Bmatrix}$
6. Calculate the hamming distance $HD_i$ between GT and each
       feature map $F_{bin(i)}$ as follows:
       $HD_i = |F_{bin(i)} - GT|$
       This results in $TS \times F_N$ hamming distances
7. Calculate the sum of all resultant hamming distances, and
       shortlist the minimum hamming distances using threshold T
8. Select appropriate feature maps according to the shortlisted
       hamming distances
**Output:** Feature maps sensitive to fire
```

```
**Algorithm 2.** Fire Localization Algorithm
**Input:** Image I of the video sequence and the proposed deep CNN model (CNN-M)
1. Select a frame from the video sequence and forward propagate it
       through CNN-M
2. **IF** predicted label = non-fire **THEN**
       No action
   **ELSE**
       a) Extract feature maps 8, 26, and 32 ($F_8$, $F_{26}$, $F_{32}$) from
          the "Fire2/Concat" layer of CNN-M
       b) Calculate mean activations map ($F_{MA}$) for $F_8$, $F_{26}$, and
          $F_{32}$
       c) Apply binarization on $F_{MA}$ through threshold T as
          follows:
          $F_{Localize} = \begin{Bmatrix} 1, & F_{MA} > T \\ 0, & Otherwise \end{Bmatrix}$
       d) Segment fire regions from $F_{MA}$
   **END**
**Output:** Binary image with segmented fire $I_{localize}$
```

III. Experimental Results and Discussion

The experiments performed to verify the performance of our approach are described in this section. Starting with the experimental details, we give information about the system specification and the datasets used for the experiments. Following this, the experimental results for various fire datasets are presented, followed by a comparison with existing approaches in terms of fire detection and localization. Finally, we describe tests verifying the superiority of our method from the perspective of robustness. Our approach is referred to as "CNNFire" throughout the experiments.

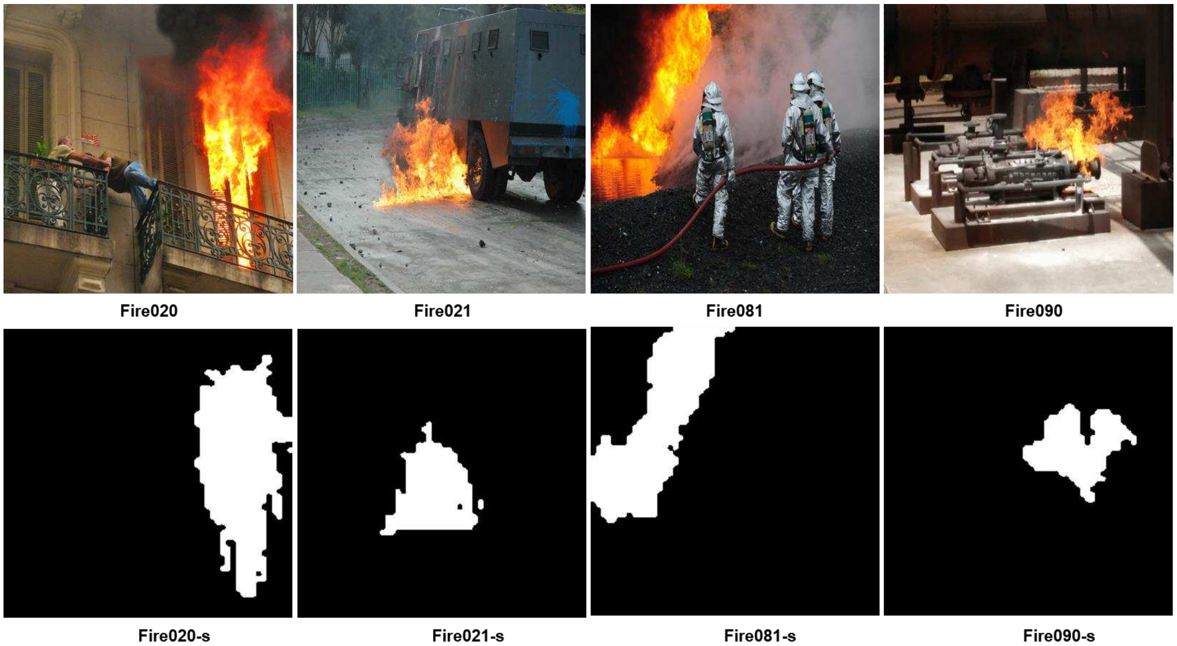


Fig. 4: Sample images and the corresponding localized fire regions using our approach. The first row shows the original images, while the second row shows the localized fire regions.

A. Experimental Setup and Datasets

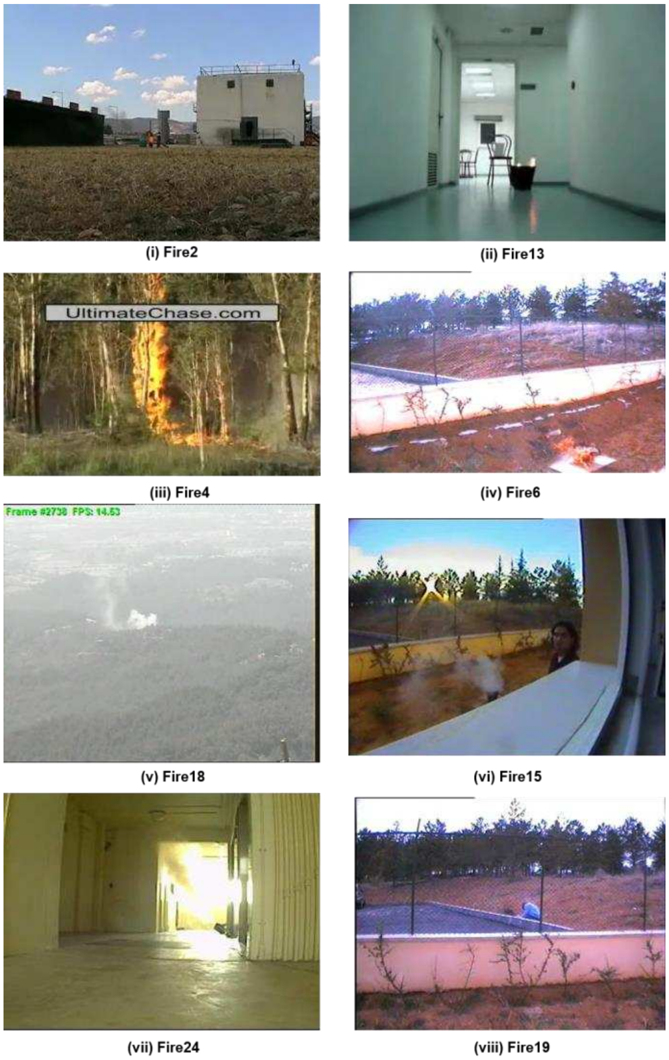
We conducted the experiments using a system with the following specifications: NVidia GeForce GTX TITAN X (Pascal) with 12 GB onboard memory using a deep learning framework [34] and Ubuntu OS installed on an Intel Core i5 CPU with 64 GB RAM. A total of 68,457 images were used in the experiments; these were obtained from well-known fire datasets including those of Foggia et al. [21] with 62,690 frames, Chino et al. [35] with 226 images, and other dataset sources [14, 36]. For the training and testing phases of the experiments, we followed the experimental strategy of [21], where 20% and 80% of the data are used for training and testing, respectively. Using this strategy, we trained our proposed SqueezeNet model with 5,258 fire images and 5,061 non-fire images, resulting in a training dataset of 10,319 images. The details of the experiments using the different fire datasets and their comparison with state-of-the-art techniques are given in subsequent sections.

B. Experiments on Dataset1

Our experiments for testing the performance of the proposed framework are mainly based on two datasets: 1) Foggia et al. [21] (Dataset1), and Chino et al. [35] (Dataset2). The reasons for using each of these datasets are provided in the relevant sections. Dataset1 contains a total of 31 videos captured in different environments. Of these videos, 14 videos include a fire, while 17 are normal videos. A variety of challenges, including its larger size compared to other available datasets, make this dataset particularly suitable for these experiments. For example, some of the normal videos include fire-like objects; this makes fire detection more challenging, and hence fire detection methods using color features may wrongly classify these frames. In addition, a set of videos are captured in mountain areas and contain clouds and fog, for which motion-based fire detection schemes may not work properly. These situations can occur in the real world, and they are therefore introduced in this dataset to make it as challenging as possible. This is the primary reason for the selection of this dataset for the experimental evaluation of our work. Further information about Dataset1 is given in Table **I**. A set of sample images from Dataset1 are given in Fig. **5**, and the collected experimental results using Dataset1 are tabulated and compared with related methods in Table **II**.

Fig. **5** shows a set of representative images from Dataset1. The top four images were taken from videos containing a fire, and the remaining four are from videos without a fire. As described at the start of this section, this dataset has many challenges, which are evident from the given set of images. The dataset contains videos captured in both indoor and outdoor environments (see Figs. 5 (ii) and (vii) for indoor and Figs. 5 (i), (iii-vi), and (viii) for outdoor examples). The distance of the camera from the fire and the size of the fire also vary a lot in the videos of Dataset1. For example, Fig. 5 (i) illustrates a video where the fire is far away and the size is very small; conversely, the size of the fire in Fig. 5 (iii) is much larger, and it is at a shorter distance. Fig. 5 (ii) represents an indoor environment with a small fire. Fig. 5 (iv) contains both a fire at a medium distance and red objects; this is similar to Fig. 5 (viii) except for the fact that the latter contains no fire. This poses a challenge and can be used to evaluate the effectiveness of color-based fire detection algorithms. Figs. 5 (v) and (vi) represent normal images with smoke and sunlight, which both look like fire. A similar effect is illustrated in the indoor scenario in Fig. 5 (vii), where the sun is rising and is reflected in the window. These variations make the dataset much more challenging for fire detection algorithms.

For a comparison of our results with state-of-the-art methods for Dataset1, we selected a total of six related works. This selection was based on criteria including the features used in the related works, their year of publication, and the dataset under consideration. We then compared our work with the selected fire detection algorithms, as shown in Table **II**.


(i) Fire2

(ii) Fire13

(iii) Fire4

(iv) Fire6

(v) Fire18

(vi) Fire15

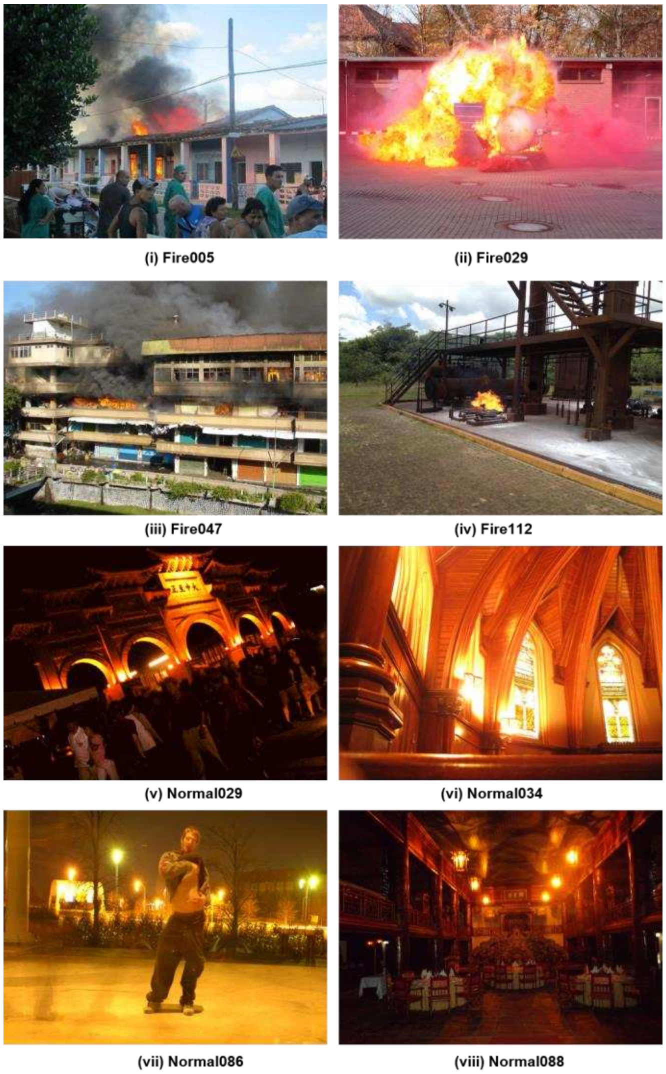
(vii) Fire24

(viii) Fire19

Fig. **5**: A set of representative images from Dataset1. The top four images are taken from videos of fires, while the remaining four images are from non-fire videos

The selected works use various low-level features and different datasets, and their year of publication ranges from 2004 to 2015. The results show that Celik et al. [18] and Foggia et al. [21] are the best algorithms in terms of false negatives. However, their results are not impressive in terms of the other metrics of false positives and accuracy. From the perspective of false positives, the algorithm of Habibuglu et al. [22] performs best, and dominates the other methods. However, its false negative rate is 14.29%, the worst result of all the methods examined. The accuracy of the four other methods is also better than this method, with the most recent method [21] being the best. However, the false positive score of 11.67% is still high, and the accuracy could be further improved.

To achieve a high accuracy and a low false positive rate, we explored the use of deep features for fire detection. We first used the AlexNet architecture without fine tuning, which resulted in an accuracy of 90.06% and reduced false positives from 11.67% to 9.22%. In the baseline AlexNet architecture, the weights of kernels are initialized randomly and these are modified during the training process considering the error rate and accuracy. We also applied the strategy of transfer learning [37] whereby we initialized the weights from a pre-trained AlexNet model with a low learning rate of 0.001 and modified the last fully connected layer according to our problem. Interestingly, we obtained an improvement in accuracy of 4.33% and reductions in false negatives and false positives of up to 8.52% and 0.15%, respectively.


(i) Fire005

(ii) Fire029

(iii) Fire047

(iv) Fire112

(v) Normal029

(vi) Normal034

(vii) Normal086

(viii) Normal088

Fig. **6**: Representative images from Dataset2. The top four images include fires, while the remaining four images represent fire-like normal images

Although the results of the proposed fine-tuned AlexNet are good compared to other existing methods, there are still certain limitations. Firstly, the size of this model is comparatively large (approx. 238 MB), thereby restricting its implementation in CCTV networks. Secondly, the rate of false alarms (false positives) is 9.07%, which is still high and would be problematic for fire brigades and disaster management teams. With these strong motivations, we explored SqueezeNet, a lightweight architecture, for this problem. We repeated the experiments for this new architecture and achieved an improvement of 0.11% in accuracy. Furthermore, the rate of false alarms was reduced from 9.07% to 8.87%. The rate of false negatives remained almost the same. Finally, the major achievement of the proposed framework was the reduction of the model size from 238 MB to 3 MB, thus saving an extra 235 MB, which can greatly minimize the cost of CCTV surveillance systems.

TABLE I
Details of Dataset1

Video Name	Resolution	Frames	Frame Rate	Modality	Description
Fire1	320×240	705	15	Fire	Fire in a bucket with person walking around it
Fire2	320×240	116	29	Fire	Fire at a comparatively long distance from the camera in a bucket
Fire3	400×256	255	15	Fire	A large forest fire
Fire4	400×256	240	15	Fire	Same description as Fire3
Fire5	400×256	195	15	Fire	Same description as Fire3
Fire6	320×240	1200	10	Fire	Fire on the ground with red color
Fire7	400×256	195	15	Fire	Same description as Fire3
Fire8	400×256	240	15	Fire	Same description as Fire3
Fire9	400×256	240	15	Fire	Same description as Fire3
Fire10	400×256	210	15	Fire	Same description as Fire3
Fire11	400×256	210	15	Fire	Same description as Fire3
Fire12	400×256	210	15	Fire	Same description as Fire3
Fire13	320×240	1650	25	Fire	An indoor environment with fire in a bucket
Fire14	320×240	5535	15	Fire	A paper box, inside which a fire is burning
Fire15	320×240	240	15	Normal	Smoke visible from a closed window with the appearance of a red reflection of the sun on the glass
Fire16	320×240	900	10	Normal	Smoke from a pot near a red dust bin.
Fire17	320×240	1725	25	Normal	Smoke on the ground with nearby trees and moving vehicles
Fire18	352×288	600	10	Normal	Smoke on the hills, far from the camera
Fire19	320×240	630	10	Normal	Smoke on red-colored ground
Fire20	320×240	5958	9	Normal	Smoke on the hills, with nearby red buildings
Fire21	720×480	80	10	Normal	Smoke at a larger distance behind trees
Fire22	480×272	22500	25	Normal	Smoke behind hills in front of UOS
Fire23	720×576	6097	7	Normal	Smoke above hills
Fire24	320×240	342	10	Normal	Smoke in a room
Fire25	352×288	140	10	Normal	Smoke at a larger distance from a camera in a city
Fire26	720×576	847	7	Normal	Same description as Fire24
Fire27	320×240	1400	10	Normal	Same description as Fire19
Fire28	352×288	6025	25	Normal	Same description as Fire18
Fire29	720×576	600	10	Normal	Red buildings covered in smoke
Fire30	800×600	1920	15	Normal	A lab with a red front wall, where a person moves, holding a red ball
Fire31	800×600	1485	15	Normal	A lab with red tables, and a person moving with a red bag and a ball

TABLE III
Comparison of different fire detection methods for Dataset2

Technique		Precision	Recall	F-Measure
Proposed Method	After FT	**0.86**	0.97	**0.91**
	Before FT	0.84	0.87	0.85
AlexNet after FT		0.82	**0.98**	0.89
AlexNet before FT		0.85	0.92	0.88
Chino et al. (BoWFire) [35]		0.51	0.65	0.57
Rudz et al. [39]		0.63	0.45	0.52
Rossi et al. [40]		0.39	0.22	0.28
Celik et al. [18]		0.55	0.54	0.54
Chen et al. [8]		0.75	0.15	0.25

TABLE II
Comparison of various fire detection methods for Dataset1

Technique	False Positives	False Negatives	Accuracy
Proposed after FT	**8.87%**	2.12%	**94.50%**
Proposed before FT	9.99%	10.39%	89.8%
AlexNet after FT	9.07%	2.13%	94.39%
AlexNet before FT	9.22%	10.65%	90.06%
Foggia et al. [21]	11.67%	**0%**	93.55%
De Lascio et al. [24]	13.33%	0%	92.86%
Habibuglu et al. [22]	5.88%	14.29%	90.32%
Rafiee et al. (RGB) [20]	41.18%	7.14%	74.20%
Rafiee et al. (YUV) [20]	17.65%	7.14%	87.10%
Celik et al. [18]	29.41%	**0%**	83.87%
Chen et al. [8]	11.76%	14.29%	87.10%

C. Experiments on Dataset2

Dataset2 consists of 226 images, with 119 fire images and 107 non-fire images. The dataset was obtained from [35], and is relatively small but contains several challenges such as red and fire-colored objects, fire-like sunlight, and fire-colored lighting in different buildings. For illustration purposes, a set of representative images are shown in Fig. **6**. It should be noted that none of the images from Dataset2 were used in the training processes of either AlexNet or our proposed model. The experimental results obtained from Dataset2 using the proposed architecture are presented in Table **III**. We compared our results with four other fire detection algorithms in terms of their relevancy, dataset, and year of publication. To ensure a fair evaluation and a full overview of the performance of our approach, we considered another set of metrics (precision, recall, and F-measure [38]) as used by [35]. In a similar way to the experiments on Dataset1, we tested Dataset2 using the fine-tuned AlexNet and our proposed fine-tuned SqueezeNet model. For the fine-tuned AlexNet, an F-measure score of 0.89 was achieved. Further improvement was achieved using our model, increasing the F-measure score from 0.89 to 0.91 and the precision from 0.82 to 0.86. It is evident from Table **III** that our work achieved better results than the state-of-the-art methods, confirming the effectiveness of the proposed deep CNN framework.

D. Fire Localization: Results and Discussion

In this section, the performance of our approach is evaluated in terms of fire localization and understanding of the scene under observation. True positive and false positive rates were computed to evaluate the performance of fire localization. The feature maps we used to localize fire were smaller than the ground truth images, and were therefore resized to match the size of the ground truth images. We then computed the number of overlapping fire pixels in the detection maps and ground truth images, and used these as true positives. Similarly, we also determined the number of non-overlapping fire pixels in the detection maps and interpreted these as false positives. One further reason for using SqueezeNet was the ability of the model to give larger sizes for the feature maps by using smaller kernels and avoiding pooling layers. This allowed us to perform a more accurate localization when the feature maps were resized to match the ground truth images.

Our system selects suitable features which are sensitive to fire using Algorithm 1, and localizes the fire using Algorithm 2. These localization results are compared with those of several state-of-the-art methods such as Chen et al. [8], Celik et al. [18], Rossi et al. [40], Rudz et al. [39], and Chino et al. (BoWFire) [35], as shown in Fig. **7**. We report three different results for our CNNFire based on the threshold T of the binarization process in Algorithm 2. It can be seen from Fig. 7 that our approach maintains a better balance between the true positive rate and false positive rate, making it more suitable for fire localization in surveillance systems.

Fig. **8** shows the results of all methods for a sample image from Dataset2. The results of BoWFire, color classification, Celik and Rudz are almost the same. Rossi gives the worst results in this case, and Chen is better than Rossi. The results from CNNFire are similar to the ground truth.

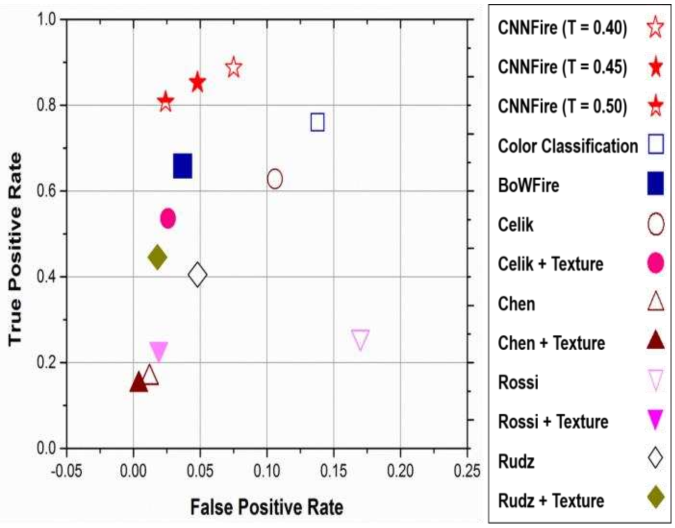


Fig. 7: Comparison of our CNNFire approach with other methods

Fig. **9** highlights the performance of all methods for another sample image, with a higher probability of false positives. Although BoWFire has no false positives for this case, it misses some fire regions, as is evident from its result. Color classification and Celik detect the fire regions correctly, but give larger regions as false positives. Chen fails to detect the fire regions of the ground truth images. Rossi does not detect fire regions at all for this case. The false positive rate of Rudz is similar to our CNNFire, but the fire pixels detected by this approach are scarce. Although our method gives more false positives than the BoWFire method, it correctly detects the fire regions which are more similar to the ground truth images.

In addition to fire detection and localization, our system can determine the severity of the detected fire and the object under observation. For this purpose, we extracted the zone of influence (ZOI) from the input image and segmented fire regions. The ZOI image was then fed forward to the SqueezeNet model, which was trained on the ImageNet dataset with 1000 classes. The label assigned by the SqueezeNet model to the ZOI image is then combined with the severity of the fire for reporting to the fire brigade. A set of sample cases from this experiment is given in Fig. **10**.

E. Robustness of the Proposed Fire Detection Method against Attacks

In addition to comparing our results with state-of-the-art methods, we tested the performance of our model against numerous attacks, i.e. all effects that can negatively affect the correct detection of a fire. Possible attacks include rotations, cropping, and noise. All attacks and their effects on performance were checked using a test image, as shown in Fig. **11** (a), which is labeled as fire with an accuracy of 99.24% by our algorithm. In Figs. 11 (b) and (e), parts of the fire are blocked by cropping a normal section from the same image and placing it over parts of the fire. The resultant images are labeled as normal with an accuracy of approximately 99% when passed through the proposed fire detection model.

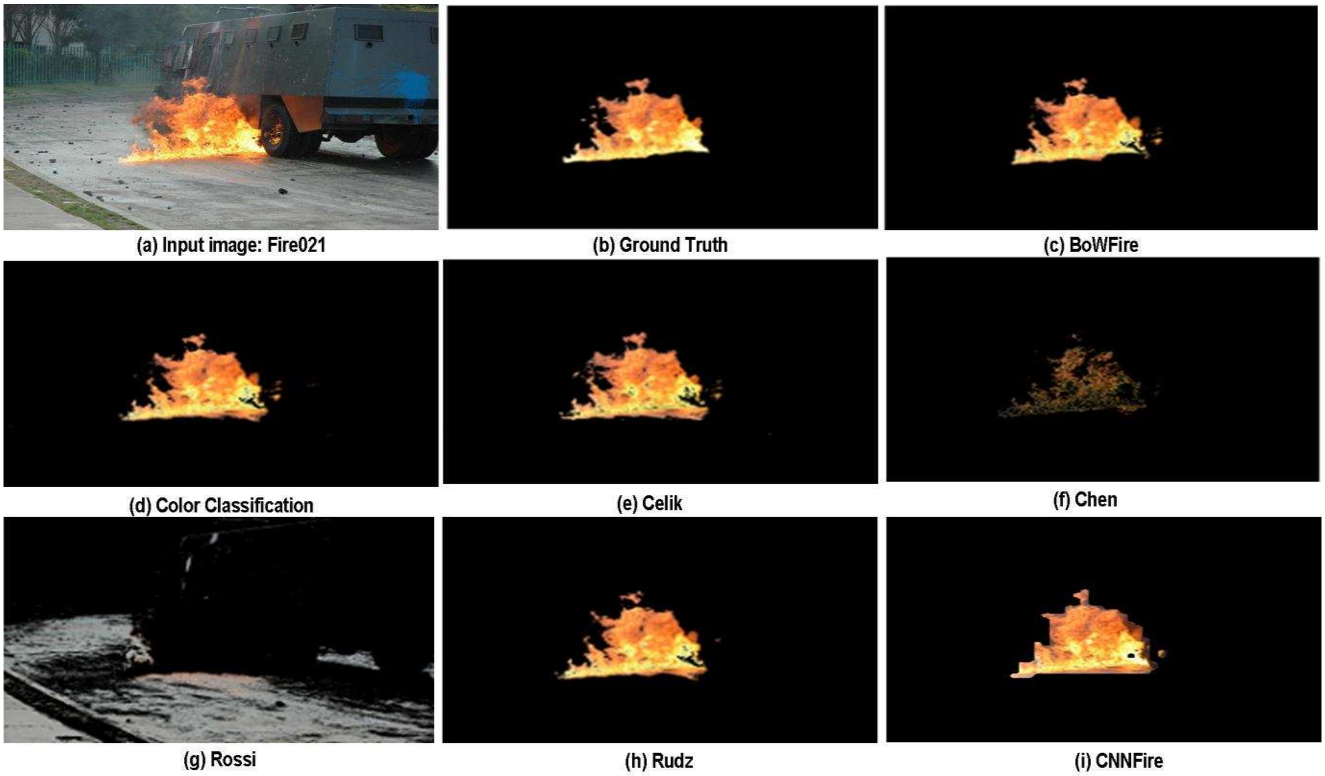


Fig. 8: Visual fire localization results of our CNNFire approach and other fire localization methods

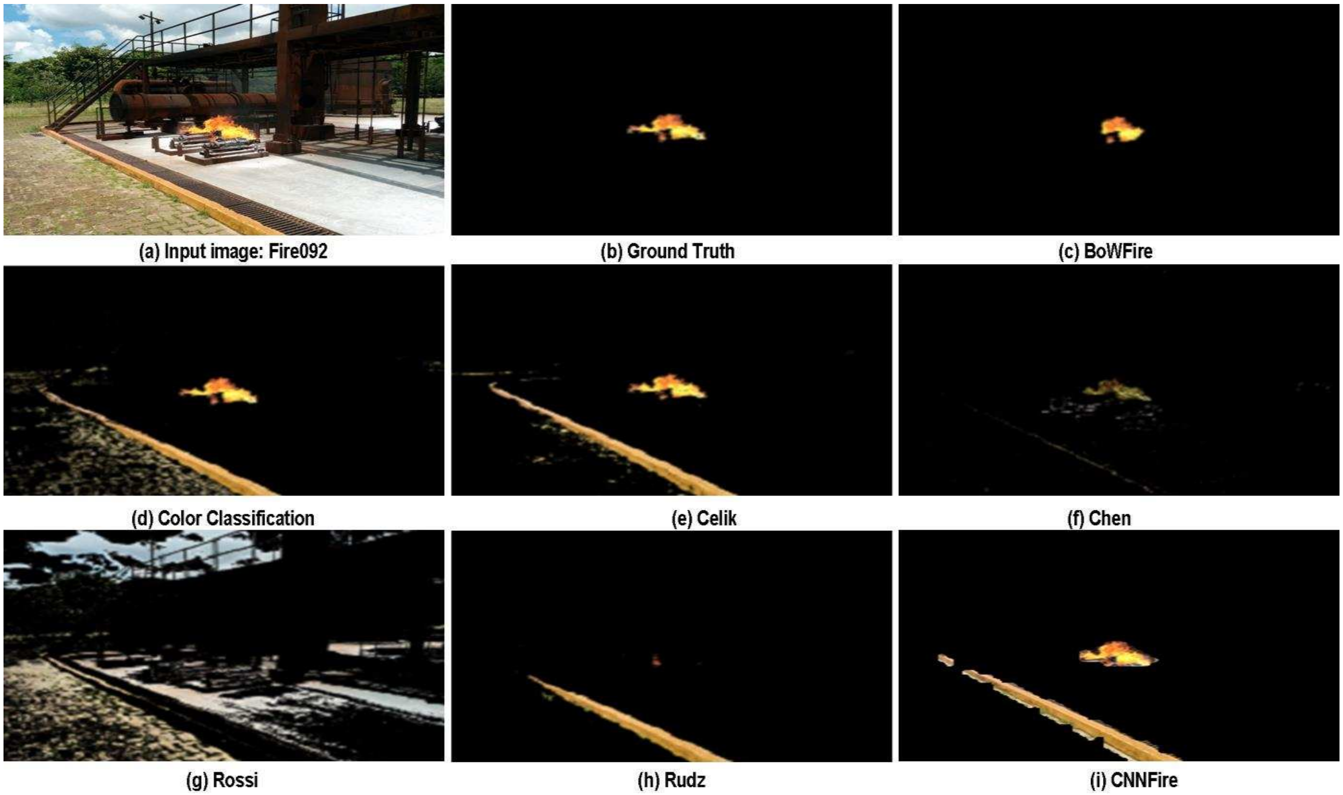


Fig. 9: Fire localization results from our CNNFire and other schemes with false positives

In Figs. 11 (c), (f), and (g), different types of noise are added to the original image, and its behavior is investigated. Interestingly, we found that the proposed model still labeled them as fire, despite a change in the quality of the images and especially the parts showing the fire. The probability scores of Figs. 11 (c) and (g) are higher than Fig. 11 (f), since the latter image of fire is more affected by the noise. Fig. 11 (d) illustrates another special test aimed at evaluating the capability of our model in terms of early fire detection. A small amount of fire is cropped from another image and is added to Fig. 11 (b). The resultant image is passed through our model, which identifies this as fire with a probability score of around 78.11%. Lastly, we investigated the behavior of the proposed model under rotation. For this purpose, we rotated the test image by 90° and 180° and passed these images through our fire detection architecture. It can be seen from Figs. 11 (h) and (i) that both images are correctly labeled as fire. We included this evaluation in experiments since in real-world surveillance systems, video frames can be exposed to different types of noise due to varying weather conditions. Thus, a fire detection system with the capability to withstand various attacks is more suitable for robust surveillance systems. Hence, our proposed architecture can be effectively used in current CCTV surveillance systems for fire detection with better accuracy and under a range of conditions, as verified by experiments.

F. Feasibility Analysis

In this section, the feasibility of the proposed fire detection method in terms of its implementation in real-world CCTV surveillance systems is investigated. For this purpose, we considered two different experimental setups with specifications as follows: 1) NVidia GeForce GTX TITAN X (Pascal) with 12 GB onboard memory using a deep learning framework [34] and Ubuntu OS installed on an Intel Core i5 CPU with 64 GB RAM (as described in Section III (A)); and 2) a Raspberry Pi 3 with 1.2 GHz 64-bit quad-core ARMv8 Cortex-A53 and a Broadcom BCM2837, equipped with 1024 MiB SDRAM [41]. Using these two specifications, our system can process 20 frames/sec and 4 frames/sec, respectively, with an accuracy of 94.50% and a false positive rate of 8.87%. It is worth noting that conventional cameras can acquire approximately 25-30 frames/sec and processing a single frame/sec for the possible detection of fire is sufficient due to the minor changes between frames. Similar work was done in [21], where they achieved 60 frames/sec using a traditional PC (Intel dual core T7300 with 4 GM RAM) and 3 frames/sec based on a Raspberry Pi B (ARM processor with 700 MHz and 512 MiB RAM). These authors reported an accuracy of 93.55% with a false positive rate of 11.67%.

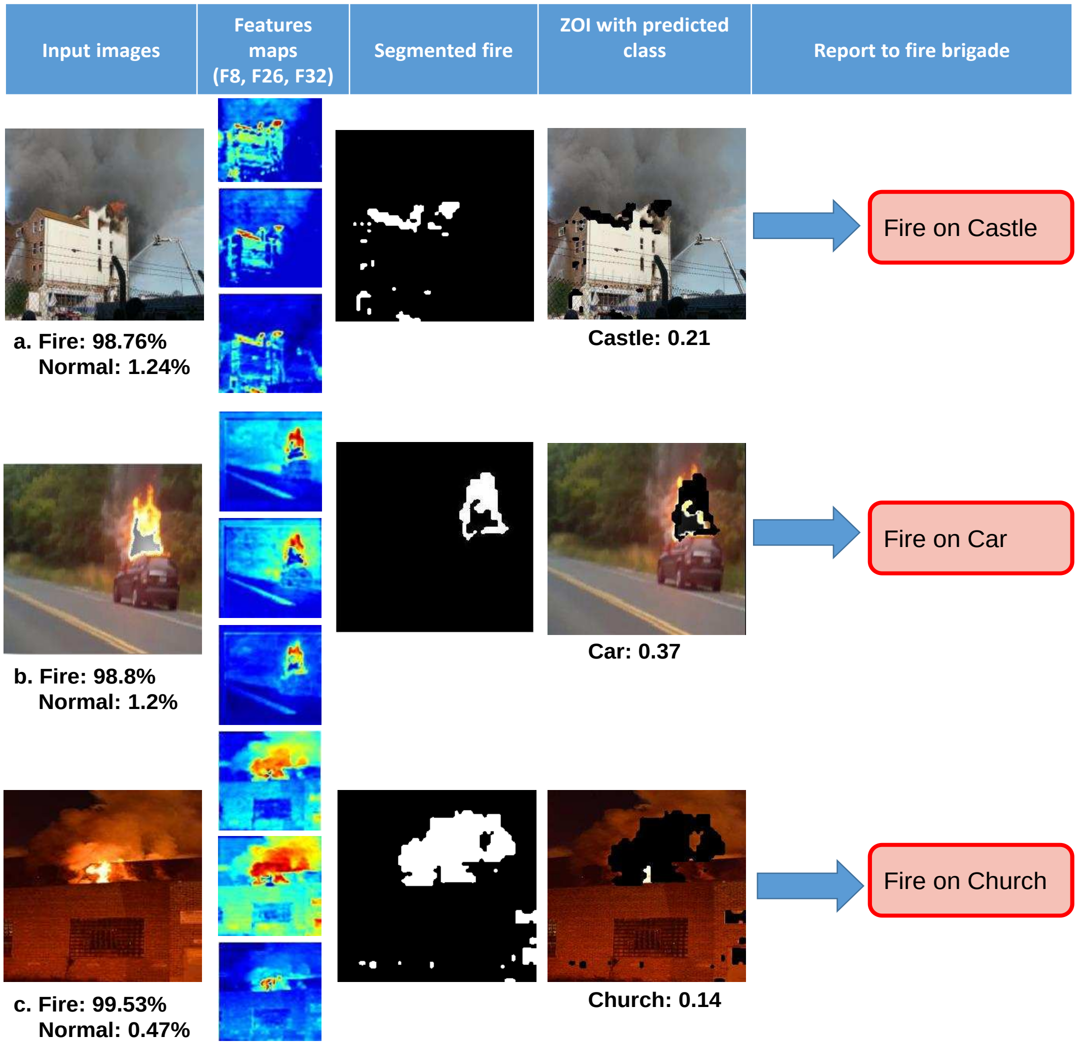
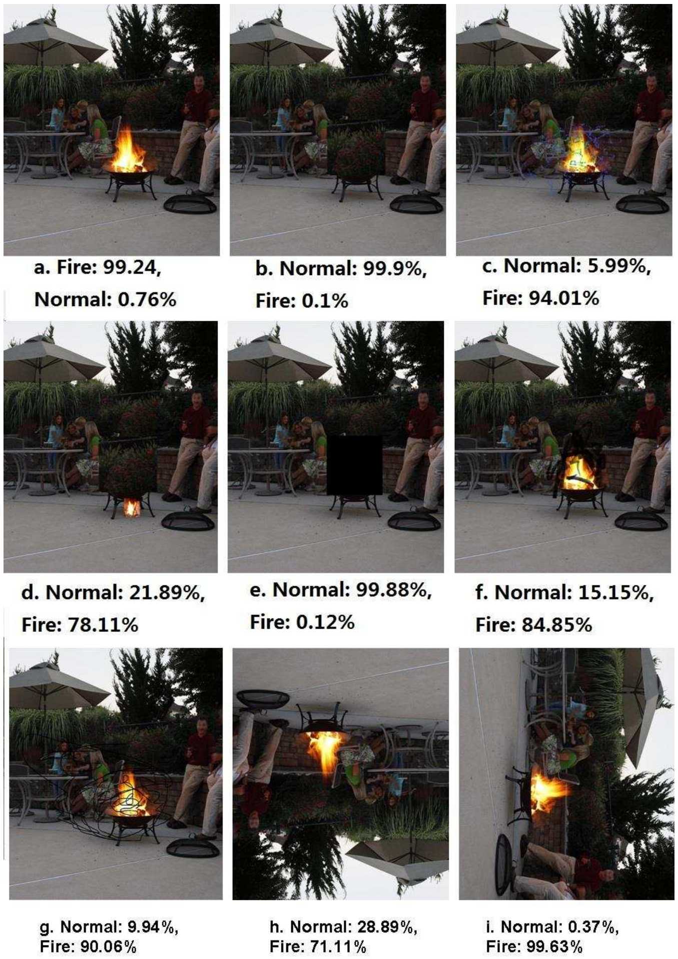


Fig. **10**: Sample outputs from our overall system: the first column shows input images with labels predicted by our CNN model and their probabilities, with the highest probability taken as the final class label; the second column shows three feature maps (F8, F26, and F32) selected by Algorithm 1; the third column highlights the results for each image using Algorithm 2; the fourth column shows the severity of the fire and ZOI images with a label assigned by the SqueezeNet model; and the final column shows the alert that should be sent to emergency services, such as the fire brigade

Related work done by the same group is reported in [24], where they obtained 70 frames/sec using the above traditional PC with 92.59% accuracy and a 6.67% false positive rate group is reported in [24], where they obtained 70 frames/sec using the above traditional PC with 92.59% accuracy and a 6.67% false positive rate. Another similar work is reported in [22], where the authors achieved 20 frames/sec with a dual core 2.2 GHz system with a 5.88% false positive rate and 90.32% accuracy. However, these scores were collected using a smaller dataset than the ones used here and in [21]. Our proposed deep CCN architecture, which has a much smaller size (3 MB) compared to the AlexNet architecture (238 MB), can successfully detect fire at an early stage with 4 frames/sec and resolution 320×240 with a 8.87% false positive rate and 94.50% accuracy. The motivation for using a Raspberry Pi 3 is its affordable price of $35 USD. In view of these statistics, it is evident that the performance of our model is better than state-of-the-art methods, and that it can be easily integrated with current surveillance systems. Finally, it is worth mentioning that our proposed model requires 0.72 GFLOPS/image compared to AlexNet's 2 GFLOPS/image, which makes it more efficient in inference, allowing it to process multiple surveillance streams.



a. Fire: 99.24, Normal: 0.76%

b. Normal: 99.9%, Fire: 0.1%

c. Normal: 5.99%, Fire: 94.01%

d. Normal: 21.89%, Fire: 78.11%

e. Normal: 99.88%, Fire: 0.12%

f. Normal: 15.15%, Fire: 84.85%

g. Normal: 9.94%, Fire: 90.06%

h. Normal: 28.89%, Fire: 71.11%

i. Normal: 0.37%, Fire: 99.63%

Fig. **11**: Evaluation of the robustness of the proposed fire detection algorithm against different attacks (noise, cropping, and rotation); images (b) and (e) are labeled as normal, and the remaining seven images are predicted as fire

IV. Conclusion and Future Work

The embedded processing capabilities of smart cameras have given rise to intelligent CCTV surveillance systems. Various abnormal events such as accidents, medical emergencies, and fires can be detected using these smart cameras. Of these, fire is the most dangerous abnormal event, as failing to control it at an early stage can result in huge disasters, leading to human, ecological and economic losses. Inspired by the great potential of CNNs, we propose a lightweight CNN based on the SqueezeNet architecture for fire detection in CCTV surveillance networks. Our approach can both localize fire and identify the object under surveillance. Furthermore, our proposed system balances the accuracy of fire detection and the size of the model using fine-tuning and the SqueezeNet architecture, respectively. We conduct experiments using two benchmark datasets and verify the feasibility of the proposed system for deployment in real CCTV networks. In view of the CNN model's reasonable accuracy for fire detection and localization, its size, and the rate of false alarms, the system can be helpful to disaster management teams in controlling fire disasters in a timely manner, thus avoiding huge losses.

This work mainly focuses on the detection of fire and its localization, with comparatively little emphasis on understanding the objects and scenes under observation. Future studies may focus on making challenging and specific scene understanding datasets for fire detection methods and detailed experiments. Furthermore, reasoning theories and information hiding algorithms [42-44] can be combined with fire detection systems to intelligently observe and authenticate the video stream and initiate appropriate action, in an autonomous way.

References

[1] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire detection based on vision sensor and support vector machines," Fire Safety Journal, vol. 44, pp. 322-329, 2009.

[2] I. Mehmood, M. Sajjad, and S. W. Baik, "Mobile-Cloud Assisted Video Summarization Framework for Efficient Management of Remote Sensing Data Generated by Wireless Capsule Sensors," Sensors, vol. 14, pp. 17112-17145, 2014.

[3] K. Muhammad, M. Sajjad, M. Y. Lee, and S. W. Baik, "Efficient visual attention driven framework for key frames extraction from hysteroscopy videos," Biomedical Signal Processing and Control, vol. 33, pp. 161-168, 2017.

[4] R. Hamza, K. Muhammad, Z. Lv, and F. Titouna, "Secure video summarization framework for personalized wireless capsule endoscopy," Pervasive and Mobile Computing, vol. 41, pp. 436-450, 2017/10/01/ 2017.

[5] P. Harris, R. Philip, S. Robinson, and L. Wang, "Monitoring Anthropogenic Ocean Sound from Shipping Using an Acoustic Sensor Network and a Compressive Sensing Approach," Sensors, vol. 16, p. 415, 2016.

[6] J. A. Khan Muhammad, Sung Wook Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," Neurocomputing, 2018.

[7] R. Chi, Z.-M. Lu, and Q.-G. Ji, "Real-time multi-feature based fire flame detection in video," IET Image Processing, 2016.

[8] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in Image Processing, 2004. ICIP'04. 2004 International Conference on, 2004, pp. 1707-1710.

[9] T. Toulouse, L. Rossi, M. Akhloufi, T. Celik, and X. Maldague, "Benchmarking of wildland fire colour segmentation algorithms," IET Image Processing, vol. 9, pp. 1064-1072, 2015.

[10] D. Guha-Sapir, F. Vos, R. Below, and S. Penserre, "Annual disaster statistical review 2015: the numbers and trends," http://www.cred.be/sites/default/files/ADSR_2015.pdf, 2015.

[11] T. Qiu, Y. Yan, and G. Lu, "An autoadaptive edge-detection algorithm for flame and fire image processing," IEEE Transactions on instrumentation and measurement, vol. 61, pp. 1486-1493, 2012.

[12] C.-B. Liu and N. Ahuja, "Vision based fire detection," in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, 2004, pp. 134-137.

[13] T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu, "Fire detection using statistical color model in video sequences," Journal of Visual Communication and Image Representation, vol. 18, pp. 176-185, 2007.

[14] B. C. Ko, S. J. Ham, and 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, vol. 21, pp. 1903-1912, 2011.

[15] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images," Fire safety journal, vol. 41, pp. 285-289, 2006.

[16] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," Pattern recognition letters, vol. 27, pp. 49-58, 2006.

[17] D. Han and B. Lee, "Development of early tunnel fire detection algorithm using the image processing," in International Symposium on Visual Computing, 2006, pp. 39-48.

[18] T. Celik and H. Demirel, "Fire detection in video sequences using a generic color model," Fire Safety Journal, vol. 44, pp. 147-158, 2009.

[19] P. V. K. Borges and E. Izquierdo, "A probabilistic approach for vision-based fire detection in videos," IEEE transactions on circuits and systems for video technology, vol. 20, pp. 721-731, 2010.

[20] A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, "Fire and smoke detection using wavelet analysis and disorder characteristics," in Computer Research and Development (ICCRD), 2011 3rd International Conference on, 2011, pp. 262-265.

[21] P. Foggia, A. Saggese, and 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, vol. 25, pp. 1545-1556, 2015.

[22] Y. H. Habiboğlu, O. Günay, and A. E. Çetin, "Covariance matrix-based fire and flame detection method in video," Machine Vision and Applications, vol. 23, pp. 1103-1113, 2012.

[23] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," IEEE Transactions on Image Processing, vol. 22, pp. 2786-2797, 2013.

[24] R. Di Lascio, A. Greco, A. Saggese, and M. Vento, "Improving fire detection reliability by a combination of videoanalytics," in International Conference Image Analysis and Recognition, 2014, pp. 477-484.

[25] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," IEEE transactions on circuits and systems for video technology, vol. 25, pp. 339-351, 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

[27] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, "ContextLocNet: Context-Aware Deep Network Models for Weakly Supervised Localization," in European Conference on Computer Vision, 2016, pp. 350-365.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," IEEE transactions on pattern analysis and machine intelligence, vol. 38, pp. 142-158, 2016.

[29] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, et al., "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," NeuroImage, vol. 108, pp. 214-224, 2015.

[30] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?," IEEE Transactions on Image Processing, vol. 24, pp. 5017-5032, 2015.

[31] Y. Li, L.-M. Po, C.-H. Cheung, X. Xu, L. Feng, F. Yuan, et al., "No-Reference video quality assessment with 3D shearlet transform and convolutional neural networks," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, pp. 1044-1057, 2016.

[32] J. Ahmad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "Saliency-weighted graphs for efficient visual content description and their applications in real-time image retrieval systems," Journal of Real-Time Image Processing, pp. 1-17.

[33] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 1MB model size," arXiv preprint arXiv:1602.07360, 2016.

[34] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675-678.

[35] D. Y. Chino, L. P. Avalhais, J. F. Rodrigues, and A. J. Traina, "BoWFire: detection of fire in still images by integrating pixel color and texture analysis," in 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, 2015, pp. 95-102.

[36] S. Verstockt, T. Beji, P. De Potter, S. Van Hoecke, B. Sette, B. Merci, et al., "Video driven fire spread forecasting (f) using multi-modal LWIR and visual flame and smoke data," Pattern Recognition Letters, vol. 34, pp. 62-69, 2013.

[37] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, pp. 1345-1359, 2010.

[38] K. Muhammad, J. Ahmad, M. Sajjad, and S. W. Baik, "Visual saliency models for summarization of diagnostic hysteroscopy videos in healthcare systems," SpringerPlus, vol. 5, p. 1495, 2016.

[39] S. Rudz, K. Chetehouna, A. Hafiane, H. Laurent, and O. Séro-Guillaume, "Investigation of a novel image segmentation method dedicated to forest fire applications," Measurement Science and Technology, vol. 24, p. 075403, 2013.

[40] L. Rossi, M. Akhloufi, and Y. Tison, "On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics," Fire Safety Journal, vol. 46, pp. 9-20, 2011.

[41] "http://elinux.org/RPi_Hardware," accesed on 12 December, 2016.

[42] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image," Multimedia Tools and Applications, vol. 75, pp. 14867-14893, 2016.

[43] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "Image steganography using uncorrelated color space and its application for security of visual contents in online social networks," Future Generation Computer Systems, p. http://dx.doi.org/10.1016/j.future.2016.11.029.

[44] K. Muhammad, M. Sajjad, and S. W. Baik, "Dual-Level Security based Cyclic18 Steganographic Method and its Application for Secure Transmission of Keyframes during Wireless Capsule Endoscopy," Journal of Medical Systems, vol. 40, p. 114, 2016.