Early Fire Detection using LSTM based Instance Segmentation and IoTs for Disaster Management

Sharaf J. Malebary *

Posted Date: 3 October 2023

doi: 10.20944/preprints202310.0065.v1

Keywords: Instance Segmentation; Key-frame Extraction; Fire detection; IoTs; Disaster Management

Article

Early Fire Detection Using LSTM based Instance Segmentation and IoTs for Disaster Management

Sharaf J. Malebary

Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, P.O. Box 344, Rabigh 21911, Saudia Arabia; smalebary@kau.edu.sa

Abstract: Fire outbreaks continue to cause damage despite the improvements in fire-detection tools and algorithms. It is still challenging to implement a well performing and optimized approach, which is sufficiently accurate, and has tractable complexity and low false alarming rate. Small amount of fire and identification of fire from a long distance is also a challenge in previously proposed techniques. In this study, we propose a novel hybrid model based on Convolutional Neural Networks (CNN) to detect and analyze fire intensity. 21 convolutional layers, 24 Rectified Linear Unit (ReLU) layers, 6 pooling layers, 3 fully connected layers, 2 dropout layers, and a softmax layer are included in the proposed 57-layer CNN model. Our proposed model performs instance segmentation in order to distinguish between fire and non-fire events. To reduce the intricacy of the proposed model, we also propose a key-frame extraction algorithm. The proposed model uses Internet of Things (IoT) devices to alert the relevant person by calculating the severity of fire. Our proposed model is tested on a publicly available dataset having fire and normal videos. The achievement of 95.25 % classification accuracy, 0.09% False Positive Rate (FPR), 0.65 percent False Negative Rate (FNR), and a prediction time of 0.08 seconds validates the proposed system.

Keywords: instance segmentation; key-frame extraction; fire detection; iots; disaster management

1. Introduction

Fire releases smoke, light, flames, heat and chemical gases as a result of combustion process [1]. Although fire has provided humans with prosperous living by providing the means for energy sources, heating and cooking, uncontrolled fire can endanger properties and human lives. National Fire Protection Association (NFPA) have reported 1.3 million fire cases in 2015, causing more than 3 thousand deaths and 15 thousand injuries [2]. Existing fire detection tools can be categorized into sensor-based, video-based and hybrid techniques utilizing video-based sensors [3]. Sensor-based techniques utilizes sensors to measure levels of carbon dioxide, carbon monoxide, temperature and smoke particles for efficiently detecting fire at early stages sensors [4]. The problem with these techniques is the cost and maintenance of these sensors. In contrast, video-based sensors utilize devices like cameras to capture the data and thermographic sensors to detect the fire pixel intensities [5]. The problem with these systems is their slow processing due to time taken during data collection, processing and triggering the alarm in severe conditions [6]. This issue can be solved by adopting an efficient technique, which not only solves the processing time by reducing the dimensionality of extracted data, but effectively processes the input data in lowest possible time. Thus, a need for reliable techniques to detect fire in early stages is essential to prevent the damage and loss of human lives.

Connecting billions of smart devices creates the Internet of Multimedia Things (IoMTs), while an increase in the number of installed sensors is leading to the emergence of the Tactile Internet (TI), which has various application in the areas of e-health [7], smart surveillance [8] and disaster management [9]. Smart surveillance includes disaster and security management, where edge intelligence has a significant role. For rapid actions in disastrous situation, it is crucial to report unusual circumstances instantly. Disaster management mainly depends on fire/smoke recognition, which can be achieved using edge computing. Fire can spread due to human errors or system failures, which is a big risk for human lives and properties. In 2015, overall damage of 3.1 billion USD is

noticed only caused by wildfire catastrophe whereas, in Europe 10,000 km^2 fertile area is affected from fire disasters yearly [10]. Color-based fire detection methods [11] have a major issue having high rate of incorrect alarms. To overcome this issue, a hybrid approach was introduced using color, shape and motion characteristics of fire [12].

Convolutional Neural Networks (CNNs) are widely used for fire identification problems [13]. CNNs have recently achieved efficient results on many other domains including agriculture [14], medical [15,16] and others [17–25]. A CNN based fire detection methods was proposed in [26] which was based on limited dataset and not compared with any of existing methods to prove their performance. Another CNN-based fire detection method utilizing VGG16 and Resnet50 models was proposed in [27], which was trained and tested on a very small dataset having 651 images and achieved accuracy of 93%. Another CNN-based fire detection technique [28] was proposed to implement smart surveillance, which was trained on two level datasets. The proposed model was huge in size (238Mb) and constrained to deploy on restricted hardware systems. In [29], an optimized tradeoff between accuracy and false rate is maintained along with keeping model size rational. Moreover, fire localization and detection network were proposed in [30] with minimized model scope, false alarm rate and high accuracy.

Early fire detection systems were proposed with machine learning to analyze sensor data and fire images to attain precise accuracy. In [31], a You Only Look Once (YOLO) based fire detection method was proposed, which was tested for flame recognition. The proposed model was learned on 196 images of fires and achieved an accuracy of 76%; however, training images were not sufficient to fully saturate the model. In [32], a smoke detection method using Deep Belief Network (DBN) was proposed, where model was trained on 482 images and achieved 95% accuracy. Using the optical flow method, a neural network of deep convolutional long recurrent networks was tested for real-time fire detection and combustion detection [33]. A dataset containing 10,000 images and 70 video frames was used to train and test the proposed method and achieved 93.3% accuracy. Results include false detection of lights and flames as fire, which could be controlled by using other sensors. In [34], A fuzzy algorithm was put forth for the purpose of detecting fires using input from several sensors. A hybrid approach for fire detection using fuzzy algorithm and CNN was proposed, which collected images from sensors and Closed-Circuit Television (CCTV) [35]. In this system, CCTV images were first preprocessed using CNN model to recognize fire, however these CNNs were unable to identify fire in blind spots where cameras can't be deployed. To remedy this issue, fuzzy logic computes the probability of fire presence by analyzing image and sensor's data. This method was named S-FDS and was more flexible, as it used static as well as rule-based algorithms.

Previously, many statistical techniques were used for data analytics. Each statistical algorithm can have unique characteristics based on its formula, the outcomes of its data analysis, and the algorithms to which it is related. Various Machine Learning (ML) techniques can be applied for data analysis too. L techniques can be categorized as either superficial learning or deep learning. Superficial learning algorithms concentrate on superficial data structures, such as SVM, Decision Trees, and K-means clustering. In contrast, deep learning algorithms deal with deep layered structures which include CNN, deep neural networks [36]. In real environments, deep learning models have shown to be more flexible and expressive, compared with shallow learning models. End-to-end recognition is difficult for DNN, because it has limited abstraction ability. Whereas CNN has high abstraction power and can analyze the image features to examine situation. In the beginning of fire, the flame is of small size and interval, in this situation it is difficult to capture image features from flame video data [37]. Fuzzy algorithms utilize membership functions to represent proximity to situations that cannot be clearly divided. The environment affects the range of the membership function of fuzzy algorithms, whereas general fuzzy algorithms disregard these variations. To overcome this limitation, an adaptive fuzzy algorithm is introduced which can update membership function. These adaptive fuzzy algorithms don't filter out exemption data cause of errors in sensor data, which affects result's accuracy.

A well performing and optimized approach is still a challenge, which should be optimally accurate, has less complexity and a low false alarming rate. Small amount of fire and identification of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

fire from a long distance is also a challenge for state-of-the-art methods. In this article, a hybrid model for classifying and detecting fire images in real-time environments is proposed. A 57-layer CNN architecture with 21 convolutional layers, 24 Rectified Linear Unit (ReLU) layers, 6 pooling layers, 3 fully connected layers, 2 dropout layers, and a softmax layer is proposed. Before training CNN model, Instance Segmentation (IS) is performed to efficiently segment the fire. To reduce the training and testing duration of the proposed model, an algorithm to extract key frames based on the correlation between consecutive frames is proposed. This article is structured as follows: section 2 describes the proposed classification and detection model and its application to real-world problems. Using publicly accessible datasets, Section 4 demonstrates the effectiveness of the proposed model. In the final section, conclusion and prospective work are presented.

2. Proposed Work

Early detection of fire becomes particularly challenging with factors like shadows, fire-like objects and changing lights. Traditional local features are inadequate to detect the fire due to the low accuracy and high false negative rate. Extracting local features for fire detection is also a time consuming and tedious task. These issues can be solved by extracting deep features using CNN models. After examining various pre-trained CNN models for target problems, a CNN model is proposed, which can classify and localize the fire at early stages. Figure 1 depicts the schematic representation of the proposed method.

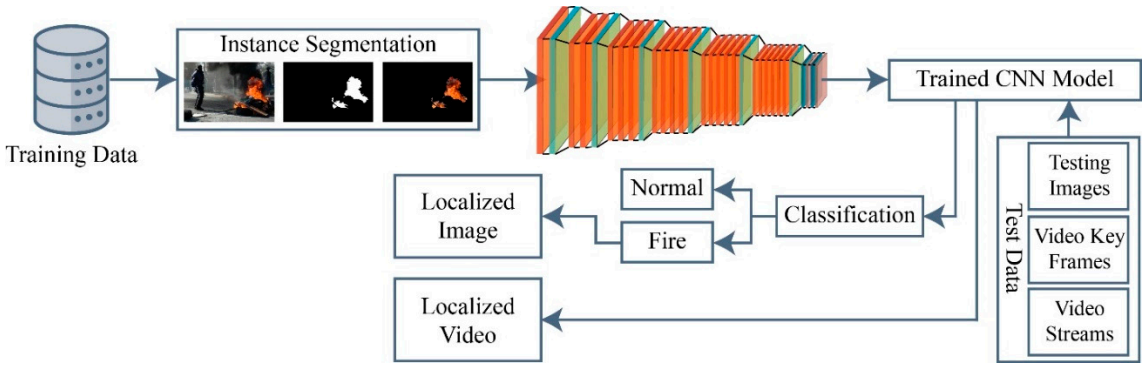


Figure 1. Schematic representation of the proposed method.

2.1. Instance Segmentation

Semantic segmentation [38] is one of the famous segmentation techniques, which deals with problems of known classes, where each pixel of image must belong to one predefined class and pixels are used to evaluate the predictions. But semantic segmentation cannot be applied to segment fire, as the instances of fire are unknown and have different shades and colors on different intensities. This problem is solved by employing instance segmentation, which is more challenging than other pixel-level techniques due to the nature of the solving problems, where classes are unknown. The evaluation of instance segmentation requires a loss function which is invariant to the assignment of pixels into different clusters. As instance segmentation is generally performed to count the objects in an image, it proves useful to count the instances of fire in an image. The approach proposed in [39] is inspired by the counting process followed by humans. Humans count the objects by keeping track of accounted locations in an accurate spatial memory. Recurrent Convolutional Neural Networks (RCNNs) were used to segment the objects while saving the current state in spatial memory. However, for the purpose of fire segmentation, the RCNNs did not perform well, as the fire instances are, sometimes too small. To overcome this issue, the RCNN is replaced by Mask-RCNN, which provides improved results. The overall structure of instance segmentation utilized in this work is illustrated in Figure 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

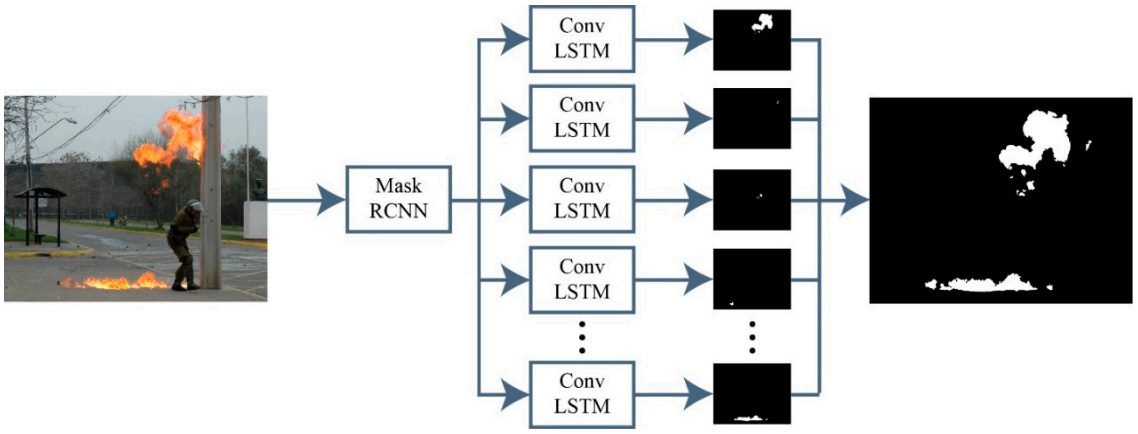


Figure 2. Structure of instance segmentation.

2.2. Deep CNN Architecture

A novel CNN model is proposed in this article, as the existing pre-trained models are trained on a large dataset ImageNet [40] containing 1000 classes. The weights and activations of pre-trained networks are adjusted according to the images in ImageNet dataset. These pre-trained models are structured in such a way that a single model can be utilized to classify multiple problems. It makes these networks too complex for classifying the simpler problems containing fewer classes. The parameters of our proposed network are updated by training it on fire and non-fire images only, which makes it more problem oriented. The proposed network contains 57 layers, including 21 convolutional, 24 ReLU, 6 pooling, 3 fully connected, 2 dropouts, and a softmax layer. The network accepts an input of size $200 \times 200 \times 3$ and softmax layer provides 1000 features. The overall structure of the proposed model is shown in Figure 3. As the input images are already segmented, the activations on each layer remain consistent and reduce gradually. The purpose of this arrangement is to learn all the possible features of fire along with different shades and intensities. The segmented images proved vital to train a network, capable of training a strong classifier and detector at the same time.

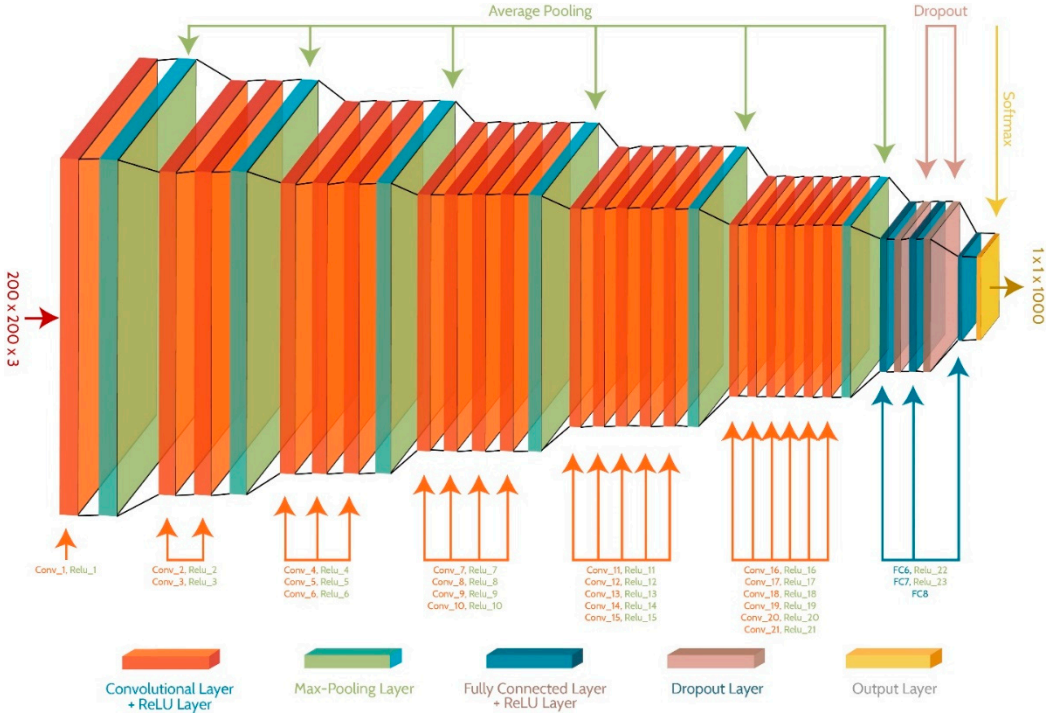


Figure 3. Architecture of proposed CNN model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

The structure of CNN model is divided into 6 blocks, where each block increases the number of convolutional and ReLU layers by 1 and ends on an average pooling layer. The input is forwarded to block 1, where only combination of convolutional and ReLU layer applies 96 filters of size 11×11 for generating 512 feature maps. Average pooling with a stride of 2 pixels is employed to shrink the size of feature map and retaining the useful attributes by discarding the less important features. In second block, 2 combinations of convolutional and ReLU layers apply 128 and 384 filters of size 5×5 and 3×3 respectively and generate 256 feature maps. The average pooling of this block reduces the feature maps to 128. Blocks 3 to 6 contains 3,4,5 and 6 combinations of convolutional and ReLU layers respectively and apply different number of filters to further convolved the input image. The average pooling of block 6 provides a descriptor map of size 64, which is forwarded to fully connected layers, where FC6 and FC7 layers extract 5000 features, while FC8 extracts 1000 features. The softmax layer provides the out of 1000 features. Detailed overview of layers along with adjusted parameters are catalogued in Table 1.

Table 1. Detailed overview of layers along with adjusted parameters.

Combinations	Filters	Total Filters	Stride Size	Weight Size	Bias Vector	Activations
Input Layer	–	–	–	–	–	200 × 200 × 3
Convolutional + ReLU	11 × 11	96	[4 × 4]	11 × 11 × 3 × 96	1 × 1 × 96	512 × 512 × 96
Max Pooling	3 × 3	–	[2 × 2]	–	–	256 × 256 × 48
Convolutional + ReLU	5 × 5	128	[1 × 1]	5 × 5 × 48 × 128	1 × 1 × 128	512 × 512 × 128
Convolutional + ReLU	3 × 3	384	[1 × 1]	3 × 3 × 128 × 384	1 × 1 × 384	512 × 512 × 384
Max Pooling	3 × 3	–	[2 × 2]	–	–	256 × 256 × 192
Convolutional + ReLU	3 × 3	192	[1 × 1]	3 × 3 × 192 × 192	1 × 1 × 192	512 × 512 × 192
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 192 × 128	1 × 1 × 128	512 × 512 × 128
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 128 × 128	1 × 1 × 128	512 × 512 × 128
Max Pooling	3 × 3	–	[2 × 2]	–	–	256 × 256 × 64
Convolutional + ReLU	3 × 3	64	[1 × 1]	3 × 3 × 64 × 64	1 × 1 × 64	256 × 256 × 64
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 64 × 128	1 × 1 × 128	256 × 256 × 128
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 128 × 128	1 × 1 × 128	256 × 256 × 128
Convolutional + ReLU	3 × 3	256	[1 × 1]	3 × 3 × 128 × 256	1 × 1 × 256	256 × 256 × 256
Max Pooling	3 × 3	–	[2 × 2]	–	–	128 × 128 × 128
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 128 × 128	1 × 1 × 128	128 × 128 × 128
Convolutional + ReLU	3 × 3	64	[1 × 1]	3 × 3 × 128 × 64	1 × 1 × 64	128 × 128 × 64
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 64 × 128	1 × 1 × 128	128 × 128 × 128
Convolutional + ReLU	3 × 3	256	[1 × 1]	3 × 3 × 128 × 256	1 × 1 × 256	128 × 128 × 256
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 256 × 128	1 × 1 × 128	128 × 128 × 128
Max Pooling	3 × 3	–	[2 × 2]	-	-	64 × 64 × 64
Convolutional + ReLU	3 × 3	512	[1 × 1]	3 × 3 × 64 × 512	1 × 1 × 512	64 × 64 × 512
Convolutional + ReLU	3 × 3	256	[1 × 1]	3 × 3 × 512 × 256	1 × 1 × 256	64 × 64 × 256
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 256 × 128	1 × 1 × 128	64 × 64 × 128
Convolutional + ReLU	3 × 3	128	[1 × 1]	3 × 3 × 128 × 128	1 × 1 × 128	64 × 64 × 128
Convolutional + ReLU	3 × 3	96	[1 × 1]	3 × 3 × 64 × 96	1 × 1 × 96	64 × 64 × 96
Convolutional + ReLU	3 × 3	192	[1 × 1]	3 × 3 × 32 × 192	1 × 1 × 192	64 × 64 × 192
Max Pooling	3 × 3	–	[2 × 2]	–	–	32 × 32 × 96
FC6 + ReLU + Dropout	–	–	–	4096 × 25088	4096 × 1	1 × 1 × 4096
FC7 + ReLU + Dropout	–	–	–	4096 × 4096	4096 × 1	1 × 1 × 4096
FC8	–	–	–	1000 × 4096	1000 × 1	1 × 1 × 1000
Softmax	–	–	–	–	–	1 × 1 × 1000

2.3. Key Frames Extraction

The amount of video data collected from surveillance increases every day. Fire events occur rarely and if fire needs to be detected on a particular day or hour, it is still a tedious task to process and verify each frame from the video. If frames are extracted from one-hour video at 30fps, there will be 108,000 frames and checking all these frames will take some serious amount of time. The execution and processing time reduces dramatically by only extracting key frames from a video. In this article, a method is utilized to extract only key frames by ignoring the duplicate frames. This is achieved by calculating the correlation between two consecutive frames. If the correlation (C) is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

greater than or equal to the threshold value (T), the relationship is considered significant, and the images are similar, else the frames are considered as key frames. The overall flow of extracting key frames is explained in Algorithm 1. Flow diagram is also shown in Figure 4.

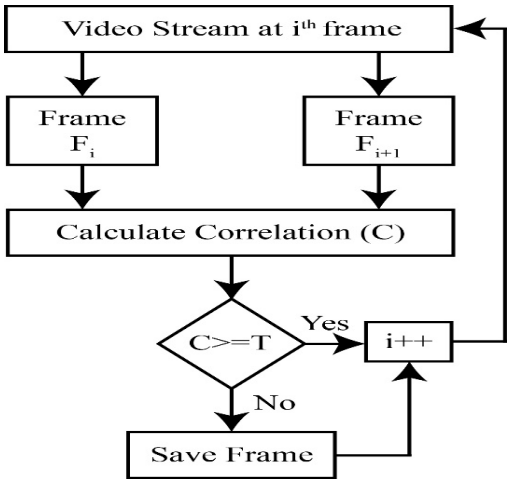


Figure 4. Flow diagram to extract key frames.

Algorithm 1 Extracting Key Frames from Video

Input: A video stream
Output: Key frames
1. $F_i \leftarrow$ All video frames
2. ***while***$(i < length(F_i))$
3. $C = correlation(F_i, F_{i+1})$
4. ***if***$(C \geq T)$ then
   $i++;$
   else
   $SaveFrame(F_i);$
   $i++;$

End

2.4. Fire Classification and Localization

The proposed CNN architecture is designed to automatically learn robust features from raw fire data in both indoor and outdoor environments. Segmented fire images are provided as training data to label the test data as fire or normal images. This decision is based on the probability score of the CNN model. Once the fire and normal images are classified, the next step is to localize the fire within an image. Algorithms 2 describes the fire classification and localization process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

Algorithm 2 Classification and Localization of Fire

Input: Trained classifier (*Classifier*), test data (*TD*), output type (*OT*) and trained CNN model (*Net*)
Output: Localized fire images or video
1. Analyze the input data (*ID*), either images (*I*) or video streams (*VS*)
2. Analyze the *OT*, either localized image (*LI*) or localized video (*LV*)
3. ***if*** $(ID == I)$
    Extract test features of ID and predict label using Net
else if $(ID == VS)$
    if $(OT == LI)$
        Extract Key Frames
        Repeat step 3
    else if $(OT == LV)$
        Resize video as per the Network Size
        Localize the Video using Net
4. Check the predicted Label
  if $(PredictedLable == Normal)$
    No action required
  else if $(PredictedLable == Fire)$
    Extract the features (FV) using FC7 layer of CNN model.
    Apply binarization using Threshold (T) as:
    $Image_{Binary} = \begin{cases} 1, & FV < T \\ 0, & Otherwise \end{cases}$
5. Localize the fire of input image using $Image_{Binary}$

By sending test data to a trained classifier, which can be an image or a video stream, fire can be localized. Features are taken from the image and its label are predicted if it is an image, and from the video frames if it is a video stream. Following the creation of a binary image utilizing the defined threshold and the predicted fire picture, the localization of the fire instances inside the image or video frame is subsequently accomplished.

2.5. Fire Analysis

At this point, the input images or videos containing fire are localized. Next step is to analyze the fire intensity and severity as many post-fire assessments are based on this information. The intensity of fire mainly depends upon the distance between the camera and burning object. This distance is calculated by performing pre-processing steps like identifying all objects in an image, measuring the distance between camera and burning object, and measuring the area of burning object. Objects are identified by training the proposed CNN model on sub-part of a famous object dataset Caltech101 [41]. The selected part of the dataset contains 23 classes which can catch fire. The dimensions of these classes are preset to a default width and height. The other step of this analysis is to predict the severity of fire for taking the post-fire actions. Categorizing the fire level can determine either to contact the house owner or fire brigade. These fire levels are regarded as low, moderate and high severity. Algorithm 3 is used to determine the intensity of fire and take necessary post-fire steps.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

Algorithm 3 Determining Intensity and Severity of Fire

Input: Labelled Image

Output: Alert concerning person/department

1. $Net \leftarrow$ Trained Proposed CNN model on 23 classes
2. $I_i \leftarrow$ Input Image
3. $O_i \leftarrow$ Extracted objects from I_i using Instance Segmentation
4. $O_f \leftarrow DetectObjectOnFire(O_i)$
5. $Labeled_O \leftarrow IdentifyObjects\ (Net, O_f)$
6. $Size_{Actual}(w,h) \leftarrow FetchPresetSize(O_f)$
7. $Size_{Pixel}(w,h) \leftarrow CalculateLocalizedSize(O_f)$
8. $Size_{Predicted}(w) = \frac{Size_{Pixel}(w)}{Size_{Actual}(w)},\ Size_{Predicted}(h) = \frac{Size_{Pixel}(h)}{Size_{Actual}(h)}$
9. $Dif(w) = \frac{Size_{Actual}(w)}{Size_{Predicted}(w)},\ Difference(h) = \frac{Size_{Actual}(h)}{Size_{Predicted}(h)}$
10. $\boldsymbol{if}(Dif > 1)$ then Object is Dif times bigger and each Dif pixels will be equal to 1 pixel

$\boldsymbol{else\ if}(Dif \leq 1)$ then Object is either equal or Dif times smaller and each 1 pixel will be equal to Dif pixels in case of smaller object

11. $Fire_{Pixels}(w) \leftarrow CountFirePixels\big(Size_{Pixel}(w)\big)$,
$Fire_{Pixels}(h) \leftarrow CountFirePixels(Size_{Pixel}(h))$
12. $Processed_{FirePixels}(w) \leftarrow ProcessPixels(Fire_{Pixels}\ (w), Dif(w))$
$Processed_{FirePixels}(h) \leftarrow ProcessPixels(Fire_{Pixels}\ (h), Dif(h))$
13. $Fire_{Effected}(w) = \frac{Size_{Actual}(w)}{Processed_{FirePixels}(w)}$,

$$Fire_{Effected}(h) = \frac{Size_{Actual}(h)}{Processed_{FirePixels}(h)}$$

14. $Effected = mean(Fire_{Effected}(w), Fire_{Effected}(h))$
15. $\boldsymbol{if}(Effected \geq 60)$ then label fire as **High Severity**.

$\boldsymbol{else\ if}(Effected \geq 15\ and\ Effected < 60)$ then label fire as **Medium Severity**.

$\boldsymbol{else\ if}(Effected < 15)$ then label fire as **Low Severity**.

The magnitude of the fire instance is used as the basis for fire analysis. Instance segmentation is first carried out to detect fire items, after which the difference between the real and anticipated objects is determined. Fire pixels are calculated to accurately anticipate the fire severity after this difference has been calculated.

3. Experimental Results and Discussion:

This section describes the investigations that are conducted to validate the proposed method. The information is described, including the experimental setup and the selected dataset. These

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

dataset's results are presented, followed by a comparison with extant techniques for fire detection and localization. Finally, a comprehensive discussion verifies the approach's robustness and efficacy.

3.1. Experimental Setup

The proposed CNN model is trained using MATLAB 2022a on an NVIDIA GeForce GTX 1080 with an overall computation capability of 6.1, a clock rate of 1,607-1,733 MHz, and 7 multiprocessors. Stochastic Gradient Descent with Momentum (SGDM) is the algorithm that represents the 64-minibatch training technique. The initial learning rate is fixed at 0.01 and decreased by a factor of 5 every 5 generations. Momentum is set at 0.7, and the utmost number of epochs is set to 150. Cross-Entropy [41] is used as a suitable loss function because it has proven to be reasonable for many multiclass problems. The data is divided according to the standard proportions of 70-15-15 for training, testing, and validation, respectively.

3.2. Experimental Results

The publicly available dataset contains 32 videos including 22 fire videos and 10 normal videos. The videos have 24fps rate, which makes a total of 64,049 frames of fires and 25,511 frames of normal images and a grand total of 89,560 frames. The complexity, size and background colors make this dataset challenging. The normal images contain fire-like objects, which makes the detection and classification even harder. Figure 5 illustrates a few test images, one frame each from all videos, while Table 2 presents basic description for this dataset.

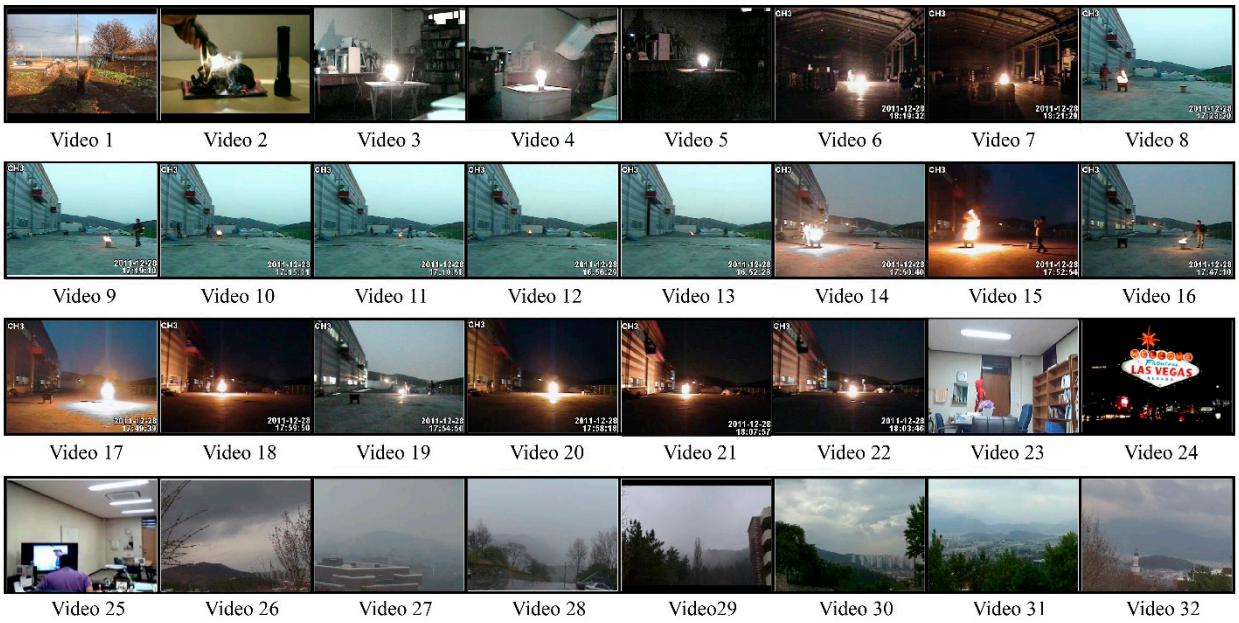


Figure 5. Sample videos from dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

Table 2. Basic description of dataset.

Video Name	Original File Name	Resolution	Frames	Modality	Total Frames
Video 1	Flame1		402	Fire	64,049
Video 2	Flame2		411	Fire	
Video 3	Flame3		613	Fire	
Video 4	Flame4		373	Fire	
Video 5	Flame5		748	Fire	
Video 6	indoor_night_20m_heptane_CCD_001		1,658	Fire	
Video 7	indoor_night_20m_heptane_CCD_002		3,846	Fire	
Video 8	outdoor_daytime_10m_gasoline_CCD_001		3,491	Fire	
Video 9	outdoor_daytime_10m_heptane_CCD_001		4,548	Fire	
Video 10	outdoor_daytime_20m_gasoline_CCD_001		3,924	Fire	
Video 11	outdoor_daytime_20m_heptane_CCD_001		4,430	Fire	
Video 12	outdoor_daytime_30m_gasoline_CCD_001		6,981	Fire	
Video 13	outdoor_daytime_30m_heptane_CCD_001		3,754	Fire	
Video 14	outdoor_night_10m_gasoline_CCD_001		1,208	Fire	
Video 15	outdoor_night_10m_gasoline_CCD_002		1,298	Fire	
Video 16	outdoor_night_10m_heptane_CCD_001		3,275	Fire	
Video 17	outdoor_night_10m_heptane_CCD_002		776	Fire	
Video 18	outdoor_night_20m_gasoline_CCD_001		5,055	Fire	
Video 19	outdoor_night_20m_heptane_CCD_001		4,141	Fire	
Video 20	outdoor_night_20m_heptane_CCD_002		1,645	Fire	
Video 21	outdoor_night_30m_gasoline_CCD_001		6,977	Fire	
Video 22	outdoor_night_30m_heptane_CCD_001		4,495	Fire	
Video 23	smoke_or_flame_like_object_1		171	Normal	25,511
Video 24	smoke_or_flame_like_object_2		530	Normal	
Video 25	smoke_or_flame_like_object_3		862	Normal	
Video 26	smoke_or_flame_like_object_4		904	Normal	
Video 27	smoke_or_flame_like_object_5		8,229	Normal	
Video 28	smoke_or_flame_like_object_6		7,317	Normal	
Video 29	smoke_or_flame_like_object_7		2,012	Normal	
Video 30	smoke_or_flame_like_object_8		8,49	Normal	
Video 31	smoke_or_flame_like_object_9		2,807	Normal	
Video 32	smoke_or_flame_like_object_10		1,830	Normal	
Total Frames					89,560

In the proposed system, the initial instance segmentation proves vital as it helps the model to learn only fire features. The parameters of Mask-RCNN are learned using backpropagation. To prevent the effect of exploding gradient, gradients are clipped to make sure that each of their elements remains under the absolute value of 3. Adam optimization algorithm [42] is applied to train the network by using initial learning rate of 10^{-4} and reducing it by 0.1 of each error. As there was no overfitting during preliminary experiments, neither l2 regularization nor dropout was utilized throughout the segmentation process. Mini-batch size was set to 8 images per batch and initial weights of Mask-RCNN were randomly initialized within the range of [-0.04 - 0.04]. The results of instance segmentation on some sample images from a smaller dataset are presented in Figure 6, while the results of proposed system are illustrated in Figure 7.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

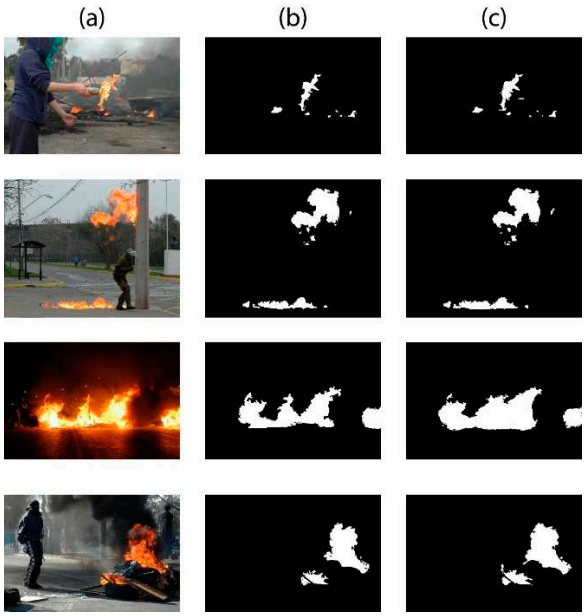
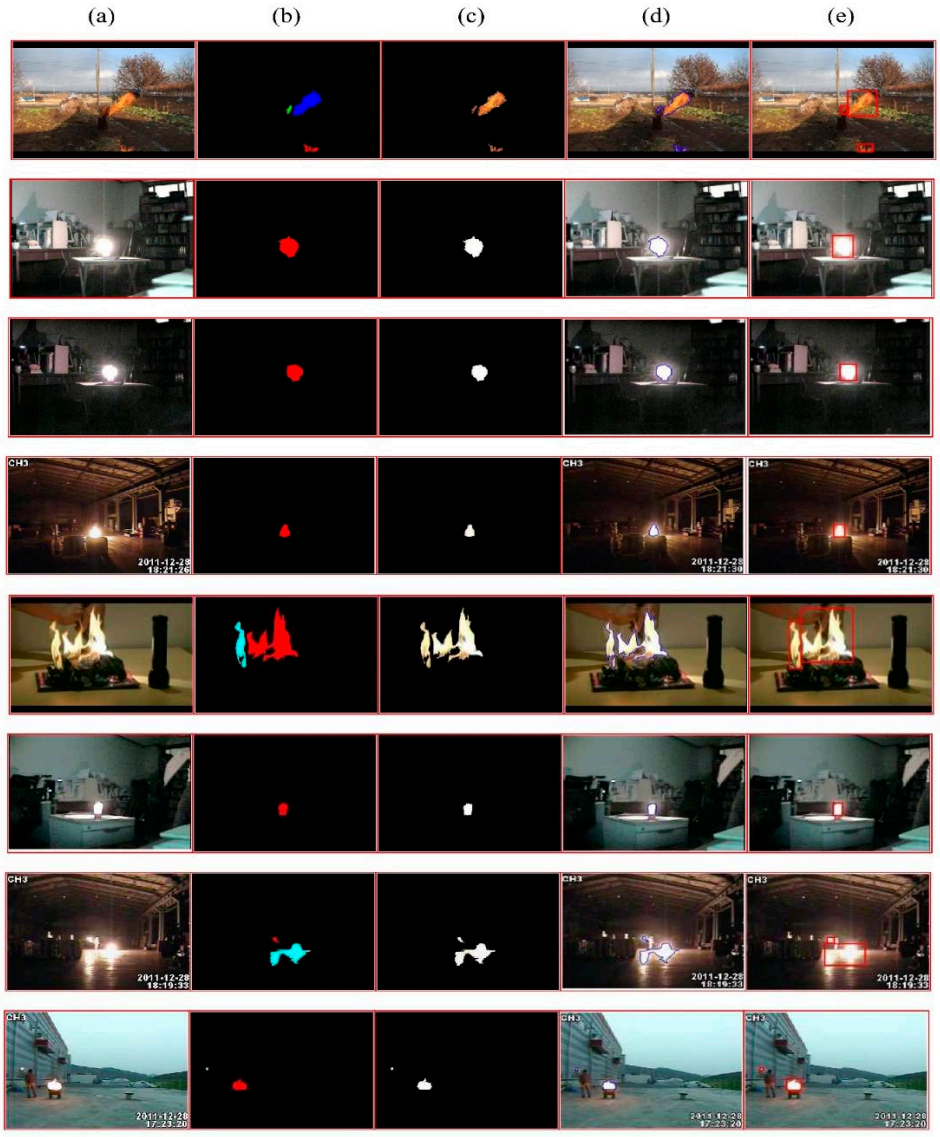


Figure 6. Results of instance segmentation. (a) Input image. (b) Ground-truth image. (c) Segmented image using instance segmentation.



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

Figure 7. Results of proposed model. (a) Original image. (b) Binary image extracted using instance segmentation. (c) Segmented image. (d) Boundary image, localized by detector. (e) Final predicted image.

The proposed CNN model performs well on this dataset by maintaining a low false positive rate and high accuracy. The training time and prediction time is also noteworthy. Different experiments are performed including utilizing pre-trained models like AlexNet [43], InceptionV3 [44] and SuqeezeNet [45] before and after the fine-tuning. All these networks are also serially used to note the impact. The proposed network is also experimented with before and after fine-tuning as well as before and after adding the instance segmentation module. The outcomes of all these experiments are shown in Table 3. It can be clearly seen that the pre-trained models, fused models and model without instance segmentation could not outperform the proposed model.

Table 3. Classification results of different experiments.

	Model	Fine Tuning		Accuracy (%)	FPR (%)	FNR (%)	Training Time (s)	Prediction Time (s)
		No	Yes					
CNN Pre-Trained Models	AlexNet	✓		78.31	41.18	14.29	78.9	1.19
			✓	86.04	13.58	7.14	114.3	1.63
	InceptionV3	✓		83.87	29.33	10.65	69.8	0.83
			✓	87.56	7.22	2.13	93.4	0.94
	SqueezeNet	✓		74.39	14.67	7.80	63.5	0.98
			✓	84.77	9.41	5.50	87.4	1.23
	Fused	✓		89.47	11.76	9.74	397.2	0.78
			✓	90.35	5.88	1.50	247.9	0.63
Proposed	Without IS	✓		91.62	3.38	2.94	**54.7**	0.32
			✓	93.84	1.82	1.43	73.5	0.18
	With IS	✓		92.40	0.65	0.84	84.3	0.12
			✓	**95.25**	**0.09**	**0.65**	100.8	**0.08**

This is notable that the training time increases when instance segmentation is applied on the proposed approach, but the FPR and FNR rates are decreased to the minimum with the lowest prediction time of 0.08 seconds. The maximum accuracy is also noted at 95.25%, which is better than existing state-of-the-art techniques.

3.3. Robustness of Proposed Model:

The success of a fire detection system lies in its robustness against well-known attacks in uncertain environments. This section investigates the robustness of the proposed system by employing different attacks like fire-blockage and noise. Figure 8 shows that the proposed system performs well in most cases under uncertain environments and weather conditions. It can be clearly seen that the proposed system achieved efficient results on certain attacks. The fire analysis is carried out by testing images from the real-world and it achieved effective results as well. Figure 9 and Figure 10 show that the algorithm provides necessary information regarding the fire intensity and object on fire.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

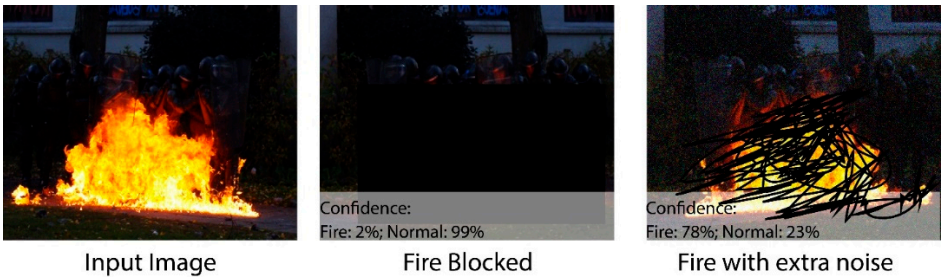


Figure 8. Robustness of proposed model on different noisy conditions.

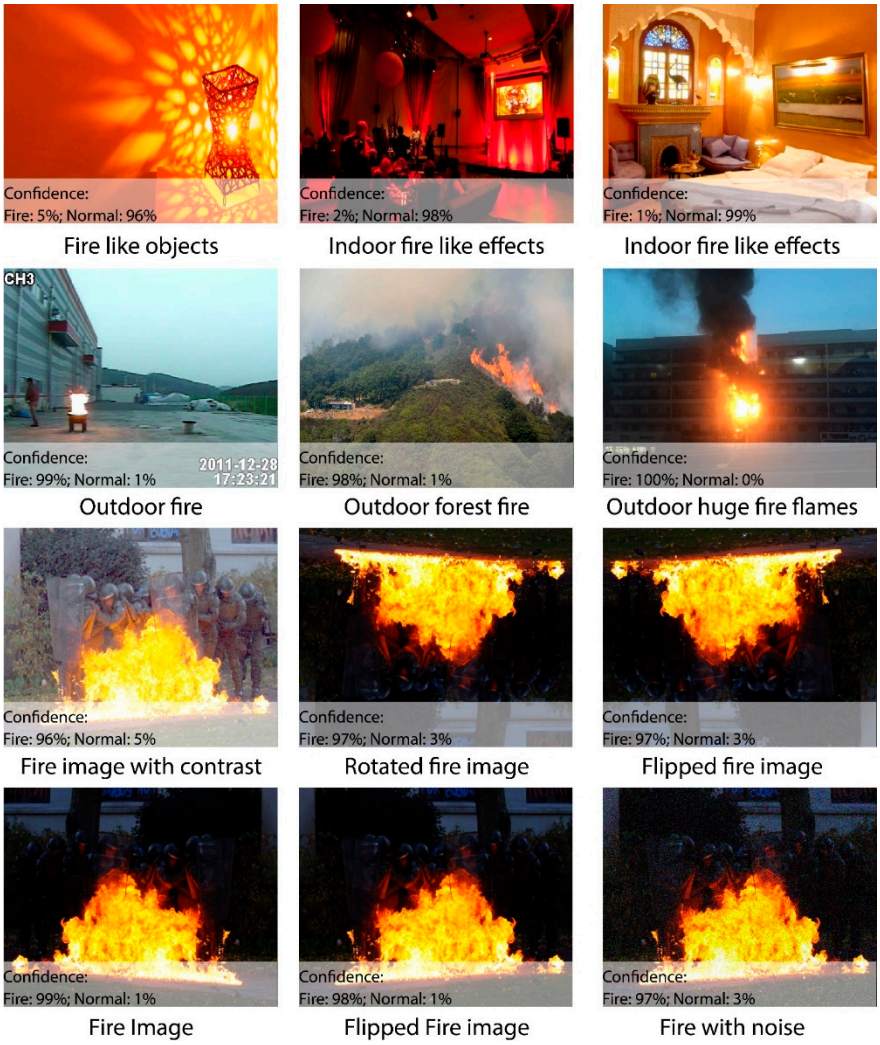


Figure 9. Output of proposed model on fire as well as fire like objects (first two rows). Output of proposed model on different kind of noises (last two rows).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2023 doi:10.20944/preprints202310.0065.v1

Figure 10. Output of proposed model on different fire and non-fire scenarios.

3.4. Discussion:

A system to detect fire on early stages was proposed utilizing CNNs and IoMTs for disaster management where a fine-tuned AlexNet model is used to detect fire with an accuracy of 94.39% and false positive rate of 9.07% [46]. Many techniques are proposed, which utilizes the color shape and motion features and achieved an overall accuracy between 87%-90% [47–49]. But these techniques proved vulnerable when fire-like objects are identified within the scene. In another technique, the moving objects were initially detected to deal with the environmental changes throughout the timespan. These objects were then preprocessed by subtracting the background to extract the fire instances. The instances were evaluated based on color, shape and difference between two consecutive frames in a video. Achieved accuracy of this technique was 95.55% with a false positive rate of 11.76% [50]. A transfer learning technique was implemented utilizing a pre-trained network AlexNet to detect the fire at early stages. The proposed model was later fine-tuned using a SqueezeNet network, which reduced the size and feasibility of the approach to achieve an accuracy of 94.50% and false positive rate of 8.87% [30]. Table 4 shows experimental results along with comparison to the previous techniques.

Table 4. Experimental results along with comparison to the previous techniques.

Technique	FPR (%)	FNR (%)	Accuracy (%)
Rafiee [47]	17.65	07.14	87.10
Habiboğlu [48]	5.88	14.29	90.32
Chen [49]	11.76	14.29	87.10
Bellavista [46]	9.07	02.13	94.39
Foggia [50]	11.76	-	93.55
Muhammad [30]	8.87	02.12	94.50
Proposed	0.09	00.65	95.25

In this work, a hybrid model is proposed, as shown in Figure 1, utilizing instance segmentation along CNN architecture as shown in Figure 2. The parameters of CNN are provided in Table 1, while the structure of CNN model is shown in Figure 3. As the proposed model is trained and tested on video datasets, an algorithm is proposed, as shown in Figure 4 and explained in Algorithm 1, to extract key frames by calculating the correlation between consecutive pixels. After the extraction of key frames, the model is trained to classify and localize the fire in an image. Initially, the CNN model is trained on dataset to classify images, while the detector is trained on a subpart of a well-known dataset Caltech-101. Detector provides information regarding the object on fire while the fire is analyzed as per the proposed algorithm. The overall procedure of classification, localization and fire analysis is explained in Algorithm 2 and Algorithm 3. Description of utilized dataset is explained in Table 2, while Figure 5 shows some samples frames from each of the dataset videos. Results of

instance segmentation are illustrated in Figure 6, while detection and localization results are shown in Figure 7. Table 4 shows classification results of different experiments. The robustness of the proposed model is checked against several attacks like injecting noise, blocking fire, rotation and flipping operations. Achieved results are shown in Figure 8, Figure 9 and Figure 10. The model achieved improved results than previously proposed state-of-the-art methods and compared in Table 4.

4. Conclusion

In this article, an automated system combining the properties of IS and CNN architecture is proposed to classify and detect fire in real-time environment. The CNN architecture is 57-layer deep, containing 21 convolutional layers, 24 ReLU layers, 6 pooling layers, 3 fully connected layers, 2 dropout layers and a softmax layer. Training in CNN architecture is optimized by employing IS, which efficiently extracts the fire from images and video frames. To minimize the training and testing time of proposed model, an algorithm is proposed to extract key frames based on correlation between consecutive frames. The robustness of the proposed model is verified by testing it on real-time data, where the model achieved improved results than state-of-the-art methods. As for future work, the CNN model with more depth can be utilized and dimensionality can be reduced by implementing feature optimizing techniques. Key-frames can also be extracted by employing methods like Genetic Algorithm (GA) to improve the output of any model.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. A. Gaur et al., "Fire sensing technologies: A review," IEEE Sensors Journal, vol. 19, no. 9, pp. 3191-3202, 2019.
2. M. Ahrens, "Trends and patterns of US fire loss," National Fire Protection Association (NFPA) report Google Scholar, 2017.
3. J. Fonollosa, A. Solórzano, and S. Marco, "Chemical sensor systems and associated algorithms for fire detection: A review," Sensors, vol. 18, no. 2, p. 553, 2018.
4. J. Li et al., "Long-range raman distributed fiber temperature sensor with early warning model for fire detection and prevention," IEEE sensors journal, vol. 19, no. 10, pp. 3711-3717, 2019.
5. P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," Case Studies in Thermal Engineering, vol. 19, p. 100625, 2020.
6. H. Wang, X. Fang, Y. Li, Z. Zheng, and J. Shen, "Research and application of the underground fire detection technology based on multi-dimensional data fusion," Tunnelling and Underground Space Technology, vol. 109, p. 103753, 2021.
7. N. Pathak, S. Misra, A. Mukherjee, and N. Kumar, "HeDI: Healthcare Device Interoperability for IoT-Based e-Health Platforms," IEEE Internet of Things Journal, 2021.
8. M. Kumar, K. S. Raju, D. Kumar, N. Goyal, S. Verma, and A. Singh, "An efficient framework using visual recognition for IoT based smart city surveillance," Multimedia Tools and Applications, pp. 1-19, 2021.
9. J. Dugdale, M. T. Moghaddam, and H. Muccini, "IoT4Emergency: Internet of Things for Emergency Management," ACM SIGSOFT Software Engineering Notes, vol. 46, no. 1, pp. 33-36, 2021.
10. D. Guha-Sapir and P. Hoyois, "Estimating populations affected by disasters: A review of methodological issues and research gaps," Brussels: Centre for Research on the Epidemiology of Disasters (CRED), Institute of Health and Society (IRSS), University Catholique de Louvain, 2015.
11. A. Khalil, S. U. Rahman, F. Alam, I. Ahmad, and I. Khalil, "Fire Detection Using Multi Color Space and Background Modeling," Fire Technology, pp. 1-19, 2020.
12. Y. Xie et al., "Efficient Video Fire Detection Exploiting Motion-Flicker-Based Dynamic Features and Deep Static Features," IEEE Access, vol. 8, pp. 81904-81917, 2020.
13. Y. Luo, L. Zhao, P. Liu, and D. Huang, "Fire smoke detection algorithm based on motion characteristic and convolutional neural networks," Multimedia Tools and Applications, vol. 77, no. 12, pp. 15075-15092, 2018.
14. O. Khudayberdiev and M. H. F. Butt, "Fire detection in Surveillance Videos using a combination with PCA and CNN," Academic Journal of Computing & Information Science, vol. 3, no. 3, 2020.
15. M. Rashid, J. H. Shah, M. Sharif, M. Y. Awan, and M. H. Alkinani, "An optimized approach for breast cancer classification for histopathological images based on hybrid feature set," Current Medical Imaging, vol. 17, no. 1, pp. 136-147, 2021.

16. M.-A. Khan et al., "A Blockchain Based Framework for Stomach Abnormalities Recognition," Computers, Materials \& Continua, vol. 67, no. 1, pp. 141--158, 2021. [Online]. Available: http://www.techscience.com/cmc/v67n1/41161.
17. M. Attique Khan, M. Alhaisoni, T. Saba, A. Rehman, and T. Iqbal, "A hybrid deep learning architecture for the classification of superhero fashion products: An application for medical-tech classification," Computer Modeling in Engineering & Sciences, vol. 124, no. 3, pp. 1017-1033, 2020.
18. M. A. Khan, A. Armghan, and M. Y. Javed, "SCNN: A Secure Convolutional Neural Network using Blockchain," in 2020 2nd International Conference on Computer and Information Sciences (ICCIS), 2020: IEEE, pp. 1-5.
19. I. M. Nasir et al., "Pearson correlation-based feature selection for document classification using balanced training," Sensors, vol. 20, no. 23, p. 6793, 2020.
20. I. M. Nasir et al., "Deep Learning-Based Classification of Fruit Diseases: An Application for Precision Agriculture," CMC-COMPUTERS MATERIALS & CONTINUA, vol. 66, no. 2, pp. 1949-1962, 2021.
21. M. Raza, J. H. Shah, M. A. Khan, and A. Rehman, "Human action recognition using machine learning in uncontrolled environment," in 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), 2021: IEEE, pp. 182-187.
22. M. Raza, J. H. Shah, S.-H. Wang, U. Tariq, and M. A. Khan, "HAREDNet: A deep learning based architecture for autonomous video surveillance by recognizing human actions," Computers and Electrical Engineering, vol. 99, p. 107805, 2022.
23. J. Tariq et al., "Fast intra mode selection in HEVC using statistical model," Computers, Materials and Continua, vol. 70, no. 2, pp. 3903-3918, 2022.
24. I. Mushtaq, M. Umer, M. Imran, G. Muhammad, and M. Shorfuzzaman, "Customer prioritization for medical supply chain during COVID-19 pandemic," Computers, Materials and Continua, pp. 59-72, 2021.
25. M. Raza, S. M. Ulyah, J. H. Shah, N. L. Fitriyani, and M. Syafrudin, "ENGA: Elastic Net-Based Genetic Algorithm for human action recognition," Expert Systems with Applications, vol. 227, p. 120311, 2023.
26. S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, 2016: IEEE, pp. 877-882.
27. J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in International conference on engineering applications of neural networks, 2017: Springer, pp. 183-193.
28. K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," Neurocomputing, vol. 288, pp. 30-42, 2018.
29. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," IEEE Access, vol. 6, pp. 18174-18183, 2018.
30. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 7, pp. 1419-1434, 2018.
31. D. Shen, X. Chen, M. Nguyen, and W. Q. Yan, "Flame detection using deep learning," in 2018 4th International conference on control, automation and robotics (ICCAR), 2018: IEEE, pp. 416-420.
32. R. Kaabi, M. Sayadi, M. Bouchouicha, F. Fnaiech, E. Moreau, and J. M. Ginoux, "Early smoke detection of forest wildfire video using deep belief network," in 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2018: IEEE, pp. 1-6.
33. C. Hu, P. Tang, W. Jin, Z. He, and W. Li, "Real-time fire detection based on deep convolutional long-recurrent networks and optical flow method," in 2018 37th Chinese Control Conference (CCC), 2018: IEEE, pp. 9061-9066.
34. F. A. Saputra, M. U. H. Al Rasyid, and B. A. Abiantoro, "Prototype of early fire detection system for home monitoring based on Wireless Sensor Network," in 2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), 2017: IEEE, pp. 39-44.
35. J.-Y. Jang, K.-W. Lee, Y.-J. Kim, and W.-T. Kim, "S-FDS: A Smart Fire Detection System based on the Integration of Fuzzy Logic and Deep Learning," Journal of the Institute of Electronics and Information Engineers, vol. 54, no. 4, pp. 50-58, 2017.
36. J. Yao et al., "Predicting the minimum height of forest fire smoke within the atmosphere using machine learning and data from the CALIPSO satellite," Remote sensing of environment, vol. 206, pp. 98-106, 2018.
37. S. shensheng Xu, M.-W. Mak, and C.-C. Cheung, "Deep neural networks versus support vector machines for ECG arrhythmia classification," in 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2017: IEEE, pp. 127-132.
38. G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," Pattern Recognition Letters, vol. 30, no. 2, pp. 88-97, 2009.
39. D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9157-9166.

40. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, 2009: Ieee, pp. 248-255.
41. T. Kinnunen, J.-K. Kamarainen, L. Lensu, J. Lankinen, and H. Käviäinen, "Making visual object categorization more challenging: Randomized caltech-101 data set," in 2010 20th International Conference on Pattern Recognition, 2010: IEEE, pp. 476-479.
42. I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," Knowl. Eng. Data Sci, vol. 2, no. 1, pp. 41-46, 2019.
43. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097-1105, 2012.
44. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818-2826.
45. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," arXiv preprint arXiv:1602.07360, 2016.
46. P. Bellavista, K. Ota, Z. Lv, I. Mehmood, and S. Rho, "Towards smarter cities: Learning from Internet of Multimedia Things-generated big data," ed: Elsevier, 2020.
47. A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, "Fire and smoke detection using wavelet analysis and disorder characteristics," in 2011 3rd International Conference on Computer Research and Development, 2011, vol. 3: IEEE, pp. 262-265.
48. 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, no. 6, pp. 1103-1113, 2012.
49. T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in 2004 International Conference on Image Processing, 2004. ICIP'04., 2004, vol. 3: IEEE, pp. 1707-1710.
50. 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, no. 9, pp. 1545-1556, 2015.