System For Real Time Fire And Smoke Intensity Detection

VIKAS MAHESHKAR[1]
AYUSH SINGH[1]
HUNY DAHIYA[2]
VANSH SINGH[3]

Netaji Subhas University of Technology
Department of Information Technology
Azad Hind Fauj Marg - Sector 3 Dwarka - New Delhi - India
[1]vikas.maheshkar@nsut.ac.in
[2]ayush.singh.ug20@nsut.ac.in
[3]huny.ug20@nsut.ac.in
[4]vansh.ug20@nsut.ac.in

Abstract. Fire poses a significant threat to daily life, causing both economic and social harm. To mitigate these damages, early detection of fire and smoke is crucial, and this paper introduces a model employing vision-based techniques. The proposed model utilizes image processing and convolutional neural networks to detect fire and smoke, providing insights into their intensity and any changes in a video. The model comprises two units for fire and smoke detection, each employing image preprocessing techniques, including rule-based color detection and motion detection, along with CNN. The calculated percentages of fire and smoke in the processed images offer detailed information about the severity of the hazards in a specific area. The model detects whether the intensity of fire and smoke is increasing, decreasing or constant.

Keywords: Fire Detection, Smoke Detection, CNN, Colour Detection, Segmentation, Motion Detection

(Received December 12th, 2023 / Accepted June 28th, 2024)

1 Introduction

Fire, as a hazardous natural occurrence, poses significant risks to daily life, potentially resulting in casualties, economic setbacks, and environmental harm. Consequently, there is a growing emphasis on fire and smoke warning systems. Over the years, various detection systems relying on sensors have been developed. However, the considerable distance between the sensor and the fire source, particularly in outdoor settings, often leads to substantial delays in detecting fire or smoke events. In contrast, vision-based fire and smoke detectors offer several advantages, including rapid response, extensive coverage, and the visualization of alarm information. Given the widespread use of video surveillance systems, vision-based detectors are increasingly recognized as a promising technology for automated fire detection.

Traditional vision-based fire and smoke detection methods that are based on various colour spaces, spatial and motion features, have been widely investigated. However, feature engineering relies on professional knowledge, and threshold parameters need to be manually adjusted through experiments. Furthermore, the handcrafted features are the shallow features of the flame. It is extremely challenging to detect fires or smoke because of the complex fire and smoke types and scenes as well as interference events in practical application.

Deep learning, a widely adopted data-driven learning approach, has proven successful across various domains [2, 1, 33, 11, 31, 10, 32], including classification, object detection, and semantic segmentation [35, 30, 12, 13, 27]. In recent times, there has been a shift in focus among researchers towards employing convolutional neural networks (CNN) for extracting features related to fire and smoke. In contrast to traditional methods, CNN-based approaches eliminate the need for manually crafted features, instead learning deep features from extensive datasets of fire and smoke. This transition has led to a substantial enhancement in the accuracy of detection using CNN-based methods. Furthermore, these methods contribute to improved environmental robustness, particularly in challenging and uncertain practical settings.

Although the performance of the CNN-based methods has been greatly improved than traditional methods, there are still some problems. Current methods based on deep learning mostly considered fire and smoke detection as a classification task, which classifies the entire image into fire or non-fire. Sometimes, these methods generate false alarms caused by the interference of fire-like objects due to the complexity of the real environment.

In this paper, we propose a fire and smoke detection model which detects fire and smoke in a given video and also reports the real-time percentage levels of fire and smoke. The model consists of 2 modules i.e fire detection and smoke detection which work in parallel. The fire and smoke detection module consists of image processing techniques such as motion detection and rule-based colour-detection along with CNN for efficient fire and smoke detection.

The main contributions of this paper are as follows:

- A Multimodal fire and smoke detection framework is designed to integrate two different models (Fire detection and smoke detection), which are designed on the basis of fire video dataset and smoke video dataset, respectively.
- The real-time percentage levels of fire and smoke in real-time are shown accurately and efficiently
- The model identifies the real-time intensities of fire and smoke and categorizes fire and smoke as either escalating, diminishing, or remaining constant.
- Fire and smoke detection model is designed by integrating motion detection, rule-based colour detection and CNN for efficient fire and smoke detection

2 Related Works

Early fire detection methods usually describe fire regions based on manual designed features. Concretely, Borges et al. [5] integrated image colour information and Bayesian Classifier for fire recognition. Another approach proposed by Foggia et al. [14] considers fire colour, shape, and motion features comprehensively, but tiny fire could hardly be identified. With the development of deep learning, CNN has brought notable improvements in wild-fire detection. Barmpoutis et al. [3] identified fire regions by modifying Faster R-CNN [29], and wildfire are detected based on the locality criterion on manifold. Recently, a branch of latest studies extensively employs YOLO series to fire detection. Initially, Li et al. [21] compared fire detection results by Faster R-CNN, RFCN, SSD and YOLO v3, and concluded that YOLO v3 achieves the most satisfactory performance. Moreover, Zhang et al. [38] combined multi-scale output mechanism and channel-wise attention with YOLO v3 [28] to increase network generalization, Huang et al. [17] improved regression box loss in YOLO v4 [4] to enhance small-scale fire detection results. However, computational cost of YOLO series is generally higher than other advanced light-weight networks (MobileNet series, ShuffleNet series, etc.) for classification due to challenging object detection task.

Recently, many scholars have paid attention to fire detection in video surveillance. Matukhina et al. [22] firstly utilized Gaussian mixture model to detect moving regions, and identify fire through RGB colour analysis. Muhammad et al. [23] developed a CNN based fire detection framework that applies dynamic channel selection to fire detection on house, forest and vehicle. Then, considering of reducing computational cost for real-world video surveillance. Muhammad et al. [24] introduce a SqueezeNet [18] based architecture for fire video detection, which can better balance the efficiency and accuracy. In [25] and [26], two video-based fire detection networks are proposed by modifying Inception-V4 [36] and MobileNet [34]. In general, although light-weight network is considered efficient in real-world video surveillance system, it will cause false alarm to a certain extent.

Chen et al. [8] utilized a color model to identify fire and smoke pixels, extracting dynamic measures of growth and disorder for verification. However, the algorithm's false detection rate becomes notably high in scenes with fire-colored moving objects. Genovese [20] et al. proposed a smoke detection method based on computational intelligence techniques, focusing on features such as movement, color, and the edge of smoke. Yu et al. [9] proposed a video smoke detection method using color and motion features, but it faces challenges in achieving real-time detection rates due to computationally expensive optical flow.

Smoke detection systems employing machine vision techniques to categorize frame sequences primarily rely on static information extracted from individual frames. These systems either incorporate dynamic characteristics of smoke or use features such as texture, shape, color, and movement, as seen in [6]. However, this category of algorithms may yield unsatisfactory outcomes when there is a minimal chromatic difference between the background and smoke pixels, potentially resulting in an excessive number of false alarms, rendering the system impractical. Consequently, in [37], the issue of background estimation and segmentation is directly addressed.

Emerging deep learning approaches can automate the feature extraction process, enhancing effectiveness in image classification and object detection [19]. Additionally, optimization techniques and model approximations can be applied to enable the deployment of medium-sized models on low-performance embedded devices, facilitating the system's use across diverse domains. Several deep learning models, including R-CNN, YOLO, and SSD networks, have been suggested for fire and smoke detection. Despite the potential for higher accuracy, deep learning methods are often more intricate than traditional algorithms and may not be suitable for low-memory embedded devices.

3 Methodology

3.1 Overall Architecture

The objective is to recognize fire or smoke, determine their respective percentages, and classify the direction of change in a given input video. Our methodology comprises two modules: a smoke detection module and a fire detection module. Initially, the video is converted into frames and these frames are sent to the smoke detection module, as well as the fire detection module in parallel. Each module provides outputs indicating the presence or absence of fire and smoke, along with the corresponding percentage.

3.2 Fire Detection Module

We propose a 3-stage fire detection framework. In the first stage, the motion feature of the flame is utilized. Pixel movements are detected by comparing consecutive frames, and the detected moving pixels are labeled. In the second stage, various image processing techniques, including HSV (Hue Saturation Value), YCbCr (Y: Luminance, Cb: Chroma (blue minus luma), Cr: Chroma (red minus luma)) filters, are employed to identify the flame. The resulting flame image is extracted from the filtered image. In the third stage, the presence of fire is determined across the entire image using a CNN algorithm. Figure 1 depicts the proposed fire detection model.

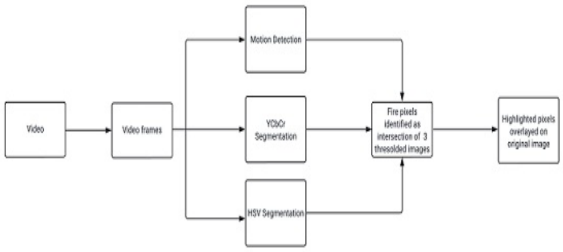

Figure 1: Proposed Fire Detection Model

3.2.1 Motion Detection

One of the swiftest and uncomplicated techniques for detecting motion in video images involves comparing two frames. This method entails examining pixels in the previous and subsequent frames of the video, marking those that exhibit variance. If the disparity is equal to or exceeds 30, the pixel is classified as being in motion. This approach effectively identifies the mobility of fire in the video. Algorithm 1 depicts the approach used for motion detection.

3.2.2 Image Processing

In computer vision applications, the human vision mechanism is imitated. At this stage of the fire detection framework, various HSV and YCbCr filters were applied to detect the fire on the images. Algorithm 2 depicts the steps carried out for performing the segmentation of fire.

HSV Filter : Video frames are converted from BGR to HSV colour space. The hue is commonly called color (mainly red, yellow, green, cyan, blue or magenta). Saturation refers to the intensity of the color between gray (low saturation or unsaturation) and pure color (high saturation). The value corresponds to the brightness of the color, between black (low value) and average saturation (maximum value). After conversion to the HSV scale we use thresholds to decide the range of pixel values that should be classified as flame pixels. Based on the thresholds, segmentation is performed to separate

Algorithm 1 Motion Detection Algorithmm for Fire Detection Model

Input : Current frame (C) and previous frame (P) of video

Output : Thresholded image (TI) after motion detection

procedure DETECTMOTION
  convert C to $grayscale$
  convert P to $grayscale$
  for each (row, col) in $greyCurr$ **do**
    calculate $difference = |greyCurr[row][col] - greyPrev[row][col]|$
    if $difference > threshold$ **then**
      $TI[row][col] = 255$
    else
      $TI[row][col] = 0$
    end if
  end for
  for $k \leftarrow 1$ to 5 **do**
    run $dilate()$ on TI
  end for
end procedure

the flame pixels from the non-flame pixels.

$$PixelValue = \begin{cases} (255, 255, 255) & \text{if } 0 \leq h \leq 60 \text{ and} \\ & 50 \leq s \leq 255 \text{ and} \\ & 100 \leq v \leq 255 \\ (0, 0, 0) & \text{otherwise} \end{cases}$$

YCbCr Filter : Next, we convert the original frame to the YCbCr scale. Y represents the brightness (luma), Cb represents the blue minus luma (B-Y) and Cr represents the red minus luma (R-Y). After conversion to the YCbCr scale, different filters are applied to segment the flame pixels from the non-flame pixels.

$$PixelValue = \begin{cases} (255, 255, 255) & \text{if } y \geq C_b \text{ and} \\ & C_r \geq C_b \text{ and} \\ & |C_r - C_b| \geq 20 \\ & \text{and } C_b \leq 120 \text{ and} \\ & C_r \geq 140 \\ (0, 0, 0) & \text{otherwise} \end{cases}$$

3.3 Smoke Detection Module

We propose a 3-stage smoke detection framework. In the first stage, motion detection is done using Kalman filters which are used to identify the foreground pixels with change in time. In the second stage, various image processing techniques, including HSV (Hue Satu-

Algorithm 2 Segmentation Algorithmm for Fire Detection Model

Input : Current frame (C) of video

Output : Thresholded image (TI) after segmentation

procedure PERFORMSEGMENTATION
  convert C to hsv and store in h
  create $mask$ for h based on light orange to dark orange thresholds
  $hsvImage = (h)$ bitwise and $(mask)$
  for each (row, col) in $hsvImage$ **do**
    if $hsvImage[row][col] > threshold$ **then**
      $hsvImage[row][col] = 255$
    else
      $hsvImage[row][col] = 0$
    end if
  end for
  convert C to YC_bC_r and store in y
  for each (row, col) in y **do**
    if $y[row][col]$ in threshold range **then**
      $y[row][col] = 255$
    else
      $y[row][col] = 0$
    end if
  end for
  for each (row, col) in (C) **do**
    if $hsvImage[row][col] == 255$ and $y[row[col] == 255$ **then**
      $TI[row][col] = 255$
    else
      $TI[row][col] = 0$
    end if
  end for
end procedure

ration Value), RGB(Red Green Blue), are employed to identify the smoke. The resulting images from the two detection methods are combined and passed through a denoising filter to reduce noise in the final image. It is then overlayed on the original image which is passed to the CNN for prediction. The proposed smoke detection model is shown in Figure 2.

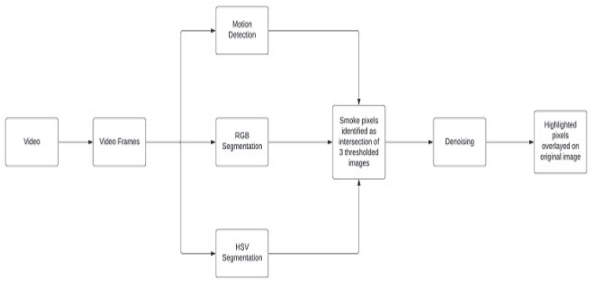

Figure 2: Proposed Smoke Detection Model

3.3.1 Motion Detection

The proposed motion detection algorithm for smoke is similar to the one in [15]. The Kalman filter is used to estimate movement within a series of frames. The motion detection algorithm detects groups of pixels that change their value over time. We allow the value of the pixel to evolve in time following a linear model. We use the Kalman filter to predict the expected value of a pixel based on its previous state history. If the difference between the predicted value and the actual value is larger than a threshold, a movement is detected and the pixel is marked accordingly. Motion detection approach for smoke is shown in Algorithm 3.

Algorithm 3 Motion Detection Algorithmm for Smoke Detection Model

Input : Current frame (C) and previous frame (P) of video

Output : Thresholded image (TI) after motion detection

procedure SMOKEMOTIONDETECTION
- calculate $backgroundIMg$ from P
- find $estimatedBgImg$ for C
- calculate $|foregroundImg = C - estimatedBgImg|$
- apply $thresholdFunc()$ to $foregroundImg$
- store thresholded image in TI

end procedure

The background prediction is given by Eq. 1, where $\widetilde{BG}_k$ is the background prediction of the current frame C, $\widehat{BG}_{k-1}$ is the background estimation at the previous frame P, and $\alpha = A/(1-\beta)$ is a weighting coefficient for the previous state of the pixel. We allow α to be dependent on the camera frame rate with the relation $\beta = 1/(1+\tau_\beta \cdot fr)$, where fr is the frame rate in FPS of the processed video and τ_β is a time constant set to 10s, and A is a constant set to 0.618.

$$\begin{bmatrix} \widetilde{BG}_k \\ \dot{\widetilde{BG}}_k \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ 0 & \alpha \end{bmatrix} \cdot \begin{bmatrix} \widehat{BG}_{k-1} \\ \dot{\widehat{BG}}_{k-1} \end{bmatrix} \tag{1}$$

The background estimation $\widehat{BG}_k$ of the frame I is obtained from Eq. 2, where $\widetilde{BG}_k$ in Eq. 1, and K_1 and K_2 are defined in Eq. 3.

$$\begin{bmatrix} \widehat{BG}_k \\ \dot{\widehat{BG}}_k \end{bmatrix} = \begin{bmatrix} \widetilde{BG}_k \\ \dot{\widetilde{BG}}_k \end{bmatrix} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} (I - \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \widetilde{BG}_k \\ \dot{\widetilde{BG}}_k \end{bmatrix}) \tag{2}$$

$$K_1 = K_2 = \Lambda \cdot FG_k + \beta \cdot (1 - FG_k) \tag{3}$$

$$FG_k = |C - \widetilde{BG}_{k-1}| \tag{4}$$

$$FG_k \geq THR_{foreg} \tag{5}$$

In the initialization phase, we set $\widetilde{BG}_k$ equal to initial frame C and $\dot{\widetilde{BG}}_k$ equal to zero. This initialization is done only when the first frame is received. According to Eq. 5, we select the pixel of the foreground FG_k if their value is higher than the threshold THR_{foreg}. In the above equations, FG_k is the foreground of the frame C, $\Lambda = 1/(1+\tau_\alpha \cdot fr)$, where τ_α is a time constant set to 16 s. The empirical threshold THR_{foreg} is set to 0.08.

3.3.2 Image Processing

In computer vision applications, the human vision mechanism is imitated. At this stage of the fire detection framework, various RGB and HSV filters were applied to detect the smoke on the images. The image segmentation algorithm for smoke is stated in Algorithm 4

RGB Filter: Video frames are captured in the BGR format and then converted into the RGB colour model. It is a standard colour model where R stands for Red, G stand for Green and B stands for Blue. RGB values vary from 0-255 and represent the various values of the pixel in 3 different channels. For classifying an image as a smoke or not smoke one, we consider the difference in the intensities of the various colour channels and compare them to a global threshold.

Algorithm 4 Segmentation Algorithmm for Smoke Detection Model

Input : Current frame (C) of video
Output : Thresholded image (TI) after segmentation

procedure PERFORMSEGMENTATION
  apply $threshold$ to saturation channel in h
  store thresholded image in $hsvImage$
  for each (row, col) in $hsvImage$ **do**
    if $hsvImage[row][col] > threshold$ **then**
      $hsvImage[row][col] = 255$
    else
      $hsvImage[row][col] = 0$
    end if
  end for
  for each $pixel$ in C **do**
    Calculate the differences $|R-G|$, $|G-B|$, $|B-R|$
    if any difference $>$ threshold **then**
      $pixel = 255$
    else
      $pixel = 0$
    end if
  end for
  store thresholded image in $rgbImage$
  $segmentedImg = hsvImage \wedge rgbImage$
end procedure

$$PixelValue = \begin{cases} (255,255,255) & \text{if } |R-G| \leq 10 \text{ and} \\ & |G-B| \leq 10 \text{ and} \\ & |R-B| \leq 10 \\ (0,0,0) & \text{otherwise} \end{cases}$$

HSV Filter: Next, we convert the original image to HSV format. HSV is a colour in which the colour values are considered in one channel only which is good for outdoor images. The hue is commonly called color (mainly red, yellow, green, cyan, blue or magenta). Saturation refers to the intensity of the color between gray (low saturation or unsaturation) and pure color (high saturation). The value corresponds to the brightness of the color, between black (low value) and average saturation (maximum value). Images containing smoke generally have a low saturation value and that is used to further classify the images as smoke or not smoke.

$$PixelValue = \begin{cases} (255,255,255) & \text{if } 0 \leq h \leq 179 \text{ and} \\ & 0 \leq s \leq 50 \text{ and} \\ & 0 \leq v \leq 255 \\ (0,0,0) & \text{otherwise} \end{cases}$$

3.4 Convolutional Neural Network

The Convolutional Neural Network (CNN) is a deep learning approach inspired by the visual processing mechanism in living organisms. It is commonly employed due to its effectiveness in various studies, including motion detection, image classification, and object detection in images. The key advantage lies in its ability to generate a comprehensive feature map using the raw image. Typically, a CNN consists of three main layers: convolutional, pooling, and fully connected layers. The Convolution layer extracts features from the input. The pooling layer is responsible for reducing the size of the extracted feature maps. After undergoing multiple convolution and pooling stages, image features are flattened in the flatten layer and then forwarded to the fully connected layer, which exhibits a neural network structure. Classification tasks are carried out within this layer. The CNN architectures for both fire and smoke smodel are shown in Figures 3 and 4 respectively.

4 Experiments

4.1 Benchmark Datasets

In the experiments, we adopt two widely used datasets VisiFire [7] and FireSense [16] for validation of fire and smoke detection.

Layer (type)	Param #	Stride	Filters	Kernel Size	Activation Function	Output
Conv2D	416	1	32	2 x 2	relu	255 x 255
MaxPool2D	0	2				127 x 127
Conv2D	8256	1	64	2 x 2	relu	126 x 126
MaxPool2D	0	2				63 x 63
Conv2D	32896	1	128	2 x 2	relu	62 x 62
MaxPool2D	0	2				31 x 31
Flatten	0					123008
Dense	125961216				relu	1024
Dense	524800				relu	512
Dense	513				sigmoid	1

Figure 3: CNN for Fire Detection Module

Layer(type)	Param#	Stride	Filters	Kernel Size	Activation Function	Output
Conv2D	34944	4	96	11x11	relu	246x246
MaxPool2D	0	2				122x122
Conv2D	614656	1	256	5x5	relu	118x118
MaxPool2D	0	2				58x116
Conv2D	885120	1	384	3x3	relu	56x114
Conv2D	1327488	1	384	3x3	relu	54x112
Conv2D	884992	1	256	3x3	relu	52x110
MaxPool2D	0	2				25x54
Flatten	0					345600
Dense	1415581696				relu	4096
Dense	8390656				relu	2048
Dense	2049				sigmoid	1

Figure 4: CNN for Smoke Detection Module

- VisiFire : The dataset for fire detection comprises of 15 videos, encompassing 13 videos featuring fires and 2 videos without fire. Among the fire videos, 7 depict forest fires, while the remaining 6 showcase outdoor fires.The dataset for smoke detection comprises of 21 videos, consisting of 10 videos featuring smoke and 11 videos without smoke.

- FireSense: The dataset for fire detection comprises 27 videos, encompassing 11 videos featuring fires and 16 videos without fire. Among the fire videos, there are 6 outdoor fire videos, 1 forest fire video, and 4 indoor fire videos. It contains 13 positive videos and 9 negative videos for smoke detection. The dataset for smoke detection consists of 22 videos, with 13 videos containing smoke scenarios and 9 videos without smoke.

4.2 Evaluation metrics

For the whole fire and smoke detection model, a result of fire or non-fire, smoke or non-smoke frame would be identified. Sensitivity is adopted here to reflect the ability that a method could identify fire video. Specificity here indicates the ability a model could exclude non-fire video. ACC shows the ratio of correctly identified frames to the total number of frames of a video. F-measure denotes the weighted harmonic average of Precision and Recall, and can give a more objective assessment results when Precision and Recall conflict. AUC provides a concise way to assess the discriminatory ability of the binary classification model, giving a single number that reflects its overall performance across different decision thresholds. Two-class confusion matrix is shown in Table 1.

		True Class	
		Negative	Positive
Predicted Class	Negative	TN	FN
	Positive	FP	TP

Table 1: Two-class confusion matrix

Performance metrics are calculated using the values on the confusion matrix. The formulas required to calculate the performance metrics used in the study are given in Table 2.

4.3 Implementation Details

In this section, the effectiveness and efficiency of the proposed framework are assessed. All experiments are conducted on the following hardware platform: Apple M1 Pro CPU (8 cores), 16GB RAM, and Apple M1 Pro built-in GPU. The software stack comprises Python 3.9.6, Jupyter Notebook 6.5.2, NumPy 1.23.2, TensorFlow 2.10.10, and OpenCV 4.5.0.

The CNN models were trained on datasets comprising numerous videos sourced from diverse online platforms. The training dataset for fire consists of a total of 2,120 images, with 1,552 depicting fire scenarios and 568 featuring non-fire contexts. The training dataset for smoke consists of 447 images, with 300 depict-

ABBREVIATION	FORMULA
ACC (Accuracy)	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
FSC (F-1 Score)	$FSC = 2 * \frac{PRE*RCL}{PRE+RCL}$
PRE (Precision)	$PRE = \frac{TP}{TP+FP}$
RCL (Recall)	$RCL = \frac{TP}{TP+FN}$
FPR (False Positive Rate)	$FPR = \frac{FP}{FP+TN}$
FNR (False Negative Rate)	$FNR = \frac{FN}{FN+TP}$

Table 2: Performance Metrics

ing smoke regions and 147 featuring non-smoke contexts. Subsequently, we evaluated the framework's performance using two well-known datasets, namely VisiFire and FireSense. The VisiFire dataset comprises 535 fire images, 38 non-fire images, 274 smoke images and 50 non-smoke images, while the FireSense dataset consists of 414 fire images, 504 non-fire images, 480 smoke images and 132 non-smoke images.

5 Results and Discussions

5.1 Detection of flame and smoke motion with motion detection

The flames of a fire and clouds of smoke are consistently in motion, making the mobility characteristic of the flames and smoke a valuable element in the stages of fire and smoke detection. By analyzing a video, the motion of the flames and smoke clouds is identified, and the areas in which movement occurs are highlighted. Figures 5 and 6 shows images after motion detection.

5.2 Extraction of the fire and smoke region from the image with image processing

Various filters were applied sequentially on the image in order to extract the fire and smoke region from the image.The sample images used in the experiments and the fire and smoke zones extracted as a result of the filters applied are shown in Figures 5 and 6.

5.3 Detecting the presence of fire and smoke with CNN

Details regarding the existence of fire and smoke within the images can be obtained through the implementation of CNN model. During the training phase of the CNN model, a dataset comprising 1,552 images depicting a

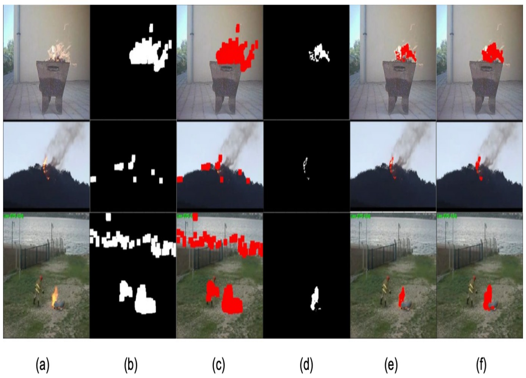

Figure 5: Examples of highlighting flame region from images (a) Input images (b) Threshold image after motion detection (c) Motion pixels highlighted (d) Thresholded image after colour based segmentation (e) Segmented pixels highlighted in original image (f) Final preprocessed image

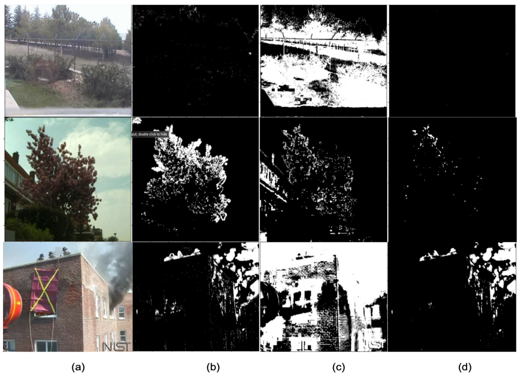

Figure 6: Examples of highlighting smoke region from images(a) original image (b) grayscale image after motion detection (c) Thresholded image after colour based segmentation (d) Final preprocessed image

fire zone and 568 images without a fire zone was utilized. Testing of the trained model was done on VisiFire and FireSense datasets. Confusion matrices of both the datasets for fire detection are shown in Tables 3 and 4. Confusion matrices of both the datasets for smoke detection are shown in Tables 5 and 6 respectively.

Various Performance Metrics such as Accuracy (Acc), F - 1 score, False Positive Rate(FPR) and False Negative Rate(FNR) of fire and smoke detection models for two popular datasets : VisiFire and FireSense are shown in Tables 7 and 8.

		True Class	
		Non - Fire	Fire
Predicted Class	Non - Fire	38	0
	Fire	1	535

Table 3: VisiFire confusion matrix for Fire Detection Model

		True Class	
		Non - Fire	Fire
Predicted Class	Non - Fire	504	2
	Fire	5	414

Table 4: FireSense confusion matrix for Fire Detection Model

		True Class	
		Non-Smoke	Smoke
Predicted Class	Non-Smoke	25	11
	Smoke	1	287

Table 5: VisiFire confusion matrix for Smoke Detection Model

Some examples of images classified correctly and incorrectly by the fire and smoke detection models are shown in Figures 7 and 8 respectively.

Graphs depicting Fire and smoke model history on accuracy on the datasets, AUC curve history for both the models and Loss degradation history on fire and smoke models are shown in 9, 10 and 11 respectively.

		True Class	
		Non-Smoke	Smoke
Predicted Class	Non-Smoke	97	2
	Smoke	2	511

Table 6: FireSense confusion matrix for Smoke Detection Model

	Accuracy	F-1 Score	FPR	FNR
VisiFire	99.83	99.91	2.56	0
FireSense	99.24	99.16	0.98	0.48

Table 7: Performance metrics of fire detection model on datasets

5.4 Calculating percentage of fire and smoke

The final preprocessed image, obtained after the application of advanced image processing and motion detection techniques on the video frame, is subsequently utilized to calculate the percentage of fire and smoke present within the image. The model also classifies the intensity of fire and smoke as either increasing, decreasing, or constant. This classification is achieved by conducting a comparative analysis between the current frame and previous frames of the video, thereby enabling a dynamic assessment of the situation. The results of this analysis are systematically presented in Table 9, which showcases various frames of videos alongside their corresponding percentages for fire and smoke, as well as the fire and smoke intensity classifications. Such detailed analysis allows for a comprehensive understanding of the behaviour and progression of fire and smoke over time, facilitating more effective monitoring and response strategies.

	Accuracy	F-1 Score	FPR	FNR
VisiFire	96.14	96.94	3.84	3.69
FireSense	96.47	96.55	2.02	0.41

Table 8: Performance metrics of smoke detection model on datasets

Frames	% of fire	class of fire	% of smoke	class of smoke
	5.38	original frame	0.53	original frame
	5.72	constant	0.56	constant
	15.26	increasing	0.71	increasing
	2.51	decreasing	0.23	decreasing
	2.65	original frame	2.77	original frame
	13.59	increasing	27.88	increasing
	14.84	constant	33.91	increasing
	7.53	decreasing	19.05	decreasing

Table 9: Table depicting percentage and class of frames

(a)

(b)

Figure 7: (a) Examples of images correctly classified as fire, (b) Examples of images incorrectly classified as no fire

(a)

(b)

Figure 8: (a) Examples of images correctly classified as smoke, (b) Examples of images incorrectly classified as no smoke

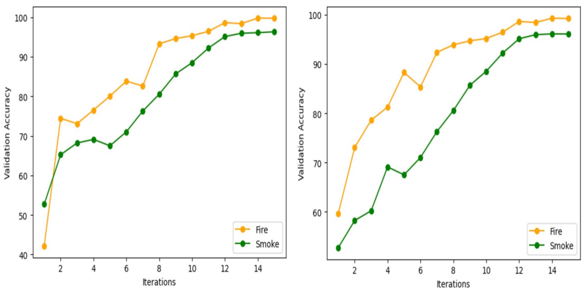

Figure 9: Testing Acc on (a)VisiFire (b)FireSense

6 Conclusions

This paper proposes a model for fire and smoke detection in videos through three methods. The first method assesses the mobility of fire and smoke using motion characteristics, but it proved insufficient alone. The second method applies image processing algorithms, utilizing RGB, HSV, and YCbCr filters for color-based isolation. Recognizing the need for multiple algorithms, a three-stage detection framework is proposed.The pivotal stage involves a CNN model trained on a web-sourced dataset, achieving high classification success rates for fire and smoke. The model classifies the intensity changes in a video, aiding timely fire management decisions.The model calculates the percentage of fire or smoke in real time. Overall, the three-method framework demonstrates satisfactory success and has the potential to enhance early fire detection and integration into automated extinguishing systems, with further improvements possible through dataset enhancement.

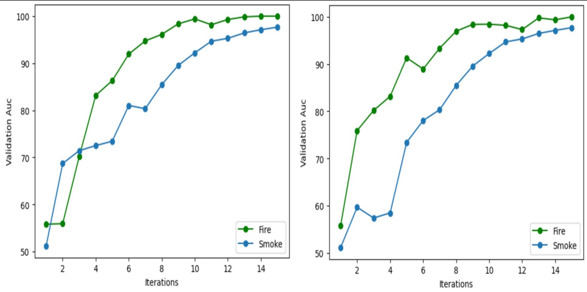

Figure 10: Testing AUC on (a)VisiFire (b)FireSense

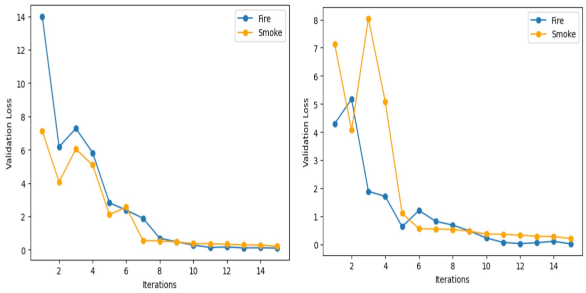

Figure 11: Testing Loss on (a)VisiFire (b)FireSense

References

[1] Affonso, E. T., Nunes, R. D., Rosa, R. L., Pivaro, G. F., and Rodriguez, D. Z. Speech quality assessment in wireless voip communication using deep belief network. *IEEE Access*, 6:77022–77032, 2018.

[2] Affonso, E. T., Rosa, R. L., and Rodriguez, D. Z. Speech quality assessment over lossy transmission channels using deep belief networks. *IEEE Signal Processing Letters*, 25(1):70–74, 2017.

[3] Barmpoutis, P., Dimitropoulos, K., Kaza, K., and Grammalidis, N. Fire detection from images using

faster r-cnn and multidimensional texture analysis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8301–8305. IEEE, 2019.

[4] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] Borges, P. V. K. and Izquierdo, E. A probabilistic approach for vision-based fire detection in videos. *IEEE transactions on circuits and systems for video technology*, 20(5):721–731, 2010.

[6] Çelik, T., Özkaramanlı, H., and Demirel, H. Fire and smoke detection without sensors: Image processing based approach. In *2007 15th European signal processing conference*, pages 1794–1798. IEEE, 2007.

[7] Cetin, E. Computer vision based fire detection dataset. *Online [Accessed 17 March 2022] http://signal. ee. bilkent. edu. tr/VisiFire/Ultimate chase. Online [Accessed 17 March 2022] http://ultimatechase. com*, 2015.

[8] Chen, T.-H., Wu, P.-H., and Chiou, Y.-C. An early fire-detection method based on image processing. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 3, pages 1707–1710 Vol. 3, 2004.

[9] Chunyu, Y., Jun, F., Jinjun, W., and Yongming, Z. Video fire smoke detection using motion and color features. *Fire Technology*, 46:651–663, 07 2010.

[10] Dantas Nunes, R., Lopes Rosa, R., and Zegarra Rodríguez, D. Performance improvement of a non-intrusive voice quality metric in lossy networks. *IET Communications*, 13(20):3401–3408, 2019.

[11] de Almeida, F. L., Rosa, R. L., and Rodriguez, D. Z. Voice quality assessment in communication services using deep learning. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6. IEEE, 2018.

[12] dos Santos, M. R., Batista, A. P., Rosa, R. L., Saadi, M., Melgarejo, D. C., and Rodríguez, D. Z. Asqm: Audio streaming quality metric based on network impairments and user preferences. *IEEE Transactions on Consumer Electronics*, 69(3):408–420, 2023.

[13] Dos Santos, M. R., Rodriguez, D. Z., and Rosa, R. L. A novel qoe indicator for mobile networks based on twitter opinion ranking. In *2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. IEEE, 2023.

[14] Foggia, P., Saggese, A., and Vento, M. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE TRANSACTIONS on circuits and systems for video technology*, 25(9):1545–1556, 2015.

[15] Gagliardi, A., de Gioia, F., and Saponara, S. A real-time video smoke detection algorithm based on kalman filter and cnn. *Journal of Real-Time Image Processing*, pages 1–11, 2021.

[16] Grammalidis, N., Dimitropoulos, K., and Cetin, E. Firesense database of videos for flame and smoke detection. *IEEE Trans Circuits Syst Video Technol*, 25:339–351, 2017.

[17] Hongyu, H., Ping, K., Li, F., and Huaxin, S. An improved multi-scale fire detection method based on convolutional neural network. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 109–112. IEEE, 2020.

[18] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[19] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[20] Labati, R. D., Genovese, A., Piuri, V., and Scotti, F. Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43:1003–1012, 2013.

[21] Li, P. and Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering*, 19:100625, 2020.

[22] Matukhina, O., Amaeva, L., and Merzlyakov, S. Fire detection system with utilization of industrial

video surveillance system. In *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pages 1–5. IEEE, 2020.

[23] Muhammad, K., Ahmad, J., and Baik, S. W. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, 288:30–42, 2018.

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

[25] Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., and Baik, S. W. Convolutional neural networks based fire detection in surveillance videos. *Ieee Access*, 6:18174–18183, 2018.

[26] Muhammad, K., Khan, S., Elhoseny, M., Ahmed, S. H., and Baik, S. W. Efficient fire detection for uncertain surveillance environment. *IEEE Transactions on Industrial Informatics*, 15(5):3113–3122, 2019.

[27] Okey, O. D., Udo, E. U., Rosa, R. L., Rodríguez, D. Z., and Kleinschmidt, J. H. Investigating chatgpt and cybersecurity: A perspective on topic modeling and sentiment analysis. *Computers & Security*, 135:103476, 2023.

[28] Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[29] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[30] Rezaie, V., Parnianifard, A., Rodriguez, D., Mumtaz, S., and Wuttisittikulkij, L. Speech emotion recognition using anfis and pso-optimization with word2vec. *J Neuro Spine*, 1(1):41–56, 2023.

[31] Rodriguez, D. Z. and Bressan, G. Video quality assessments on digital tv and video streaming services using objective metrics. *IEEE Latin America Transactions*, 10(1):1184–1189, 2012.

[32] Rodriguez, D. Z. and Junior, L. C. B. Determining a non-intrusive voice quality model using machine learning and signal analysis in time. *INFOCOMP Journal of Computer Science*, 18(2), 2019.

[33] Rodríguez, D. Z., Rosa, R. L., Almeida, F. L., Mittag, G., and Möller, S. Speech quality assessment in wireless communications with mimo systems using a parametric model. *IEEE Access*, 7:35719–35730, 2019.

[34] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[35] Silva, D. H., Maziero, E. G., Saadi, M., Rosa, R. L., Silva, J. C., Rodriguez, D. Z., and Igorevich, K. K. Big data analytics for critical information classification in online social networks using classifier chains. *Peer-to-Peer Networking and Applications*, 15(1):626–641, 2022.

[36] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[37] Vijayalakshmi, S. and Muruganand, S. Smoke detection in video images using background subtraction method for early fire alarm system. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, pages 167–171. IEEE, 2017.

[38] Zhang, X., Qian, K., Jing, K., Yang, J., and Yu, H. Fire detection based on convolutional neural networks with channel attention. In *2020 Chinese Automation Congress (CAC)*, pages 3080–3085. IEEE, 2020.