



Adaptive Flame Detection Using Video Image Processing and Optimization Algorithm

Zahraa Shihab AL Hakeem^{1*}, Hiba J. Aleqabie²

¹ Artificial Intelligence Engineering Department, College of Engineering and Information Technology, Al-Zahraa University for Women, Karbala, Iraq

² Artificial Intelligence Engineering Department, College of Engineering and Information Technology, Al-Zahraa University for Women, Karbala, Iraq
E-mail: zahraa.shihab@alzahraa.edu.iq

Received: 05/09/2025

Revised: 23/03/2026

Accepted: 05/09/2026

Available online: 13/09/2026

Abstract— Fires in farms, wheat fields, and homes have increased as a result of rising temperatures. Conventional smoke sensors are ineffective in open environments. Video Image Detection (VID) provides a compelling approach for detecting flames. This paper presents a fire detection system that employs a video camera able of adapting to environmental changes in color characteristics and illuminance. The proposed flame detection system merges two methods to achieve accurate detection. While the second method uses motion detection for flames, the first method concentrates on color identification. Using a one-level Lifting Wavelet Transform (LWT) to the Red Green Blue (RGB) input video reduces the processed data by 75% while preserving the essential fire features. Then, the frames are classified and extracted from the video with the help of some feature detectors. Using the firing threshold criterion, the area of fire is calculated and boxed accordingly. The proposed system for colour detection has two types of thresholds, which are adaptive thresholds of Kapur and Otsu criteria and Forest Optimization Algorithm (FOA) based objective functions. Based on the experimental results, the system is quite accurate. In real-time situations, the adaptive threshold flame detection system manages to achieve 97.5% indoors and 92.4% outdoors. The system works on time, i.e., within 0.6s. The system is performing robustly in different environmental situations and can identify fire inside buildings as well as outside.

Keywords— Fire Detection; Forest Optimization Algorithm; Video Image Detection.

1. INTRODUCTION

Fire is a serious risk that causes loss of property and life. In 2022, Iraq registered an estimated 32,477 fire incidents [1]. They caused significant damage and loss of life in outdoor farms. Moreover, they caused damage in woodlands and wheat fields. The fire inside hotels, restaurants, and offices in government and private buildings. Nevertheless, fire hazard zoning development and monitoring of ignition occurrences [2] These are the first steps toward minimizing the consequences of devastating fire. Old methods using cameras and CCTV often do not give proper alarms and detections. Traditional systems consist of flame detection functions visual data from fixed cameras.

For Fire Detection of Smart Home Environment, the paper proposes an adaptive threshold-based fire detection system. The suggested system incorporates color-based flame detection with motion analysis to perform detection with high reliability and low false alarm. Forest optimization algorithm is the method to find optimum multilevel thresholds using Kapur and Otsu objective functions. Moreover, the Integer Lifting Wavelet Transform is applied in the

preprocessing stage to reduce the computational cost by reducing the processed data almost 75% while keeping significant flame characteristics.

The fire detection systems based on video have made good progress but many challenges still remain especially in achieving high detection accuracy while maintaining low computational complexity in real-time applications. Many contemporary processes depend on deep learning models, which need massive training datasets and high computational power. This limits their use in real-time environments.

To solve these problems, this paper suggests a hybrid fire detection model that uses a combination of colour models (HSV, YCbCr), motion detection and optimized threshold selection. The main contributions of this paper can be summarized as follows:

1. A reliable fire detection system with a combination of HSV color model, YCbCr color model, and motion detection mechanism, which improves the reliability of the detection system and reduces false alarms .
2. The Forest Optimization Algorithm (FOA) focuses on the calculation of optimal multilevel thresholds utilizing Otsu and Kapur objective functions.
3. Reducing the computational complexity using the Integer Lifting Wavelet Transform (ILWT), which reduces the size of the processed data by nearly 75%.
4. Introduction to a frame-selection strategy to detect fire with fewer processed frames, along with depth detection accuracy. Experimental assessment that the video and benchmark dataset demonstrate high detection accuracy and fast detection speed of the proposed approach. The presented approach is suitable and competent for fire monitoring applications.

2. RELATED WORKS

Fire detection is vital for safety and helps prevent devastation and loss of life. Reviewing progress in fire detection is certainly an important task. Furthermore, key studies help provide useful knowledge that could be beneficial for R&D for fire detection systems. Thus, this paper looks at reviewing the progress in fire detection with reference to previous work. Gong et al. [3] developed a system identifying fires using frame differences and color analysis. By figuring out the mass center of the fire in every frame, they extracted spatial, shape, and area variability, reducing false positive rates. Despite its effectiveness, Gong et al.'s methodology is not commonly applied. Using a decision unit, an autonomous threshold mechanism, and local and global feature analysis, Hsu Ting Wei [4] proposed a system with minimal false alarms and high detection, though its reaction time is longer due to information buffering .Wahyono et al. [5] tackled this difficulty by employing probabilistic models with Gaussian multiples and moment-invariant analysis to identify dynamic fire motions from color features, achieving a high TPR of 89.92%, though camera placement remains a practical obstacle. Li et al. [6], which detects fires using a fully convolutional one-stage CNN. Wang et al. [7] proposed DSS-YOLO, a lightweight real-time fire detection model based on the YOLOv8 architecture, achieving improved performance on flame detection in video streams while reducing computational complexity. This suggests the potential of optimized deep learning models for real-time applications, although such models may still require careful dataset preparation and resource considerations. While this approach achieves high accuracy with a compact model size, it requires a substantial amount of information for training in order to cover various indoor and outdoor scenarios. In the field of fire detection video research, multi-domain technologies have

been utilized to overcome the limitations of existing systems, though they still present some challenges. Although this method attains high precision with a small model size, it necessitates a significant amount of data for training to encompass diverse indoor and outdoor situations. In fire detection video research, multi-domain technologies have been employed to address the shortcomings of current systems, although challenges remain.

3. PROPOSED METHOD

3.1. Pre-Processing Stage

Captured online datasets [8], [9]. This phase utilizes a median filter and a wavelet transform as in Fig. 1. Wavelets are mathematical basis functions used to estimate signals; the Discrete Wavelet Transform (DWT) decomposes signals into time-frequency components [10]. The Integer Lifting Wavelet Transform (ILWT) gears DWT with integer arithmetic; the Haar wavelet applies dilation and shifting [11]. In the (Int-to-Int) lifting arrangement, approximation quantities are averages of adjacent pixels, and the features are differences. With correlated pixels, the detail power is low. Since input samples (S_j) are integers, the prediction simplifies to $S_j(e) = \text{floor}(S_j(o))$, enabling efficient calculation of the full coefficients as in Equation 1.

$$\text{Int} - \text{to} - \text{int} - \text{HLWT prediction: } d_j - 1 = S_j(o) - S_j(e) \quad (1)$$

Here, $S_j(o)$ and $S_j(e)$ denote odd/even inputs [12]. The frame is decomposed into LL, HL, LH, and HH, where LL contains key detection information. ILWT reduces input size by 75%. The median filter replaces each pixel with the median of neighbours, reducing noise while preserving features [13]. In Fig. 1, the new frame, which has been compressed with little noise, is referred to as Frame_n .

3.2. Color Detection

The color of a flame is its most important feature; therefore, the second stage identifies fire regions using YCbCr and HSV with multilevel-thresholding, as shown in Fig. 1.

3.2.1. HSV Based on the Multi-Thresholding Method

This method distinguishes flame indoors and outdoors and adapts to light changes. Multilevel thresholding with Otsu and Kapur criteria optimized by FOA is applied, and flame color is detected through the steps in Fig. 1.

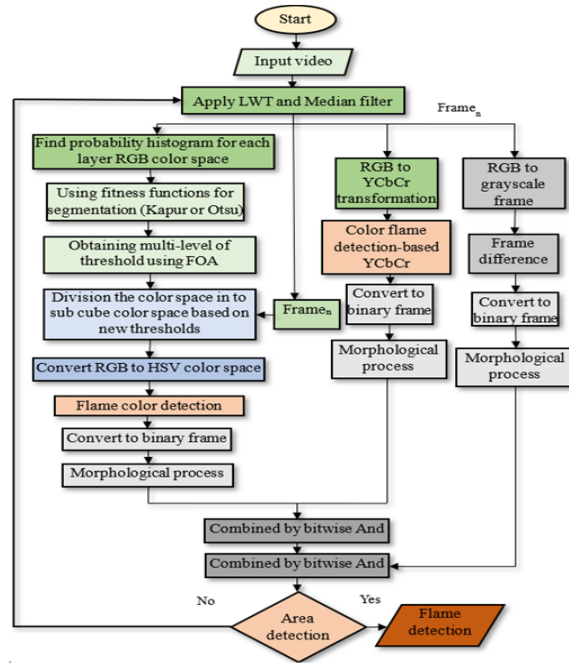


Fig. 1. Flowchart of the flame color detection

A. For any gray level i^{th} between 0 and $L-1$, find the probability histogram for each layer (R, G, and B). The likelihood of a particular gray level (i) can be expressed as:

$$p_i = h_i / (M \times N) \quad (2)$$

The frame's dimensions are denoted by M and N , and the number of pixels with gray level i , $0 < i \leq (L - 1)$, is denoted by h_i .

B. The two most widely used thresholding functions are Otsu's between-class variance technique [14] and Kapur's entropy criterion approach [15]. Using a target function maximized Kapur's entropy criterion for multilevel thresholding t_m [16]:

$$U(t_1, t_2, \dots, t_m) = H_0 + H_1 + H_2 + \dots + H_m \quad (3)$$

Where the partial entropies range from H_0 to H_m .

C. Multilevel thresholds for any grayscale image can also be found using Otsu's between-class variance approach. Thus, an objective function with m dimensions is described as follows [16]:

$$U(t_1, t_2, \dots, t_m) = \sigma_0 + \sigma_1 + \sigma_2 + \dots + \sigma_m \quad (4)$$

where $t_1 < t_2 < t_3 < \dots < t_m$.

D. The Forest Optimization Algorithm [17] is a nature-inspired method for multilevel threshold selection, simulating tree survival and reproduction. The procedure used to obtain the optimal threshold values is summarized in Algorithm 1. Parameter settings are in Table1.

Algorithm 1 FOA-based Multilevel Threshold Optimization

Input: Image histogram, number of thresholds m

Output: Optimal threshold values $T = \{t_1, t_2, \dots, t_m\}$

- 1: Initialize forest population with random threshold sets
- 2: Evaluate the fitness of each tree using the Otsu or Kapur objective function
- 3: when the termination condition is not satisfied, do

- 4: Perform Local Seeding
- 5: Modify threshold values of selected trees
- 6: Evaluate new trees using the fitness function
- 7: Perform Global Seeding
- 8: Generate new candidate solutions
- 9: Remove trees exceeding age or area limits
- 10: Update the best solution
- 11: end while
- 12: Return optimal threshold values

Table 1 The FOA parameters in use.

Parameter	Description	Value
Transfer_rate	Percentage of candidate population for global seeding	10
LSC (Local Seeding Change)	Number of features modified in local seeding	1
Ub (Upper Bound)	Maximum pixel intensity threshold	255
area_limit	Maximum number of trees in the forest	200
T-max	Maximum number of algorithm iterations	200
GSC (Global Seeding Change)	Number of features modified in global seeding	1
Lb (Lower Bound)	Minimum pixel intensity threshold	1

The ideal threshold values are determined using the FOA parameters in use. The designated optimization approaches are used for image histogram segmentation. The procedure goes like this:

- i. Using the optimization algorithm's lower and upper bounds as a guide, determine the maximum and minimum threshold limits.
- ii. Establish the m-dimensional position of each population member, with dimension values denoting the number of thresholds in the sequence $t_1 < t_2 < t_3 < \dots < t_m$.
- iii. The goal function of each population member should be evaluated using equations (3) and (4).
- iv. Update the threshold values for every member.
- v. If the termination requirements are met, the process is finished; if not, return to step iii.
- vi. The appropriate threshold values are those that match the global best member.

F. Thresholds for each channel divide the color cube into sub-cubes, grouping pixels into clusters. These subdivisions vary per image, affecting pixel distribution and cluster shape; larger clusters are less homogeneous. Increasing thresholds improves homogeneity, with maximum clusters:

$$c_m = (m + 1)^3 \quad (5)$$

For $m=1$, there are 8 sub-clusters, $m=2$ gives 27, and $m=3$ gives 64. As the threshold value increases, the pattern continues to occur.

G. RGB color space conversion to HSV color space [18]. Figs. 2 and 3 show how three thresholds are used per channel during the HSV color space conversion for Otsu, Kapur, and FOA in both indoor and outdoor scenarios. The blue-sky color represents the area of fire color, R_{color} .

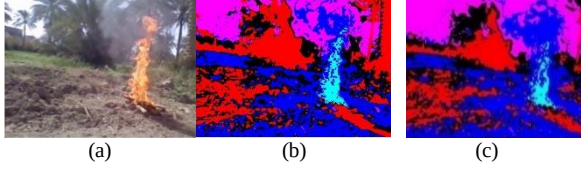


Fig. 2. Flame detection using HSV clusters with three thresholds for outdoor scenes: (a) RGB frames, (b) results using FOA-Otsu, and (c) results using FOA-Kapur.

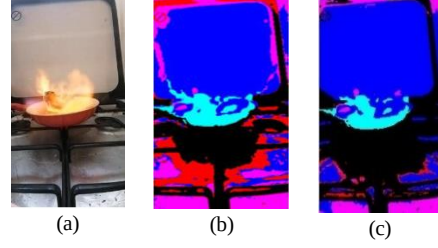


Fig. 3. Flame detection using HSV clusters with three thresholds for indoor scenes: (a) RGB frames, (b) results using FOA-Otsu, and (c) results using FOA-Kapur.

3.2.2. YCbCr Color Space

The YCbCr model separates luminance (Y) from chrominance (Cb, Cr), giving better brightness-color discrimination than RGB [19]; converting RGB to YCbCr improves detection [20]. Each frame is decomposed into Y, Cb, and Cr; fire frames show higher Y than Cb and $Cr > Cb$ (Fig. 4).

$$Rule\ 1 : F(M, N) = \begin{cases} 1, & \text{if } Y(M, N) > Cb(M, N) \cup Cr(M, N) > Cb(M, N) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Since flames are the brightest regions, their Y exceeds the frame mean, while Cb is lower and Cr is higher [21]. This leads to the following rule:

$$Rule\ 2: F(x, y) = \begin{cases} 1, & \text{if } Y(M, N) > Y_{mean} \cup Cb(M, N) < Cb_{mean} \cup Cr(M, N) > Cr_{mean} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$Y_{mean} = \frac{1}{q} \sum_{ii=1}^q Y(J_{ii}, K_{ii}), \quad Cb_{mean} = \frac{1}{q} \sum_{ii=1}^q Cb(J_{ii}, K_{ii}) \text{ and } Cr_{mean} = \frac{1}{q} \sum_{ii=1}^q Cr(J_{ii}, K_{ii}) \quad (8)$$

The pixel coordinates (M, N) denote spatial position, with Y-mean, Cb-mean and q the total pixels. Combining Rule 1 and Rule 2 yields the flame color area R_{color} , as shown in Fig. 5.

$$Rule\ 3 = Rule\ 1 \cap Rule\ 2 \quad (9)$$

The binary frames and morphological processing [22] over the binary frames to get them ready to combine with the YCbCr color space are displayed in Figs. 6 and 7.

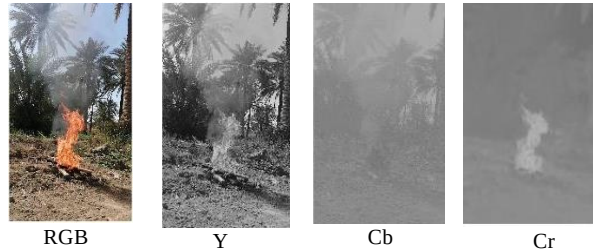


Fig. 4 The flame Frame_n and Separated Y, Cb, Cr Layers for Outdoor Scenes

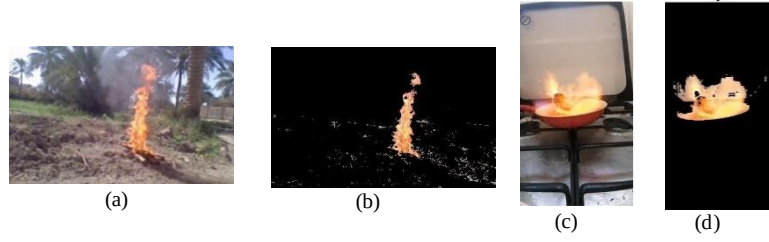


Fig. 5. Flame detection using YCbCr: (a) Outdoor Framen and detected flame regions, (b) Indoor Framen and detected flame regions.



Fig. 6. Morphological process for outdoor flame detection: (a) FOA-Otsu based, (b) FOA-Kapur based.

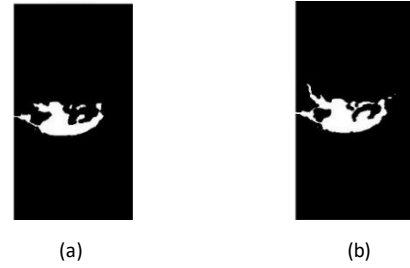


Fig. 7. Morphological process for indoor flame detection: (a) FOA-Otsu based, (b) FOA-Kapur based.

In the last stage, a binary AND operator is used to combine the HSV and YCbCr color space rules to create a complete two-color model. To identify fire zones as regions of interest, this model is applied to every frame of the video. $R_{color}(M, N, j)$, which denotes the regions of interest in terms of their spatial coordinates within the frame, is the precise definition of these fire zones.

3.3. Motion Detection (Frame Difference)

Frame difference [23] detects motion by subtracting consecutive frames, but this is unsuitable for flames due to irregular movement. Instead, a time-delay frame difference is used, subtracting frames spaced by 8. These better capture flame pixel changes. Frames are first converted to grayscale to reduce complexity, then the absolute differential is computed as:

$$I_{d(j,j+8)} = |I_{(j+8)} - I_j| \quad (10)$$

In the video sequence, the value of the j th frame is denoted as I_j . Similarly, the value of the $(j+8)$ th frame is represented as $I_{(j+8)}$. $I_{d(j,j+8)}$ denotes the absolute difference between the two frames.

An 8-frame interval was chosen experimentally to balance capturing noticeable flame changes and maintaining stable motion detection. The frame difference technique, based on time delay and absolute frame differencing, effectively highlights regions of flame motion for accurate analysis.

The binary frames from the earlier stages are shown in Fig. 8, along with the use of morphological processing methods. These morphological operations [22] in are intended to improve the binary frames and get them ready to be combined with the two-color technique.

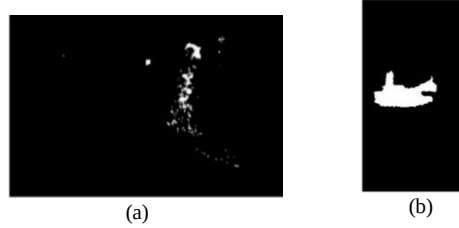


Fig. 8 Morphological process for motion detection: (a) outdoor fire, (b) indoor fire.

3.4. Combining Frame Differences and Two-color

Preprocessing enhances the suitability of an image on which fire detection will be performed. To accurately detect fire in a video sequence, it is paramount to address the shortcomings of relying on frame differences or two-color detection of fire attributes; Hence, it is a must to combine the two. the fire area denoted as $R_{fire}(M, N, j)$ using Equation (11):

$$R_{fire}(M, N, j) = R_{color}(M, N, j) \cap Id(j, j + 8)(M, N, j) \quad (11)$$

The binary image from the region of interest found using two-color detection and the image difference between frames $Id(j, j+8)(M, N, j)$ at a time interval of 8 frames are represented by this equation. Furthermore, the result of the combination is no longer sequential because of the improved frame difference method. Instead of the typical 30 frames per second, this enhanced method, known as chosen frame N_s , allows fire detection to be accomplished in just four frames. This progression pointedly hastens processing time, ensuring both accuracy and prompt alarm notifications.

Figs. 9 and 10 illustrate the integrated method, where flame regions are enclosed with green bounding boxes. A region is classified as fire if the area difference between the original frame and the detected fire boundary exceeds a predefined threshold (≥ 55).

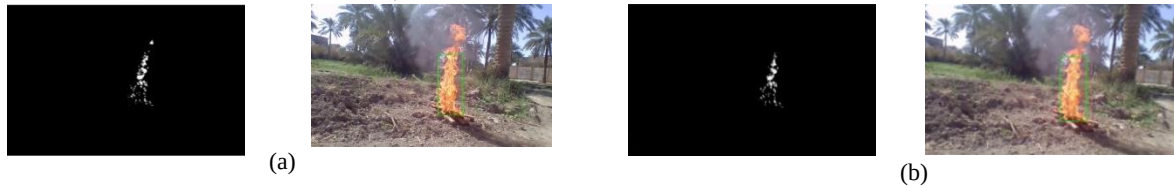


Fig. 9 combination between color detection and motion detection, which results in a green box for outdoor (a) FOA-Otsu (b) FOA-Kapur.



Fig. 10 combination between motion detection and color detection, which results in a green box for indoor (a) FOA-Otsu (b) FOA-Kapur.

4. Results and Discussion

MATLAB R2021b was used to implement the suggested system on a Windows 10 computer with an Intel Core i7 2.70 GHz CPU and 16 GB RAM, and flame detection experiments were conducted on recorded and real-time videos under indoor and outdoor conditions. Real-time testing used a laptop camera at 1280×720 and 30 FPS, with datasets including self-recorded footage, VisiFire, FireSense, and publicly available sources[8], [9]. The tested videos covered diverse scenes such as real fires and fire-like objects (e.g., sunlit fields or flame-colored

clothing) in various indoor and outdoor environments, providing a wide range of evaluation scenarios.

4.1. Tests of the Real-time Flame Detection

Five videos having various backgrounds and conditions in Fig 11. were used for real-time flames detection testing in indoor and outdoor environments. 11. The flames detected through our proposed algorithm are shown in the green box. As illustrated in Table 2, these videos exhibit an average detection accuracy. The average detection accuracy is achieved using the FOA approach on an Otsu and Kapur objective function as calculated using equation (12). The measure of accuracy, which falls between 0 and 1, provides insight into the reliability of tests. When the test's results correspond closely with the actual results of the test subject, we can say that the accuracy of the assay is highly reliable.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+TP} \quad (12)$$

In this equation, the TP, FP, TN, and FN show true positive, false positive, true negative, and false negative, which is correct. The good clip has fire, whereas the bad clip does not have fire. In the positive video, the detection that occurs correctly is TP, and the detection that fails is FP. The negative video will be classified incorrectly, which refers to the FN and not.



Fig. 11. Real-time fire detection outcomes for outdoor and indoor scenarios

The Otsu algorithm outperformed the Kapur algorithm in flame detection probability as recorded in Table 3. That is to say, Otsu achieves over 96% flame accuracy although Kapur achieves only 93% flame accuracy. This is because Otsu's maximization of the between-class variance enhances separation. This, in turn, enhances detection between flame and background. On the other hand, Kapur's entropy-based approach is less effective for fire features.

Indoor detection accuracy surpasses outdoor accuracy as the flame is better seen indoors compared to outdoors in low-light conditions. As illustrated in Fig. The system (V4, V5) knows the difference between the colours of the sunlight and flame and does not get confused with misleading light sources (V1). It can also detect small flames (V3).

Table 2 Real-time flame detection accuracy using FOA with Otsu and Kapur objective functions

Video Name	Frame number	Ns	Otsu				Kapur					
			TN	FP	TP	FN	ACC	TN	FP	TP	FN	ACC
V 1	246	49	0	1	48	0	97.9	0	2	46	1	100
V 2	330	41	0	1	40	0	97.5	0	0	40	1	97.5
V 3	270	23	0	0	23	0	100	0	0	22	1	95.6
V 4	640	80	0	2	74	4	92.5	0	3	73	4	91.2
V 5	390	31	0	0	30	1	96.7	0	1	28	2	84.8
Total	1876	224	0	4	215	5	95.9	0	6	209	9	93.3

Indoor Total	846	113	0	2	111	0	98.2	0	2	108	3	95.5
Outdoor Total	1030	111	0	2	104	5	93.6	0	4	101	6	90.9

*Note: Ns denotes the number of the selected and tested frame

In practice, the system helps in detecting flame instantly; the response time is 0.6 seconds or less. Also, the proposed system achieves detection by adopting the lifting wavelet transform. It reduces overall detection time by half and helps in detecting flame instantly.

4.2. Performance Evaluation

Two datasets, VisiFire [8] and FireSense [9], which included films with various resolutions, scenes, and fire conditions, and backgrounds for a robust assessment, were used to evaluate the system. Table 3 details the datasets: The following non-fire films are from FireSense [9], whereas the first 11 fire videos are from VisiFire [8]. Detection performance was measured using TPR, TNR, FPR, and FNR.

Table 3 Dataset used for evaluating the proposed flame detection method.

Video	Frame rate	Resolution	Fire	Frames	Notes
V 1	15	400 × 256	Yes	219	Forest4
V 2	15	400 × 256	Yes	208	Controlled3
V 3	15	400 × 256	Yes	200	Forest1
V 4	15	400 × 256	Yes	246	Controlled2
V 5	2	320 × 240	Yes	241	fBackYardFire
V 6	30	640 × 368	No	541	negsVideo7.864
V 7	15	400 × 256	Yes	216	Forest5
V 8	10	320 × 240	No	160	negsVideo3.860
V 9	10	320 × 240	No	155	negsVideo2.859
V 10	25	360 × 288	No	246	negsVideo10.1072
V 11	15	400 × 256	Yes	245	Forest2
V 12	25	320 × 240	No	180	negsVideo11.1073
V 13	30	480 × 368	No	439	negsVideo5.862
V 14	24.46	352 × 288	NO	272	negsVideo11.1075
V 15	29.97	320 × 240	Yes	140	40m PanFire 20060824
V 16	25	640 × 480	No	645	negsVideo9.866
V 17	18.51	1600 × 1200	No	196	negsVideo16.1077
V 18	5	240 × 230	Yes	236	Fire1
V 19	15	400 × 256	Yes	260	Controlled1
V 20	15	400 × 256	Yes	255	Forest3

To comprehensively evaluate performance, two groups of metrics were employed. The first set, including TPR and FNR, compared the proposed system with Meth1 [24], Meth2 [25], and Meth3 [5]. , as shown in Table 4, based on Otsu and Kapur functions with FOA optimization. Table 8 presents TNR and FPR performance compared to Wahyono [5] and other FOA-based methods.

Table 4 Comparison of FNR and TPR results on the VisiFire

Video	V 1	V 2	V 3	V 4	V 5	V 7	V 11	V 15	V 18	V19	V 20	Avg
Method												
Meth1 [24]	TPR	-	73.7	-	87.5	-	-	-	-	34	-	65.1
	FNR	-	10	-	4.9	-	-	-	-	0	-	4.96
	TPR	-	95	-	-	-	-	-	-	94.9	-	94.9

Meth2 [25]	FNR	-	5	-	-	-	-	-	-	-	5.02	-	5.01
Meth3 [5]	TPR	100	100	100	100	98.6	98.6	100	88.8	74.8	100	92.2	95.7
	FNR	0	0	0	0	1.4	1.4	0	11.2	25.2	0	7.8	4.3
FOA- Otsu	TPR	100	100	100	100	100	98.9	100	94.8	73.9	100	95.4	96.6
	FNR	0	0	0	0	0	1.1	0	5.2	26.1	0	4.6	3.4
FOA- Kapur	TPR	100	100	100	100	100	98.7	100	92.4	73.7	100	93	96.2
	FNR	0	0	0	0	0	1.3	0	7.6	26.3	0	7	3.8

Due to limited resources, our approach was compared only with methods evaluated on the same videos used by Wahyono [5]. Testing 11 VisiFire videos, our method achieved higher TPR than Wahyono's (Table 4) and outperformed Töreyn [23] and Ko [26]. on three videos. On non-fire FireSense videos, our method achieved TNR above 96% versus Wahyono's 38.73%, with FPR below 3.5% (Table 5).

Table 5 Comparison of FPR and TNR results on the FireSense

Video	The method of Wahyono [5]		Proposed methods			
			FOA-Otsu		FOA-Kapu	
	FPR	TNR	FPR	TNR	FPR	TNR
V 6	66.54	33.46	0	100	0	100
V 8	79.38	20.62	0	100	0	100
V 9	72.26	27.74	0	100	0	100
V 10	65.45	34.55	6.45	93.55	6.45	93.55
V 12	33.58	66.42	0	100	0	100
V 13	61.28	38.72	0	100	0	100
V 14	40.56	59.44	0	100	0	100
V 16	78.29	21.71	8.68	91.32	7.5	92.5
V 17	54.08	45.92	16.67	83.33	16.67	83.33
Avrg	61.27	38.73	3.533	96.467	3.402	96.598

5. CONCLUSION

In conclusion, the proposed flame detection framework demonstrates high effectiveness and robustness across both indoor and outdoor environments, including multi-fire conditions. The integration of Int-to-Int HLWT in the preprocessing stage successfully reduced computational overhead by nearly 50% while maintaining detection accuracy. Further, the systems proposed are still able to detect flames with high accuracy in the presence of sunlight and other fire-colored backgrounds due to the optimization-based color threshold. The use of the adaptive fire detection system has also brought about a significant reduction of 5% in false fire alarm activation, which is very helpful in minimizing the operational costs of the detection system. This is needed in order to increase the overall reliability of the fire detection system, which is also able to lower system false alarm rates. This is key to preserving the user's attention, minimizing the interruptions caused by unnecessary alarm triggers.

Despite its promising performance, the system is limited by reliance on color and motion features, making it sensitive to lighting conditions and similar fire-like objects, and it

does not utilize thermal or multispectral data. Future work will focus on integrating deep learning and extending the system for real-time, large-scale and embedded applications.

REFERENCES

- [1] "Iraqi Ministry of Interior," 2023. <https://moi.gov.iq/?page=4417> (accessed Jan. 05, 2023).
- [2] B. Pradhan, M. D. H. Bin Suliman, and M. A. Bin Awang, "Forest fire susceptibility and risk mapping using remote sensing and geographical information systems (GIS)," *Disaster Prev. Manag. An Int. J.*, vol. 16, no. 3, pp. 344–352, 2007, doi: 10.1108/09653560710758297.
- [3] F. Gong *et al.*, "A real-time fire detection method from video with multifeature fusion," *Comput. Intell. Neurosci.*, vol. 2019, p. 18, 2019, doi: 10.1155/2019/1939171.
- [4] T. W. Hsu *et al.*, "An early flame detection system based on image block threshold selection using knowledge of local and global feature analysis," *Sustain.*, vol. 12, no. 21, pp. 1–22, Nov. 2020, doi: 10.3390/su12218899.
- [5] Wahyono, A. Harjoko, A. Dharmawan, F. D. Adhinata, G. Kosala, and K. H. Jo, "Real-Time Forest Fire Detection Framework Based on Artificial Intelligence Using Color Probability Model and Motion Feature Analysis," *Fire*, vol. 5, no. 1, p. 23, 2022, doi: 10.3390/fire5010023.
- [6] Y. Li, J. Shang, M. Yan, B. Ding, and J. Zhong, "Real-Time Early Indoor Fire Detection and Localization on Embedded Platforms with Fully Convolutional One-Stage Object Detection," *Sustainability*, vol. 15, no. 3, p. 1794, 2023, doi: 10.3390/su15031794.
- [7] H. Wang, X. Fu, Z. Yu, and Z. Zeng, "DSS-YOLO : an improved lightweight real-time fire detection model based on YOLOv8," *Sci. Rep.*, vol. 15, no. 1, p. 8963, 2025, doi: <https://doi.org/10.1038/s41598-025-93278-w>.
- [8] A. E. Çetin, "COMPUTER VISION BASED FIRE DETECTION SOFTWARE," 2014. <http://signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html> (accessed Dec. 20, 2022).
- [9] N. Grammalidis, K. Dimitropoulos, and E. Cetin, "FIRESSENSE database of videos for flame and smoke detection," Jul. 31, 2017. <https://zenodo.org/record/836749> (accessed Feb. 10, 2023).
- [10] Z. Váňa, S. Privara, J. Cigler, and H. A. Preisig, "System identification using wavelet analysis," *Eur. Symp. Comput. Aided Process Eng.*, vol. 29, pp. 763–767, 2011, doi: 10.1016/B978-0-444-53711-9.50153-X.
- [11] M. Ramalingam and N. A. M. Isa, "Video steganography based on integer Haar wavelet transforms for secured data transfer," *Indian J. Sci. Technol.*, vol. 7, no. 7, pp. 897–904, 2014, doi: 10.17485/ijst/2014/v7i7.4.
- [12] H. I. Shahadi, R. Jidin, and W. H. Way, "High Performance FPGA Architecture for Dual Mode Processor of Integer Haar Lifting-Based Wavelet Transform," *Int. Rev. Comput. Softw.*, vol. 8, no. 9, pp. 2058–2067, 2013, [Online]. Available: <https://www.researchgate.net/publication/259753239%0AHigh>.
- [13] H. Hwang and R. A. Haddad, "Adaptive Median Filters: New Algorithms and Results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, 1995, doi: 10.1109/83.370679.
- [14] N. OTSU, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man Cybern.*, no. 1, pp. 62–66, 1979, doi: <https://doi.org/10.1109/TSMC.1979.4310076>.
- [15] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram," *Comput. VISION, Graph. IMAGE Process.*, vol. 29, no. 3, pp. 273–285, 1985, doi: [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2).
- [16] T. R. Farshi, R. Demirci, and M. R. Feizi-Derakhshi, "Image clustering with optimization algorithms and color space," *Entropy*, vol. 20, no. 4, p. 296, 2018, doi: 10.3390/e20040296.
- [17] M. Ghaemi and M. R. Feizi-Derakhshi, "Forest optimization algorithm," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6676–6687, 2014, doi: 10.1016/j.eswa.2014.05.009.
- [18] A. R. Smith, "Color Gamut Transform Pairs," in *SIGGRAPH 78 Conference Proceedings*, 1978, no. 2, pp. 12–19.
- [19] C. E. Premal and S. S. Vinsley, "Image processing based forest fire detection using YCbCr colour model," in *2014 International Conference on Circuits, Power and Computing Technologies, ICCPCT 2014*, 2014, pp. 1229–1237, doi: 10.1109/ICCPCT.2014.7054883.
- [20] T. Çelik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Saf. J.*, vol. 44, no. 2, pp. 147–158, 2009, doi: 10.1016/j.firesaf.2008.05.005.
- [21] R. A. Khan, J. Uddin, S. Corraya, and J.-M. Kim, "Machine vision-based indoor fire detection using static and dynamic features," *Int. J. Control Autom.*, vol. 11, no. 6, pp. 87–98, 2018, doi: <https://doi.org/10.14257/ijca.2018.11.6.09>.
- [22] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing ,Second edition*. New Jersey: Parson, 2009.
- [23] N. Singla, "Motion Detection Based on Frame Difference Method," *Int. J. Inf. Comput. Technol.*, vol. 4, no. 15, pp. 1559–1565, 2014, [Online]. Available: http://www.ripublication.com/irph/ijict_spl/ijictv4n15spl_10.pdf.
- [24] B. U. Töreyn, Y. Dedeoğlu, U. Güdükbay, and A. E. Çetin, "Computer vision based method for real-time

- fire and flame detection," *Pattern Recognit. Lett.*, vol. 27, no. 1, pp. 49–58, 2006, doi: 10.1016/j.patrec.2005.06.015.
- [25] T. X. Truong and J.-M. Kim, "Fire flame detection in video sequences using multi-stage pattern recognition techniques," *Eng. Appl. Artif. Intell.*, vol. 25, pp. 1365–1372, 2012, doi: 10.1016/j.engappai.2012.05.007.
- [26] B. C. Ko, K. H. Cheong, and J. Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Saf. J.*, vol. 44, pp. 322–329, 2009, doi: 10.1016/j.firesaf.2008.07.006.