Experimental Study on Multiple Fire Detection Algorithms in Images and Videos

$YingjieLiu$[1] Quan Gu[2]
[1]Université Paris-Saclay [2]Université Paris-Saclay
yingjie.liu@etu − upsaclay.fr, quan.gu@etu − upsaclay.fr

Abstract

Fire detection plays a critical role in safety monitoring and environmental protection. Traditional color-threshold-based detection methods are easy to implement and can achieve high recall, but they often suffer from high false positives under complex backgrounds. This report explores various improvement strategies on top of color-based methods, such as enhanced color range, morphological operations, multi-scale analysis, Gabor texture features, frequency-domain analysis (FFT), and machine learning methods (SVM, CNN). Experiments were conducted on 206 test images (and 42 images in a separate test for CNN) to compare accuracy, precision, recall, and F1-score. Our findings show that integrating color, texture, frequency, and motion information significantly improves overall performance while maintaining a high recall rate. We also discuss an optical-flow-based approach for fire detection in video scenarios. The results suggest that multi-modal methods can better balance false alarms and missed detections in complex environments.

1 Introduction

Fire remains a major threat to human life and property, especially in high-risk areas such as forests, factories, and residential buildings. A timely and reliable fire detection system is of great significance to prevent fire incidents. Traditional fire detection techniques often involve smoke sensors or color thresholding, but such methods can suffer high false alarms under complex lighting and cluttered backgrounds. Recent research indicates that fusing multiple features (e.g., color, texture, frequency-domain, motion) or applying machine learning techniques (SVM, CNN) can improve the robustness and accuracy of fire detection.

In this paper, we begin with a baseline color-threshold approach, followed by various enhancements: optimized color ranges, Gabor texture analysis, FFT-based frequency-domain analysis, SLIC superpixel segmentation, SVM, and a CNN-based approach. We also present an outline for video-based fire detection using optical flow.

2 Problem Definition

We focus on accurately detecting fire in static images and video frames while minimizing false negatives and false positives. Our objectives are:

- **Input:** An image or a video frame (RGB/BGR).
- **Output:** Binary decision (fire / no fire) and, optionally, bounding regions of fire.
- **Constraints:**
  - Low false alarm rate.
  - Robustness to various lighting and background conditions.
  - For video, the method should incorporate temporal cues to reduce frame-by-frame misclassifications.

3 Related Work

Conventional fire detection often relies on color thresholding in RGB/HSV spaces but can easily mistake red-like objects for fire. Texture-based methods, such as Gabor filters, have been proposed to capture the characteristic patterns of flames. Meanwhile, frequency-domain approaches (e.g., FFT) can filter out large smooth red areas. Deep learning methods (Alex-Net [1], YOLO [2], etc.) show promising results but can be data-hungry and computationally intensive. In addition,T. Celik proposed a rule-based general color model for flame pixel classification. The proposed algorithm uses the YCbCr color space, which separates luminance and chrominance more effectively than RGB or similar color spaces. [3]. B. U. Töreyin et al. proposed a flame detection algorithm that detects the quasi-periodic variations in flame boundaries using temporal wavelet transform and analyzes the color variations in moving flame-colored regions using spatial wavelet transform. [4]

3.1 Color Thresholding

Early-stage methods detect fire pixels via HSV or RGB thresholds. Although they achieve high recall, they are sensitive to red or yellow objects in the background, leading to numerous false positives.

3.2 Texture Analysis with Gabor

Gabor filters are used to extract orientation-selective texture features and can help distinguish the dynamic, irregular edges of flames from static red objects.

3.3 Frequency-Domain Analysis

Fourier transform (FFT) reveals frequency components of an image region. Active flames usually present more high-frequency components than smooth red objects (e.g., red paper or walls).

3.4 Machine Learning and Deep Learning

SVM and CNN-based methods have been proposed to learn discriminative features for fire vs. non-fire. While potentially more accurate, they often require significant labeled data and computational resources.

4 Methodology

Our study examines multiple methods, starting from a baseline color-threshold approach and incrementally adding improvements.

1. **Baseline (Color Threshold):** Convert BGR to HSV, apply a fixed range to detect red, use morphological dilate/erode, and connected-component analysis.

   In this section, Firstly, We convert the image from BGR format to HSV format. It is because that in fire detection, the HSV color space is more stable than RGB and easier to process.

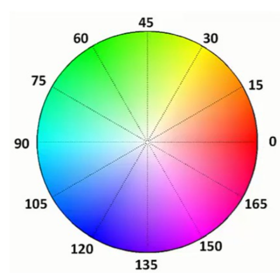

Figure 1: H channel:

   Next, We use the Thresholding method to extract specific color regions of the flame (red/orange). It generates a binary mask for subsequent morphological processing and connected region analysis. we could easily understand the change of matrix through Thresholding.(0 - 10)

$$A = \begin{bmatrix} 5 & 8 & 12 & 170 & 175 & 1 \\ 3 & 10 & 14 & 165 & 178 & 182 \\ 6 & 9 & 11 & 172 & 180 & 190 \\ 50 & 55 & 60 & 20 & 30 & 40 \\ 52 & 58 & 59 & 18 & 25 & 35 \\ 51 & 56 & 57 & 22 & 28 & 8 \end{bmatrix}$$

$$B = \begin{bmatrix} 255 & 255 & 0 & 0 & 0 & 255 \\ 255 & 255 & 0 & 0 & 0 & 0 \\ 255 & 255 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255 \end{bmatrix}$$

For Morphorlogical operations, we just need to know there are two ways we need to implement: dilation and erosion. Dilation: If any pixel within the kernel is 1, all pixels covered by the kernel will become 1 (expanding the foreground). Erosion: Only if the entire 5×5 region is 1, the center pixel will remain 1 (shrinking the foreground).

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \underset{\text{erosion}}{\overset{\text{dilation}}{\rightleftarrows}} D = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Figure 2: dilation and erosion

Connected component analysis: firstly, we need to judge if the area is a connectivity point or not. If any of the masks of the 8-connectivity or 4-connectivity neighboring points of the center point whose mask is 255 has a value of 255. The center point is a connectivity point.

$$\begin{bmatrix} 0 & 0 & 0 & 255 & 255 & 255 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 0 \\ 0 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 & 255 & 255 & 255 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 255 & 255 & 255 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 0 \\ 0 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 & 255 & 255 & 255 & 0 & 0 \end{bmatrix}$$

Figure 3: 4 connectivity and 8 connectivity

Next, suppose the point is a connectivity point and any of the neighboring points also is a connectivity point. In that case, we will expand the connectivity area, and if the size of the connectivity area exceeds the threshold, the algorithm returns the true.

The images below respectively show the original image, the grayscale image of the H channel after HSV conversion, the binary image after threshold segmentation, and finally, the binary image after morphological processing.

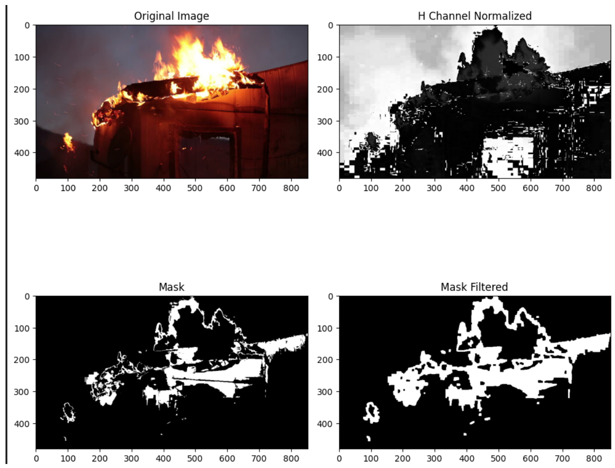

Figure 4: basline:Data Process

2. **Baseline + Optimized Color Range:** Extend red ranges (e.g., [0,10] and [160,180]) and refine morphological operations (open/close), adding an upper area threshold to filter oversized red regions.

   In this section, the most optimization is that we use the better morphological operations. After binarization, an image may contain noise or holes in the target region. To optimize the binary image, we first perform opening followed by closing. This helps reduce false detections and ensures a more complete target shape. What is Opening? What is Closing?

   Opening = Erosion → Dilation

- Mainly used to remove small noise points (e.g., isolated small red dots).
- Erosion: first shrinks or removes small noise points.
- Dilation: then restores the integrity of the main target.

Closing = Dilation → Erosion

- Mainly used to fill small gaps or holes in the target region.
- Dilation: first expands the small gaps, filling them.
- Erosion: then restores the original shape of the target boundary. [5]

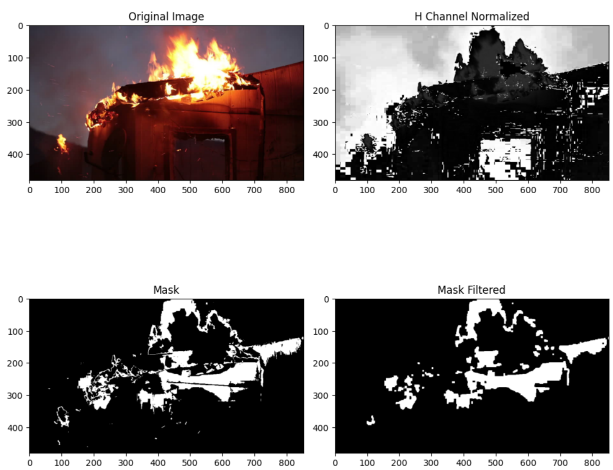

Figure 5: optimized baseline: Data Process

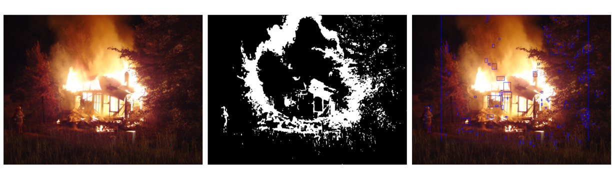

Figure 6: Left: the original image. Middle: the binarized mask after HSV thresholding. Right: detected flame regions marked with rectangular boxes on the original image.

3. **Color + Gabor Texture:** Extract Gabor features from suspicious regions to filter out non-flame textures.

   For this method, we just realize the Gabor filter. Since flames often exhibit random flickering and irregular edges, they can, in some cases, be distinguished from static red objects based on texture characteristics.

   Since the Gabor filter will change the mask value of the image, so it is important to find a suitable threshold. Otherwise, this operation even could decrease the performance of Algorithm.

4. **Multi-Scale + FFT(or FFT):** We use multi-scale detection to enable the algorithm to better adapt to flames of different sizes. The main reason for performing frequency domain analysis is to eliminate errors caused by smooth red objects, such as red clothing.

5. **SLIC Superpixel Fusion:** Segment images into superpixels, then check color, Gabor, and FFT features per superpixel, followed by morphological cleaning.

   We can reduce computational complexity by merging similar pixels, thereby decreasing the number of processing units. Additionally, extracting region-level features such as color, texture, and edges helps minimize noise and improve the accuracy of flame detection.

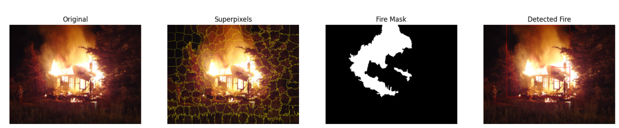

Figure 7: The picture was processed by superpixels

6. **SVM:** Train an SVM for classification. In this algorithm, we just the the pre-process way about Hop and hist/lbp.

7. **CNN:** Employ a convolutional neural network for end-to-end feature extraction and classification.

8. **Video (Optical Flow):** For consecutive frames, compute optical flow to detect dynamic regions, combine with color thresholding, and filter out static red areas.

5 Experiments and Evaluation

We tested on a dataset of 206 images for most methods, with additional 42 images for CNN evaluation. Below, we summarize each method's performance in terms of True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), Accuracy, Precision, Recall, and F1-score.

5.1 Baseline: Color Threshold

Table 1: Baseline: Color Threshold Results

Total	TP	FN	FP	TN	Acc	Prec	Rec	F1
206	110	0	83	13	0.60	0.57	1.00	0.73

Analysis: The method detects all actual flames (Recall=1.00) but yields many false positives (FP=83). Precision is 0.57, leading to an overall Accuracy of 0.60.

5.2 Baseline + Optimized Color Range

Table 2: Baseline + Optimized Color Range

Total	TP	TN	FP	FN	Acc	Prec	Rec	F1
206	104	31	65	6	0.66	0.62	0.95	0.75

Analysis: Extending the red range and improving morphological operations reduce FP from 83 to 65, at the cost of a slightly increased FN (6). Precision improves to 0.62, recall remains high (0.95).

5.3 Color + Gabor Texture

Table 3: Color + Gabor Texture

Total	TP	TN	FP	FN	Acc	Prec	Rec	F1
206	106	31	65	4	0.67	0.62	0.96	0.75

Analysis: Gabor texture analysis slightly reduces misses (FN=4), achieving a recall of 0.96, with accuracy at 0.67. In our experiment, we found that multiple parameters of the Gabor filter, including its threshold (Gabor threshold) and the FFT threshold, significantly impact the final recognition performance. However, due to the high computational cost of performing a comprehensive search for the optimal parameters, we only conducted an optimal parameter search for the Gabor threshold in the Gabor+baseline algorithm. Ultimately, we found that this optimization improved the F1-score from the default value of 0.75 to 0.78. However, in the final comparison experiment, we did not use this optimized F1-score as the benchmark for comparison, as the parameters of other algorithms had not undergone the same optimization process, ensuring the fairness of the experiment.

5.4 Multi-sacle FFT(FFT)

Table 4: FFT

Total	TP	TN	FP	FN	Acc	Prec	Rec	F1
206	104	34	62	6	0.67	0.63	0.95	0.75

Analysis: Multi-scale analysis helps detect smaller or distant flames; FFT filters smooth red regions. Precision rises to 0.63, recall remains at 0.95. Actually, if we just use the FFT, we don't change other parameters, we even could get a higher f1-score(0.77), so finding the suitable scale list is an important task!

5.5 Superpixel Segmentation

Table 5: Superpixel

Total	TP	FN	FP	TN	Acc	Prec	Rec	F1
206	108	2	56	40	0.718	0.659	0.982	0.788

Analysis: The SLIC-based method exhibits a high recall of 0.982, with improved accuracy (0.718). Precision is 0.659, showing fewer false positives compared to earlier methods. For superpixel-based methods, when applying Gabor+superpixel, FFT+superpixel, Gabor+FFT+superpixel, or superpixel alone, we found that this approach introduces greater uncertainty. In certain cases, it may improve the F1-score, but if the parameters are not set appropriately, it can lead to a significant performance decline. Moreover, the superpixel operation itself incurs a substantial computational overhead, and its mask calculation is highly complex and time-consuming. Finding the appropriate tuning parameters also requires considerable effort. Compared to other methods, the performance improvement brought by superpixel-based approaches is not particularly significant. Therefore, we do not recommend using this method.

5.6 SVM

Table 6: SVM

Total	TP	FP	TN	FN	Acc	Prec	Rec	F1
206	110	92	4	0	0.553	0.545	1.000	0.705

Analysis: While recall is perfect (1.000), the false positives are high (FP=92), causing a precision drop to 0.545 and overall accuracy of 0.553. In our task, SVC has very high requirements for data preprocessing because, unlike the previous algorithms, we did not apply feature processing operations such as Gabor and FFT in this case. We found that its performance was even worse than the baseline. Additionally, in this task, we were unable to effectively incorporate PCA for dimensionality reduction due to its high computational cost. Compared to the subsequent CNN-based methods, SVC offers almost no advantages, and therefore, we do not recommend using it.

5.7 CNN

Analysis: CNN achieves a high F1-score (0.816) on this small test set, indicating strong overall performance. However, it demands more training data and higher computation.

Table 7: CNN Evaluation

Acc	Prec	Rec	F1
0.786	0.690	1.000	0.816

Although CNNs require longer training time, their classification performance far surpasses traditional basic algorithms, such as SVC, Gabor filtering, FFT analysis, etc. Notably, in our experiments, we did not even perform additional preprocessing (such as Gabor filtering or FFT transformation) on the input data to enhance flame features, yet CNNs were still able to automatically learn and extract effective features, achieving superior classification performance.

While this experiment did not adopt more advanced deep learning architectures like Movenet or YOLOv5, CNNs still significantly outperformed traditional CCA algorithms. This demonstrates that CNNs have a stronger end-to-end feature learning capability, enabling them to directly extract valid information from raw pixel data without relying on manually designed feature engineering.

Compared with traditional methods, the main advantages of CNNs are: [5]

- Automatic Feature Learning: No need for manual feature extraction, avoiding dependency on parameters in methods like Gabor and FFT.
- Strong Adaptability: CNNs can still learn effective recognition patterns even without specific flame feature preprocessing.
- Higher Classification Performance: Even without using more complex architectures (such as YOLOv5 and Movenet), CNNs still significantly outperform in terms of F1 score.

This fully illustrates the strength of CNNs—even a relatively simple CNN model performs better than traditional methods in the flame detection task. Although CNNs have relatively high computational costs, their end-to-end feature extraction capabilities make their advantages in complex visual tasks irreplaceable. [5]

5.8 Analysis

Now, we could summarized the results:

Although, in theory, the classification ability of the first few algorithms gradually improves, there is little difference in actual performance, which may be due to the small size of the dataset.

Another important reason is that both Gabor and DFT operations require very well-tuned threshold values; otherwise, it is difficult to achieve optimal performance and may even lead to adverse effects. Depending on different datasets, multiple experiments should be conducted to obtain the best results.

Addmittedly, there are also many issues with our experiment, as we did not actually fine-tune the parameters of each algorithm. As a result, we did not obtain the optimal results for each model but rather a relative outcome.

We found that, compared to traditional digital image processing algorithms and machine learning algorithms, deep neural networks demonstrate significant advantages in image recognition tasks. They have a lower dependency on image preprocessing and typically require minimal parameter tuning (especially when using pre-trained models) to achieve high classification performance.

Moreover, another advantage is that CCA recognition based on traditional digital image processing methods relies heavily on parameter tuning, making it difficult to directly optimize all parameters to achieve the best recognition performance.

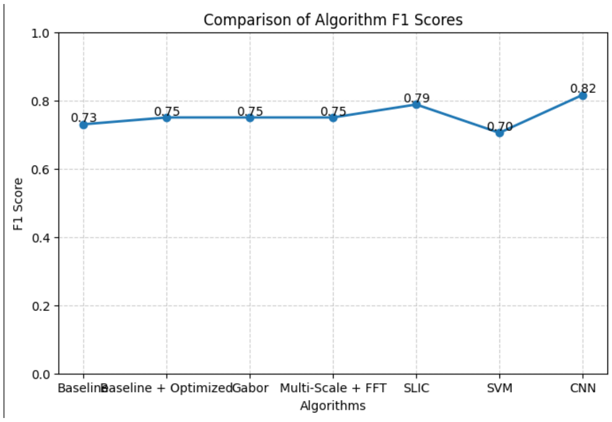

Figure 8: comparison F1 scores

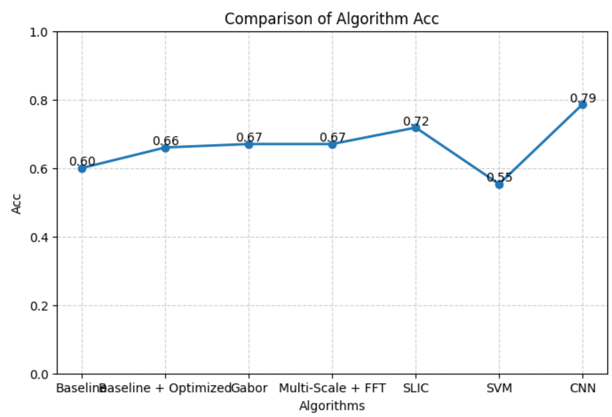

Figure 9: Accu

5.9 Video Fire Detection (Optical Flow)

For video scenarios, each pair of frames undergoes optical flow calculation (e.g., Farneback), filtering out small magnitudes as noise. The color thresholding (HSV red range) and the motion mask are combined via logical AND. Subsequent morphological operations and connected-component analysis yield candidate flame regions. Although feasible, practical issues like camera shakes and global motion require further considerations.

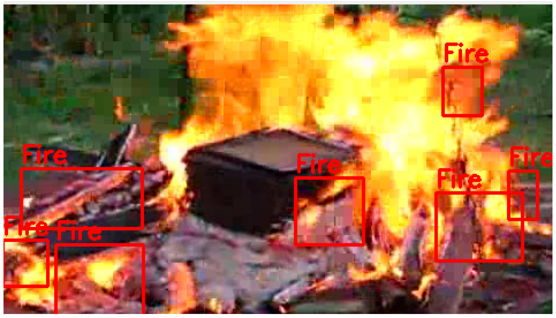

Figure 10: optical algo for video

6 Conclusion

Our experiments show that:

- The baseline color-threshold approach achieves high recall but yields many false positives.
- Enhanced color ranges, morphological refinement, and multi-feature integration (texture, FFT, superpixels) help boost precision while keeping recall high.
- Multi-scale detection benefits small or distant flames, and frequency-domain analysis filters out smooth red objects.
- CNN methods demonstrate potential for further accuracy improvements, albeit with higher complexity.
- For video, optical flow can capture the dynamic nature of flames, but additional measures are needed to address global motion and camera instability.

Overall, combining color, texture, frequency, and motion cues in a multi-modal framework offers a promising solution for robust fire detection. Future work may involve lightweight deep networks and advanced spatiotemporal features for real-time deployment. In addition, unfortunately, we don't use the mentioned algorithm in the realted work part, such as Alexnet and Yolov. Moreover, in recent years, fire detection algorithms, whether deep learning-based or morphology-based, have focused on audio-based fire detection or fire disaster recognition. These algorithms are more advanced. In our project, we just use the FFT, basing on previous work, we know that the SWT is better.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1097–1105. [Online]. Available: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-netw

[2] H. Xu, B. Li, and F. Zhong, "Light-yolov5: A lightweight algorithm for improved yolov5 in complex fire scenarios," *Applied Sciences*, vol. 12, no. 23, p. 12312, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/23/12312

[3] T. Celik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Safety Journal*, vol. 44, no. 2, pp. 147–158, 2009.

[4] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Çetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865505001819

[5] OpenAI, "Chatgpt: Gpt-4," 2023. [Online]. Available: https://openai.com