

Fire Smoke Identification Based on Genetic Algorithm Optimizing BP Neural Network

Qian Qi¹, Anton Louise De Ocampo^{2,*}, Yanwu Hong³

Submitted: 10/03/2024 Revised: 25/04/2024 Accepted: 02/05/2024

Abstract: With the continuous improvement of people's living standards, the frequency of home fires is increasing. This paper proposes a method for identifying smoke-type fires involving image processing of monitoring screens. Suspected smoke areas are roughly identified based on the gray value of images. Then the local binary variance and the relative energy of high and low frequencies in suspected smoke areas are extracted using the LBP algorithm and wavelet transform algorithm. Using the collected fire smoke images, extract corresponding feature data using the algorithm in this article, and manually calibrate the fire label to obtain the corresponding training data. BP neural network is employed in this paper for smoke and fire recognition. However, under limited training data, the recognition results using only the neural network are poor, with experimental results showing a prediction accuracy of only 64.5% for the pure BP neural network. Given this situation, this paper utilizes a genetic algorithm to optimize the internal topology of the BP neural network. By encoding the weight thresholds of the network structure into binary code as population individuals of the genetic algorithm and performing continuous iterative optimization, experimental results demonstrate that the prediction accuracy of fire recognition by the optimized BP neural network is improved to 94%.

Keywords: Genetic Algorithm, BP Neural Network, Home Fire Recognition, Intelligent Recognition System.

1. Introduction

With the increasing population density, especially in urban areas, the problems brought about by population density are becoming increasingly apparent. The decrease in per capita land area means accommodating more people in the same space. Home fires can be triggered by various reasons such as electrical faults, cooking accidents, smoking, ignition sources, electrical wiring issues, and more. Regardless of the cause, fires can spread rapidly, posing a threat to home safety.

In the field of fire detection, numerous studies have focused on enhancing accuracy by analyzing the color characteristics and dynamic behavior of flames. Calderara et al. combined smoke color and energy features, employing a novel Bayesian method to identify smoke regions in images, and used a Gaussian mixture model to construct an image energy statistical model, thereby improving smoke detection effectiveness [1]. Although Haar-like features are commonly employed together with other feature extractor such as a cascaded Haar [2], Kim et al. extracted Haar-like features and statistical features from images, formulating smoke recognition rules based on an AdaBoost classifier [3]. Yuan F et al. improved the detection of small smoke

targets in forest fires by using Haar-like features and statistical features, coupled with step search techniques and a dual-threshold adaptive enhancement algorithm [4]. Zhao et al. analyzed motion regions extracted from adjacent frames of smoke videos and used the CS Adaboost algorithm to process spatial color, motion, flicker, and image energy features of smoke, proposing a method for early detection of thin smoke in forests [5]. Ye et al. developed an algorithm that detects both smoke and flames in static videos, enhancing detection stability through complex steps including morphological processing and chaotic motion estimation [6]. In traditional machine learning, researchers relied on manually extracted flame features and machine learning classifiers to identify flames. Frizzi and Zhang et al. developed efficient end-to-end fire detection systems that significantly improved detection efficiency and accuracy. These systems included training with deep generative adversarial networks combined with spatiotemporal flame evolution data, further reducing false alarm rates [7][8]. Chi Yuan et al. identified the distribution characteristics of smoke pixels in the RGB color space and input these features into a Kalman filter and morphological algorithm to determine the presence of smoke in images [9]. Zeng et al. constructed a forest fire smoke recognition model based on Faster RCNN, SSD, and R-FCN algorithms and conducted comparative analysis using public datasets. The experimental results showed that the SSD model combined with MobileNet had a faster smoke recognition speed but lower accuracy [10]. Lee et al. proposed a robust Faster RCNN algorithm based on local and global features of smoke images [11]. Saima Majid et al. trained a shallow

¹ College of Engineering, Batangas State University, Alangilan Campus.
Batangas City 4200, Philippines
ORCID ID: 0009-0000-2286-1424

² College of Engineering, Batangas State University, Alangilan Campus.
Batangas City 4200, Philippines
ORCID ID: 0000-0002-6280-6259

³ College of Engineering, Batangas State University, Alangilan Campus.
Batangas City 4200, Philippines
ORCID ID: 0000-0001-8714-5832

* Corresponding Author Email: antonlouise.deocampo@ieee.org

neural network model using an attention-based CNN and attention mechanisms to improve model performance, achieving more accurate visualization and localization of smoke in forest fire images [12]. These advancements not only mark significant technological leaps but also provide powerful tools for early fire detection.

However, fires can occur for various reasons. In situations where flames are not visible, smoke continues to accumulate, eventually evolving into a tragic fire incident. The aforementioned flame detection algorithms and methods become ineffective. Therefore, this paper proposes an algorithm for identifying and detecting smoke-type fires. By extracting feature data such as local binary variance and relative energy of high and low frequencies from suspected smoke areas, the accuracy of smoke detection and identification is enhanced using a method that optimizes BP neural networks with genetic algorithms.

2. Research Methodology

The process of intelligent monitoring of home fires proposed in this paper is illustrated in Figure 1. In cases where flames are not detected, further detection and identification of smoke are necessary. This paper specifically elaborates on the process of smoke-fire recognition as in Figure 1:

2.1. Extraction of feature data from fire smoke images

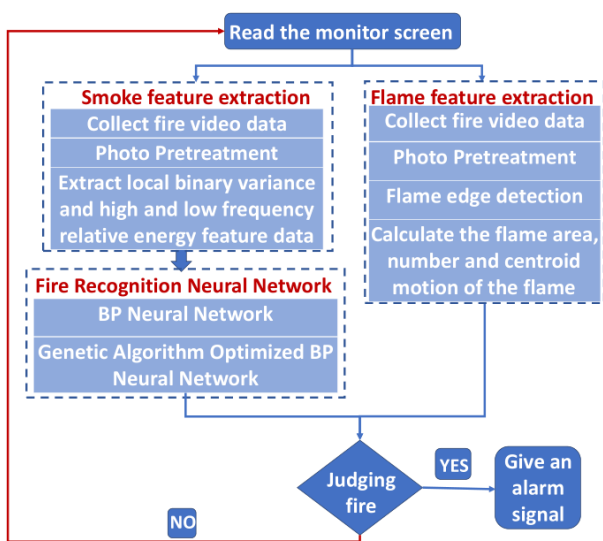


Fig. 1. Diagram of the Intelligent Recognition System for Home Fires

2.1.1. Collection of training data for fire smoke images

The fire smoke training data mainly comes from two aspects: first, laboratory video data collection, and second, monitoring data collection of actual fire scenes.

1. Laboratory video data collection focuses on simulating various types of fire scenarios by controlling environmental conditions, to obtain diverse flame and smoke image data. The laboratory video collection scene is illustrated in Figure

2.



Fig.2. Laboratory Data Collection Scene

2. Data collection in actual fire scenes focuses on capturing the characteristics of flames and smoke under real fire conditions, enhancing the model's adaptability and recognition accuracy to real-world scenarios. The data collection scene at fire sites is depicted in Figure 3.



Fig. 3. Fire Scene Data Collection Scene

2.1.2. Image preprocessing

After obtaining the initial fire training data, data quality is improved through image preprocessing steps, including image denoising, color space conversion, gray-scale processing, and binarization. Image noise reduction involves using median filtering techniques to reduce random noise during the image capture process. The color space conversion transforms the RGB color space into the HSI color space, which is more suitable for extracting flame and smoke features. Subsequently, further simplification of the image data is achieved through gray-scale processing and binarization, highlighting the morphological features of flames and smoke.

1. Image filtering

Median filtering is a non-linear smoothing technique. Its basic principle is to sort the pixel values around the central point of the image by magnitude, replacing the value with the median of that domain. Its major advantages lie in its simplicity, speed, and effective noise reduction. It significantly eliminates isolated noise points. In this paper, a 3×3 template is used for sliding [13].

2. RGB to HSI conversion

HSI color space, derived from the human visual system, describes colors using Hue, Saturation (or chroma), and Intensity (or brightness). The HSI color space can be described using a cone-shaped model. While this cone

model to describe the HSI color space is quite complex, it effectively illustrates variations in hue, brightness, and color saturation [14].

The conversion from RGB to HSI model involves transforming from a unit cube based on the Cartesian coordinate system to a double cone based on cylindrical coordinates. The conversion formula is as in:

$$r = \frac{R}{R + G + R \cdot g} = \frac{G}{R + G + B}, b = \frac{B}{R + G + B}$$

$$h = \cos^{-1} \left\{ \frac{0.5[(r-g)+(r-b)]}{[(r-g)^2+(r-b)(g-b)]^{1/2}} \right\} \quad h \in [0, \pi] \text{ for } b \leq g \quad (1)$$

$$h = 2\pi - \cos^{-1} \left\{ \frac{0.5[(r-g)+(r-b)]}{[(r-g)^2+(r-b)(g-b)]^{1/2}} \right\} \quad h \in [\pi, 2\pi] \text{ for } b > g$$

$$s = 1 - 3 \cdot \min(r, g, b): s \in [0,1] \quad (3)$$

$$I = (R + G + B) \neq (3 \cdot 255): i \in [0,1] \quad (4)$$

$$H = h \times 180/\pi: S = s \times 100: I = i \times 255$$

3. Filtering of suspected smoke areas

Based on the color characteristics of smoke, we utilize the HSI color matrix and RGB color matrix to filter suspected smoke areas in the monitoring footage[15]. Firstly, filtering is performed in the HSI color space using the formula shown in (5), and then filtering is conducted in the RGB color space using formula (6):

$$80 \ll I_{(xy)} \ll 200 \quad (5)$$

$$\max(R, G, B) - \min(R, G, B) \ll 20 \quad (6)$$

By using the above formulas, the approximate identification of suspected smoke areas in the monitoring footage is achieved, and non-suspected smoke areas are set to black, for subsequent feature data extraction.

4. Image Morphological Processing

Image morphological processing extracts and separates regions of interest by using dilation and erosion operations. Dilation is used to fill small holes, connect neighboring elements, or enlarge boundaries, while erosion "erodes" objects in the image to remove small objects, break object bridges, or shrink boundaries [16]. This paper primarily employs the closing operation, which involves dilation followed by erosion.

2.1.3. Extraction of Smoke Features

Extraction of smoke features focuses on analyzing gray distribution, texture, and motion patterns. It identifies

smoke features from subtle texture variations using the Local Binary Patterns (LBP) algorithm and wavelet transform technique.

Smoke exhibits characteristics of uniform gray distribution and slow diffusion outward. After preprocessing the images, we extract the texture images of smoke using the LBP detection algorithm, then proceed to extract local binary variance feature data and relative energy feature data in high and low frequencies.

1. Extraction of Local Binary Variance Feature Data

(1) LBP algorithm

The basic LBP operator can only cover a fixed region and is not suitable for textures of different scales. Therefore, extending the LBP operator to arbitrary-sized circular neighborhoods achieves gray and rotation invariance, adapting to texture features of different scales [17]. The improved LBP operator can define multiple evenly spaced sampling points within a circular neighborhood of any radius, as shown in Figure 4.

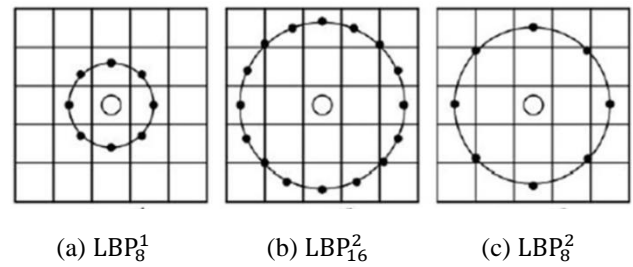


Fig. 4. Circular LBP Operator

Figure 4-(a) shows the LBP operator with a radius of 1 and sampling points of 4, denoted as LBP_8^1 ; Figure 4-(b) shows the LBP operator with a radius of 2 and sampling points of 16, denoted as LBP_{16}^2 ; Figure 4-(c) shows the LBP operator with a radius of 2 and sampling points of 8, denoted as LBP_8^2 . The operator radius and sampling points can be any number. The calculation formula for sampling points is as follows :

$$x_p = x_c + R \cos \frac{2\pi p}{P} \quad (7)$$

$$y_p = y_c - R \sin \frac{2\pi p}{P} \quad (8)$$

In the formula:

- (x_p, y_p) represents the coordinates of the p-th sampling point.
- (x_c, y_c) represents the coordinates of the window center.
- p denotes the p-th sampling point.
- P denotes the total number of sampling points.
- R denotes the radius of the neighborhood.

Since the sampling points in the circular LBP algorithm are distributed along the circumference, it cannot guarantee that

the coordinates of each sampling point are integers. Therefore, bilinear interpolation is needed to address this issue. For points with non-integer coordinates, their coordinates are rounded up and down, denoted as x_0, x_1, y_0, y_1 , resulting in four coordinates: $(x_0, y_0), (x_1, y_0), (x_0, y_1)$, and (x_1, y_1) . The calculation is done according to the following formula9:

$$f(x_p, y_p) = [x_1 - x_p \quad x_p - x_0] \times \begin{bmatrix} f(x_0, y_0) & f(x_0, y_1) \\ f(x_1, y_0) & f(x_1, y_1) \end{bmatrix} \begin{bmatrix} y_1 - y_p \\ y_p - y_0 \end{bmatrix} \quad (9)$$

For pixels with integer coordinates, calculating them using the above formula can significantly reduce programming difficulties. The final LBP image is shown in Figure 5:



(a) The original image (b)The LBP image

Fig.5. Original Image and LBP Image

(2) Local Binary Variance Feature Data Extraction

The LBPV operator combines local variance as weights to extract local variance features from images, addressing the issue of contrast being ignored when describing image spatial structures using the LBP operator. The LBP operator represents the texture frequency of the image, while the LBPV operator introduces variance to reflect contrast. High variance regions indicate significant texture variations, thus providing more detailed texture information. The formula10 for variance is:

$$VAR = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - u)^2 \quad (10)$$

In the equation:

$$- u = \frac{1}{P} \sum_{p=0}^{P-1} g_p$$

- g_p represents the gray value of the central pixel.

- P represents the number of neighborhood pixels.

2. Extraction of High-Low Frequency Relative Energy Feature Data

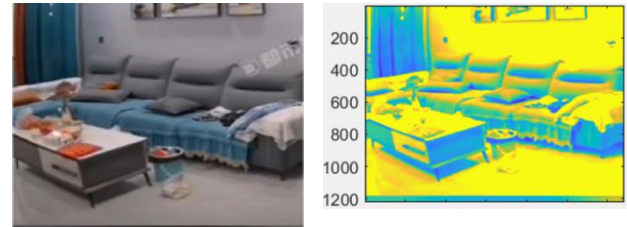
(1) Wavelet Transform Feature Data Extraction

The wavelet transform algorithm decomposes signals at different scales to obtain a series of approximation and detail signals, representing the characteristics of the signals at various scales. Analyzing these features helps in understanding the intrinsic structure and developmental trends of the signal. The reconstructed image using wavelet transform is shown in Figure 6. Given the scaling function $\varphi(x)$ and the wavelet function $\Psi(x)$ at a given scale, a two-

dimensional scaling function and three two-dimensional wavelet functions are combined, as shown respectively in equations (11) and (12):

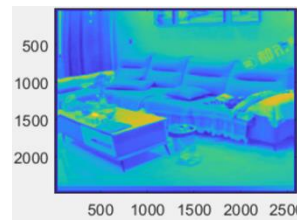
$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (11)$$

$$\Psi^H(x, y) = \Psi(x)\Psi(y), \Psi^V(x, y) = \Psi(x), \Psi(y)\Psi^D(x, y) = \Psi(x)\Psi(y) \quad (12)$$

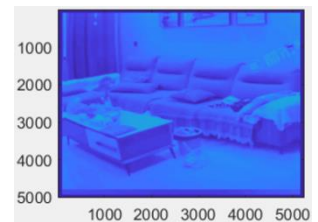


(a) Original image

(b) Single Wavelet Reconstructed Image



(c) Double Wavelet Reconstructed Image



(d) Triple Wavelet Reconstructed Image

Fig. 6. Wavelet Reconstructed Image Results

(2) Extraction of High-Low Frequency Relative Energy Feature Data

Extraction of high-low frequency energy features is an important step in signal processing and pattern recognition. By analyzing the energy distribution of signals across different frequency ranges, we can understand the frequency characteristics and structures of signals. A commonly used method for this purpose is wavelet transform. Wavelet transform decomposes signals into components at different scales and frequencies. High-frequency wavelet coefficients correspond to rapidly changing parts of the signal, while low-frequency wavelet coefficients correspond to slowly changing parts of the signal. By calculating the energy within each frequency band (scale), we can obtain features of high and low-frequency energy.

2.2. Determination and Optimization of Neural Networks

This paper aims to use a BP neural network to identify smoke-type fire incidents, based on two extracted feature data: local binary variance and high-low frequency relative energy as input parameters, and fire incident labels as output parameters [18]. Due to limited training data, the BP neural network is prone to local convergence, resulting in large prediction errors in the final training network structure. Given that the initial weights and thresholds of the BP neural

network have a significant impact on the final training results, this paper adopts a genetic algorithm to optimize the weights and thresholds of the BP neural network. The individual with the highest final prediction accuracy is selected as the network structure model for smoke and fire incident recognition in this paper. The flowchart of the genetic algorithm optimizing BP neural network is shown in Figure 7.

The above flowchart mainly consists of three modules:

- Creation of the topology structure of the smoke recognition neural network.
- Training of the neural network and calculation of prediction errors.
- Iterative optimization of initial weights and thresholds using genetic algorithm after binary encoding.

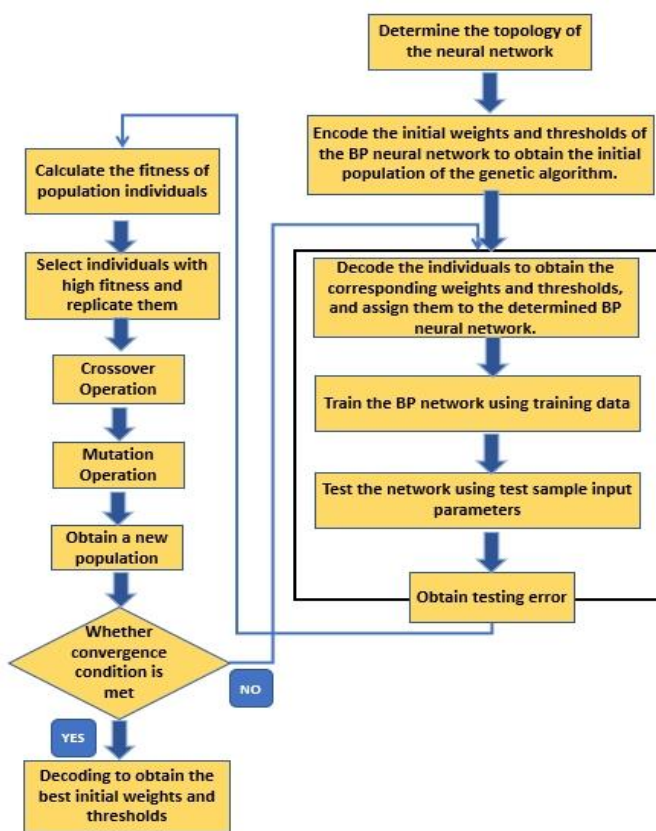


Fig. 7. Flowchart of Genetic Algorithm Optimizing BP Neural Network

2.2.1. Creation of the Topology Structure of the Smoke Recognition Neural Network

1. Creation of the Neural Network

Using a three-layer neural network model (input layer, hidden layer, and output layer)[19], the formula for the relationship between the number of neurons n_2 in the hidden layer and the number of neurons n_1 in the input layer is shown in equation (13):

$$n_2 = 2 \times n_1 + 1 \quad (13)$$

In the fire smoke incident recognition system model, the input parameters are local binary variance and high-low frequency relative energy, thus the number of neurons in the input layer is $n_1 = 2$. The output parameters are the fire recognition result (0/1), so the number of neurons in the output layer is $n_3 = 1$. Using formula (13), we can calculate that the number of neurons in the hidden layer is $n_2 = 5$. Therefore, the structure of this neural network is 2-5-1. From this structure, it can be inferred that the total number of connection weights within the neural network is: $2 \times 5 + 5 \times 1 = 15$, and the number of thresholds is: $5 + 1 = 6$. The number of parameters within this neural network is shown in Table 1.

Table 1 Topological Structure Data of Smoke Recognition System

The input layer and hidden layer connection weights	10
The number of hidden layer thresholds	5
The number of connection weights between the hidden layer and the output layer	5
The number of output layer thresholds	1

2. Initialization of Neural Network Weights and Thresholds

According to the neural network topology data in Table 1, there are a total of 21 ($10+5+5+1=21$) weights and thresholds, the initial weights and thresholds of the neural network are randomly set between $[-0.5, 0.5]$. Since the initial weights and thresholds have a significant impact on the final convergence results of neural network training, all weights and thresholds are concatenated into a single column and used as individuals in the genetic optimization algorithm. This string is shown in Figure 8.

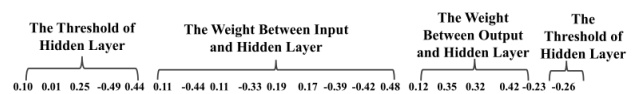


Fig. 8. String of Weights and Thresholds

2.2.2. Training of the neural network and calculation of prediction errors

Neural network training involves continuously adjusting weights and thresholds to reduce output errors, while automatically adjusting the learning rate. The training is conducted using the *transig* function and an adaptive learning rate method. The parameters are set as follows: training for 1000 epochs, the learning goal of 0.01, the initial learning rate of 0.05, and an adaptation parameter of 1.05. In this study, a total of 86 sets of smoke and fire incident images were collected. Among them, 40 sets were randomly selected as training data for the neural network, 5 sets were randomly selected for prediction error calculation, which

primarily serves as the fitness function for the subsequent genetic algorithm. The remaining 31 sets were used for calculating the accuracy of smoke recognition.

The training images are subjected to feature extraction, resulting in corresponding feature data of 40 sets of local binary variance and high-low frequency relative energy, along with their respective fire incident labels. The neural network is trained based on the set training parameters. The feature data of the test samples are inputted into the trained neural network, and the prediction errors of the neural network are calculated based on the predicted results and actual fire incidents. The formula for calculating prediction error is shown in equation (14):

$$\varepsilon = \sum_{i=1}^5 |T_{1i} - T_{2i}|/5 \quad (14)$$

In the equation:

- T_{1i} represents the predicted fire incident result.
- T_{2i} represents the actual fire incident result.
- i denotes the sequence number of the prediction error calculation sample.

2.2.3. The training Image Labeling

The neural network topology structure is based on the input parameters of local binary variance and high-low frequency relative energy, obtained through image preprocessing and the LBPV algorithm, as well as wavelet transform. The output parameter represents whether a fire incident occurs or not. This is manually labeled, where a fire is labeled as 1 and no fire is labeled as 0 [20].

2.2.4. Model Creation for Genetic Algorithm Optimized BP Neural Network

1. Population initialization

Based on the smoke and fire feature data created above and the neural network topology for smoke and fire, the string formed by the initialized weights and thresholds of the neural network serves as individuals in the genetic algorithm population. The initialized weight and threshold strings are encoded into binary, with each value encoded using 10 binary bits. The length of the string representing weights and thresholds is 21 bits, so after decoding, each individual is a binary string of length 210 bits. The binary representation of an individual is illustrated in Figure 10, where:

- Bits 1 to 100 represent the decoding of connection weights between the input layer and the hidden layer.
- Bits 101 to 150 represent the decoding of the thresholds of the hidden layer.
- Bits 151 to 200 represent the decoding of connection weights between the hidden layer and the output layer.
- Bits 201 to 210 represent the decoding of the thresholds of

the output layer.

The population size of the genetic algorithm is set to 50. Therefore, the binary string representation of the population individuals and the results of the population initialization are illustrated in Figure 9 and Figure 10.

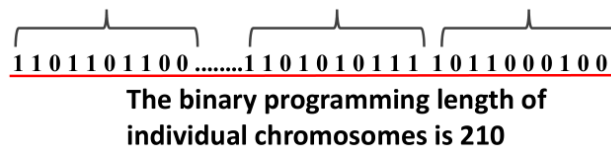


Fig. 9. Schematic Representation of Population Individual Binary Strings

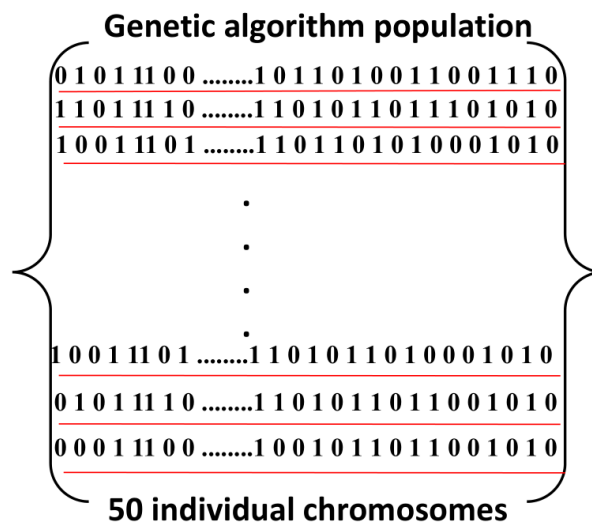


Fig. 10. Population Initialization Illustration

2. Fitness Function

The fitness value of a population individual is an important criterion for assessing the excellence of an individual in the genetic algorithm optimization process [21]. After each individual is decoded, the decoded values are sequentially assigned to the newly created BP neural network, and training is conducted using the training data. The error test data from groups 41-45 is then input into the trained network, and the testing error is used as the fitness function for that individual, as shown in equation (15):

$$f_i = \varepsilon_i \quad (15)$$

In the equation:

- f_i represents the fitness value corresponding to the individual.
- ε_i represents the prediction error of the compiled neural network corresponding to the individual.

3. Optimization Operators of Genetic Algorithm

Crossover Operator and Mutation Operator: After calculating the fitness of each individual using the above formula, crossover and mutation operators are applied iteratively to ensure the diversity of the population. The

process of crossover and mutation operators is illustrated in Figure 11.

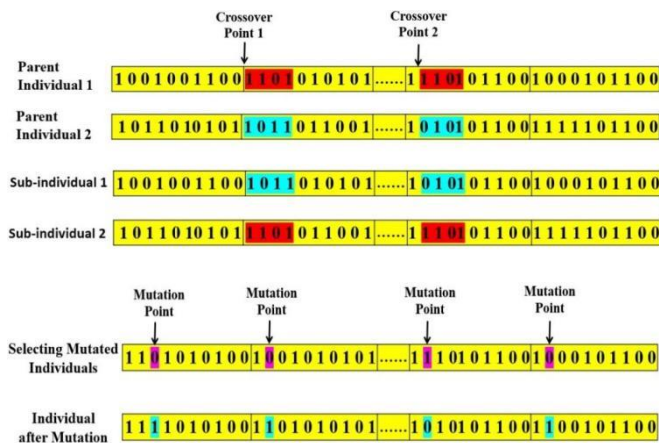


Fig. 11. Flowchart of Crossover and Mutation Operators

2.2.5. Genetic Algorithm Parameter Settings

The genetic algorithm utilizes the genetic toolbox developed with Sheffield, where the toolbox itself provides the code for operations such as selection, crossover, and mutation. The fitness function is autonomously coded based on the formula. The internal optimization parameter settings for the multi-population genetic algorithm are shown in Table 2.

Table 2 Internal Optimization Parameter Settings for Multi-Population Genetic Algorithm

Population Size	50
Maximum Iteration Count	50
Individual Length	210
Crossover Probability	0.7
Mutation Probability	0.02
Selection Parameters	0.95

3. Experimental Validation

According to the method flow described above, the following experimental results are extracted.

3.1. Smoke Feature Extraction Results

When flames cannot be detected, further smoke detection is required in the monitoring scene, as illustrated in Figure 12.

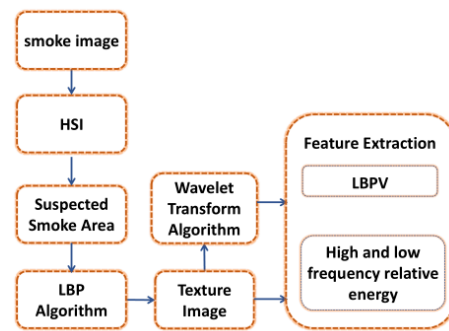


Fig. 12. Fire Smoke Feature Extraction Process

3.1.1. Preprocessing Results of Fire Smoke Images

From the original image to the gray image, and then to the denoised filtered image, each step aims to clearly display the presence of smoke and reduce false alarms. This is crucial for designing effective fire alarm systems, as it can improve the accuracy of early fire detection. The results are shown in Figure 13.



Fig. 13. Image preprocessing of current frame image

3.1.2. RGB to HSI Color Space Conversion and Determination of Suspected Smoke Areas

As shown in Figure 14, suspected smoke areas are initially filtered based on the I matrix of the HSI image, which better separates the color characteristics because HSI is closer to human color perception. In this image, hue is used to represent the type of color, saturation indicates the purity of color, and brightness reflects the lightness of the color. This processing is helpful for identifying and analyzing specific color features, such as flames or smoke. Regions with I values exceeding 220 or falling below 80 are set to black, preliminarily screening out suspected smoke areas, as shown in Figure 15. The image displays the areas suspected of containing smoke, separating the smoke from other areas. Smoke typically appears in images as regions with specific textures and gray levels, different from the pixel intensities of other areas.

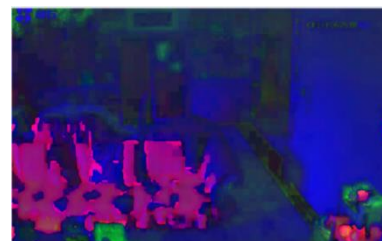


Fig. 14. Conversion from RGB Image to HSI Image

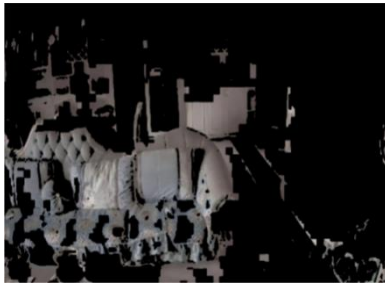


Fig. 15. Suspected Smoke Areas

3.1.3. Calculation Results of Local Binary Variance

Through the LBP algorithm, the images of suspected smoke areas are processed to obtain corresponding texture images. The identification of suspected smoke areas is shown in Figure 16. LBP can highlight the irregular textures of smoke, contrasting them with other areas. This helps the system identify smoke or fire, especially in fire detection systems, by assisting in detecting abnormal patterns in the images and issuing early warnings.



(a) Original image

(b) Suspected Smoke Area

Fig. 16. The result of LBP Algorithm Processing on The Suspected Smoke Area

Based on the texture images of suspected smoke areas obtained from the figure 16, the Local Binary Pattern Variance (LBPV) algorithm is applied to calculate the local binary variance of the images. Subsequently, the vertical mean of the local binary variances is computed, which serves as the feature data for smoke recognition. These computed LBP variance values, as shown in Table 3, represent the degree of local texture variation. Variance values close to 0 indicate minimal local texture variation, while higher values (e.g., 6.81, 6.1, 6.62, 6.83) indicate significant texture variation in that area. From the numerical perspective, most regions exhibit LBP variance values close to 0, suggesting minimal texture variation in those areas. However, regions indexed 15-21, 22-28, 36-42, and 43-49 show significant local texture variation, indicating the richest or most complex texture areas in the image, and potentially representing suspected smoke or other abnormal conditions.

Table 3 Vertical Mean of Local Binary Variance

Number	Local binary variance value						
1-7	0	0.30	0	0	0.71	0	5.38
8-14	0	0	0	0	0.66	0	4.79
15-21	0	6.810	0	0	0	0	0
22-28	0	0.14	0	6.1	0	6.06	0
29-35	0.46	0	0	0	0	0	0
36-42	0	0	4.61	0	6.62	0	0.6
43-49	0	0	0	0	0	0	6.83
50-56	0	0.49	0	0	0	0	0.6
Local binary variance	8242						

3.1.4. Calculation Results of High-Low Frequency Relative Energy

The Uniform Distribution of Low-Frequency Coefficients in the Suspected Smoke Area after Wavelet Transform, as shown in Figure 17, may indicate consistent texture or color characteristics of the background. Meanwhile, significant texture differences are exhibited in the high-frequency coefficients, suggesting the presence of smoke. Coefficient images in horizontal, vertical, and diagonal directions are depicted in Figure 18 (a), (b), and (c). Any high-frequency fluctuations occurring in these areas may indicate smoke interference. Applying these findings in practice offers a method to enhance the accuracy of early fire detection, aiding in the development of more sensitive and timely fire alarm systems. The application of this technique is not limited to smoke identification; it is equally applicable in various image processing domains, including but not limited to image enhancement, image de-noising, and feature extraction, demonstrating the potential of wavelet transforms as a versatile tool.

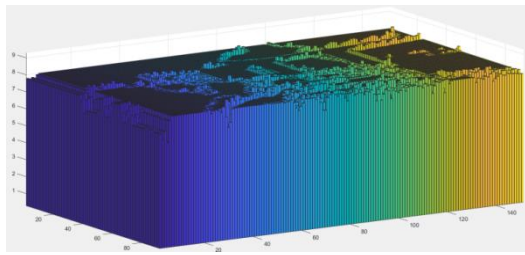
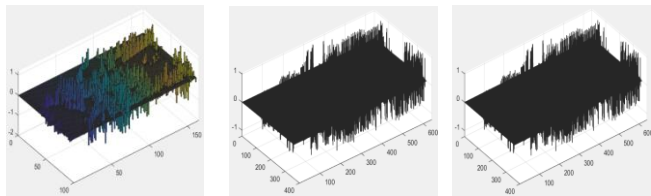
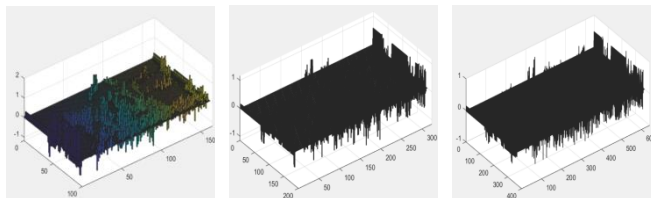


Fig. 17. Low-frequency Coefficient Result Image



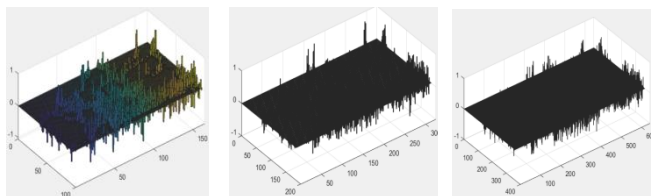
Horizontal Level 1 Horizontal Level 2 Horizontal Level 3

(a) High-frequency Horizontal Directional Coefficient Image



Vertical Direction 1 Vertical Direction 2 Vertical Direction 3

(b) High-Frequency Vertical Direction Coefficient Image



Oblique Direction 1 Oblique Direction 3 Oblique Direction 3

(c) High-Frequency Oblique Direction Coefficient Image

Fig. 18. High-Frequency Coefficient Result Image

By calculating the low-frequency energy coefficient and high-frequency energy coefficient, corresponding high-frequency energy N_h and low-frequency energy N_l are computed. The relative energy value of high and low frequencies N_r is then calculated using the formula $N_r = N_h / N_l$. These values are sequentially used as feature data for suspected smoke areas. The calculation results are shown in Table 4.

Table 4 Relative Energy Feature Values of High and Low Frequencies

N_h	N_l	N_r
6679.4	1002935	0.0068

3.2. Neural network training and error calculation results

3.2.1. Obtain training data for recognizing smoke neural networks

To obtain training images and corresponding feature data for the neural network of real-life smoke and fire recognition scenarios, a segment of monitoring video containing smoke and fire incidents is selected. Forty frames are randomly chosen from it as training samples, and two frames are selected as test samples for training error. The specific information of the monitoring video for smoke and fire incidents is provided in Table 5.

Table 5 Information of Smoke and Fire Incident Monitoring Video

Duration	107 s
Contains Flame Situation	NO
Contains Smoke Situation	YES
Total Frames	2157







The process described above is used to extract local binary variance and high-low frequency relative energy feature data from the randomly selected 40 frames. The results are shown in Table 6 .

Table 6 Extraction Table of Local Binary Variance and High-Low Frequency Relative Energy Feature Data

Frame Number	Local Binary Variance	High-Low Frequency Relative Energy	Frame Number	Local Binary Variance	High-Low Frequency Relative Energy	Frame Number	Local Binary Variance	High-Low Frequency Relative Energy
1	1071	0.0433	27	1210	0.0433	652	1240	0.0448
3	1078	0.0433	29	1234	0.0435	702	1175	0.0449
5	1121	0.0437	102	1196	0.0436	752	1159	0.0440
7	1139	0.0435	152	1246	0.0460	802	1143	0.0453
9	1159	0.0436	202	1184	0.0438	852	1206	0.0460
11	1191	0.0439	252	1264	0.0445	902	1126	0.0459
13	1207	0.0444	302	1190	0.0434	952	1211	0.0462
15	1208	0.0442	352	1232	0.0442	1002	1107	0.0454
17	1202	0.0443	402	1181	0.0442	1052	1158	0.0471
19	1216	0.0442	452	1187	0.0441	1102	1134	0.0464
21	1149	0.0424	502	1117	0.0443	1152	1144	0.0468
23	1164	0.0425	552	1203	0.0450	1202	1102	0.0468
25	1187	0.0429	602	1150	0.0436	1252	1168	0.0482
26	1195	0.0435						

Based on the selected smoke images, manual annotations are made for the smoke and fire situation, where fire incidents are labeled as 1 and no fire incidents are labeled as 0. Due to the large number of training images, only a few annotated sample images are displayed. The results are shown in Table 7.

Table 7 Illustration of Fire Incident Annotation Results for Training

Training Images	Fire Situation	Training Images	Fire Situation
	0		1
	0		1
	0		1

3.2.2. Neural Network Training Results

The obtained 40 sets of local binary variance and high-low frequency relative energy feature data, along with their corresponding fire incident annotations, are imported into the created neural network for training. The training parameters are set according to the specifications in section 2.2 above. The training results are shown in Figures 19 and 20.

In Figure 19, the variation of the error metric during the training process is displayed. The annotation "Best Training Performance" at the last epoch indicates that although the error has decreased, it did not reach the initial training goal of 0.01. Training stopped after reaching the maximum number of training cycles. Figure 20 is the linear regression plot of the training results, which is crucial for evaluating the model's generalization performance. A higher regression coefficient indicates a better fit of the neural network model to all training data. The regression coefficient of the neural network alone is only 0.62.

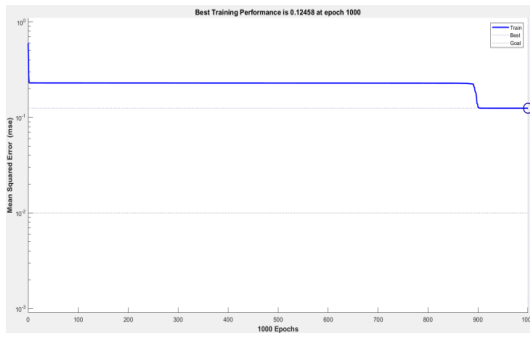


Fig. 19. Training Error Iteration Graph

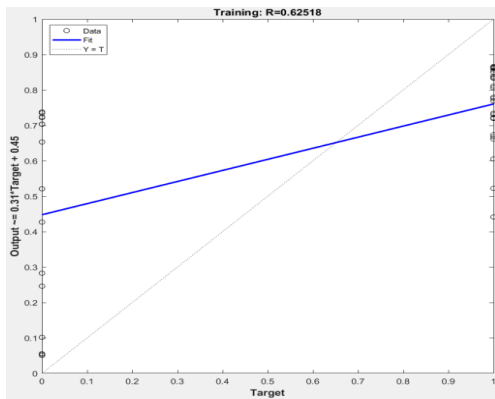


Fig. 20. Linear Regression Plot of Training Results

3.2.3. Prediction Accuracy Calculation Results

Import 31 sets of test data into the trained neural network for prediction, compare the predicted smoke and fire results with the actual smoke and fire results, and calculate the prediction accuracy of the network. The feature values of the test images, along with the actual fire situation and the predicted fire situation, the results are shown in Table 8.

Table 8 Comparison Table of Predicted and Actual Values for 31 Test Samples

Frame Number	Local Binary Variance	High-Low Frequency Relative Energy	Annotation	Prediction	Frame Number	Local Binary Variance	High-Low Frequency Relative Energy	Annotation	Prediction
3	1079	0.0434	0	0	759	1205	0.0445	1	1
66	1218	0.0446	1	0	822	1133	0.0459	1	1
129	1218	0.0451	1	1	885	1164	0.0460	1	1
192	1249	0.0446	1	1	948	1171	0.0464	1	0
255	1276	0.0446	1	1	1011	1146	0.0457	1	1
318	1250	0.0450	1	1	1074	1168	0.0458	1	1
381	1163	0.0445	1	1	1137	1185	0.0480	1	1
444	1133	0.0439	1	0	1200	1141	0.0474	1	1
507	1202	0.0452	1	1	1263	1111	0.0477	1	0
570	1168	0.0448	1	1	1326	1099	0.0447	1	1
633	1237	0.0436	1	0	1389	1076	0.0463	1	0
696	1189	0.0451	1	1	1452	1115	0.0473	1	0
1	1072	0.0433	0	0	10	1179	0.0439	0	1
4	1093	0.0435	0	0	13	1207	0.0444	0	1
7	1139	0.0435	0	0	16	1204	0.0444	0	1
19	1217	0.0442	0	1					

The comparison between predicted fire incidents and actual fire incidents is illustrated in Figure 21.

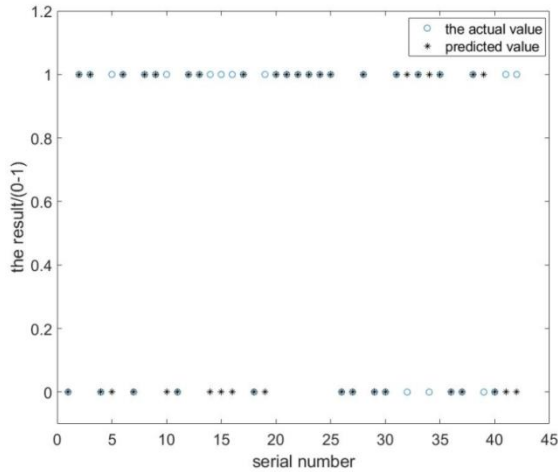


Fig. 21. Comparison Plot of Predicted Fire Incidents vs. Actual Fire Incidents

From Figure 21 and Table 8, it can be observed that out of 31 test samples, there are 11 instances where the predicted results do not match the actual results. Thus, the prediction accuracy is only 64.5%. This indicates that relying solely on neural networks for prediction yields poor accuracy and may converge prematurely.

3.3. Genetic Algorithm Optimization of Neural Networks

3.3.1. Optimal Neural Network Structure Parameters

According to the set parameters, optimization is performed to obtain the final optimal neural network structure. The internal weights and thresholds of the optimal neural network structure are shown in Tables 9 and 10.

Table 9 The Internal Weights

Input layer and hidden layer connection weights	
-0.57	994.77
-0.14	18.23
-0.01	-393.65
-0.14	-1309.96
-0.03	1056.72

Table 10 Thresholds

Hidden Layer Thresholds				
424.86	92.94	-298.14	-407.41	11.90
Hidden layer and output layer connection weights				
642.69	162.78	24.76	223.57	-42.87
Output Layer Thresholds				
-132.2				

During the iterative optimization process of genetic algorithm, the fittest individual, i.e., the one with the lowest testing error, is selected at each generation. After 50 generations of evolutionary iterations, the optimal neural network structure parameters mentioned above are obtained. The best individual during the iteration process is depicted in Figure 22.

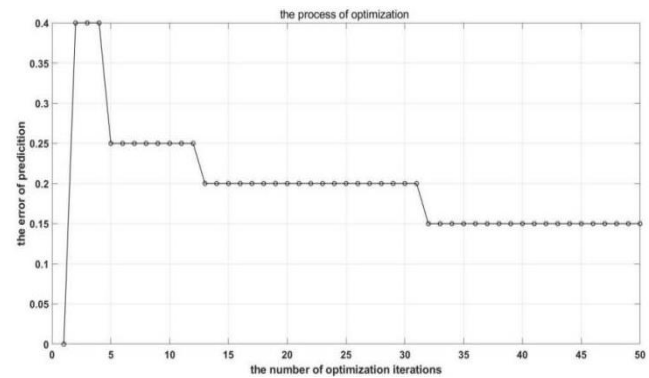


Fig. 22. Optimization Iteration Process Diagram

3.3.2. Validation of Optimized Neural Network Fire Smoke Prediction Accuracy

The 31 predicted samples shown in the table are imported into the optimized neural network for prediction accuracy calculation. The results are shown in Table 11.

Table 11 Comparison Table of Predicted and Actual Values for 31 Test Samples

Frame Number	Local Binary Variance	High-Low Frequency Relative Energy	Annotation	Prediction	Frame Number	Local Binary Variance	High-Low Frequency Relative Energy	Annotation	Prediction
3	1079	0.0434	0	0	759	1205	0.0445	1	0
66	1218	0.0446	1	1	822	1133	0.0459	1	1
129	1218	0.0451	1	1	885	1164	0.0460	1	1
192	1249	0.0446	1	1	948	1171	0.0464	1	1
255	1276	0.0446	1	1	1011	1146	0.0457	1	1
318	1250	0.0450	1	1	1074	1168	0.0458	1	1
381	1163	0.0445	1	1	1137	1185	0.0480	1	1
444	1133	0.0439	1	1	1200	1141	0.0474	1	1
507	1202	0.0452	1	1	1263	1111	0.0477	1	1
570	1168	0.0448	1	1	1326	1099	0.0447	1	1
633	1237	0.0436	1	0	1389	1076	0.0463	1	1
696	1189	0.0451	1	1	1452	1115	0.0473	1	1
1	1072	0.0433	0	0	10	1179	0.0439	0	0
4	1093	0.0435	0	0	13	1207	0.0444	0	0
7	1139	0.0435	0	0	16	1204	0.0444	0	0
19	1217	0.0442	0	0					

The comparison between predicted fire incidents and actual fire incidents is illustrated in Figure 23.

From Table 11 and Figure 23, it can be observed that out of the 31 test samples, there are 2 instances where the predicted results do not match the actual results. Thus, the prediction accuracy is approximately 94%. This indicates that the smoke and fire recognition neural network optimized using a genetic algorithm has significantly improved prediction accuracy.

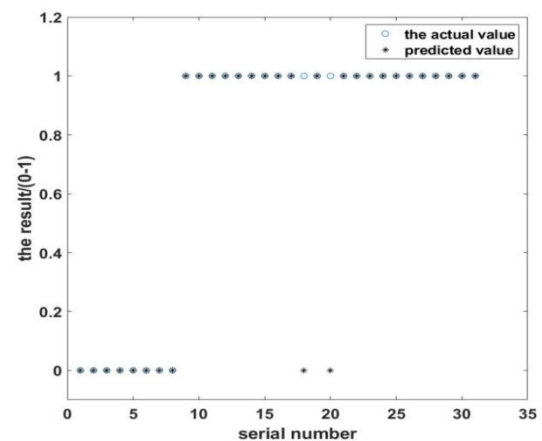


Fig. 23. Comparison Plot of Predicted and Actual Values

4. Conclusion

This study proposes a method for smoke and fire recognition, with the following innovations:

1. Smoke Feature Extraction: Through preprocessing of smoke and fire images and conversion to the HSI color space, local binary variance and high-low frequency relative

energy feature data of smoke images are extracted using the LBP algorithm and wavelet transform algorithm. Smoke recognition is performed using a BP neural network.

2. Genetic Algorithm Optimized BP Neural Network: Utilizing a genetic algorithm to optimize the weights and thresholds of the neural network effectively addresses the issues of local optima and premature convergence in BP neural networks, significantly improving the accuracy of smoke and fire recognition. Experimental results show that the recognition accuracy of the neural network alone is only 64.5%, whereas the recognition accuracy of the neural network optimized by genetic algorithm can reach 94%.

3. This intelligent recognition system can improve the accuracy of fire and smoke detection systems by compensating for situations where there is no flame, thus reducing the probability of fire occurrence.

Author contributions

Qi Qian initiated the research topic and guided the project. **Anton Louise De Ocampo** did a strict review of the whole article and provided helpful insights and suggestions on various aspects of writing the paper. **Hong Yanwu** actively participated in the Neural network model design of the system. All authors read and approved the final version of the article.

Conflicts of interest

The authors declare no conflicts of interest.

References

- [1] S. Calderara, P. Piccinini, and R. Cucchiara, "Vision based smoke detection system using image energy and color information," *Machine Vision and Applications*, vol. 22, no. 4, pp. 705–719, 2011.
- [2] A. L. P. De Ocampo, "Haar-CNN Cascade for Facial Expression Recognition," *2023 International Electrical Engineering Congress (iEECON), Krabi, Thailand, 2023*, pp. 89–92, doi: 10.1109/iEECON56657.2023.10126902.
- [3] H. Kim, D. Ryu, and J. Park, "Smoke detection using GMM and adaboost," *International Journal of Computer and Communication Engineering*, vol. 3, no. 2, pp. 123, 2014.
- [4] F. Yuan, Z. Fang, S. Wu, et al., "Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis," *IET Image Processing*, vol. 9, no. 10, pp. 849–856, 2015. Fig. 23. Comparison Plot of Predicted and Actual Values
- [5] Y. Q. Zhao, Q. J. Li, and Z. Gu, "Early smoke detection of forest fire video using CS Adaboost algorithm," *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 19, pp. 2121–2124, 2015.
- [6] S. Ye, Z. Bai, H. Chen, et al., "An effective algorithm to detect both smoke and flame using color and wavelet analysis," *Pattern Recognition and Image Analysis*, vol. 27, no. 1, pp. 131–138, 2017.
- [7] S. Frizzi, R. Kaabi, M. Bouhouicha, et al., Convolutional neural network for video fire and smoke detection, 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 2016, pp. 877–882.
- [8] K. He, X. Zhang, S. Ren, et al., Deep residual learning for image recognition, IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 2016, pp. 770–778.
- [9] C. Yuan, Z. Liu, and Y. Zhang, "Learning-based smoke detection for unmanned aerial vehicles applied to forest fire surveillance," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1, pp. 337–349, 2019.
- [10] J. Zeng, Z. Lin, C. Qi, et al., "An Improved Object Detection Method Based On Deep Convolution Neural Network For Smoke Detection," *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1, 2018, pp. 184–189.
- [11] Y. Lee and J. Shim, "False positive decremented research for fire and smoke detection in surveillance camera using spatial and temporal features based on deep learning," *Electronics*, vol. 8, no. 10, pp. 1167, 2019.
- [12] S. Majid, F. Alenezi, S. Masood, et al., "Attention based CNN model for fire detection and localization in real-world images," *Expert Systems With Applications*, vol. 189, pp. 116114, 2022.
- [13] Z. Ke, "A New Image Filtering Algorithm for Impulse Noise," *Fire Control and Command Control*, 2011.
- [14] J. Li, K. Wang, and D. D. Zhang, "A new equation of saturation in RGB-to-HSI conversion for more rapidity of computing," *IEEE*, 2002.
- [15] E. Welch, R. Moorhead, and J. K. Owens, "Image processing using the HSI color space," *Southeastcon '91, IEEE Proceedings of IEEE*, 1991.
- [16] M. J. Thurley and V. Danell, "Fast Morphological Image Processing Open-Source Extensions for GPU Processing With CUDA," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 7, pp. 849–855, 2012.
- [17] W. H. Alobaidi, et al., "Face detection based on probability of amplitude distribution of local binary patterns algorithm," *IEEE*, 2018.
- [18] X. P. Zeng, et al., "The Research of Using BP Neural

Network in Prediction of Woods Fire Insurance Grade,” *Journal of Chongqing University (Natural Science Edition)*, vol. 28, no. 1, pp. 73-76, 2005.

- [19] J. Zhou, W. Ma, and J. Miao, “Fire Risk Assessment of Transmission Line Based on BP Neural Network,” *International Journal of Smart Home*, vol. 8, no. 3, pp. 119-130, 2014.
- [20] C. Rocha, et al., “Calibration of Forest Fire Risk Combined Index - ICRIF,” *3rd EJIL - LAETA Young Researchers Meeting*, 2015.
- [21] M. A. Matos, et al., “A Genetic Algorithm for Forest Firefighting Optimization,” *International Conference on Computational Science and Its Applications*, Springer, Cham, 2022.