№ 2-3(5), 2023 ISSN: 2534 – 8507 (print) ISSN: 2534 – 8515 (online)

RESEARCH AND BUILDING A FIRE-DETECTING MODEL BASED ON DATA PREPROCESSING AND CONVOLUTIONAL NEURAL NETWORK

Dinh Cong Tung[1] **Do Thi Huyen**[2] **Dimitar Borisov**[3]

[1]Faculty of Information Technology, University of Transport and Communications

[2]Faculty of Information Technology, East Asia University of Technology

[3]University of Chemical Technology and Metallurgy, Sofia, Bulgaria

tungdc@utc.edu.vn; huyendt@eaut.edu.vn; shtain@uctm.edu

Abstract: **This paper proposes a method to improve the accuracy of fire detecting deep learning model through data preprocessing. Firstly, images are converted to an HSV color model to highlight fire features, then the FAST corner detection algorithm is used to extract regions of interest (ROI). For each identified ROI, we use the pre trained VGG16 network to classify whether there is a fire object present. Experimental results show that the accuracy of the proposed model in this article reaches over 92%, higher than other deep learning models.**

Keywords: **Fire detecting, HSV color model, FAST algorithm, VGG-16.**

I. INTRODUCTION

Today, the risks of fire and explosions are becoming more and more of a threat, causing serious damage to both people and properties. One of the main reasons for this is late fire detection. Late fire detection leads to severe consequences such as uncontrolled fire growth, increasing fire-damaged areas, causing significant property damage, and even loss of human lives. Therefore, early fire detection is needed for firefighting and rescue teams to respond promptly preventing the fire from spreading and minimizing the loss of lives and property caused by fire and explosions.

To address this issue, various methods have been proposed. Some common fire detection methods have been introduced, including using temperature sensors and smoke detectors. However, sensor-based systems have limitations, such as a small monitoring area and sensor lifespan issues due to environmental and geographical conditions. Satellite-based alert systems can cover larger areas but suffer from low image resolution and high operational costs, which are also greatly affected by weather conditions. In practice, a highly regarded method for fire detection is relies on images and videos collected from surveillance cameras. Therefore, image processing techniques are extensively researched and applied in fire detection.

Recently, deep learning algorithms have gained attention due to their ability to rapidly and accurately detect fires. Image recognition algorithms based on Convolutional Neural Networks (CNNs) are efficient at automatically learning and extracting complex image features. Consequently, researchers have applied Convolutional Neural Networks (CNN) to detect fire through images. In [1], a research group evaluated four popular deep learning models for fire detection and found that the ResNet152-V2 model achieved the highest accuracy. In [2], a method was applied to the Haar Cascade Classifier machine learning algorithm, adapted from the YOLOv3 model to help robots identify and manage fires. The paper [3] introduced a multi-level forest fire detection method using data generated from GAN models and Adaboost based on HOG. The authors then used CNN and Support Vector Machines (SVMs) to detect forest fires. The paper [4] proposed a CNN fire detection algorithm called Yolo-Edge. The paper [5] presented the DAI-YOLO model, an improved fire detection method based on an enhancing YOLOv3, including structural improvements, larger input image sizes, separable convolution structures, and improved detection features. The paper [6] researched a home fire detection solution using computer vision to provide information on fire location, spread direction, and scale, along with methods to enhance training data and improve the model. In [7], the authors proposed a fire detection method based on deep learning models such as AlexNet, GoogLeNet, and VGG-16. The paper [8] introduced a CNN structure inspired by SqueezeNet, improving performance and accuracy in fire detection. In [9], an enhanced YOLOv5 model with augmented training data, dynamic region pooling, and an attention mechanism was proposed to detect smoke and provide early fire warnings. Domain knowledge about smoke was applied for region-of-interest selection, along with transfer learning from Alexnet, Inception v3, and pre-trained ResNet models to enhance smoke detection accuracy in videos [10]. The paper [11] proposed a synthetic learning method, combining YOLOv5 and EfficientDet. YOLOv5 performed fire detection, and the EfficientNet model synthesized results for accurate identification. The paper [12] used a Dirichlet Process Gaussian mixture model in preprocessing to address diversity in flame color, texture, and shape. In 2020, LBP color features combined with deep learning were used to detect smoke and fire from overhead camera data [13]. The paper [14] introduced the SR-Net model for smoke and fire detection using satellite imagery, combining CNN and the Lightweight Vision Transformer (Lightweight ViT). Background subtraction methods were applied to reduce computation and false detections, such as adaptive background subtraction [15], GMM-based background subtraction [16], and motion history images [17]. The paper [18] proposed the ForestResNet model, a deep learning model for fire detection based on ResNet50. The paper [19] presented a CNN model for smoke detection in fires, optimizing and improving classification accuracy using batch normalization and multi-convolutions.

In this paper, we introduce a method to enhance the accuracy of fire detection based on images collected from cameras. Figure 1 shows an overview of the methods used in the paper. The input to the model is an RGB color image. Firstly, the image is converted to the HSV color space, where regions with the potential presence of fire are detected. Next, we use the FAST algorithm to mark areas that are likely to contain fire. Finally, these regions are classified as either having fire or not using the VGG-16 model.

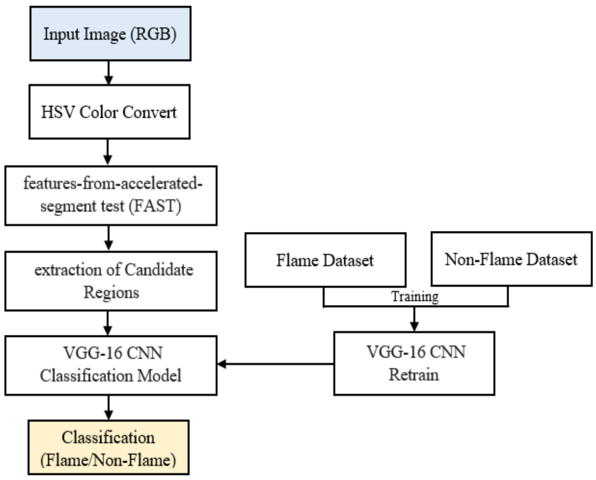

Fig. 1. Overall processing workflow in the paper

II. PROPOSED METHOD

A. Data preprocessing

In this paper, our first preprocessing step is to convert the color space from RGB to HSV. The HSV color space describes the hue, saturation, and value (brightness) of an image. Hue and saturation values are particularly useful because they closely resemble how humans perceive colors, making them a useful foundation for image processing algorithms. Hue represents color distribution based on red, while saturation indicates the level of white light, and value is used to describe the intensity of light in an image. These attributes of the HSV color space make it an ideal tool for developing image processing algorithms based on color sensor attributes. To highlight fire characteristics, we propose specific color range limits to filter suspicious fire-existence pixels based on hue, saturation, and value, applied in the HSV color space as follows: 5-90 for hue, 40-255 for saturation, and 220-255 for value.

Figure 2 shows the HSV color conversion in the paper. Specifically, Figure 2a corresponds to the original fire image, Figure 2b represents the image after the conversion to the HSV color space, and Figure 2c presents the resulting areas suspected to contain fire based on the pixels, following the completion of the HSV color space conversion process.

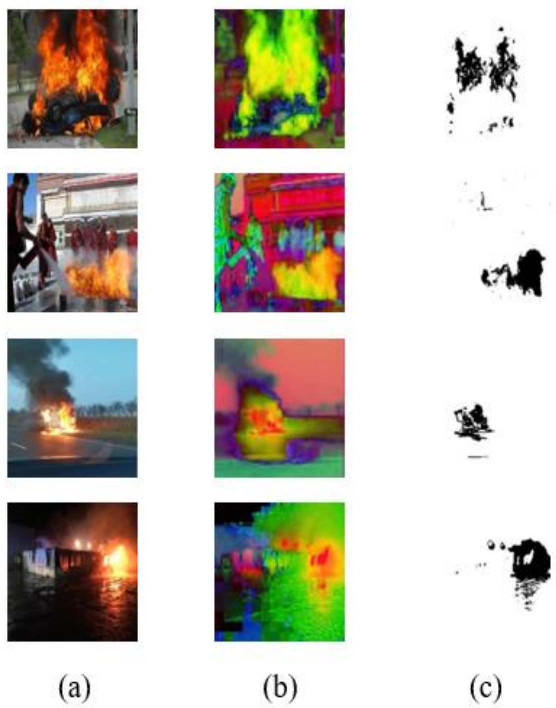

Fig. 2. Results after performing the HSV color space conversion

After converting to the HSV color space, we utilize the FAST algorithm to mark suspected fire regions. FAST proposed by Rosten and Drummond in 2006, follows these steps.

1. Choose any pixel point (p) within the image. Assume the intensity (I) for the pixel point (p) as the candidate of interest.
2. Set the intensity threshold (T) at 25% of the pixel value.
3. Draw a Bresenham circle around the candidate pixel point with a 16-pixel radius (equal to a radius of 3).
4. Among the N neighboring pixels outside the circle, a pixel is considered of interest if its intensity is brighter (threshold + I) or darker (I - Threshold). (Usually, N = 12)
5. Check the pixel intensities (I1, I5, I9, and I13) around the circle (clockwise) and compare them with (I). To speed up detection, at least three out of four pixels must meet the threshold criteria (T). Thus, the point of interest will be detected.
6. Check the pixels (I1 and I9) to see if they are darker or brighter. If so, check (I5 and I13). To be considered a point of interest, it must have a minimum of (3) out of (4) pixels that are brighter (I+) or darker (I - Threshold). If none meet this criteria, the pixel point (p) is not considered of interest.
7. Repeat these steps for each pixel point [20].

This algorithm has some drawbacks as described below. Firstly, when (N) is less than 12, the algorithm does not perform well because it results in an excessive number of detected points of interest. Secondly, arranging the 16 pixels can limit the algorithm's speed. To fix this, machine learning methods have been incorporated into the FAST algorithm [21].

Figure 3 illustrates the regions of interest after applying the FAST corner detector. The top row shows the original image, the second row shows the image after converting to the HSV color space, as described in section A. The third row represents the regions of interest marked in blue after using the FAST corner detector. These regions will be further classified as containing fire or not, based on the VGG-16 model.

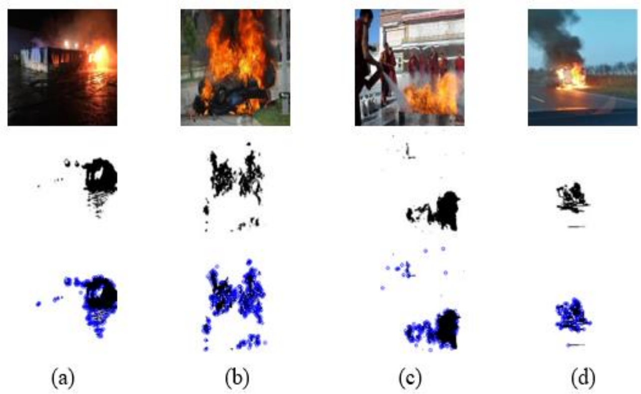

Fig. 3. Results after using FAST corner detector

B. VGG-16 model

When applying the fire detection method based on HSV color information, objects that are not fire but have a similar yellowish color to fire may still exist, leading to a significant decrease in accuracy. Furthermore, in the image preprocessing step, a region where the fire object may exist is extracted using the FAST algorithm, but it cannot determine the object type. Therefore, to address this, the final step in the paper is to use the VGG-16 deep learning model to detect fire in the regions of interest extracted in the preprocessing step.

VGG-16, introduced in 2014, was a helpful improvement for previous models like AlexNet and LeNet. In it's architecture, VGG-16 features 13 convolutional layers, convolutional blocks, stacked CNN layers, and max-pooling layers, unlike previous models that alternated between just one CNN and max-pooling layer. It also employs ReLU activation after each CONV layer. VGG-16 uses only 3x3 sized filters, reducing the number of model parameters and improving computational efficiency. Based on the VGG-16 model, we designed the CONV layers following the input image with the following parameters: a 3x3 convolutional layer with padding=1, stride=1, and a 2x2 max-pooling layer. The layer structure is described in Figure 4. In the input layer, we employ 3 filters, each with a size of 3x3. Based on the VGG-16 model, we design CONV layers placed behind the input image with the following parameters: Convolutional layer with size 3x3, filter with size 3x3, padding=1, stride=1, max pooling with the layer is 2x2 in size. The max pooling functions are applied with the purpose of reducing the size of the input matrix according to a window of size 2x2, after 2 to 3 convolution layers. The layers of VGG16 are depicted in Figure 4.

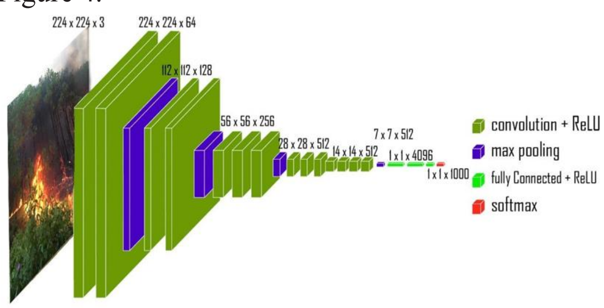

Fig. 4. VGG-16 model

The training process with the VGG-16 model was conducted using a dataset of 10003 images of Rohan Roy, categorized into two classes: with fire and without fire. The training dataset included 3003 fire images and 3000 non-fire images. The validation dataset comprised

1000 fire and 1000 non-fire images. The testing dataset also included 1000 fire images and 1000 non-fire images. Table 2 describes the dataset as presented in the paper.

Table 1. Data division in the paper.

Data	Fire	Non-fire	Total
Training	3003	3000	6003
Test	1000	1000	2000
Validation	1000	1000	2000
Total	5003	5000	10003

After training with the VGG16 algorithm, we obtained a model capable of fire detection. During testing, for each input image, we perform preprocessing using the proposed method in section A to extract the regions of interest (ROI) and apply the trained model to classify whether there is fire within those regions of interest.

III. Experimental Results

Our method was implemented and trained on a computer with CORE I7-12700F (20 CPUs) 2.10GHz, 32GB RAM, running Windows 10, and using Python 3.6 as the development environment. To demonstrate the effectiveness of the proposed method, we performed training and testing with three popular deep learning models, AlexNet, Inception-V3 and VGG-16, using unprocessed test data. After experimenting with these three deep learning models and the proposed method, we found that our method achieved the highest accuracy (92.70%). In comparison, the VGG-16 method without preprocessing achieved an accuracy of 87.75%, the Inception-V3 method had an accuracy of 85.05%, and the AlexNet method had an accuracy of 74.60%. To further clarify the effectiveness of the proposed method, we conducted statistical analysis and compared four values when training the models: Precision, Recall, F1-Score, and Accuracy. Specifically, Accuracy is the ratio of correctly predicted cases to the total number of test data, calculated using the formula (9). Precision is the ratio of correctly predicted cases in a class to the total number of cases predicted in that class, according to the formula (10). Recall is the ratio of the number of correctly predicted cases in a class to the total actual data in that class, calculated using the formula (11). The F1-score is the harmonic mean determined based on the precision and recall measures, according to the formula (12). The comparison results of the accuracy of the proposed method with other deep learning models are shown in Table 2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{12}$$

Table 2. Accuracy comparision between algorithms

Model	Accuracy	Precision	Recall	F1-Score
Proposed model	92.70%	91.69%	93.9%	92.78%
VGG-16	87.75%	86.40%	89.60%	87.97%
Inception-V3	85.05%	84.46%	85.90%	85.17%
AlexNet	74.60%	74.84%	74.10%	74.47%

To provide a more visual representation of the accuracy of predictions with the proposed model, a Confusion Matrix was used. In the Confusion Matrix, the vertical axis represents the actual labels corresponding to the two classes, "Fire" and "Non- Fire" and the horizontal axis represents the predicted labels, also for the same two classes. For the proposed model, the Confusion Matrix is shown in Figure 5a. According to this matrix, the model correctly predicted 939 cases of "Fire" out of a total of 1024 predicted cases. There were 85 cases where the model incorrectly predicted "Fire" when, in reality, there was no fire. Additionally, there were 61 cases of actual "Fire" that the model failed to predict. Out of the total 1000 actual "Fire" cases, the model correctly predicted 939 cases. For the VGG-16 model, using the test dataset non preprocessing, the Confusion Matrix is shown in Figure 5b. In this case, the model correctly predicted 896 cases of "Fire" out of a total of 1037 predicted cases. There were 141 cases where the model incorrectly predicted "Fire" when, in reality, there was no fire. Additionally, there were 104 cases of actual "Fire" that the model failed to predict. Out of the total 1000 actual "Fire" cases, the model correctly predicted 896 cases. For the Inception-V3 model, also using the test dataset without preprocessing, the Confusion Matrix is shown in Figure 5c. According to this matrix, the model correctly predicted 859 cases of "Fire" out of a total of 1017 predicted cases. There were 158 cases where the model incorrectly predicted "Fire" when, in reality, there was no fire. Additionally, there were 141 cases of actual "Fire" that the model failed to predict. Out of the total 1000 actual "Fire" cases, the model correctly predicted 859 cases. For the AlexNet model, also using the test dataset without preprocessing, the Confusion Matrix is shown in Figure 5d. According to this matrix, the model correctly predicted 741 cases of "Fire" out of a total of 990 predicted cases. There were 249 cases where the model incorrectly predicted "Fire" when, in reality, there was no fire. Additionally, there were 259 cases of actual "Fire" that the model failed to predict. Out of the total 1000 actual "Fire" cases, the model correctly predicted 741 cases, we gathered a model which is proposed to classify lung abnormalities based on X-ray.

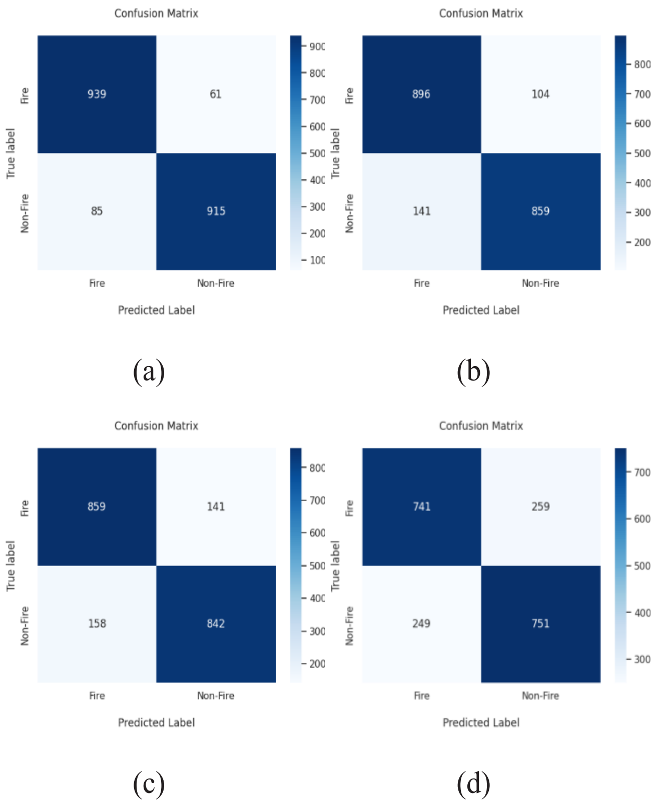

Fig. 5. Confusion Matrix of each model

It can be seen that the choice of training model has a great influence on the proposed method. When a training model with high accuracy is selected, the classification of suspicious areas presented in section A will also be more accurate. If the training model for fire classification has low accuracy, the effectiveness of the proposed method will be affected.

CONCLUSIONS

This paper proposed an appropriate preprocessing method with a deep CNN model for fire detection based on images collected from cameras. The preprocessing methods utilized the HSV color space to highlight fire-related features, and the FAST corner detector was used to scan and extract Regions of Interest (ROI). After that, a deep CNN model was trained to detect the presence of fire within these ROIs. Through the proposed method, the accuracy of fire detection improved by approximately 5% compared to a model without preprocessing, and it is outperformed to other deep learning models. In the future, we intend to further develop and enhance the deep learning model to achieve even higher accuracy. Additionally, we plan to expand the scope of detection to include smoke and integrate infrared cameras to ensure faster and more effective fire detection.

ACKNOWLEDGEMENTS

This research is funded by University of Transport and Communications (UTC) under grant number T2023-CN-004.

REFERENCES

[1]. Cong Tung Dinh, Thu Huong Nguyen, Huyen Do Thi, Nam Anh Bui, 2023. Research and Evaluate some Deep Learning Methods to Detect Forest Fire based on Images from Camera. Proceeding of The 12th Conference on Information Technology and its Applications; Image and Natural Language Processing, 181-191.

[2]. Ramasubramanian, S. A. Muthukumaraswamy, A. Sasikala, 2020, Fire Detection using Artificial Intelligence for Fire-Fighting Robots, 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 180-185.

[3]. Zhaochun Liu, Kai Zhang, Chenyang Wang, Siyu Huang, 2020, Research on the identification method for the forest fire based on deep learning, Optik, chapter 223, 165491.

[4]. W. Li, Z. Yu, 2021, A Lightweight Convolutional Neural Network Flame Detection Algorithm, IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC).

[5]. Chengnuo Li, Dapeng Cheng, Yajie Li, 2022, Research on fire detection algorithm based on deep learning, International Conference on Cloud Computing, Performance Computing, and Deep Learning (CCPCDL 2022), chapter 12287, 1228722.

[6]. James Pincott, Paige Wenbin Tien, Shuangyu Wei, John Kaiser Calautit, 2022, "ndoor fire detection utilizing computer vision-based strategies, Journal of Building Engineering, chapter 61, 105154.

[7]. G. Son, J. -S. Park, B. -W. Yoon, J. -G. Song, 2018, Video Based Smoke and Flame Detection Using Convolutional Neural Network, 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 365-368.

[8]. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, S. W. Baik, 2019, Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications, IEEE Transactions on Systems, Man, and Cybernetics: Systems, chapter 49, no 7, 1419-1434.

[9]. Wang Z, Wu L, Li T, Shi P, 2022, A Smoke Detection Model Based on Improved YOLOv5, Mathematics, chapter 10, no 7, 1190.

[10]. Yang Jia, Weiguang Chen, Manjiang Yang, Liangwu Wang, Dongcai Liu, Qixing Zhang, 2021, Video smoke detection with domain knowledge and transfer learning from deep convolutional neural networks, Optik, chapter 240, 166947.

[11]. R. Xu, H. Lin, K. Lu, L. Cao, Y. Liu, 2021, "A Forest Fire Detection System Based on Ensemble Learning", Forests, chapter 12, no 2, pp 217.

[12]. Z. Li, L. Mihaylova, L. Yang, A deep learning framework for autonomous flame detection, Neurocomputing, Volume 448, 11 August 2021, Pages 205-216.

[13]. Hossain, F.M.A.; Zhang, Y.M.; Tonima, M.A. Forest fire flame and smoke detection from UAV-captured images using fire-specific color features and multi-color space local binary pattern. J. Unmanned Veh. Syst. 2020, 8, 285–309.

[14]. Zheng Y, Zhang G, Tan S, Yang Z, Wen D and Xiao H (2023) A forest fire smoke detection model combining convolutional neural network and vision transformer. Front. For. Glob. Change 6:1136969. doi: 10.3389/ffgc.2023.1136969

[15]. Z. Li, O. Isupova, L. Mihaylova, et al, Autonomous flame detection in video based on saliency analysis and optical flow, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2016, 10.1109/MFI.2016.7849492.

[16]. F. Yuam, et al, A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with AdaBoost for video smoke detection, Pattern Recognition, Volume 45, Issue 12, December 2012, Pages 4326-4336.

[17]. M. Mueller, P. Karasev, I. Kolesov, A. Tannenbaum, Optical flow estimation for flame detection in videos, IEEE transactions on image processing, vol. 22, no. 7, july 2013.

[18]. Y. Tang, H. Feng, J. Chen. ForestResNet: A Deep Learning Algorithm for Forest Image Classification. J. Physics Conf. Ser. 2021, 2024, 012053.

[19]. X. Sun, Y. Huang. Forest fire smoke recognition based on convolutional neural network. J. For. Res. 2020, 32, 1921–1927.

[20]. E. Rosten, R. Porter, and T. Drummond, FASTER and better: A machine learning approach to corner detection" in IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, pp. 105-119 14 Oct 2008.

[21]. E. Rosten and T. Drummond, Fusing points and lines for high-performance tracking, in Proc. Int. Conf. Computer Vision, vol. 2, pp. 1508–1511, October 2005.