Fire Detection Based on Improved-YOLOv5s

Mengdong Zhou, Jianjun Li, and Shuai Liu(✉)

Central South University of Forestry and Technology, Changsha 410004, China
xifang99@126.com

Abstract. Forest fires have a very bad impact on the natural environment and human beings. To protect the environment and enhance human safety, it is important to detect the source of a fire before it spreads. The existing fire detection algorithms have a weak generalization and do not fully consider the influence of fire target size on detection. To enhance the ability of fire detection of different sizes, ground fire data and Unmanned Aerial Vehicle (UAV) forest fire data are combined in this paper. To improve the detection accuracy of the model, a cosine annealing algorithm, label smoothing, and multi-scale training are introduced. The experimental results show that the Improved-YOLOv5s model proposed in this paper has strong generalization and a good detection effect for different sizes of fires. The mean Average Precision (mAP) value reaches 88.7%, 8% higher than that of YOLOv5s mAP. The proposed model has the advantages of strong generalization and high precision.

Keywords: Fire detection · YOLOv5 · Cosine annealing

1 Introduction

Forests are known as the "lungs of the earth". As a kind of natural ecosystem, forests can regulate climate, purify the air and maintain ecological balance [1]. However, in recent years, frequent forest fires have inflicted devastating blows on many forest ecosystems, not only burning trees, affecting soil quality, but also having a significant impact on human society [2, 3]. After the flame spreads, it is difficult to carry out fire fighting work, so it is very important to find the fire source in time [4].

There are two approaches to fire detection: sensor-based and vision-based. Sensor-based fire detection requires close activation of sensors to collect sensitive information on the fire scene. The fire alarm detection system designed by Dasari [5] uses a smoke sensor and a flame sensor to detect the flame and uses the global mobile communication system to notify the user remotely. Noureddine [6] uses a unimodal approach to detect fires (the sensed data is scalar in nature, such as temperature and humidity). Due to the high cost of sensor-based detection systems and complex detection conditions [7], vision-based detection systems have fast response, wide-coverage, and low deployment costs [8]. Therefore, vision-based fire detection has attracted more and more attention.

In the early development of computer vision, machine learning algorithms flourished, and the development of fire detection was accompanied by machine learning methods. Mithira et al. [9] used a Bayesian classifier and Support Vector Machine (SVM) classifier

E. Pimenidis et al. (Eds.): ICANN 2022, LNCS 13532, pp. 88–100, 2022.
https://doi.org/10.1007/978-3-031-15937-4_8

to analyze the features of the Histogram of Oriented Gradient (HOG) extracted from images. To achieve higher detection accuracy; Jin and Lu [10] extracted the fire motion information and color information and improved the Local Binary Pattern (LBP) features in the color information to achieve better detection performance. Mishra et al. [11] achieved accurate and efficient prediction by combining RGB, YC_bC_r, and CIE-LAB color features. Wahyono et al. [12] proposed a new framework for fire detection based on color, motion, and shape features combined with machine learning techniques to accelerate detection speed while maintaining accuracy. Although machine learning can achieve good detection results, it still cannot get rid of the shortcomings of manual extraction. Fire detection is faced with the problem of background complexity and target diversity, which makes it impossible for humans to discover the deep features hidden in pixel information with naked eyes [13, 14]. The emergence of the deep learning method brings a new development path to fire detection.

Deep learning methods can realize automatic feature extraction, which makes flame detection based on image classification more efficient [15]. Existing object detection algorithms can be roughly divided into two categories: two-stage algorithms and one-stage algorithms. The algorithm of the two-stage is to locate first and then classify; the algorithm of the one-stage completes two tasks at the same time. Zhang [16] adopted Faster Regional Convolutional Neural Network (Faster R-CNN) to detect wildland forest fire smoke, avoiding the complex manual feature extraction process in traditional video smoke detection methods. Barmpoutis [17] combined Faster R-CNN with multi-dimensional texture analysis, and the detection accuracy was improved. Saponara [18] developed a detection technique using R-CNN to measure smoke and fire features in restricted video surveillance environments. These two-stage algorithms, although high in accuracy, are slow in detection. To address this problem, efficient object detection frameworks (one-stage algorithms) are proposed, such as You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD). Wu [19] judged that the one-stage algorithm has a better real-time performance by comparing the ability of Faster R-CNN, YOLOv3, and SSD to detect fire. Wang [20] proposed a lightweight fire detection model using MobileNetV3 to replace the backbone network of YOLOv4, and the inference speed was accelerated by 3 times. Xu [21] integrated YOLOv5 and EfficientDet to complete the fire detection process, while using EfficientNet to learn global information, the detection performance was improved by 2.5%–10.9%. Since forest fire detection needs to take into account the accuracy and speed, this paper improves the YOLOv5 network on this premise.

The main contributions of this paper are as follows:

- To improve the detection accuracy, the Focus module in the backbone network is replaced by the Convolution layer, the BatchNorm layer, and the Sigmoid Weighted Liner Unit layer (CBS);
- To speed up the inference process of the model, the Spatial Pyramid Pooling (SPP) module is upgraded to the Spatial Pyramid Pooling Faster (SPPF) module;
- Use the cosine annealing algorithm and the warm-up learning rate decay strategy to replace the original linear decay strategy to make the model more effective;
- Label smoothing is used to prevent overfitting;
- Use multi-scale training to improve the detection accuracy of the model.

2 Method

2.1 Data Collection and Preprocessing

Data Collection. To enhance the generalization ability of the model, the dataset in this paper contains flames of different shooting heights, different forest scenes, different times, and different burning ranges. The large number of pictures taken by drones included in it ensure the model's ability to detect small flames.

Due to the scarcity of public fire datasets, this paper collected three sets of public datasets for experiments, namely the public datasets of the Computer Vision and Pattern Recognition Laboratory of Keimyung University in Korea [22], the FIRESENSE database [23], and the public datasets of Durham University. The dataset [24], with a total of 2059 images (Fig. 1a). To enhance the generalization effect of the model and improve the detection ability of the model for small target flames, the data set also contains 1853 pieces of UAV forest flame data published by Northern Arizona University, Fire Luminosity Airborne-based Machine learning Evaluation (FLAME) [25] (Fig. 1b), and 1549 flame data of the drone data taken by ourselves (Fig. 1c). The final experimental data has a total of 5461 flame pictures.

Images are marked with "fire" using LabelME software to generate XML files. In the experiment, the data set is divided into the training set, validation set, and test set according to the ratio of 8:1:1.

a b c

Fig. 1. Dataset example, where a is ordinary fire data, b is FLAME drone data, and c is our drone data.

Data Preprocessing. To further enhance the generalization of the model, we preprocess the images using mosaic data augmentation. Mosaic data enhancement is to stitch four pictures into one picture, and these four pictures are randomly flipped, scaled, color gamut changed, and panned (Fig. 2). The probability of image scaling and flipping is 50%, the probability of hue, saturation, and brightness in the color gamut change is 1.5%, 70%, and 40%, respectively, and the probability of panning is 10%.

The uneven distribution of small objects in the dataset on the images will lead to insufficient training. After using mosaic data enhancement, the distribution of small targets will become uniform, so mosaic data enhancement can improve the detection ability of small targets.

Fig. 2. Mosaic data augmentation.

2.2 Network Model

YOLOv5 is the latest YOLO network, which achieves the effect of fast detection speed and high detection accuracy. It is mainly composed of the backbone network, neck, and head. In this experiment, we fine-tuned the network structure of YOLOv5, replacing the Focus module in the backbone network with the CBS module, and the SPP module in the neck with the SPPF module. The overall network model is shown in Fig. 3.

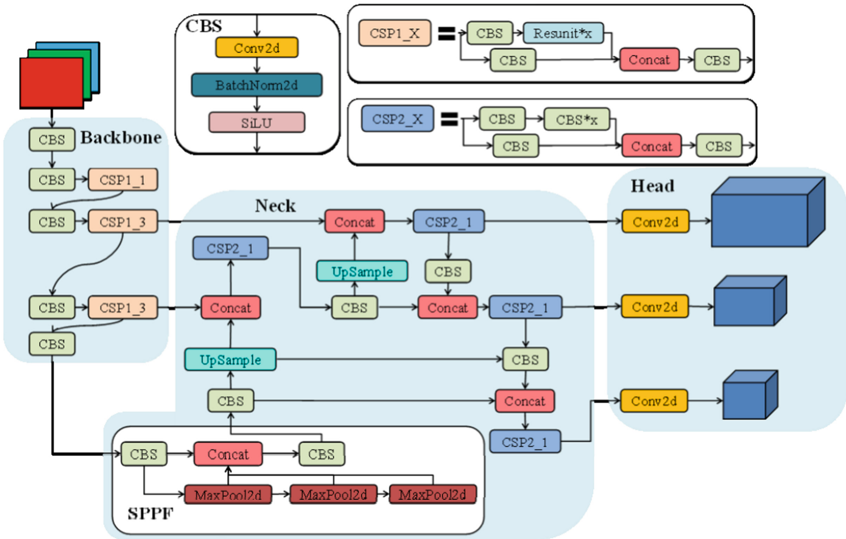

Fig. 3. Overall network structure diagram.

SPPF. The SPPF module is called fast spatial pyramid pooling, which can transform feature maps of arbitrary size into feature vectors of fixed size. Compared with the SPP module, SPPF (see Fig. 4) divides the features obtained through the maximum pooling layer into two parts, one is used for final splicing and the other is continued pooling, and features of different levels are obtained through different sub-pooling. The final effect of both SPP and SPPF is to obtain eigenvectors of fixed size, but SPPF is about 4 times faster.

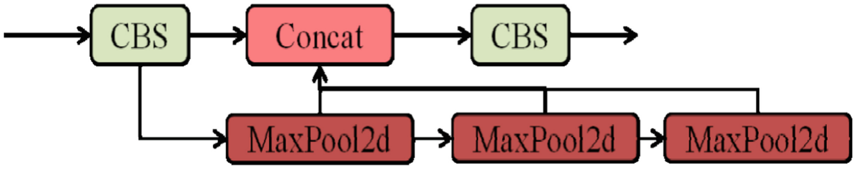

Fig. 4. SPPF.

2.3 Cosine Annealing + Warm-Up

In the training of the deep learning algorithm, the method of gradient descent is adopted to optimize the model, and the standard weight update formula is Eq. 1:

$$W+ = \alpha * gradient \tag{1}$$

where W is the weight; α is the learning rate.

The learning rate α controls the step of gradient update. The larger α is, the faster the weight changes and the faster it reaches the optimal point. If α is 0, the network will stop updating. In the whole process of gradient descent, assuming that the learning rate is constant if the learning rate is set too small, the gradient descent speed will be too slow; if the learning rate is set too large, the model will finally be difficult to converge and hover around the minimum value. So it's very important to change the learning rate dynamically. So model training introduces the concept of learning rate decay strategy.

The cosine annealing algorithm (Eq. 2) is a learning rate decay strategy. The principle of the cosine annealing algorithm is to reduce the learning rate from an initial value following a cosine function to zero. Slowly reduce the learning rate at the beginning, almost linearly reduce the learning rate in the middle, and slowly reduce the learning rate again at the end.

$$\eta_t = \eta_{min}^i + \frac{1}{2}\left(\eta_{max}^i - \eta_{min}^i\right)\left(1 + cos\left(\frac{T_{cur}}{T_i}\pi\right)\right) \tag{2}$$

where, i represents the index times, η_{max}^i and η_{min}^i represent the maximum and minimum values of the learning rate respectively. T_{cur} represents how many epochs have passed since the last restart. T_i represents how many epochs need to be trained for the i th restart.

Warm-Up is a way to warm up the learning rate. Since the model weight is randomly generated at the beginning of training, if the learning rate is too large at this time, the model will be oscillated (unstable). If Warm-Up is selected, the learning rate at the beginning of training will be small, and the model will gradually stabilize. The pre-set learning rate is used for training, which makes the model convergence speed faster and the model effect better. We combine the two methods to obtain the learning rate transformation of this experiment (see Fig. 5).

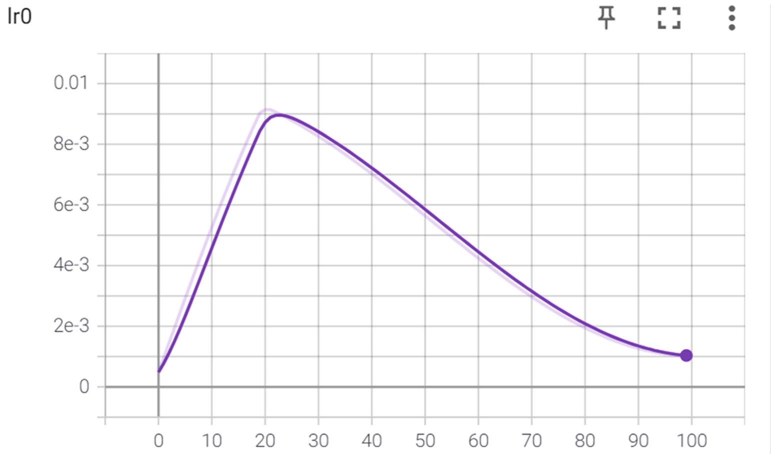

Fig. 5. Learning rate curve.

2.4 Label Smoothing

Because the samples of the dataset are manually labeled, there are usually a small number of incorrect labels, which can affect the prediction effect. Label smoothing is to assume that there may be errors in the label in the training process, avoid "excessive" trust in the label of the training sample, and improve the problem of poor generalization ability.

Label smoothing (Eq. 3–4) combines uniform distribution and replaces the traditional one-hot encoded label vector y_{hot} with the new label vector $\hat{y}_i$:

$$\hat{y}_i = y_{hot} * (1 - \alpha) + \alpha / K \tag{3}$$

$$\hat{y}_i = \begin{cases} 1 - \alpha, \ i = \text{ target} \\ \alpha / K, \ \ i \neq \text{ target} \end{cases} \tag{4}$$

where K is the number of label categories and α is the hyperparameter that determines the smoothness quantity, α is a small hyperparameter (generally 0.1).

Distribution of label smoothing is equivalent to adding noise to the real distribution, which prevents the model from being too confident about the correct labels and reduces the difference between the output values of positive and negative samples, thus avoiding over-fitting and improving the generalization ability of the model.

2.5 Multi-scale

Multi-scale training has been proven to be an effective way to improve performance. The size of the input image has a great influence on the performance of the detection model. In the basic network part, the feature image is often generated tens of times smaller than the original image, resulting in the feature description of small objects not easy to be captured by the detection network. By inputting larger and larger images for training, the robustness of the detection model to object size can be improved to a certain extent.

Multi-scale training refers to setting several different image input scales and iterations randomly selecting a scale during training. In this way, the trained model has strong robustness, which can accept images of any size as input, and the test speed will be faster by using images of small scale.

3 Result

The experiment is deployed on a computer with an I9-7920X CPU and a GeForce RTX 2080Ti GPU. We used Pytorch deep learning framework for modeling and called CUDA, Cudnn, and OpenCV libraries to train and test the forest fire detection model. In the experiment, the training and test pictures were randomly scaled to 640 × 640. The SGD optimizer was used to optimize the network, and the setting of hyperparameters was shown in Table 1.

Table 1. Hyperparameter settings.

Parameter value	Parameter value
Learning rate (lr)	0.01
Warm Up epochs	3.0
Warm Up momentum	0.8
Warm Up bias lr	0.1
Momentum	0.937
Batch size	16
Number of iterations	100

3.1 Evaluation Index Calculation Formula

To compare the difference in detection effect among different models, three evaluation indexes, precision (Eq. 5), recall (Eq. 6), and mAP (Eq. 7), are introduced to quantitatively compare model performance.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$mAP = \sum\nolimits_{k=1}^{N} P(k)\Delta R(k) \tag{7}$$

Among them, True Positive (TP) means that the sample is positive and the prediction result is positive, False Negative (FN) means that the sample is positive and the prediction result is negative, and FP (False Positive) means that the sample is negative and the prediction result is positive. In Eq. 7, N represents the number of all pictures in the test set, P(k) represents the value of Precision when k pictures can be recognized, and ΔR(k) represents that the number of recognized pictures changes from k −1 to k time (by adjusting the threshold) the change of the recall value.

3.2 Results Presentation

To compare the progress of each modification on the experiment, we conducted five control experiments, and each control experiment was modified by one step based on the previous one. Since adding the modified method name directly after the model name will cause the name to be too large. Therefore, letters A~E are used to represent different control experiments. The correspondence between letters A~E and experiments is shown in Table 2.

Table 2. Correspondence between letters and models

Letter	Model
A	YOLOv5s
B	YOLOv5s + CBS + SSPF
C	YOLOv5s + CBS + SSPF + cosine annealing
D	YOLOv5s + CBS + SSPF + cosine annealing + label smoothing
E (Our model)	YOLOv5s + CBS + SSPF + cosine annealing + label smoothing + multi-scale

Figure 6 is a curve of the evaluation index as the number of training iterations increases. As can be seen from the figure, Experiment E as a whole is the best performing model in terms of precision, recall, and mAP. The original model is the worst in any aspect, so it can be concluded that our modification is very helpful to improve the detection effect.

To show the improvement of the model more intuitively, Table 3 shows the result data of the four evaluation indicators, where mAP_0.5 represents the average accuracy of the model when the threshold is 0.5, and mAP_0.5:0.95 represents the average accuracy of the model when the threshold is 0.95. It can be seen from the table that compared with the previous set of experiments, the precision of experiments B, C, and E is improved, but the recall remains unchanged, which proves that modifying the model structure, using the cosine annealing algorithm and setting multi-scale can reduce the false detection rate, that is, the target that is not a flame is not detected as a flame, which enhances the detection ability of similar targets. The recall value of experiment D is increased and the precision is decreased, which proves that label smoothing improves the overall detection level, and thus improves the false detection rate, but overall the improvement of flame detection is still stronger, so the mAP increases. In general, mAP is 8% higher than that of YOLOv5s, which is a great improvement. Among them, the cosine annealing algorithm contributes the most to the improvement of the model detection effect.

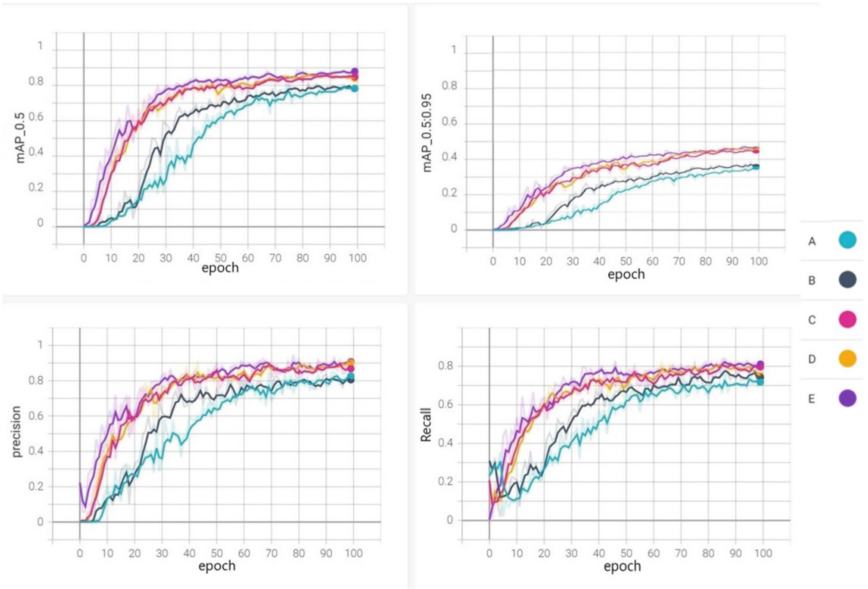

Fig. 6. Change curve of evaluation indexes.

Table 3. Comparison of evaluation index results.

Model	Precision	Recall	mAP_0.5	mAP_0.5:0.95
YOLOv5s	0.77804	0.7834	0.80621	0.35018
YOLOv5s + CBS + SSPF	0.81124	0.78409	0.81004	0.36408
YOLOv5s + CBS + SSPF + cosine annealing	0.90271	0.78409	0.85961	0.44868
YOLOv5s + CBS + SSPF + cosine annealing + label smoothing	0.87013	0.81439	0.86947	0.45218
YOLOv5s + CBS + SSPF + cosine annealing + label smoothing + multi-scale (Our model)	0.94685	0.81061	0.88712	0.45584

To demonstrate the detection ability of the model for objects of different sizes in different environments, Tables 4 and 5 show the detection effects of the five models on common datasets and UAV datasets. Among them, Table 4 shows the detection effect of ordinary flame data, and Table 5 shows the detection effect of UAV data. From the table, it can be seen that experiment E is better in terms of both the number of detections and the accuracy. Especially in the UAV image, the flame target is too small and easy to be missed, and the missed detection rate of the model in this paper is the lowest.

Table 4. Detection results of normal data.

Model	Detection results		
A			
B			
C			
D			
E			

4 Discussion

In Sect. 3.2, we discuss the application performance of Improved-YOLOv5s in different scenarios. The Improved-YOLOv5s model proposed in this paper has achieved good results in detecting targets of different scales, targets in different environments, and targets from different angles. It not only provides real-time detection but also has good robustness in practical applications. Nonetheless, we found that there are still problems of false detection of fire-like objects and missed detection of severely occluded objects during testing. This phenomenon may be caused by the variability of flames and the complexity of fire spread in the actual environment. It is worth mentioning that this is also an urgent problem to be solved by current object detection models [26]. Encouragingly, these difficulties are not insurmountable. A more powerful feature extraction network [27, 28] or an attention mechanism [29] can be selected to enhance the learning ability of the model. In future work, the structure of the Improved-YOLOv5s model will be further optimized, and the focus will be on the image feature extraction stage. It is hoped

Table 5. Detection results of UAV data.

Model	Detection results
A	
B	
C	
D	
E	

that the generalization ability of Improved-YOLOv5s can be further improved through transfer learning.

5 Conclusion

Forest is an important natural resource, it is very important to grasp the prevention and control of forest fire. In this paper, we put forward a forest fire detection model with strong generalization, which is convenient for the follow-up fire fighting work. To improve the detection effect of the model, the Focus module and SPP module are modified structurally. Cosine annealing algorithm, label smoothing, and multi-scale training are introduced in the training. Compared with the original network, the mAP of the modified network is improved by 8%. In the future, we will further study the characteristic differences between fires and similar targets to further improve the model accuracy and speed up detection.

References

1. Muys, B.: Forest Ecosystem Services. Encyclopedia of the UN Sustainable Development Goals (2020)
2. Holden, S.R., Rogers, B.M., Treseder, K.K., Randerson, J.T.: Fire severity influences the response of soil microbes to a boreal forest fire. Environ. Res. Lett. **11**, 035004–035004 (2016). https://doi.org/10.1088/1748-9326/11/3/035004
3. Matin, M.A., Chitale, V.S., Murthy, M.S.R., Uddin, K., Bajracharya, B., Pradhan, S.: Understanding forest fire patterns and risk in Nepal using remote sensing, geographic information system and historical fire data. Int. J. Wildland Fire **26**, 276–286 (2017). https://doi.org/10.1071/wf16056
4. Bo, P.: Research on Classification of Forest Fire Risk Based on GIS Technology in Xichang City, Sichuan Province (2021)
5. Premsai, D., Reddy, G.K.J., Gudipalli, A.: Forest fire detection using wireless sensor networks. Int. J. Smart Sens. Intell. Syst. **13**, 1–8 (2020). https://doi.org/10.21307/ijssis-2020-006
6. Noureddine, H., Bouabdellah, K.: Field Experiment Testbed for Forest Fire Detection using Wireless Multimedia Sensor Network (2020). https://doi.org/10.2174/2210327909666190219120432
7. Varela, N., Díaz-Martinez, J.L., Ospino, A., Zelaya, N.A.L.: Wireless sensor network for forest fire detection. FNC/MobiSPC (2020). https://doi.org/10.1016/j.procs.2020.07.061
8. Bouakkaz, F., Ali, W., Derdour, M.: Forest fire detection using wireless multimedia sensor networks and image compression. Immunotechnology **20**, 57–63 (2021). https://doi.org/10.18280/i2m.200108
9. Mithira, S., Kavi, S., Ilakiya, S.: Efficient Fire Detection Using Hog Feature Extraction In Machine Learning (2020)
10. Jin, S., Lu, X.: Vision-based forest fire detection using machine learning. In: Proceedings of the 3rd International Conference on Computer Science and Application Engineering (2019). https://doi.org/10.1145/3331453.3361659
11. Mishra, R., Gupta, L., Gurbani, N., Shivhare, S.N.: Image-based forest fire detection using bagging of color models. In: Khanna, A., Gupta, D., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds.) International Conference on Innovative Computing and Communications. AISC, vol. 1394, pp. 477–486. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-3071-2_38
12. Wahyono, Harjoko, A., Dharmawan, A., Adhinata, F.D., Kosala, G., Jo, K.-H.: Real-time forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis. Fire (2022). https://doi.org/10.3390/fire5010023
13. Nazarenko, E., Varkentin, V., Polyakova, T.: Features of application of machine learning methods for classification of network traffic (features, advantages, disadvantages). In: 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), pp. 1–5 (2019). https://doi.org/10.1109/fareastcon.2019.8934236
14. Bhoomika, C.H., Rakshitha, B.H.: A survey on machine learning. Int. J. Eng. Appl. Sci. Technol. (2021)
15. Goodfellow, I.J., Bengio, Y., Courville, A.C.: Deep learning. Nature **521**, 436–444 (2015)
16. Zhang, Q.X., et al.: Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. Procedia Eng. **211**, 441–446 (2018)
17. Barmpoutis, P., Dimitropoulos, K., Kaza, K., Grammalidis, N.: Fire detection from images using faster R-CNN and multidimensional texture analysis. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8301–8305 (2019)

18. Saponara, S., Elhanashi, A.E., Gagliardi, A.: Exploiting R-CNN for video smoke/fire sensing in antifire surveillance indoor and outdoor systems for smart cities. In: 2020 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE (2020)
19. Wu, S., Zhang, L.: Using popular object detection methods for real time forest fire detection. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), vol. 1, pp. 280–284 (2018). https://doi.org/10.1109/iscid.2018.00070
20. Wang, S., et al.: Forest fire detection based on lightweight yolo. In: 2021 33rd Chinese Control and Decision Conference (CCDC), pp. 1560–1565 (2021). https://doi.org/10.1109/ccdc52312.2021.9601362
21. Xu, R., Lin, H.-X., Kang, L., Cao, L., Liu, Y.: A forest fire detection system based on ensemble learning. Forests **12**, 217 (2021). https://doi.org/10.3390/f12020217
22. Ko, B.C., Ham, S.J., Nam, J.Y.: Modeling and formalization of fuzzy finite automata for detection of irregular fire flames. IEEE Trans. Circuits Syst. Video Technol. **21**, 1903–1912 (2011). https://doi.org/10.1109/tcsvt.2011.2157190
23. Grammalidis, N., Dimitropoulos, K., Cetin, E.: FIRESENSE Database of Videos for Flame and Smoke Detection. Zenodo (2017). https://doi.org/10.5281/zenodo.836749
24. Dunnings, Andy Fire Image Data Set for Dunnings 2018 Study - PNG Still Image Set. Durham University
25. Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fulé, P., Blasch, E.: The FLAME dataset: aerial imagery pile burn detection using drones (UAVs). https://doi.org/10.1016/j.comnet.2021.108001
26. Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., Baik, S.W.: Convolutional neural networks based fire detection in surveillance videos. IEEE Access **6**, 18174–18183 (2018). https://doi.org/10.1109/ACCESS.2018.2812835
27. Chen, Z., Yang, J.-C., Chen, L., Jiao, H.: Garbage classification system based on improved ShuffleNet v2. Resources, Conservation and Recycling (2022)
28. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: RepVGG: making VGG-style ConvNets great again. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13728–13737 (2021)
29. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9992–10002 (2021)