Research Article

An Improved YOLO-Based Framework for Real-Time Fire and Smoke Detection

Sai Prabhu, Vannelaganti[1], Dr. K. Kiran Kumar[2]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, Guntur(D) 2401050189cse@gmail.com
[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, Guntur(D) kiran.klu5434@gmail.com

Abstract

Fire and smoke detection is highly relevant in terms of protecting people, buildings, and the environment as it helps people to be aware of dangerous events in time. Conventional vision-based detection algorithms rely heavily on hand-crafted features, which do not always work well in challenging environments with variable light conditions, cluttered backgrounds and high smoke content. To break these limitations, a deep learning-based fire and smoke detector system is developed based on advanced YOLO models. An image gallery consisting of images of fire and smoke scenarios is used, which is bound by annotation in the form of an object localization (YOLO). Image normalization, organizing the data, and establishing a single data file to be used with the classification as well as the detection process are all aspects of preprocessing. We train and test various different versions of YOLO, including YOLOv5, YOLOv8, YOLOv9, YOLOv10, YOLOv11s, and a better version of YOLOv11-DH3. The latest YOLOv11x model is also provided to determine whether the performance has improved. Models are measured by using standard metrics such as precision, recall, and mAP. The experimental results indicate that YOLOv11x is the most effective, and its precision is 0.930, recall is 0.981 and mAP is 0.967. It is also linked to a Flask-based web interface in such a way that it is capable of detecting smoke and fire in real time. There is also an automated system of email alert. It provides real time email messages with the count of detections, confidence, and the geographical location of fire and smoke events. This allows the need to respond swiftly and implement the system in a more efficient manner.

Keywords: Fires, Feature extraction, Shape, YOLO, Accuracy, Detectors, Monitoring, Image color analysis, Web and internet services, Deep learning.

1. Introduction

Fire incidences continue to pose a serious challenge to human life, infrastructure and the environment, particularly in areas that are rapidly developing and industrial areas. The occurrence of fires is on the rise due to factors such as electric issues, industrial operations, fireworks, and poor weather, which significantly contribute to the likelihood of large-scale disasters [1]. Even a small fire can easily become a disaster in cities with a lot of people and areas with a lot of trees, unless it is found early [2]. History has revealed that a delay too long to search may be disastrous, costly to both human life and property, and in the long term, the environment itself may suffer [3]. These issues demonstrate the significance of having rapid and reliable fire detection systems, which are able to perform in real-time scenarios [4].

Most of the conventional methods of fire detection relies on sensors such as heat, smoke, and gas sensors which are usually limited by their functionality in various environments as well as by their reach to different areas [5]. Such systems are less dependable in real life scenarios due to factors that are not fires like dust, fog and lighting changes [6]. In addition, traditional vision-based methods, which use hand-crafted features, find it difficult to change settings which change without notice such as color, shape, and intensity of fire [7]. Such drawbacks make current approaches more difficult in giving accurate and timely identification particularly in real-life situations that are complicated by various backgrounds and lighting conditions [8].

The recent advances in DL and computer vision have enabled them to extract features and locate objects more reliably, allowing them to solve these problems [9]. More recent detection models can directly learn complex patterns directly out of visual data, and can thus be more accurate and can be applied in a broader context. However, there still remains a need to have all-inclusive systems that are not only effective in detecting fire and smoke but also use real time response systems to effectively implement them [10].

This paper provides an intelligent fire and smoke detection framework based on sophisticated YOLO architectures to overcome these difficulties. The desired outcomes are a high level of detection, the capability to perform in a variety of scenarios as well as the ability to process information in real time. The key findings of the study include the implementation and comparison of a few YOLO models, the development of a more effective YOLOv11-DH3 architecture, and the inclusion of a real-time email alerts system to respond to emergencies. It is also suitable to real world fire monitoring applications because it has a web interface based on Flask which allows users to interact with the system and lets the system grow as required.

The rest of this study is set up like this: Section II contains a full assessment of the literature, looking at current methods for detecting fire and smoke and pointing out their flaws. Section III discusses the materials and methods, including how the datasets were collected, how they were preprocessed, and how a number of YOLO-based models were used. In the IV section, we discuss the results of the experiment and their performance in terms of precision, recall, and mAP. Section V wraps up the study by going over the main results, the system's effectiveness, and the real-world effects. Lastly, Section VI discusses future development, which dwells upon the possible improvements, more advanced integrations, and how the proposed fire and smoke detection system can be applied in the real world.

2. Literature review

Recent advances in computer vision have rendered fire and smoke detectors far more efficient because now they can learn the complex visual patterns without much effort. The initial approach to DL involved applying a combination of both spatial and temporal cues in order to detect better. As an example, G. Lin et al. erected a three-dimensional convolutional neural networks-based smoke detection system in the form of a video. Such networks performed very well in identifying motion and temporal dynamics of video sequences [11]. This technique helped locate items in dynamic environments, and was too difficult to implement in real time as it was too complex to calculate. H. Li et al. also came up with a generative learning-based method for filtering smoke that used an upgraded CycleGAN framework to make images clearer in forest fire situations [12]. This is a good method to enhance visual quality and focus on preprocessing rather than directly detecting fires, which makes this method less useful in itself.

Researchers have investigated lightweight and sequential learning models to ensure that things work better. M. Jeong et al. designed a lightweight recurrent neural network based on LSTM networks to detect wildfire smoke in real-time. They aimed at making the system consume less processing power and making decisions quicker [13]. Although recurrent models are performing well in their operations, they normally have difficulty in correctly positioning the fire zones in space. In another sense, Y. Zhao et al. studied the characteristics of wildfire identification based on the combination of time-series satellite data with deep GRU networks, demonstrating the potential of remote sensing technologies to provide extensive surveillance [14]. The satellite-based systems, however, have issues with temporal resolution, atmospheric disturbances, and too slow response times, which makes them less useful in quickly finding hazards.

The conventional methods have been used in the development of the fire detecting techniques. M. Jamali et al. proposed a saliency-based technique, which relies on both texture and color information to locate areas of fire. This is less difficult to comprehend and it is less expensive to operate [15]. But feature-based approaches that are hand-crafted are highly sensitive to environmental changes and are too weak to cope with complex situations in the real world. To overcome these limitations, the latter studies have adopted end-to-end learning systems. J. Huang et al. introduced a compact-target smoke detecting model with deformable transformer structure structure, which has an enhanced sensitivity to distant and low-intensity smoke patterns [16]. Such models are excellent, and they frequently require a substantial amount of computational resources and large datasets with annotations.

Single-stage object detection frameworks have become a promising solution since they are fast and accurate at the same time. J. Lian et al. enhanced the effectiveness of fire and smoke detection by the advanced methodology based on YOLOv7, achieving higher real-time detection performance [17]. These models are based on the work made by J. Redmon et al. that made a great leap into the field creating the YOLO framework of unified and real-time object detection. This enabled objects to be easily and quickly found in only one pass [18]. The advances in architecture such as deformable convolutional networks have enabled detection to be more accurate. Dai and colleagues enhance the flexibility of space and allow the models to better understand geometric changes in more complex environments [19]. R. has also considered sophisticated learning structures such as deep and cross networks to ensure that feature interaction and generalization in big systems are improved. Wang et al. [20].

Despite these enhancements, achieving a good balance in accuracy, speed, and robustness remains a challenge. Many of the existing solutions either have a high consumption of computing power in their architecture, or cannot adapt to environmental changes or generalize to different fire scenarios. These limitations demonstrate that we require a detection model that can effectively be used in the real world and can be extended or shrunk as the need be. This work builds on these existing methods by adding advanced YOLO architectures, improving detection performance, and adding real-time alert systems to make fire and smoke monitoring systems more useful and faster to respond to.

3. Materials And Methods

The suggested system develops an advanced system of fire and smoke detection through the object detection algorithm based on deep learning. This enables the speedy and precise finding of hazards. A well selected collection of labeled fire and smoke photos is employed and bounding boxes are drawn in the YOLO format to accurately locate the object. To learn powerful spatial features across diverse environments, the system uses a range of YOLO designs, including YOLOv5, YOLOv8, YOLOv9, YOLOv10, YOLOv11s and a proprietary YOLOv11-DH3 model. To address the issue of feature representation and feature finding accuracy, the YOLOv11-DH3 model provides improvements by making changes to the architecture based on the latest YOLO advances [24]. The assurance of effective and consistent training is guaranteed through structured preprocessing methods such as normalization, labeling and unified setup. We examine the accuracy, recall, and mean average precision measures to observe the performance of the model. It is also constructed with an automatic email notification system to send real-time email notifications with information about the detection that makes emergency response more efficient. The use of a Flask-based web interface allows easy interaction between two individuals and enables real-time inferences. The system is grounded on older fire detection techniques used on surveillance applications [22] and uses new fire monitoring datasets and fire detection algorithms to make it more reliable in a variety of settings [23].

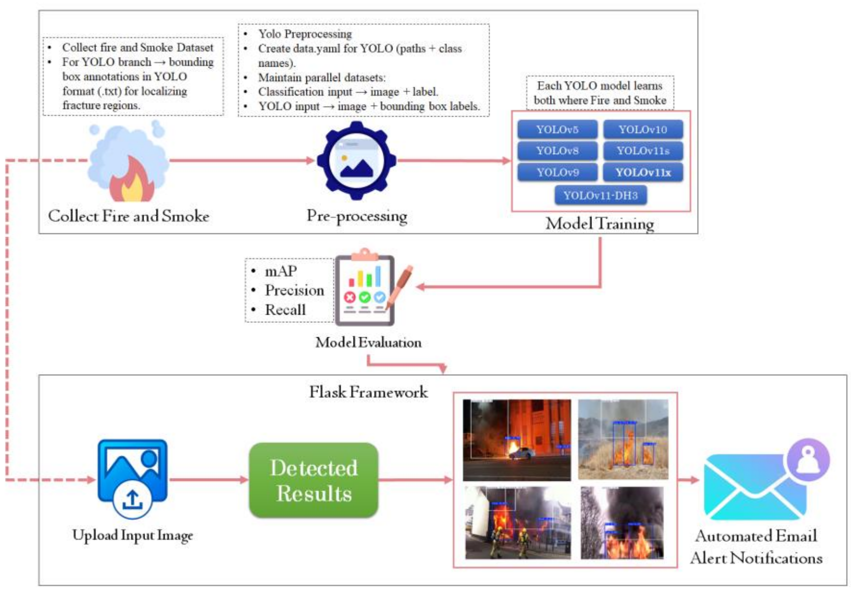

Fig. 1. Proposed Architecture

Fig. 1 depicts a full system of automatically locating fires and smoke with the help of a multi-generational YOLO structure. The method starts with gathering and labeling localized fire and smoke datasets. Then, it has a rigid pre-processing stage to prepare YAML settings and parallel labels. Training and testing various models, such as YOLOv5 and YOLOv11, are measured using the following metrics: precision, recall, and mAP. Lastly, the system is implemented with the help of a Flask web application that allows uploading images in real-time, object detection, and automated email notifications immediately.

A) Dataset Collection

The data that is utilized in this paper is that gathered in a publicly available online database of annotated fire and smoke images which could be used in the object detection task. It contains numerous various visual samples that have been taken under different conditions such as the various lighting, weather, outdoor and indoor locations, and busy backgrounds. The pictures of fires of different severities, smoke concentration, and the size of objects in the dataset allow the model to gain robust and generalized feature representations. With bounding boxes of the YOLO format, all images are thoroughly marked up, allowing the fire and smoke regions to be precisely identified. To assist in the systematic model building and testing, the data is divided into three subsets training, validation and testing. To ensure consistency in the work and make the training process more efficient, preprocessing tasks, such as shrinking pictures, normalizing them, and label checking, are completed. The quality and diversity of the dataset are highly relevant to enhance the potential of DL models to locate things, particularly in challenging real-life conditions. Recent investigations on fire detection datasets and algorithms [22] indicate that one should use such complete datasets to enhance the level of fire detection and make it operational in diverse conditions.

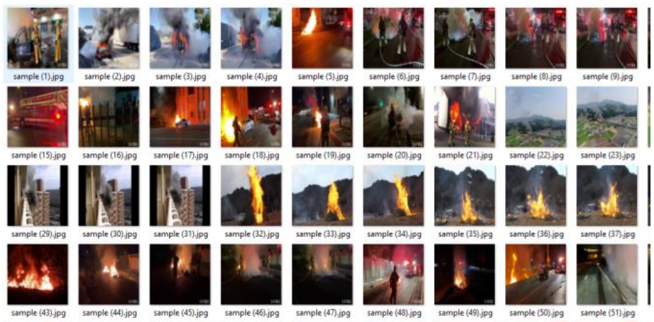

Fig. 2. Dataset

B) Pre-Processing

Another step that is important and can be observed in the suggested fire and smoke detection system is referred to as pre-processing. It ensures that the input data is uniform, well-structured and ready toDLmodels. It includes the preparation of photographs and notes through the normalization, resizing, labeling, and ranking datasets. Efficient preprocessing may help accelerate model convergence, reduce noise, enhance feature learning, and ultimately help to increase the accuracy and reliability of detection.

a) Image Preprocessing: Image processing is extremely relevant towards making data more easily digestible and capable of training deep learning models which can detect smoke and fire. The raw images of the dataset in this system are transformed in various ways to ensure that they all appear similar and work with structures based on Yolo. The initial thing to do is to resize all of the pictures so that they are equal in size. This will be significant in batch processing and maintaining the dataset in a consistent manner. This also aids in retaining important visual elements whilst simplifying the math.

Normalization is employed to bring pixel values to some standard range, typically between 0 and 1. This renders the training more stable and accelerated convergence. The stage helps to minimize the differences that arise out of the adjustment of lighting and makes the model more competent to work in various environments. Also, noise reduction methods can be used to get rid of unwanted distortions while keeping important features like fire edges and smoke patterns.

Image processing involves labeling data which is an important aspect of processing images. Using the YOLO format, which includes class names and normalized coordinates, each image has bounding boxes around the areas of fire and smoke. This structured annotation allows the model to learn at the same time both how to classify and where to find things. The data is divided into various folders to be used in training, testing, and validation. This ensures that the models are constructed and experimented in a predetermined manner.

To render the model even more powerful, data augmentation techniques such as flipping, rotating, scaling, and changing the brightness are employed. These techniques increase the size of the dataset and provide the model with additional opportunities to view various situations, which helps the model to work more effectively in the real world.

The other file is known as data.You may also code up in yaml where the data sets are and what classes they are named. This works well with the YOLO training pipeline.

The image processing pipeline ensures that input data is of high quality, eliminates inconsistencies and enhances feature representation. This renders the fire and smoke detection system far superior with regards to functionality, precision as well as reliability.

C) Algorithms

YOLOv5: YOLOv5 is a one-stage object detection system that simultaneously and quickly predicts both probability distributions and bounding boxes of objects. It has a lightweight architecture and efficient feature extraction that enables it to make quick inferences, but at the same time is highly accurate. The method uses convolutional layers to evaluate input photos and find spatial and contextual characteristics. It is able to locate things between the beginning and the end and is effective in a broad spectrum of visual environments. YOLOv5 can identify objects of varying sizes and orientations on using multi-scale feature maps and anchor-based localization. It is suitable to real-time danger monitoring since it is a good balance between speed and accuracy. This significantly enhances the detection performance and the operational durability.

YOLOv8: YOLOv8 To enhance object detection, it enhances backbone architecture and neck architecture and focuses on improving feature aggregation and multiscale detection. The architecture combines detailed spatial elements with global contextual information to assist discover and sort objects in difficult situations. We ensure that the model can generalize well to a broad set of illumination, background, and occlusion conditions by both quick inference and fine-tuning with bounding box regression. YOLOv8 demonstrates a better strength and flexibility through the use of anchor-free detection systems and the use of superior features fusion. Its quick end-to-end workflow renders it easy to locate objects in real time, and ensure that both fast detection and strong generalization work across a broad spectrum of visual conditions.

YOLOv9: YOLOv9 is an application that is capable of detecting and identifying items of the correct type and size using powerful feature extraction and multi-scale detection modules. The system is more concerned with accurate bounding box regression, as well as improved feature representation, that it performs well in challenging visual scenarios. The input photos pass through a series of convolutional and feature aggregation layers that select spatial and contextual patterns to distinguish things in the presence of busy backgrounds. YOLOv9 can be used effectively in real-time monitoring applications since it offers an excellent balance between speed and accuracy. This is advantageous as it can be used in situations where hazard detection should be accurate and reliable as well as the object can be of a variety of sizes, orientations, and environmental conditions.

YOLOv10: YOLOv10 makes improvements to the architecture that emphasize on speed and accuracy, which makes feature extraction and multiscale detection better. The model employs spatial hierarchies and contextual relations to make the objects in the various kinds of environments easier to find and classify them. It is able to process images through a single pipeline, generating bounding boxes and confidence scores in a single run. This makes inference faster. The powerful design of YOLOv10 provides it to be effective in a broad spectrum of moving and crowded visual scenes. It can be reliably and correctly identified by combining more effectively detecting heads with feature fusion methods. This assists in enhancing operational performance in real time monitoring and hazard analysis.

YOLOv11s: YOLOv11s is a simplified version of the YOLOv11 architecture that simply works on fast inference, but can still find things correctly. The model rapidly accumulates hierarchical features and incorporates the information at the various scales to discover objects with different sizes and shapes. It employs an end-to-end approach that predicts bounding boxes and object classes simultaneously, thereby increasing speed and accuracy in detecting objects. The algorithm performs efficiently in more complex and dynamic visual scenes, and it remains reliable even when objects are occluding or colliding, the light is varying, or the background is cluttered. YOLOv11s improves the generalization and operational stability, assists in rapidity to locate important objects and offers a good trade-off between speed and detection accuracy.

YOLOv11-DH3: The better features of YOLOv11-DH3 include enhancements to the architecture that enable it to better represent features and detect them in more complex visual scenarios. It is more accurate in capturing the spatial and contextual patterns by enhancing the backbone and feature fusion layers. The system takes as input multi-scale feature maps to analyze the images and simultaneously predict the bounding box and confidence score. This allows one to precisely find objects. Improved learning methods make it easier to generalize across different contexts, and enhanced detecting heads make sure that performance is stable even in tough situations. YOLOv11-DH3 is useful in providing higher accuracy in detection, localization accuracy and stability in its operation. It also facilitates real time monitoring and hazard detection and it is more sturdy compared to the previous versions of YOLO.

YOLOv11x: The latest in the line of YOLOv11 is known as the YOLOv11x. It is more accurate and robust with its enhanced feature extraction, multi-scale representation, and enhanced detection heads. It works with photos in a one-stage pipeline, which is effectively estimating the boxing and class probabilities and also able to provide detailed spatial and contextual information. The design aims at enhanced generalization and aptitude to confront alterations in the size, shape, and environment of objects. YOLOv11x gets excellent accuracy and dependable localization in complicated situations by combining better feature fusion and detection methods. Its design enables it to be used in real-time detection and high stability of its operations making it ideal in monitoring applications that consume a lot of power.

4. Experimental Results

Precision: Precision examines whether the number of the samples or case which were labeled as positive were actually the correct ones. The equation to determine the accuracy is:

$$\boldsymbol{Precision} = \frac{\textbf{True Positive}}{\textbf{True Positive } + \textbf{False Positive}} \, \textbf{(1)}$$

Recall: Recall in ML is a value that measures the ability of a ML model to identify all the relevant examples of a particular class. It is the ratio of positive observations that have been predicted correctly to the actual positive observations. This gives you an idea of how well a model is able to find instances of a certain class.

$$\boldsymbol{Recall} = \frac{\textbf{TP}}{\textbf{TP } + \textbf{ FN}} \, \textbf{(2)}$$

mAP: MAP is a means of quantifying the goodness of a ranking. It considers the number of good recommendations there are and their position on the list. The MAP at K is the average of the Average Precision (AP) at K for all users or queries.

$$\boldsymbol{mAP} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{k=1}}^{\boldsymbol{k=n}} \boldsymbol{AP_k} \, \textbf{(3)}$$

Table. 1. Performance Evaluation

ML Model	Precision	Recall	mAP
Yolo v5	0.925	0.983	0.964
Yolo v8	0.925	0.979	0.962
Yolo v9	0.929	0.973	0.961
Yolo v10	0.909	0.959	0.960
Yolo v11s	0.927	0.980	0.963
Proposed Yolo 11-DH3	0.792	0.822	0.852
Extension Yolo v11x	0.930	0.981	0.967

To compare the various YOLO models, the performance evaluation is based on precision, recall and mAP. YOLOv11x achieves the most overall performance, which implies that it is more effective at detecting things, is more stable, and works better overall than previous versions.

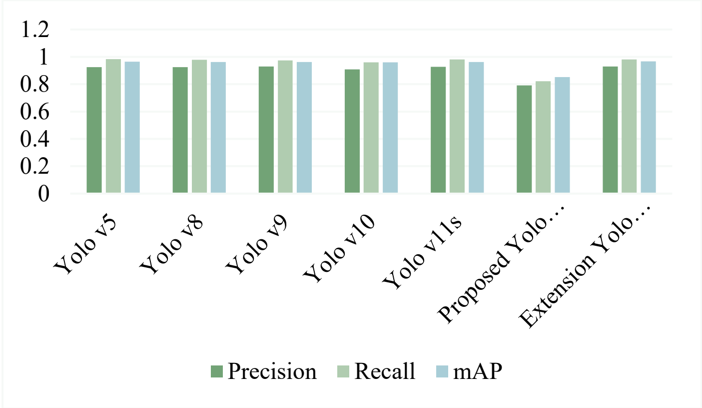

Fig. 3. Comparison Graph

The comparison graph shows the precision, recall and mAP values of various YOLO models. The best model is YOLOv11x and the worst of the models that were tested is YOLOv11-DH3.

5. Conclusion

To sum up, this paper is on how to create an effective and reliable fire and smoke detection system. It will assist people in locating hazards faster and rendering intricate settings more secure. The proposed technique involves a well-selected set of labeled fire and smoke images and state-of-the-art deep learning-based object detection algorithms to locate and identify objects. We thoroughly train and test several YOLO architectures, including YOLOv5, YOLOv8, YOLOv9, YOLOv10, YOLOv11s, and a better YOLOv11-DH3 model, to observe how well they can see things. The system uses uniform preprocessing, structured data configuration, and strong training procedures to make sure that all models are consistent and reliable. Even the latest YOLOv11x version is not left behind to achieve a more accurate detection and more successful generalization. In order to simplify it to be used in reality, a lightweight Flask-based web interface is programmed in. This allows the users to post photos and receive the results of the detection in real time on a convenient to use platform. There is also an automated email alert system which issues real time notifications with detection alerts and data such as the number of detections, confidence levels and geographic coordinates of fire and smoke incidents. This will ensure that individuals are able to react promptly and that emergency communication is enhanced. The proposed system has superior performance compared to the former with the YOLOv11x achieving a precision of 0.930, a recall of 0.981 and a mAP of 0.967. Overall, the framework designed is a reliable, efficient, and scalable solution to automated fire and smoke detection. This can be of great assistance in safety monitoring and real world decision support systems.

Future studies can examine how multi-modes of sensing methodologies can be amalgamated, and optical data can be integrated with thermal, infrared, or IoT based sensor inputs to enhance detection reliability in unfavourable environments, such as low-visibility or extreme weather. Some more advanced model optimization methods that can be employed to simplify it to make it easier to deploy on edge and embedded devices to run real-time applications. The inclusion of explainable AI techniques may bring things even more understandable and credible as it provides users with understandable information about the way detection judgments are arrived at. It might be possible to make the framework better by adding the possibility to analyze video streams continuously and identify possible fires before they occur. Large scale smart safety systems can be brought to a manageable scale, tracked, and accessed by cloud-based architecture and the capacity to connect mobile applications.

References

1. Alkhammash, E. H. (2025). A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection. Fire (2571-6255), 8(1).
2. Li, Y., Nie, L., Zhou, F., Liu, Y., Fu, H., Chen, N., ... & Wang, L. (2025). Improving Fire and Smoke Detection with You Only Look Once 11 and Multi-Scale Convolutional Attention. Fire, 8(5), 165.
3. Hu, J., He, Y., Zeng, M., Qian, Y., & Zhang, R. (2024). Smoke and fire detection based on yolov7 with convolutional structure reparameterization and lightweighting. IEEE Sensors Letters, 8(8), 1-4.

4. Gao, P. (2024). A Fire and Smoke Detection Model Based on YOLOv8 Improvement. International Journal of Advanced Computer Science & Applications, 15(3).
5. Geng, X., Su, Y., Cao, X., Li, H., & Liu, L. (2024). YOLOFM: An improved fire and smoke object detection algorithm based on YOLOv5n. Scientific Reports, 14(1), 4543.
6. Government Set to Revise Total Number of Hectares Destroyed During Bushfire Season, 9News, Sydney, NSW, Australia, Feb. 2020.
7. 58 People Died in the High-Rise Residential Fire in Jing'an District, Shanghai, China News Service, Beijing, China, 2010.
8. Theon-Site Rescue Work of the Fire and Explosion Accident at Jilin Dehui Poultry Industry Company Has Basically Ended, Xinhua News Agency, Beijing, China, 2013.
9. V. K. R. Kodur and A. M. Shakya, ''Factors governing the shear response of prestressed concrete hollowcore slabs under fire conditions,'' Fire Saf. J., vol. 88, pp. 67–88, Mar. 2017, doi: 10.1016/j.firesaf.2017.01.003.
10. N. Zheng, X. Guo, Y. Tie, N. Dong, L. Qi, and L. Guan, ''Incremental generalized multiple maximum scatter difference with applications to feature extraction,'' J. Vis. Commun. Image Represent., vol. 55, pp. 67–79, Aug. 2018, doi: 10.1016/j.jvcir.2018.04.012.
11. G.Linetal.,''Smokedetectiononvideosequencesusing3Dconvolutional neural networks,'' Fire Technol., vol. 55, pp. 1827–1847, 2019, doi: 10.1007/s10694-019-00832-w.
12. H. Li and X. Li, ''Research on smoke filtering algorithm for forest fire images based on improved CycleGAN,'' Fire Sci. Technol., vol. 43, no. 11, pp. 1596–1602, 2024, doi: 10.20168/j.1009-0029.2024.11.1596.07.
13. M. Jeong, M. Park, J. Nam, and B. C. Ko, ''Light-weight student LSTM for real-time wildfire smoke detection,'' Sensors, vol. 20, no. 19, p. 5508, Sep. 2020, doi: 10.3390/s20195508.
14. Y. Zhao and Y. Ban, ''GOES-R time series for early detection of wildfires with deep GRU-network,'' Remote Sens., vol. 14, no. 17, p. 4347, Sep. 2022, doi: 10.3390/rs14174347.
15. M.Jamali, N. Karimi, and S. Samavi, ''Saliency based fire detection using texture andcolorfeatures,'' in Proc.28thIranianConf.Electr.Eng.(ICEE), Aug. 2020, pp. 1–5.
16. J. Huang, J. Zhou, H. Yang, Y. Liu, and H. Liu, ''A small-target forest f ire smoke detection model based on deformable transformer for end to-end object detection,'' Forests, vol. 14, no. 1, p. 162, Jan. 2023, doi: 10.3390/f14010162.
17. J.Lian,X.Pan,andJ.Guo,''Animprovedfireandsmokedetectionmethod based on YOLOv7,'' in Proc. 32nd Int. Conf. Comput. Commun. Netw. (ICCCN), Jul. 2023, pp. 1–7.
18. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
19. J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, ''Deformable convolutional networks,'' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 764–773, doi: 10.1109/ICCV.2017.89.
20. R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, ''DCN v2: Improved deep & cross network and practical lessons for web scale learning to rank systems,'' in Proc. Web Conf. (WWW), Ljubljana, Slovenia, Apr. 2021, pp. 1785–1797.
21. H. Li, Y. Zhang, Y. Zhang, H. Li, L. Sang, and J. Zhu, ''DCNv3: Towards next generation deep cross network for CTR prediction,'' 2024, arXiv:2407.13349.
22. P. Foggia, A. Saggese, and M. Vento, ''Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion,'' IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 9, pp. 1545–1556, Sep. 2015, doi: 10.1109/TCSVT.2015. 2392531.
23. S. Yang, Q. Huang, and M. Yu, ''Advancements in remote sensing for active fire detection: A review of datasets and methods,'' Sci. Total Environ., vol. 943, May 2024, Art. no. 173273, doi: 10.1016/j.scitotenv.2024.173273.
24. R. Khanam and M. Hussain, ''YOLOv11: An overview of the key architectural enhancements,'' 2024, arXiv:2410.17725.
25. B. Chen, M. Wei, J. Liu, H. Li, C. Dai, J. Liu, and Z. Ji, ''EFS-YOLO: A lightweight network based on steel strip surface defect detection,'' Meas. Sci. Technol., vol. 35, no. 11, Nov. 2024, Art. no. 116003.