14th International Scientific Conference
Science and Higher Education in Function of Sustainable Development – SED 2025

APPLICATION OF REAL-TIME OBJECT DETECTION ALGORITHMS IN FIRE AND SMOKE IDENTIFICATION

Nikola Stanić[1], MSc; Stefan Ćirković[2], MSc; Aleksa Iričanin[3], MSc
[1] University of Kragujevac, Faculty of Technical Sciences Čačak, Čačak, SERBIA, nikola.stanic@ftn.kg.ac.rs
[2] University of Kragujevac, Faculty of Technical Sciences Čačak, Čačak, SERBIA, stefan.cirkovic@ftn.kg.ac.rs
[3] University of Kragujevac, Faculty of Technical Sciences Čačak, Čačak, SERBIA, aleksa.iricanin@ftn.kg.ac.rs

Abstract: *Fires pose a significant threat to human lives, property, and natural resources. The increasing urbanization and industrialization, combined with climate change, contribute to more frequent and intense fires. Traditional fire detection systems, which rely on sensors and human intervention, often fail to provide timely detection, particularly in hard-to-reach or vast areas. This study explores the application of advanced real-time object detection algorithms for fire and smoke detection, utilizing YOLOv11 and RCNN50 models demonstrating their effectiveness. The proposed methodology can significantly contribute to reducing fire-related risks by enabling automated early warning systems, particularly in high-risk areas.*

Keywords: *Fire detection, Smoke detection, Real-time object detection, Deep learning, Early warning system*

1. INTRODUCTION

Fires pose a serious challenge to the protection of human lives, property, and natural resources. The rapid growth of urban and industrial areas, combined with changes in climatic conditions, contributes to more frequent and intense fires. Traditional fire detection systems, which rely on sensors and human intervention, are often insufficient for the rapid identification of fires, especially in hard-to-reach or large areas.

Official statistics for this heating season indicate a worsening situation compared to the previous one, as the number of fires has increased by ten percent. Saša Jeršić, an inspector from the Fire and Rescue Unit Directorate, stated for Euronews Serbia that since the beginning of the heating season, 2,330 fires have been recorded in residential buildings across Serbia, emphasizing that "this is certainly not a negligible number." [1]

A 2020 report presented the development of a "static" fire hazard map for the territory of Serbia, using orographic factors (slope, aspect, elevation), vegetation types (coniferous, deciduous), and anthropogenic factors (population density, distance to settlements and roads), applying methods such as logistic regression and Random Forest. By utilizing these data, a static map was developed to assess fire hazard levels in different areas. Following the creation of the static fire hazard map, areas within a specific region of Serbia with an increased fire risk were identified. These areas will be subject to further analysis, and a network of cameras for monitoring and early fire detection will be designed. For areas not covered by cameras due to complex terrain, drone surveillance routes will be planned. [2]

By applying advanced object detection algorithms, cameras can be effectively utilized for detecting smoke and fire in real time, enabling the early identification of fires. This approach facilitates timely notification of firefighting services, significantly increasing response speed and reducing potential damage.

The integration of such models with a network of cameras for early fire detection can aid in preventing large-scale fires by enabling rapid detection in hard-to-reach areas and high-risk zones. This solution represents a crucial step toward improving the fire monitoring and protection system in Serbia.

In the study [3], an improved fire and smoke detection algorithm based on the YOLOv7 model is proposed, incorporating features like an attention mechanism, multi-scale feature fusion, a sample matching strategy, and a weighted attenuation loss function to enhance accuracy and speed. The model structure is optimized with the CDPB, PSA, and RCFPN-PAN structures to improve detection efficiency, handle scale variations, and reduce model parameters. Using a dataset with 13,843 training images and 1,061 validation images, the improved model achieved a mean average precision (APval) of 62.5% and a frame rate (FPS) of 113.6, outperforming other methods. Unlike high-accuracy models based on

Transformers that require large datasets, this approach offers a more feasible solution for fire smoke detection with limited data.
In the paper [4], a collaborative region detection and grading framework for forest fire and smoke is introduced, using a weakly supervised fine segmentation model and a lightweight Faster R-CNN. The framework detects fire and smoke regions and grades their severity, with the fire-segmentation model trained only on image-level labels. A distillation strategy is employed to simplify the Faster R-CNN. The proposed method achieves 99.6% detection accuracy and 70.2% segmentation accuracy, outperforming state-of-the-art models, while maintaining a latency of just 151ms.
To further improve the model's adaptability to real-world scenarios, a concept of collaborative cloud and edge device architecture was developed. Combined with transfer learning strategies, this enables performance enhancement through user feedback. This architecture contributes to a significant reduction in false positives and an increase in mAP values, which is essential in complex environments with high interference [5].
Jin et al. [6] reviewed various deep learning approaches for video fire detection, focusing on datasets, methodologies, and the future directions of this technology. They highlighted the importance of large and diverse datasets in training accurate models, noting that deep learning algorithms such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are increasingly being utilized for real-time fire detection in video footage. The study also underscored the challenge of detecting fire in complex and dynamic environments, where traditional methods often fall short. Furthermore, the authors identified several future research directions, including the need for more robust and adaptable fire detection systems capable of handling varying environmental conditions, different video qualities, and the integration of multi-modal sensors. This review provides valuable insights into the evolving landscape of video-based fire detection and supports the need for improved models to address the limitations of current technologies.
Particular attention should be given to models like Fire-YOLO, which focus on detecting small targets and hard-to-see visual entities under various lighting conditions. By utilizing an improved feature pyramid and reducing the number of parameters, this model enables fast and accurate identification of smaller fire and smoke sources, making it particularly suitable for forest and natural ecosystems [7].
As emphasized in studies addressing early fire detection in Serbia, there is a strong need for systems that enable precise, scalable, and automated real-time fire detection [2].
The aim of this paper is to explore and compare the efficiency of modern real-time object detection algorithms, specifically YOLOv11 and RCNN50, in identifying fire and smoke from static video footage. The paper seeks to evaluate the accuracy and detection speed of these models across different scenarios and lighting conditions, using a pre-collected dataset of video material from real-world environments.

2. ADVANCED OBJECT DETECTION ALGORITHMS

YOLO (You Only Look Once) is a popular real-time object detection algorithm. The primary characteristic of the YOLO algorithm is its ability to detect objects in a single pass through the image, making it exceptionally fast compared to other object detection methods. YOLO utilizes deep learning to recognize and classify objects in images while simultaneously estimating their locations using bounding boxes.
Unlike traditional methods that treat images as a set of smaller segmented regions, YOLO processes the entire image as a whole, enabling more efficient computation and faster object detection. The algorithm predicts multiple bounding boxes and their classes directly from the image in a single pass through the network, resulting in shorter processing times, which is particularly beneficial for real-time detection applications.
The study [8] presents a detailed performance analysis of different YOLO algorithms. Compared to previous versions, YOLO11 was analyzed on three different datasets: traffic signs, African wildlife, and ships. These datasets were carefully selected to cover a wide range of object characteristics, including object size, aspect ratio, and object density. The study highlights the strengths and weaknesses of each YOLO version, analyzing various metrics such as precision, recall, mean average precision (mAP), processing time, number of GFLOPs, and model size.
The results indicate that the YOLO11 family was the most consistent, with YOLO11m achieving an optimal balance between accuracy, efficiency, and model size. Although YOLOv10 had slightly lower accuracy compared to YOLO11, it excelled in speed and efficiency, making it a suitable choice for applications requiring fast processing. YOLOv9 also performed well, particularly when working with smaller datasets. These findings provide valuable insights for both industry and academia, aiding in the selection of the most suitable YOLO algorithm and guiding future research and improvements.
Based on this analysis, YOLO11 stands out as the best choice for applications that require a balance between accuracy, efficiency, and model size.
RCNN50 is a deep learning-based object detection model that belongs to the family of Region-Based Convolutional Neural Networks (R-CNNs). It is a variant of the standard R-CNN approach, utilizing a ResNet-50 backbone for feature extraction. This model is designed to accurately detect and classify objects in images by first identifying potential object regions and then refining their classification and localization.

In study [9], the performance of YOLO and RCNN algorithms was evaluated on a custom dataset containing various objects. The Intersection over Union (IoU) metric was used to compare the detection accuracy, with YOLO showing slightly better performance, particularly in speed. The research also demonstrates the application of object detection in an interactive language-learning model, where real-time object recognition is integrated with IoT (Raspberry Pi) and NLP technologies. This system provides an engaging, gamified learning experience through object detection and recognition.
RCNN50 follows a multi-step process for object detection:

- Region Proposal – Instead of processing the entire image at once, the algorithm first generates region proposals that may contain objects. This is typically done using a selective search or a similar region proposal method.
- Feature Extraction – Each proposed region is passed through a Convolutional Neural Network (CNN) based on ResNet-50, which extracts high-level features from the image.
- Classification and Localization – A fully connected layer processes the extracted features to classify objects and refine their bounding box coordinates.

Key differences:

- Speed vs. Accuracy – YOLO is optimized for real-time detection, whereas RCNN50 prioritizes accuracy over speed.
- Processing Method – YOLO treats the image as a whole, making it efficient but sometimes less precise. RCNN50, in contrast, carefully examines proposed regions, leading to better localization but at a higher computational cost.
- Use Cases – YOLO is ideal for real-time applications like traffic monitoring and fire detection, while RCNN50 is better suited for tasks requiring fine-grained object recognition, such as medical imaging or detailed security analysis.

3. METHODOLOGY

The dataset used in this study is an enhanced version of the original D-Fire dataset and is publicly available on Kaggle [10]. It has been curated and published by Sayed Gamal with the aim of facilitating research in the field of smoke and fire detection using object detection models such as YOLO. The dataset has been restructured to include clearly separated training and validation splits, thereby improving usability and supporting standardized model evaluation.
For training the model, a dataset containing more than 21,000 images was used, with the following distribution by categories:

- Fire only: 1,164 images
- Smoke only: 5,867 images
- Fire and smoke: 4,658 images
- No fire or smoke: 9,838 images

The dataset (Figure 1) consists of 14,122 images for training and 3,099 images for validation. This distribution allows for efficient model training, providing sufficient data for recognizing and classifying different scenarios in fire and smoke detection.

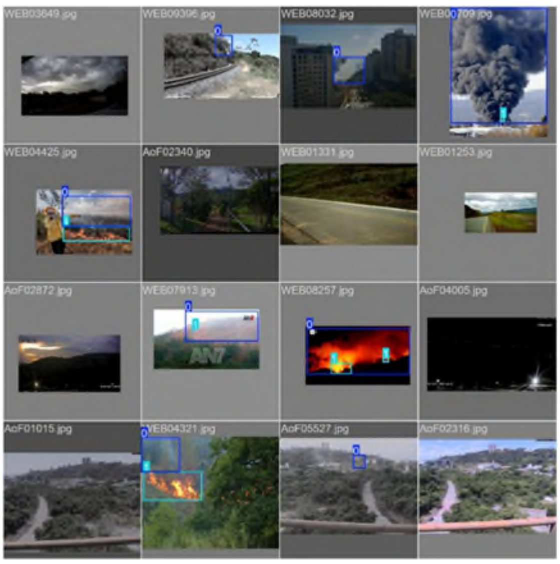

Figure 1. Data set

Each image in the dataset was individually annotated using the Labelme tool, which is available as a Python package. After annotation, each image has a corresponding JSON file with the same name, containing the coordinates of points marking the relevant parts of the image.

Once the dataset was prepared, the YOLOv11 algorithm was trained using these data. The model training was conducted on an NVIDIA GeForce 4050 graphics card, with the following parameters: 100 epochs and an image size of 640. The total training time was 4 hours and 32 minutes.

For the RCNN model, the annotations were first converted into a format compatible with RCNN. Once the conversion was complete, the RCNN algorithm was trained using the same dataset. The model training was performed with 100 epochs, and the total training time was 5 hours and 15 minutes.

Before presenting the results, it is necessary to define the evaluation metrics used to assess the performance of the YOLOv11 and RCNN models. The key metrics are precision, recall, and F1-score, each of which is derived from the confusion matrix and provides different insights into the models' effectiveness in detecting fire, smoke, and background instances.

Precision quantifies the proportion of positive identifications that were actually correct. It is given by the formula:

$$Precision = \frac{TP}{TP + FP} \qquad (1)$$

Recall measures the ability of a model to identify all relevant instances (true positives) from the dataset:

$$Recall = \frac{TP}{TP + FN} \qquad (2)$$

F1-score is the harmonic mean of precision and recall, providing a single metric that balances both values. It is particularly useful when dealing with imbalanced classes:

$$F1 - score = 2\ x\ \left(\frac{Precision\ x\ Recall}{Precision + Recall}\right) \qquad (3)$$

In the formulas above:

- TP (True Positive) represents the number of correctly predicted positive instances,
- FP (False Positive) denotes the number of incorrect positive predictions,
- FN (False Negative) is the number of actual positives that were incorrectly predicted as negatives.

These formulas serve as the foundation for analyzing the classification performance of the models and are used throughout the subsequent sections.

4. RESULT AND INTERPRETATION

In this chapter, the results and their interpretations obtained after training both the YOLOv11 and RCNN models are presented. Based on the confusion matrix shown in Table 1 for YOLOv11, and in Table 2 for RCNN, performance metrics were analyzed. The matrix illustrates how the models classify images into three categories: fire, smoke and neither. Each cell of the matrix displays the number of instances correctly or incorrectly classified by the models.

YOLOv11 model correctly classified 1450 instances as smoke, but made an error by classifying 14 fire instances as smoke and 451 background instances as smoke. As for fire instances, the model correctly identified 1548, but misclassified 8 smoke and 734 background instances as fire. The model did not identify any background instances, which is acceptable.

Table 1. YOLOv11 Confusion Matrix

	Smoke (True)	Fire (True)	Background(True)
Smoke (Pred.)	1450	14	451
Fire (Pred.)	8	1548	734
Background(Pred.)	298	614	0

RCNN model correctly classified 748 instances as smoke, but made an error by classifying 144 fire instances as smoke and 128 background instances as smoke. As for fire instances, the model correctly identified 574, but misclassified 107 smoke and 23 background instances as fire. As for background instances, the model correctly identified 1257, but misclassified 104 smoke and 19 fire instances as background.

Table 2. RCNN Confusion Matrix

	Smoke (True)	Fire (True)	Background(True)
Smoke (Pred.)	748	144	128
Fire (Pred.)	107	574	23
Background(Pred.)	104	19	1257

The discrepancy in the number of processed instances between the YOLOv11 and RCNN models is the result of both minor data loss during annotation format conversion and fundamental differences in how the two algorithms operate. During the transformation of annotation files from YOLO to the format required by RCNN, a small number of files became unusable due to conversion errors, leading to the exclusion of the corresponding images from the RCNN training and evaluation sets. In addition, YOLOv11, as a single-stage detector, processes the entire image in one pass and tends to detect a larger number of instances, including those with lower confidence or partial visibility. RCNN, on the other hand, is a two-stage detector that generates region proposals and then filters and classifies them, often omitting ambiguous or less distinct regions. As a result, YOLOv11 processed and recognized more instances overall, which is reflected in the differences observed in the confusion matrices.
Figure 2 displays a chart comparing the Precision-Confidence curves of YOLO (left) and R-CNN (right). The YOLO model achieves higher peak precision (1.00) but drops sharply at lower confidence levels, while R-CNN maintains more stable but lower overall precision (0.84), reflecting their architectural trade-offs between speed and robustness. It is particularly useful in scenarios where the false positive rate is high.

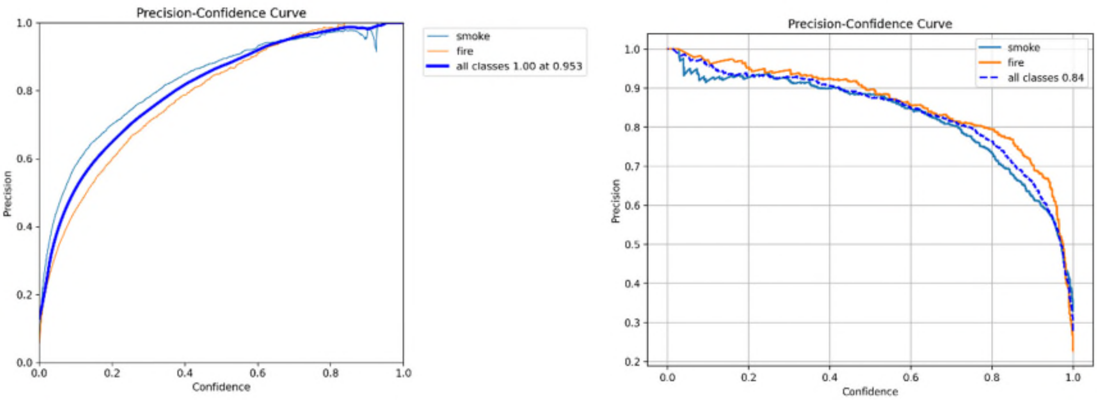

Figure 2 YoloV11 and RCNN Precision Confidence Curve

Figure 3 displays the recall, which measures the model's ability to identify all positive instances. This metric becomes particularly significant when the false negative rate is high. In the example, the YOLOv11 model achieves a recall of 0.93 (93%) and RCNN model achieves a recall of 0.85 (85%) when the confidence threshold is 0.000. This means that the YOLOv11 model successfully identifies 93% and RCNN 85% of true positive instances.

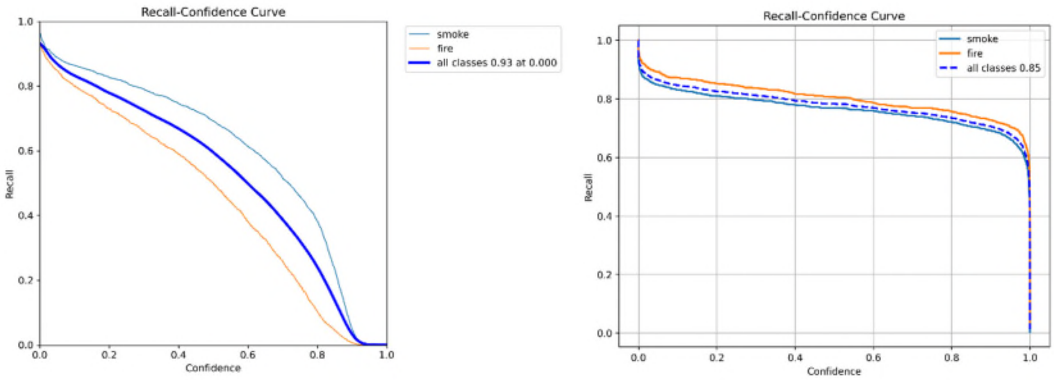

Figure 3. YoloV11 and RCNN Recall-Confidence Curves

On Figure 4, a graph is shown comparing the F1-scores of the YOLO and RCNN models. The left side of the figure represents the YOLO model, which achieved an overall F1-score of 0.74, and on right the RCNN model is depicted, reaching an F1-score of 0.78 for all classes.. The F1-score is the harmonic mean between precision and recall and is used as a measure of overall model performance, especially when there is class imbalance.

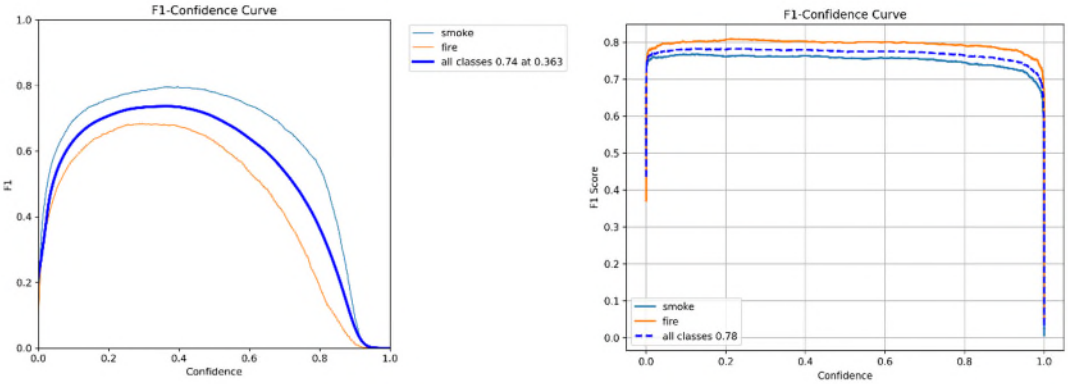

Figure 4. YoloV11 and RCNN F1 Confidence curve

Table 3 presents a comparative analysis of the performance of the YOLO and RCN models in detecting smoke and fire. The evaluation metrics include precision, recall, and F1-score, which provide insights into the effectiveness of each model. The results indicate that for smoke detection, YOLOv11 achieved a slightly higher F1-score (79%) compared to RCNN (76%), primarily due to better precision (82% vs. 78%). In contrast, for fire detection, RCNN significantly outperformed YOLOv11 with an F1-score of 80% compared to 68%, mainly as a result of RCNN's substantially higher recall (82% vs. 62%). Considering the overall performance across all classes, both models demonstrated equal precision (78%), while RCNN maintained a slightly higher F1-score (78%) than YOLOv11 (74%), suggesting that RCNN offers a more balanced trade-off between precision and recall.

Table 3. Comparison of YOLO and RCN Performance

	YOLO Precision	YOLO Recall	YOLO F1-score	RCNN Precision	RCNN Recall	RCNN F1-score
Smoke	82%	77%	79%	78%	73%	76%
Fire	75%	62%	68%	78%	82%	80%
All	78%	69%	74%	78%	77%	78%

The Figure 5 presents examples of the YOLOv11 model applied to validation images, where it detected fire and smoke. These results demonstrate the model's performance in identifying fire and smoke instances, including both correct detections and potential misclassifications.

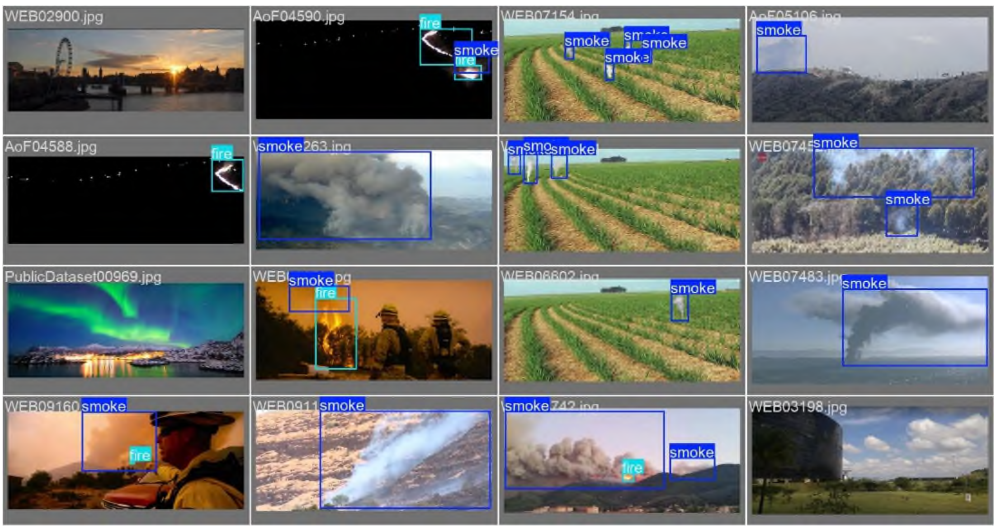

Figure 5. YOLOv11 model detections on validation images

The Figure 5 presents examples of the RCNN model applied to validation images, where it detected fire and smoke. These results demonstrate the model's performance in identifying fire and smoke instances, including both correct detections and potential misclassifications.

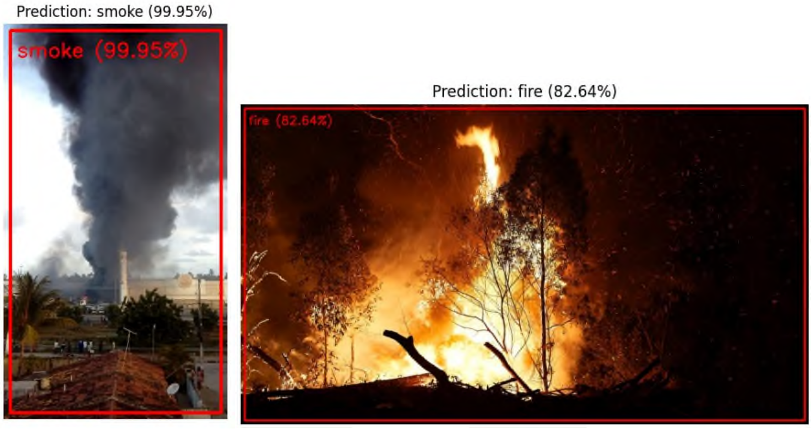

Figure 6. RCNN model smoke detections on validation images

5. CONCLUSION

The comparative analysis of the YOLO and RCNN models highlights their respective strengths and trade-offs in fire and smoke detection. YOLO achieves higher precision in detecting smoke (0.82 vs. 0.78), indicating that it generates fewer false positives in such cases. However, RCNN demonstrates superior recall in fire detection (0.82 vs. 0.62), meaning it successfully identifies a larger number of actual fire occurrences.
In terms of overall performance, YOLO maintains a higher F1-score for smoke detection (0.79 vs. 0.76), while RCNN achieves a better F1-score in fire detection (0.80 vs. 0.68). Despite these variations, their overall F1-scores remain comparable (0.74 for YOLO and 0.78 for RCNN), suggesting that the choice between the two depends on the specific application requirements.
YOLO is more suitable for scenarios that demand high precision, minimizing false alarms, whereas RCNN is preferable when maximizing recall is essential for detecting as many fire incidents as possible. Future work could explore hybrid approaches that leverage the strengths of both models to optimize real-time fire and smoke detection systems.

ACKNOWLEDGEMENTS

This study was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, and these results are parts of the Grant No. 451-03-136/2025-03/ 200132 with University of Kragujevac - Faculty of Technical Sciences Čačak.

REFERENCES

[1] Radio Slobodna Evropa, "Više od 1.500 požara u Srbiji od početka godine: Šta je uzrok?", N1, January 30, 2024. [Online]. Available: https://n1info.rs/vesti/vise-od-1-500-pozara-u-srbiji-od-pocetka-godine-sta-je-uzrok/ [Accessed: Feb. 21, 2025].

[2] S. Milanović, A. Mihalović, N. Marković, S. Milanović, D. Pamučar, and L. Gigović, "Prognoziranje opasnosti i rana detekcija požara na području Republike Srbije". Beograd, Ugovor br. 401-00-1713/2019-10, 2020

[3] C. Jin, A. Zheng, Z. Wu, and C. Tong, "Real-time fire smoke detection method combining a self-attention mechanism and radial multi-scale feature connection," *Sensors (Basel)*, vol. 23, 2023.https://doi.org/10.3390/s23063358

[4] J. Pan, X. Ou, and L. Xu, "A collaborative region detection and grading framework for forest fire smoke using weakly supervised fine segmentation and lightweight Faster-RCNN," *Forests*, vol. 12, 2021, https://doi.org/10.3390/F12060768

[5] M. Yang, S. Qian, and X. Wu, "Real-time fire and smoke detection with transfer learning based on cloud-edge collaborative architecture," *IET Image Process.*, vol. 18, pp. 3716–3728, 2024, https://doi.org/10.1049/ipr2.13187

[6] C. Jin, T. Wang, N. Alhusaini, S. Zhao, H. Liu, K. Xu, and J. Zhang, "Video fire detection methods based on deep learning: Datasets, methods, and future directions," Fire, vol. 6, no. 8, p. 315, 2023, https://doi.org/10.3390/fire6080315.

[7] L. Zhao, L. Zhi, C. Zhao, and W. Zheng, “Fire-YOLO: A small target object detection method for fire inspection,” *Sustainability*, vol. 14, no. 9, 2022, https://doi.org/10.3390/su14094930

[8] N. B. Vargafik, J. A. Wiebelt, and J. F. Malloy, "Radiative transfer," in *Convective Heat*. Melbourne: Engineering Education Australia, 2011, ch. 9, pp. 379–398.

[9] S. Borde, C. Patil, C. Sonawane, and M. Singh, "Object recognition based on deep learning algorithms using embedded IoT with interactive interface," 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), 2023, pp. 1581-1586, doi: 10.1109/ICICCS56967.2023.10142821.

[10] Kaggle, "Smoke fire detection YOLO dataset.", Available: https://www.kaggle.com/datasets/sayedgamal99/smoke-fire-detection-yolo [Accessed: Feb. 10, 2025].