



Advancing Fire Detection: A One-Stage Object Detection Approach Using YOLOv5 and YOLOv8 Models

Maroua Cheknane^{1,*}, Saida Sarra Boudouh¹, and Tahar Bendouma¹

University of Laghouat, LIM Laboratory, Algeria
m.cheknane.inf@lagh-univ.dz, s.boudouh@lagh-univ.dz,
t.bendouma@lagh-univ.dz

* Corresponding author

Abstract. Fire accidents present considerable risks on a global scale, leading to considerable losses in life, property, and the environment. Traditional sensing technologies face challenges in effectively detecting fires, particularly in large areas. Deep learning approaches have been explored for fire detection systems, but challenges remain, particularly in scenarios like indoor and forest fires, and distinguishing between fires with or without smoke. These challenges lead to environmental losses and long recovery periods. In this paper, our objective was to tackle these challenges by presenting a solution utilizing a one-stage object detection method for identifying flames and smoke, where we focused on covering indoor, outdoor, and forest fires. We employed YOLOv8 and YOLOv5 models in several gathered datasets, aiming for an accurate model. Evaluation yields a mAP@0.5 of 93% with YOLOv8. Based on the results, the best-obtained model was integrated into the implementation of a live stream-detecting application.

Keywords: Fire Detection · Deep learning · YOLOv5 · YOLOv8.

1 Introduction

Fire is a major factor causing substantial losses in lives, property, and economic damage. In 2020, the International Association of Fire and Rescue Services documented 36,480 incidents of structure fires across eighteen chosen cities globally. Additionally, 34,880 fire-related casualties were documented in 52 cities around the world over the four years from 2016 to 2020. Electrical fires are a leading cause of fires globally. Forests, which cover about 4 billion hectares or roughly 29% of the Earth's land area, also suffer significant losses from incidents like forest fires and tunnel fires. These fires, whether caused by humans or natural events, pose serious threats, often resulting in catastrophic damages and losses. Forest fires, in particular, lead to substantial environmental damage that can take decades to recover from. Early fire detection technologies primarily used particle-activated 'point sensors,' designed based on fire characteristics like heat, gas, flame, and

© The Author(s) 2024

C. A. Kerrache et al. (eds.), *Proceedings of the International Conference on Emerging Intelligent Systems for Sustainable Development (ICEIS 2024)*, Advances in Intelligent Systems Research 184,

https://doi.org/10.2991/978-94-6463-496-9_4

smoke. These sensors activate when fire particles reach the sensor, causing a delay. While effective in small rooms, they are not suitable for large, open spaces, as particles need to travel farther to reach the sensor. The static and dynamic characteristics of fire flames and smoke include attributes like color and movement, which point sensors do not utilize for detection. Deep learning, particularly Convolutional Neural Networks, has become increasingly popular across various fields of machine learning, particularly in tasks like image segmentation, classification, and retrieval [2, 9, 1]. However, the challenge of limited datasets and the large amount of data required for training hinder optimal model accuracy. Transfer learning, proposed by Sinno Jialin Pan and Qiang Yang in 2009, offers a potential solution to this issue [17, 1]. Object detection, a critical computer vision task, is divided into one-stage and two-stage detection models. One-stage models predict object presence and placement in a single pass, while two-stage models involve suggesting regions and refining bounding boxes for precise identification [13, 6]. These deep learning techniques are now widely applied in fire detection systems. Intending to fill several existing gaps, this study introduces a fire detection method utilizing the one-stage object detection technique. Hence, we aim to propose a precise model capable of detecting both flames and smoke, and we strive to integrate our proposed models into all fire detection systems, encompassing various indoor and outdoor scenarios. The rest of the paper is organized as follows: The related work section and our proposed detection method are presented in the third section, following the obtained results and discussion, and finally the conclusion.

2 Related Work

Flame detection is crucial, especially in low-light conditions where smoke can cover flames. Dou et al. [5] compared eight object detection models using CNNs on flame and smoke datasets. They optimized YOLOv5s by integrating CBAM, BiFPN, and lightweight networks like MobileNetV3. This enhanced model achieved an 82.1% mAP with 5.9M parameters and 8.1G FLOPs. Yandouzi et al. [16] employed UAVs for real-time forest fire detection, using YOLOv6, v7, v8, and Faster R-CNN with different backbones. Their dataset of 4236 Fire and Smoke images, captured by ground cameras and drones, underwent augmentation. Ground cameras provided detailed real-time visuals of forest fires. Using YOLO, they achieved 89.45% mAP@0.5, while Faster R-CNN reached 90.57% mAP@0.5. Dewangan et al. [4] introduced FlgLib, a dataset of nearly 25,000 labeled wild-fire smoke images from fixed-view cameras. They proposed SmokeyNet, a deep learning model for real-time smoke detection, achieving 83.49% image accuracy. SmokeyNet, based on ResNet34 with two input frames, demonstrates fast performance, processing images in 51.6 milliseconds.

Various machine and deep learning-based methods have been proposed to develop dependable fire detection systems. However, these approaches vary depending on their specific objectives and the techniques they employ. Some stud-

ies focused more on detecting either flame or smoke only. Meanwhile, others neglected indoor scenarios.

3 Proposed Detection System

In this research, we introduced a method for detecting fires that includes both flame and smoke detection (Fig 1). the approach starts with a data-gathering step, along with a pre-processing stage. Further, the detection phase was a one-stage object detection technique, where we applied YOLOv5/v8. Several trials resulted in trained models that were evaluated after, in order to determine the most suitable one.

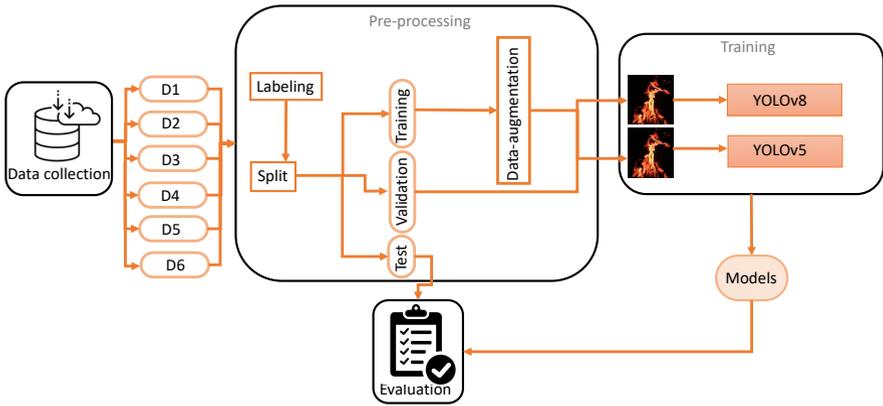


Fig. 1. Proposed One-Stage Detection Approach

3.1 Dataset

Even though several publicly available fire image datasets exist, a few of them cover all possible cases, such as indoor, outdoor, forest outdoor, flame, and smoke. In this study, we gathered our own datasets based on existing datasets, in order to guarantee variety in data as well as the maximum amount of it (Tab.1).

3.2 Pre-processing

The pre-processing phase started with the labeling process. The majority of the collected images were not already labeled. The process begins with locating the

Table 1. Collected datasets

Dataset	Images	Annotations	Indoor	Outdoor
D1	Fire:around 1000	-	✓	
D2	Fire:around 500	-	✓	
D3	Fire:1367	1500	✓	✓
D4	Smoke:737	1200		✓
D5	Fire:400 smoke:600	-	✓	✓
D6	Fire:1052 Smoke:1670	Fire:1862 Smoke:1699	✓	✓

presence of each flame and smoke with a bonding box, which was converted to a numeric value and stored in a text file with the same name as the image, containing the annotations (bounding boxes) for the corresponding image file, that is object class, object coordinates, height, and width. Furthermore, the images were resized to (640×640). Afterward, the collected datasets were split into 80%, 13%, and 7% for training, testing, and validating respectively. Augmenting images in the training set serves various objectives, such as increasing diversity, mitigating over-fitting, enlarging the dataset, improving generalizability, and addressing diverse variations. In our study, we employed several data augmentation techniques, as outlined in Table 2.

Table 2. Data Augmentation

Blur	Up to 0.75px
Rotation	Between -15° and +15°
Crop	0% Minimum Zoom, 20% Maximum Zoom

3.3 Detection Architecture

One-stage object detection refers to a type of model that predicts the presence and location of objects in an image with a single pass through the input image. In our approach, we worked with YOLO(you only look once) with two different versions: YOLOv8n and YOLOv5n as demonstrated in Fig.1.

Essentially, the YOLO network comprises three fundamental components: the backbone, responsible for feature extraction; the neck, which focuses on aggregating features; and the head, tasked with taking output features from the neck as input and producing detections [7].

YOLO-v5 marked the initial native release of architectures within the YOLO family that were coded in PyTorch [8], departing from the traditional Darknet framework. Deep learning technology employed in YOLOv5 enables real-time and efficient object detection tasks. Compared to its predecessor YOLOv4,

YOLOv5 has seen enhancements in model architecture, training methodology, and overall performance [12].

The YOLOv5 model utilizes the CSP (CrossStage Partial) network structure, which efficiently decreases redundant computations and enhances computational efficiency. Nevertheless, YOLOv5 exhibits certain limitations. For instance, it struggles with detecting small objects and requires improvement in detecting dense objects. Furthermore, its performance in challenging scenarios like occlusion and pose variations still requires enhancement [7, 12].

In January 2023, YOLOv8 was proposed as a solution combining the best features of existing object detectors. YOLOv8 incorporates the CSP concept from YOLOv5 [15], the feature fusion method (PAN-FPN) [10, 11], and the SPPF module (Fig. 2). The C2f module, inspired by the ELAN concept in YOLOv7 and combining C3 with ELAN, was introduced in YOLOv8 [14]. This integration allows YOLOv8 to acquire a wealth of gradient flow information while still maintaining its lightweight nature [14, 12]. The CBS block is composed of a convolutional layer, batch normalization, and a SiLu activation function. Meanwhile, the SPPF block includes: a CBS block, and three Max-pooling (5×5) layers that their output concatenates with the CBS, then followed by another CBS block.

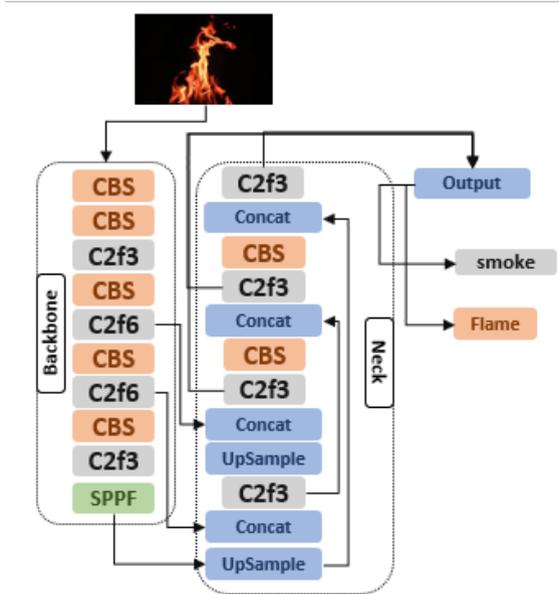


Fig. 2. Detection Architecture

4 Results and Discussion

We employed TensorFlow 2.10.1 and Python 3.10 for the execution. Yet, in the pre-processing stage, we employed the RoboFlow tool for annotation. In the training phase for each one of D3, D5, and D6 we can notice that the model's training classification loss stabilized starting from the 80th, 20th, 70th, and 130th epochs respectively, while the box loss didn't stabilize (Figs. 3, 4, 5, and 6). As for the validation classification loss values, the stabilization started from the 90th epoch for D3, at the 10 epoch for D4, the 50th epoch for D5, and the 70th for D6, however, the box loss didn't stabilize for D4, D3, and D5 but for D6 we can say that it did stabilize a bit starting from the 120th epoch.

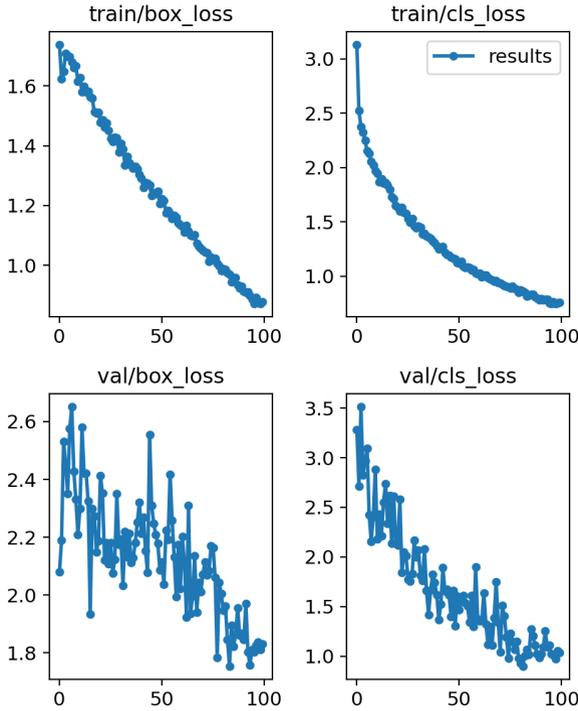


Fig. 3. D3 training results

Considering the IoU threshold, $\mathbf{a} = 0.5$

TP : $\text{IoU} \geq 0.5$

FP : $\text{IoU} < 0.5$

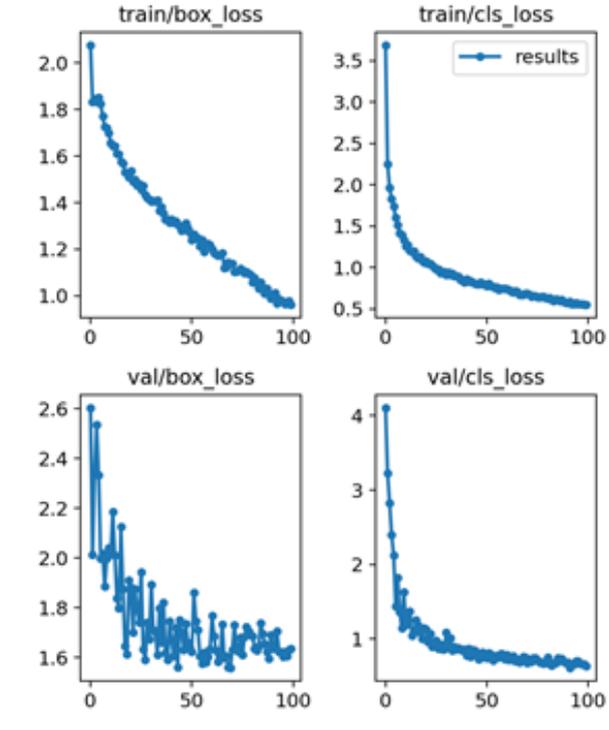


Fig. 4. D4 training results



Fig. 8. Live stream detection

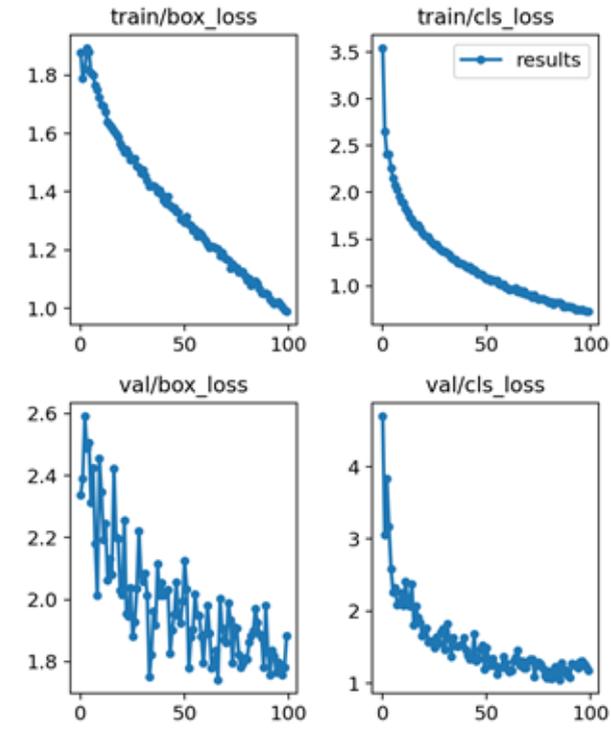


Fig. 5. D5 training results

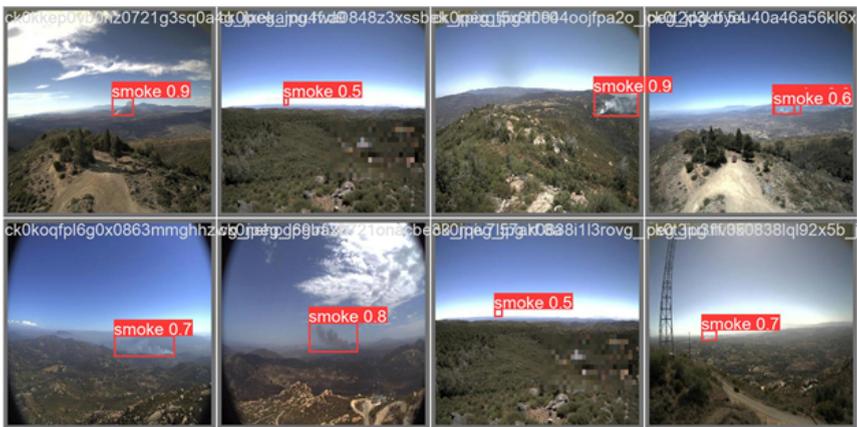


Fig. 9. Example images from the test dataset for D6

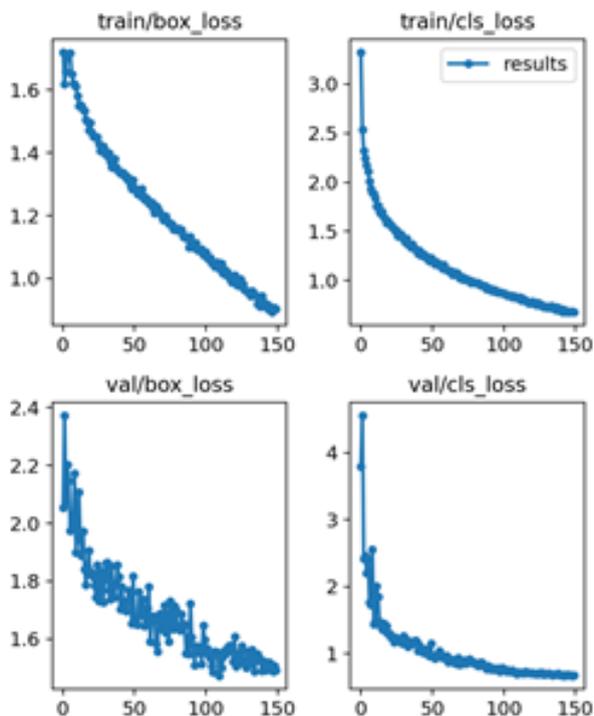


Fig. 6. D6 training results

Intersection over Union (IoU) This metric measures the degree of overlap between the identified object and the actual object present in the ground truth data. It serves as a reliable measure to verify the accuracy of the detected object with the ground truth object [16, ?] we calculate the union of the two bounding boxes and the intersection between them, with two rectangles, their coordinates are $(x1, y1, x2, y2)$ and $(x3, y3, x4, y4)$, the IoU is :

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Diagram of overlapping rectangles}}{\text{Diagram of union of rectangles}}$$

$$P1 = \text{Min}(x2, x4) - \text{Max}(x1, x3)$$

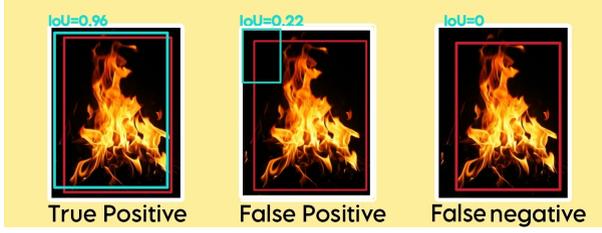


Fig. 7. Intersection Over Union.

Table 3. Yolo test results

Model	Dataset	Class	Precision	Recall	mAP@0.5	mAP@0.5-0.95
YOLOv5n	D1	All	0.64	0.36	0.34	0.131
	D2	All	0.62	0.46	0.43	0.18
YOLOv8n	D3	All	0.81	0.86	0.92	0.42
	D4	All	0.94	1	0.99	0.56
	D5	Fire	0.65	0.73	0.68	
		Smoke	0.86	0.82	0.73	
		All	0.87	0.78	0.70	
	D6	Fire	0.84	0.76	0.88	0.44
Smoke		0.95	0.92	0.97	0.56	
All		0.89	0.84	0.93	0.50	

$$P2 = \text{Min}(y2, y4) - \text{Max}(y1, y3)$$

$$AoI = P1 \times P2$$

$$P3 = (x2 - x1)(y2 - y1)$$

$$P4 = (x4 - x3)(y4 - y3)$$

$$AoU = P3 + P4 - AoI$$

IoU values vary from 0 to 1, with 0 indicating no overlap and 1 indicating perfect prediction and overlap. The IoU measure is useful because of thresholding, which means we need a threshold \mathbf{a} to verify whether the detection is correct. **Average precision** $\text{AP@}\mathbf{a}$ denotes the Area Under the Precision-Recall Curve (AUC-PR) at the IoU threshold. It is formally defined in Eq. 1. **Mean Average Precision (mAP)** The mean Average Precision (mAP) (Eq. 2) is the average of AP values over all classes, the higher the mAP the better the model's detection is [3].

$$\text{AP@}\mathbf{a} = \int_0^1 p(r) dr \quad (1)$$

$$\text{mAP@}\mathbf{a} = \frac{1}{n} \sum_n \text{AP}_i \quad (2)$$

Table 3 presents test results for YOLOv5n and YOLOv8n models across diverse datasets and classes. YOLOv8n consistently outperforms YOLOv5n, as reflected

in higher precision, recall, and mean Average Precision (mAP) scores. For instance, in dataset D6, for both classes, YOLOv8n achieved a precision of 0.89 and recall of 0.84, resulting in a mAP@0.5 score of 0.93, while YOLOv5n attained a precision of 0.64, recall of 0.46, and mAP@0.5 of 0.43. These variations highlight the importance of dataset selection and model choice for optimizing fire and smoke detection performance.

Based on the obtained evaluation outcomes, we determined that YOLOv8 using the D6 dataset had the best results. Therefore, the model was selected and implemented for fire detection in a live stream camera. Fig 8 demonstrates some fire-detected instances from the live stream.

5 Conclusion

This paper presented and evaluated a one-stage detection approach utilizing YOLOv8n and YOLOv5n for fire and smoke detection in various environments. The evaluation demonstrated the superior efficiency of YOLOv8 in real-time object detection, surpassing YOLOv5 with a mAP score of 93%. The study emphasized the importance of diverse data instances and detection architectures for enhancing model efficacy in real-world settings. Looking ahead, there are promising opportunities to apply these models practically. For example, integrating them into drone systems for forest fire detection shows potential. Additionally, incorporating these object detection frameworks into CCTV-based systems could enhance indoor fire detection capabilities. These future initiatives aim to leverage technology advancements to improve fire detection systems across different landscapes, enhancing safety and protection against fire hazards.

References

1. Boudouh, S.S., Bouakkaz, M.: Enhanced breast mass mammography classification approach based on pre-processing and hybridization of transfer learning models. *Journal of Cancer Research and Clinical Oncology* **149**, 14549 – 14564 (2023), <https://api.semanticscholar.org/CorpusID:260839918>
2. Boudouh, S.S., Bouakkaz, M.: New enhanced breast tumor detection approach in mammogram scans based on pre-processing and deep transfer learning techniques. *Multimedia Tools and Applications* (2023), <https://api.semanticscholar.org/CorpusID:261481873>
3. Chekneane, M., Bendouma, T., Boudouh, S.S.: Advancing fire detection: two-stage deep learning with hybrid feature extraction using faster r-cnn approach. *Signal, Image and Video Processing* (2024), <https://api.semanticscholar.org/CorpusID:269843519>
4. Dewangan, A., Pande, Y., Braun, H.W., Vernon, F.L., Pérez, I., Altintas, I., Cottrell, G., Nguyen, M.H.: Figlib & smokeynet: Dataset and deep learning model for real-time wildland fire smoke detection. *ArXiv abs/2112.08598* (2021), <https://api.semanticscholar.org/CorpusID:245218539>
5. liang Dou, Z., Zhou, H., Liu, Z., Hu, Y., Wang, P., Zhang, J., Wang, Q., Chen, L.J., Diao, X., Li, J.: An improved yolov5s fire detection model. *Fire Technology* (2023), <https://api.semanticscholar.org/CorpusID:264178846>

6. Enoch Arulprakash, M.A.: A study on generic object detection with emphasis on future research directions, *Journal of King Saud University - Computer and Information Sciences*, volume 34, issue 9, (2022)
7. Hussain, M.: Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines* (2023), <https://api.semanticscholar.org/CorpusID:259717436>
8. Jocher, G.R., Stoken, A., Borovec, J., NanoCode, ChristopherSTAN, Changyu, L., Laughing, tkianai, yxNONG, Hogan, A., lorenzomamma, AlexWang, Chaurasia, A., Diaconu, L., Marc, wanghaoyang, ml ah, Doug, Durgesh, Ingham, F., Frederik, Guilhen, Colmagro, A., Ye, H., Jacobsolawetz, Poznanski, J., Fang, J., Kim, J., Doan, K.N., , L.Y.: ultralytics/yolov5: v4.0 - nn.silu() activations, weights & biases logging, pytorch hub integration (2021), <https://api.semanticscholar.org/CorpusID:244999743>
9. Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review* pp. 1 – 62 (2020)
10. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 936–944 (2016), <https://api.semanticscholar.org/CorpusID:10716717>
11. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8759–8768 (2018), <https://api.semanticscholar.org/CorpusID:3698141>
12. Lou, H., Duan, X., Guo, J., Liu, H., Gu, J.J., Bi, L., Chen, H.: Dc-yolov8: Small-size object detection algorithm based on camera sensor. *Electronics* (2023), <https://api.semanticscholar.org/CorpusID:258852089>
13. Shivang Agarwal, Jean Ogier du Terrail, F.J.: Recent advances in object detection in the age of deep convolutional neural networks (2019)
14. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7464–7475 (2022), <https://api.semanticscholar.org/CorpusID:250311206>
15. Wang, C.Y., Liao, H.Y.M., Yeh, I.H., Wu, Y.H., Chen, P.Y., Hsieh, J.W.: Cspnet: A new backbone that can enhance learning capability of cnn. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 1571–1580 (2019), <https://api.semanticscholar.org/CorpusID:208310312>
16. Yandouzi, M., Grari, M., Berrahal, M., Idrissi, I., Moussaoui, O., Azizi, M., Ghomid, K., Elmiad, A.K.: Investigation of combining deep learning object recognition with drones for forest fire detection and monitoring. *International Journal of Advanced Computer Science and Applications* (2023), <https://api.semanticscholar.org/CorpusID:257966512>
17. Zhou, J., Yang, X., Zhang, L., Shao, S., Bian, G.: Multisignal vgg19 network with transposed convolution for rotating machinery fault diagnosis based on deep transfer learning. *Shock and Vibration* **2020**, 1–12 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

