A Comparative Evaluation of YOLOv11 and Faster R-CNN for Multiclass Defect Detection in Metal Casting Images

Helaine P. Neves[1], Luiz H. C. da Costa[1], Lucas A. Barbosa[1], Alex A. B. Santos[1]
[1] SENAI CIMATEC University, Salvador, Bahia, Brazil.
*Corresponding author(s). E-mail(s): **helaine.neves@fieb.org.br**;
Contributing authors: **luiz.correa@fbest.org.br**;
lucas.barbosa@fieb.org.br; **alex.santos@fieb.org.br**;

Abstract: This article presents a comparative evaluation of two state-of-the-art object detection architectures—Faster R-CNN and YOLOv11—applied to the task of classifying and localizing defects in images of metal castings, contributing to the automation of inspection in Industry 4.0. This research adapts a public database with annotated images of castings and performs data augmentation, exclusion and reclassification of classes, resulting in a set of 2,273 images divided into training, validation and testing. Both models were rigorously assessed via evaluation metric standards, including precision, recall, F1-score, and average precision (AP) over an IoU threshold of 0.5. YOLOv11 showed better performance in terms of precision and F1- score, standing out as more efficient and balanced for industrial environments that prioritize agility and a low false positive rate. On the other hand, Faster R-CNN obtained better results in terms of recall and mean average precision (mAP), being more suitable in critical scenarios where complete defect detection is essential, even with higher computational cost. The research highlights that the choice between models should consider the industrial context and the impacts of false positives or negatives on the production process.

Keywords*: Automated inspection, Convolutional neural networks, Defect detection, Faster R-CNN, Performance evaluation, YOLOv11*

1. Introduction

Object recognition is an inherently complex task, influenced by various factors such as scene constancy, image-model space variability, the number of objects in the model database, object multiplicity within images, and the presence of occlusions, among others [1]. To enable artificial intelligence models to recognize thousands of objects across millions of images, substantial learning capacity and extensive annotated datasets are required. In this context, machine learning algorithms, particularly those developed for computer vision, have been widely adopted to automate industrial tasks, including defect detection and the classification of materials and mechanical components [2][3].

Such technological advances align closely with the goals of Industry 4.0, particularly in enhancing quality control through automated surface defect identification—an activity that typically demands a degree of visual cognition. Over the past decades, object detection network architectures have undergone significant evolution. Deep learning-based techniques have emerged as the dominant approach, categorized primarily into convolutional neural network (CNN)-based and transformer-based models [4]. While transformer architecture, originally developed for natural language processing, have recently been adapted for vision tasks, such as in the Detection Transformer (DETR), which reformulates object detection as an end-to-end task through a transformer encoder-decoder mechanism [5], CNNs remain more efficient in terms of parameterization and training complexity [6].

CNNs have been employed in image recognition since the 1980s, and with the advent of increased computational power, they have demonstrated superhuman performance in complex visual applications, including autonomous driving, image retrieval, and video analysis [7]. In object detection, CNN-based models are typically categorized into two main types based on architectural design: two-stage and single-stage detectors [4]. Two-stage models—such as those based on the Region-Based Convolutional Neural Network (R-CNN)—first generate region proposals and subsequently perform classification and bounding box refinement, resulting in high detection accuracy. In contrast, one-stage models, such as You Only Look Once (YOLO), unify detection and classification into a single regression task, enabling real-time performance with reduced computational demands [8].

This study presents a comparative analysis of two CNN-based object detectors—Faster R-CNN (two-stage) and YOLOv11 (single-stage)—applied to the task of identifying and classifying casting defects. The primary contributions of this work include:

- A rigorous comparison between single-stage and two-stage detectors in an industrial defect detection context;
- A comprehensive performance evaluation using metrics such as Precision, Recall, F1-Score, Average Precision (AP), and mean Average Precision (mAP), based on the Pascal VOC protocol;
- A practical discussion on the suitability of each model for different industrial applications.

1. Faster R-CNN

Faster R-CNN significantly improved both the efficiency and accuracy of the original R-CNN framework by minimizing computational overhead. Its architecture adopts a two-stage object detection strategy. In the first stage, a convolutional neural network (CNN) functions as a backbone for feature extraction, capturing salient image attributes such as edges, textures, and structural patterns. These features are encoded into a spatial feature map, which is subsequently processed by a Region Proposal Network (RPN). The RPN employs anchor boxes of various scales and aspect ratios to identify candidate regions that are likely to contain objects. Each region is assigned an objectness score, reflecting the probability of containing a valid object. Regions with high scores are retained and passed to the second stage for further classification and refinement, while those with low scores are suppressed.

The second stage of the Faster R-CNN architecture is responsible for classification and refinement. In this phase, each region proposed by the RPN is assessed to determine the presence of an object. If an object is detected, the network assigns it a class label. Additionally, the bounding boxes generated in the previous stage are refined in terms of position and scale to improve localization accuracy. By integrating the RPN with the classification and regression layers, Faster R-CNN achieves an effective balance between detection precision and computational efficiency, establishing itself as a robust solution for multiclass object detection tasks

[9]. Prior research has demonstrated the applicability of Faster R-CNN for detecting surface defects in various materials, including steel [10], wood [11], and textiles [12].

2. YOLOv11

YOLO has transformed the field of object detection by offering a fast, efficient, and real-time solution. Unlike traditional multi-stage approaches, YOLO employs a single convolutional neural network to simultaneously predict bounding boxes and class probabilities, which greatly enhances its computational performance. This architectural simplicity, coupled with its flexibility, has established YOLO as a leading method in both academic research and industrial applications [8].

The YOLO architecture comprises three primary components: (i) the backbone, which is typically a pretrained CNN used to extract features from input images; (ii) the neck, which enhances feature representation through mechanisms such as Feature Pyramid Networks (FPNs) and Spatial Attention Modules (SAMs); and (iii) the head, responsible for predicting bounding boxes and class scores using fused features and multiscale anchor boxes to improve detection across different object sizes. The most recent version, YOLOv11, incorporates innovative modules—namely the C3k2 block and the C2PSA block—which further enhance feature extraction and processing efficiency.

3. Evaluation Metrics

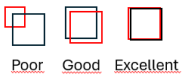
The first indicator to be considered is the intersection over union (IoU), which measures how close the boxes related to the detections are to the corresponding truth boxes or, in other words, how accurate the model is in terms of positioning its detections compared to the real position [14], as shown in Figure 1. There is a way to set a specific threshold for the IoU, below which the detection boxes are not considered good enough and should be disregarded. This study uses 0.5 as the IoU threshold, which means a minimum overlap of 50% between the ground truth box and the detection box for the detection to be considered relevant. However, the IoU is not the only parameter to be considered in the evaluation of multiclass object detection, as the model also needs to correctly predict the corresponding class of detection. For the problem studied, there are three types of possibilities for the result of each detection that can be observed in a confusion matrix [14] [15]:

- True positive (TP): corresponds to the correct detection of an existing object.
- False positive (FP) corresponds to an incorrect detection of an existing object or a detection of a nonexistent object.
- False negative (FN): corresponds to a failed detection of an existing object.

Other indicators used to evaluate the performance of object detection algorithms are as follows:

Figure 1: IoU definition

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$



Precision (Pr) corresponds to a percentage of correct predictions made in relation to the total predictions made by the model and can be calculated as:

$$\Pr = \frac{\sum_{n=1}^{N} TP_n}{TP + FP} \quad (1)$$

Recall (Re) or sensitivity corresponds to a percentage of correct predictions made in relation to the total number of existing possibilities and can be calculated as:

$$Re = \frac{\sum_{n=1}^{N} TP_n}{TP + FN} \quad (2)$$

F1 score: represents the harmonic mean between precision and recall:

$$F1 - Score = \frac{2 \cdot \Pr \cdot Re}{Pr + Re} \quad (3)$$

Considering the activity that is the object of this research, precision reflects the model's ability to correctly identify instances of defects without making too many errors. High precision is crucial in this type of task since false positives can lead to rejection of nondefective parts, increasing operational costs and reducing the efficiency of the production process. On the other hand, the sensitivity of the model represents its ability to detect all relevant instances of defects in an image. High sensitivity ensures that defects are identified and reduces the possibility of accepting defective parts as healthy, which is a critical factor in quality control scenarios where undetected defects can compromise the safety and reliability of the product. In multiclass problems, it is important to analyze these metrics by class type to assess how the model fits in different types of detection. Notably, for each detection made by the algorithm, a probability function is associated that estimates the certainty of the prediction made. This is the confidence level of detection. Similarly, it is necessary to establish a minimum threshold for the confidence level (confidence level 6 threshold), below which the detections are disregarded by the model because they have a greater probability of being false positives. A standard metric used in competitions and benchmarks is average precision (AP). It measures the quality of a model in terms of precision and recall, providing a consolidated view of performance. To calculate the average precision, it is necessary to list the detections by confidence level, in decreasing order, and calculate the accumulated precision and recall of the model. After this, the precision × recall curve (PR curve) is plotted. The AP is the area below this curve. According to [14], a good object detector should be considered good if its precision remains high as its recall increases, that is, if it can locate all

relevant objects without making many errors. A larger area under the PR curve tends to mean high precision and high recall; therefore, the higher the AP of a model is, the better it is at the activity. To calculate the AP, one can perform the integral of the curve or make an approximation through the interpolation of all points that can be calculated via the formula below:

$$AP = \sum (R_n - R_{n-1}) \cdot P_n \quad (4)$$

In multiclass problems, the average AP of each class is calculated, resulting in the mAP indicator, which provides a comprehensive assessment of the model's effectiveness in identifying defects in all specified classes:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \quad (5)$$

Where, N is the number of classes.

Methodology

The chosen dataset was originally downloaded from the Roboflow platform from the project called the "Casting detection Computer Vision Project" [16] and consists of 4,278 images of castings appropriated in 7 different classes and with their corresponding annotation text files. The following adjustments were made: 1. Exclusion of images that contain more than 4 markings; 2. Exclusion of two classes of defects: scratches and deformations; 3. Renaming of the polished class to avoid defects. By changing and omitting conditional classes in the original dataset, a web application was developed using ES6 JavaScript to reannotate these images according to the new database configuration. The images were also standardized and resized to 640x640 pixels to maintain a consistent input size. With this reorganization, the new database configuration has 2,273 images divided into three groups according to the 7 model implementation phase: training (composed of 1958 images), validation (composed of 205 images) and testing (composed of 110 images). There are images with more than one class's annotation, and the database balance by classes follows the distribution illustrated in Table 1. There are five types of labels per appointment, as shown in Figure 2.

Figure 1: Examples of parts with labels according to the identified dataset classes.

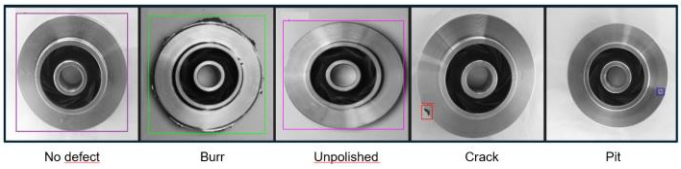


Table 1: Database configuration, after adjustments

Id.	Class	Training	Validation	Testing
0	Burr	223 (8%)	21 (7%)	9 (6%)
1	Crack	773 (29%)	94 (33%)	41 (27%)
2	Pit	481 (18%)	54 (19%)	28 (19%)
3	Unpolished	420 (16%)	48 (17%)	26 (17%)
4	No Defect	780 (29%)	68 (24%)	45 (30%)

The confusion matrices of the results obtained after processing by the Faster R-CNN and YOLO v11 models are shown in Figures 5 and 6 at the end. In the main diagonal, the first number represents the number of true positives followed

by the result and recall indicators for each label. The "background" label in this matrix was added to map the false positives related to the detection of nonexistent objects or objects that were positioned incorrectly (identified in the background column) and the false negatives (identified in the background row).

The results of the precision, recall, F1 score, and average precision (AP) indicators are presented in Table 2. In general, except for the crack and pit classes, YOLOv11 presented superior performance than Faster R-CNN. The precision × recall (PR) curves for each class are presented in Figures 3 and 4. For both models, the precision decreases rapidly with increasing recall for cracks and pits, which demonstrates an opportunity to improve them in identifying these classes.

Table 2: Results of indicators after testing the YOLOv11 and Faster R-CNN Models

Metric	Model	Burr	Crack	Pit	Unpolished	No Defect
Precision	YOLO	**1.000**	**0.815**	0.671	**0.932**	**0.982**
	Faster	0.909	0.694	**0.845**	0.677	0.653
Recall	YOLO	1.000	0.675	0.724	0.767	0.941
	Faster	1.000	**0.807**	**0.732**	**0.851**	**1.000**
F1-Score	YOLO	**1.000**	0.738	0.696	**0.841**	**0.961**
	Faster	0.952	**0.747**	**0.784**	0.754	0.790
AP	YOLO	1.000	0.518	0.608	**0.753**	**0.987**
	Faster	1.000	**0.629**	**0.681**	0.740	0.981
F1-Score Avg.	YOLO	**0.847**				
	Faster	0.806				
mAP	YOLO	0.773				
	Faster	**0.806**				

Overall, although YOLOv11 managed to achieve better precision in identifying almost all classes (except for pit), Faster R-CNN achieved better recalls in all classes (except for burrs, which both models were able to correctly identify all images of parts that had this class), suggesting that it is more sensitive and detects more positive cases. Regarding the F1 score, there is a clear advantage of YOLOv11 over Faster RCNN, with the exception of the pit class, which means that it presents a better balance between precision and recall. On the other hand, considering the results obtained for average precision (AP), Faster R-CNN has an advantage, which suggests better aggregate performance per class at different confidence thresholds. The main objective in an automated inspection task is to detect defects with high reliability, avoiding both:

- False negatives: defects that go unnoticed (serious consequences such as dissatisfaction, risk of accidents, rework).
- False positives: good parts discarded for no reason (waste).

YOLOv11 delivers superior performance in terms of precision and F1 score, which is ideal for avoiding unnecessary discarding of good parts, in addition to having a better balance between detecting defects and not misclassifying them. On the other hand, Faster R-CNN showed better sensitivity and the ability to detect more defects, which can be considered in more critical inspections (for example, parts that will be used in medicine or aeronautics), where shipping a

defective part has more serious consequences than discarding a part in good condition.

Figure 3: PR curves for the Burr, crack, pit, unpolished and no defect classes according to the Faster R-CNN model.

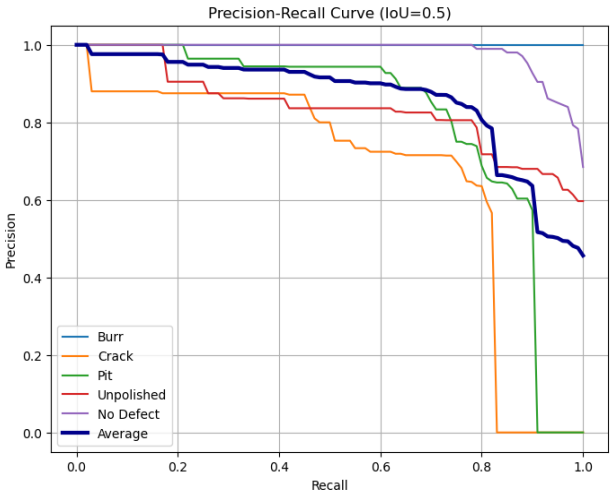
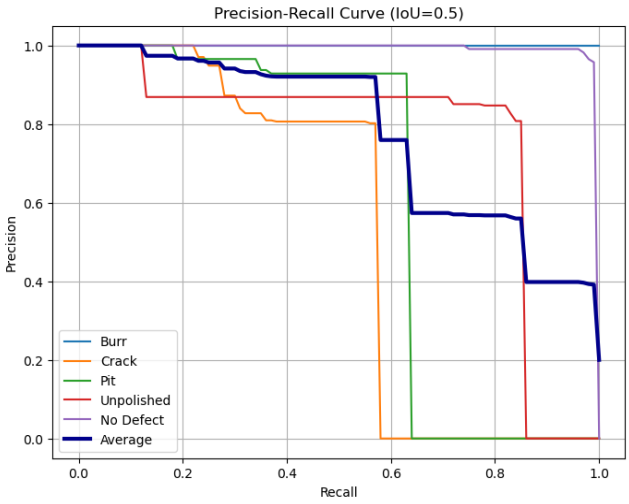


Figure 4: PR curves for the Burr, crack, pit, unpolished and no defect classes according to YOLOv11.



Conclusion

This study compared the performance of the Faster R-CNN and YOLOv11 models in the detection and classification of defects in castings, highlighting their performance in terms of precision, recall, F1 score and average precision. In general, especially in industrial environments with large-scale production, accuracy and speed are essential. In this way, YOLOv11 delivers superior performance, making it more reliable for automatic classification. In addition, the lower false positive rate helps reduce the cost of rework and the waste of good parts. However, if the cost of a single failure going unnoticed is very high, Faster R-CNN can still be considered, even with a higher total cost. For future work, techniques to expand the training database should be used to improve the performance of the models studied, with the aim of increasing data quality by reviewing current annotations, balancing classes and expanding the dataset. It is also recommended to evaluate hybrid or transformer-based models, where an improvement in the overall performance in defect detection is expected.

Figure 5: Confusion matrix of the Faster R-CNN model.

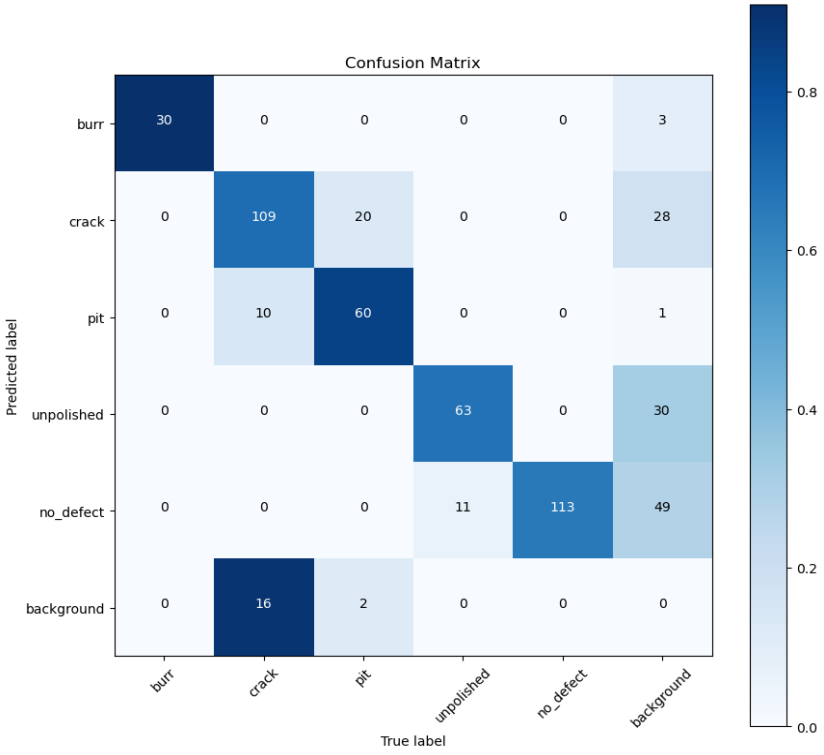
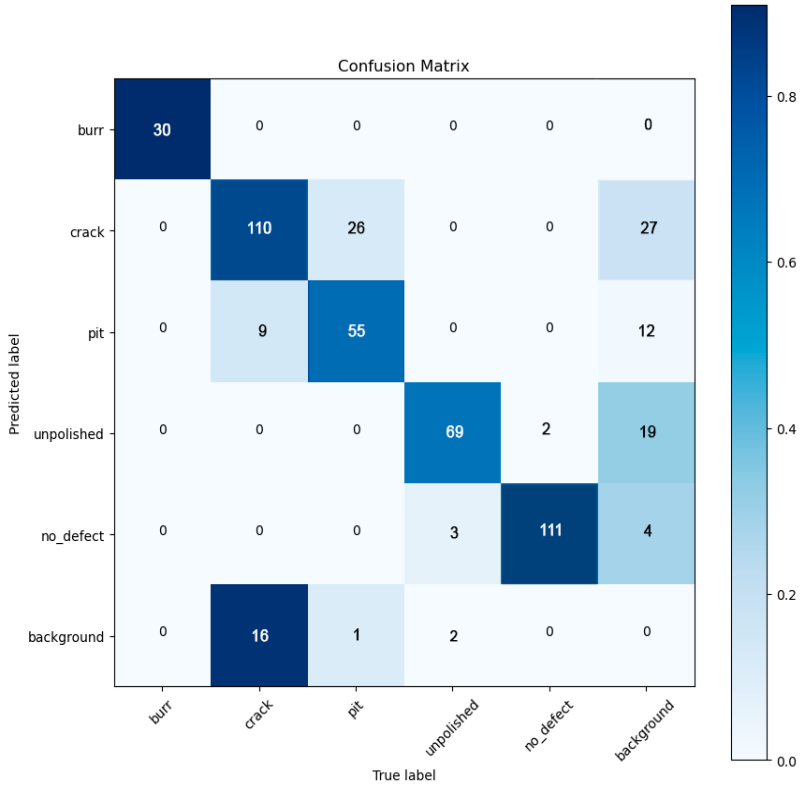


Figure 6: Confusion matrix of the YOLOv11 model.



References

[1] Jain R, Kasturi R, Schunck BG. *Machine vision. Chapter 15: Object recognition*. New York: McGraw-Hill Science; 1995.

[2] Ahmed I, Jeon G, Piccialli F. *From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where*. IEEE Trans Industr Inform. 2022;18(8):5031-42.

[3] Villalba-Diez J, Schmidt D, Gevers R, Ordieres-Meré J, Buchwitz M, Wellbrock W. *Deep learning for industrial computer vision quality control in the printing industry 4.0*. Sensors. 2019;19(18):3987.

[4] Yibo S, Zhe S, Weitong C. *The evolution of object detection methods*. Eng Appl Artif Intell. 2024;133:108256.

[5] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. *End-to-end object detection with transformers*. In: Proceedings of the European Conference on Computer Vision (ECCV); 2020. p. 213-229.

[6] Krizhevsky A, Sutskever I, Hinton GE. *ImageNet classification with deep convolutional neural networks*. Commun ACM. 2017;60(6):84-90.

[7] Géron A. Mãos à Obra: Aprendizado de Máquina Com Scikit-Learn, Keras TensorFlow. São Paulo: Altas Books; 2021.

[8] Ali ML, Zhang Z. *The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection*. Computers. 2024;13(7):171.

[9] Ahmad HM, Rahimi A. *Deep learning methods for object detection in smart manufacturing: A survey*. J Manuf Syst. 2022;64:491-510.

[10] Shi X, Zhou S, Tai Y, Wang J, Wu S, Liu J, et al. *An improved faster R-CNN for steel surface defect detection*. In: IEEE 24th International Workshop on Multimedia Signal Processing (MMSP); 2022. p. 1-6.

[11] Kodytek P, Bodzas A, Bilik P. *A large-scale image dataset of wood surface defects for automated vision-based quality control processes*. F1000Research. 2021;10:581.

[12] Liu Q, Wang C, Li Y, Gao M, Li J. *A fabric defect detection method based on deep learning*. IEEE Access. 2022;10:4284-4294.

[13] Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. *The Pascal Visual Object Classes (VOC) challenge*. Int J Comput Vision. 2010;88(2):303-338.

[14] Padilla R, Passos WL, Dias TLB, Netto SL, Da Silva EAB. *A comparative analysis of object detection metrics with a companion open-source toolkit*. Electronics. 2021;10(3):279.

[15] Padilla R, Netto SL, Da Silva EAB. *A survey on performance metrics for object-detection algorithms*. In: Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP); 2020. p. 237-242.

[16] New-workspace-kmz9b R. *Casting detection Computer Vision Project* [Internet]. Roboflow Universe; [cited 2024]. Available from: https://universe.roboflow.com/new-workspace-kmz9b/castingdetection-leboi