ARCHIVES
of
FOUNDRY ENGINEERING

ISSN (2299-2944)
Volume 2026
Issue 1/2026

165 – 176

10.24425/afe.2026.157983

20/1

Published quarterly as the organ of the Foundry Commission of the Polish Academy of Sciences

Detection of Surface Defects in Metallic Materials Using Convolutional Neural Networks with YOLO Architecture

R. Bumbálek, T. Zoubek, J. Majerník *, J. de Dieu Marcel Ufitikirezi, S.N. Umurungi, K. Šramhauser, F. Špalek
University of South Bohemia in České Budějovice, Faculty of Agriculture and Technology, Czech Republic
* Corresponding author: E-mail address: jmajernik@fzt.jcu.cz

Received 30.07.25; accepted in revised form 08.10.25; available online 30.03.2026

Abstract

The early detection and classification of surface defects in metallic materials is essential for ensuring product quality and reliability in industrial production. In this study, we evaluated and compared five recent versions of the YOLO convolutional neural network architecture (YOLOv8x, YOLOv9e, YOLOv10x, YOLOv11x, and YOLOv12x) applied to five publicly available datasets specialized in metallic defect detection (AlcastXray, Castings, GC10-DET, NEU-SEG, and Severstal SDD). All models were trained under uniform conditions and assessed using Precision, Recall, mAP50, mAP50–95, and deployment-oriented efficiency metrics, including inference speed, throughput, GPU memory usage, and power draw. The results show that YOLOv9e consistently achieved the highest detection accuracy (mAP50 up to 0.982), while YOLOv8x provided the fastest inference (>47 FPS), making it suitable for real-time applications. YOLOv10x, the most lightweight model, delivered a favorable trade-off between accuracy and computational efficiency, suggesting strong potential for edge deployment. These findings provide practical insights for selecting YOLO-based architectures according to specific industrial requirements in metallic surface defect inspection.

Keywords: Computer vision, Defect detection, CNN, Industry 4.0, NDT

1. Introduction

Computer vision, which is a fundamental component of the Industry 4.0 concept, is utilized in various parts of the production process, such as automation and optimization of partial activities, object detection and tracking, robotics, and inspection or quality control of products [1]. Manual defect inspection is highly time-consuming, costly, and inefficient; however, using image processing methods based on deep learning, such as convolutional neural networks, can eliminate these disadvantages [2, 3]. He et al. (2020) [4] divide automatic defect inspection into defect classification, which identifies whether defects are present in the image, and defect detection, which localizes individual defects in the image, marks them with bounding boxes, and assigns them to specific categories. In defect inspection, segmentation convolutional neural networks (CNNs) can also be applied, which divide the image into subregions to differentiate the observed objects from the background, thereby precisely determining their boundaries [5].

The deep learning process can be divided into supervised, semi-supervised, and unsupervised learning. The supervised approach requires extensive training datasets of images with annotations and is widely used; its most well-known algorithms include Fast R-CNN, Mask R-CNN, FCN, and YOLO [6]. In contrast, unsupervised methods use unannotated datasets for

training, where models automatically extract data characteristics and classify them based on various image features. Examples include GANs, DBNs, and Self-organizing Maps. The semi-supervised, or weakly supervised, approach combines the two, with only a small portion of the images in the training dataset being annotated [7]. Unlabelled images are gradually annotated by the trained model and reused for retraining [8], or the student-teacher paradigm is applied, as in the case of ANP-Net [9]. CNN-based detection methods are classified based on the localization and classification process into one-stage methods, which analyze the entire image at once, and two-stage methods, which first generate region proposals that are then further processed [10]. Among the most prominent two-stage detectors are R-CNN and R-FCN networks, while the most significant one-stage architecture is YOLO, which also includes models like TD-Net [11, 12].

Computer vision methods for non-destructive defect detection are applied across a wide range of industries; from electronics, where they are used for detecting defects in PCBs [13], wafers [14], photovoltaic cells [15], or electrical insulators [16]; to the food industry, for detecting defects in apples [17, 18], kiwi [19], or soybean seeds [20]; to mechanical engineering and metallurgy, where they are primarily applied in detecting surface defects in steel components [21–27], welds [28], aluminum product surfaces [29–30], and other metal parts such as wind turbine blades [31] or bearings [32]. Beyond defect detection, machine vision and deep learning are also being investigated for microstructure recognition and classification tasks in metallurgical contexts, as demonstrated by Szatkowski et al. (2023) [33], who analyzed the potential of various algorithms for microstructure type classification. These methods can also be used to detect defects in wood materials [34], tires [35], fibers [36], or in construction for detecting road defects [37–39].

Computer vision implementing deep learning methods also plays a major role in casting defect inspection, where X-ray images are often used to monitor internal material defects [40, 41], while RGB images are used for inspecting surface anomalies [42, 43]. Convolutional neural networks and image transformers are commonly applied. For example, Zhang et al. (2024) [44] developed the MFF attention-based DETR, which uses a multi-scale feature fusion module to link multi-dimensional semantic information with low-level representative features to detect defects in turbine blades of aircraft engines from X-ray images. Their method achieved a mean Average Precision at a threshold of 0.50 (mAP50) of 89.3, surpassing the original DETR's mAP50 of 86.3. Wang et al. (2025) [45] also focused on turbine blade defects, using their own dataset of 2,280 images of defective objects and applying their Spatial-cross Dual Attention Pyramid Vision Transformer Detector (SDA-PVTDet) architecture, which achieved an mAP50 of 93.6, outperforming DAB-DETR, RT-DETR with ResNet-101 backbone, and Mask R-CNN with Swin-S backbone. They also compared their model against the public Al-Cast dataset and models like YOLOv5, outperforming YOLOv5x by 4.5 in mAP50. Tang et al. (2021) [46] used a convolutional autoencoder to process unstructured X-ray image data of castings, containing both defect-free and defective products, achieving an accuracy of 97.45% in identifying defective parts. Du et al. (2021) [47] introduced a two-stream CNN for semantic segmentation that highlights pixel-level anomalies and consists of a two-stream encoder module (TSEM), pyramid pooling and gated multilayer fusion modules (PPM and GMFM), and a decoder. To enhance defect visibility in input images, they implemented the CLAHE method, which adjusts contrast locally without increasing noise. Applied to a dataset with six types of automotive cast parts, their model outperformed segmentation networks like Deeplabv3+ and UNet++, achieving a mean Intersection over Union (mIoU) of 42.2 compared to 36.5 and 40.1, respectively. Automotive parts were also the focus of Pu et al. (2024) [48], who targeted surface defects using RGB images and a detection transformer-based enhanced method called Casting-DETR, which achieved an mAP50 of 91.2, significantly outperforming DETR's 56.3, while maintaining a similar inference time (91.5 ms vs. 90.1 ms). A lower mAP50 was recorded using an unspecified YOLOv8 model (88.1) but it had a significantly faster inference time of just 29.1 ms. An improved model, OHSW-YOLO, was derived from YOLOv8n and developed by Wang and Ye (2025) [49] for a custom RGB image dataset containing six classes of aluminum casting surface defects. The model achieved an mAP50 of 93.9, outperforming YOLOv8-v11, whose best result was model v8 with mAP50 92.4, but still slower in detection with 11 ms inference time compared to 8.7 ms of OHSW-YOLO. Wu et al. (2025) [50] also enhanced YOLO architecture, creating RBS-YOLO based on YOLOv5s by implementing ShuffleNetV2 instead of the native backbone, using the CBAM attention mechanism, and replacing the SiLU activation function with ReLU. This model was applied to an RGB image dataset of iron and aluminum castings, including engine blocks and cranks and bionic hand models. The modifications reduced the parameter count from 13.7M to 7.6M, though the number of processed frames per second and the mAP50 did not increase. The new model achieved an mAP50 of 0.784 at 58.82 fps, while the original model achieved 0.828 at 76.92 fps.

Although YOLO architectures are among the most prominent one-stage detectors and have achieved state-of-the-art performance in many object detection tasks, their direct application to metallic surface defects poses challenges. Surface defects are often characterized by low contrast, irregular geometries, and small size relative to the overall image resolution, which can reduce detection accuracy. As a result, performance is not uniform across datasets or defect categories, and it is necessary to evaluate how successive YOLO versions address these limitations. In this study, we benchmarked five recent YOLO architectures (YOLOv8x, YOLOv9e, YOLOv10x, YOLOv11x, and YOLOv12x) across multiple publicly available metallic defect datasets. The aim of this work was to compare their relative strengths and weaknesses in terms of detection accuracy, generalization ability, and computational efficiency, thereby providing practical insights into model selection for industrial inspection workflows.

2. Methodology

In this section, the datasets used in this study (Al-cast, GDXray-Castings, GC10-DET, NEU-SEG, and Severstal: Steel Defect Detection) are introduced in detail, along with the training process, the configuration of training hyperparameters, and the metrics used for evaluating the accuracy of the trained models.

2.1. Description of the datasets used

To train and evaluate deep learning models for defect detection in metallic materials, five publicly available datasets were used. These datasets cover various metallic materials (aluminum and steel), imaging modalities (X-ray and grayscale), and defect types (internal and surface), thus providing a comprehensive foundation for training robust models. The images themselves, which make up the individual datasets, were not modified in any way, for example by augmentation. However, images without defects were removed and only classes relevant to this study were selected. The images in the datasets were then divided into training and validation sets, and the annotations were converted to YOLO format. A more detailed description of the modifications made is provided below for each dataset.

The Al-Cast dataset contains X-ray images of seven different high-pressure aluminum castings. 374 images that did not contain any defects were removed from the original dataset. A total of 3,024 images were used, with 2,592 designated for training and 432 for validation. The division of images into training and evaluation sets was random. The original resolution of the images was 1000 × 1000 pixels, which was resized to 640 × 640 pixels for training purposes. The dataset includes two defect classes relevant to industrial contexts: gas porosity (2,072 instances—1,776 for training and 296 for validation) and shrinkage porosity (3,737 instances—3,224 for training and 513 for validation). These defects are commonly found in HPDC (High Pressure Die Casting) processes and indicate casting quality issues such as trapped gases or incomplete solidification. Parlak and Emel (2023) [51] introduced this dataset in their publication focused on classifying casting defects using deep learning techniques.

The GDXray-Castings dataset is a subset of the broader GDXray database. It consists of X-ray images of aluminum castings forming automotive components such as wheels and joints, which exhibit visible defects. From the original 2,727 images in the "Castings" category, 997 defective samples were selected—854 for training and 143 for validation. The division of images into training and evaluation sets was random. All images were resized to 640 × 640 pixels. The dataset includes annotations for a single generic class labelled "defect," with a total of 3,273 instances (2,774 for training and 499 for validation). The original annotation file was in *.txt format and consisted of five columns. The first column recorded the image number; the second and third columns indicated the x-coordinates of the bounding box; and the fourth and fifth columns contained the y-coordinates of the bounding box. This format was then converted to YOLO format. This dataset was first introduced by Mery et al. (2015) [52] and is commonly used for benchmarking defect detection algorithms in non-destructive testing scenarios.

The GC10-DET dataset focuses on detecting surface defects in hot-rolled strip steel. The original dataset comprises 3,570 grayscale images covering ten defect categories; however, for this study, five representative defect classes were selected: crescent gap (277 instances, 245 for training and 27 for validation); oil spot (569 instances, 511 for training and 58 for validation); punched hole (340 instances, 302 for training and 38 for validation); water spot (361 instances, 316 for training and 45 for validation); and welding line (488 instances, 468 for training and 20 for validation). The resulting subset includes 1,245 annotated images (1,128 for training and 117 for validation), all resized to 640 × 640 pixels. The original dataset already included the division of images into training and validation data. The COCO annotations were converted to the YOLO format. These defects typically arise during rolling or finishing processes in steel manufacturing. The dataset was introduced by Lv et al. (2020) [53] and serves as a benchmark for evaluating the accuracy and reliability of neural networks in surface defect detection.

The NEU-Seg dataset version from Roboflow (2023) is derived from the original NEU dataset for surface defects and contains grayscale images of hot-rolled steel strip surfaces. The subset used includes 1,744 images, originally with a resolution of 200 × 200 pixels, which were resized to 640 × 640 pixels for uniformity. The dataset covers three defect categories: inclusions (1,764 instances—1,544 for training and 220 for validation), patches (1,160 instances—970 for training and 190 for validation), and scratches (1,242 instances—1,014 for training and 228 for validation The original dataset already included the division of images into training and validation data. While the original dataset includes pixel-level segmentation masks, this study considers only image-level labels. Only the bounding boxes were extracted from the original COCO annotation format when it was converted to YOLO. The original dataset was introduced by Dong et al. (2020) [54], and the version used here is publicly available on the Roboflow platform [55].

The Severstal: Steel Defect Detection dataset was originally created for a Kaggle competition [56], from where it is available for download. It contains high-resolution grayscale images (1600 × 256 pixels) of steel surfaces annotated with five different defect types. Only images containing defects were used in this study. A sliding window technique with a 128-pixel stride was applied to extract 256 × 256 pixel patches, which were then resized to 640 × 640 pixels. This preprocessing resulted in 23,208 training images and 5,803 validation images. The dataset includes the following defect classes, as named in the publication by Demir and Parlak (2025) [57]: rolled-in scale (5,103 instances in training and 1,386 in validation images), cracks (518 instances in training and 117 in validation images), scratches (26,963 instances in training and 6,755 in validation images), and patches (4,077 instances in training and 967 instances in validation images). The original dataset already included the division of images into training and validation data. The original annotation file was in *.csv format and consisted of a table with three columns. The first column contained the image name; the second represented the defect class; and the third encoded the coordinates of the pixel mask in the format c_1, n_1, c_2, n_2, ..., c_n, n_n. Here, c_1, c_2, ..., c_n represent the order of pixels from the top-left corner. When processing columns from left to right, pixel numbers are counted first within the column, from top to bottom. This means that the first pixel in the next column will have a value one greater than the last pixel in the previous column. The variables n_1, n_2, …, n_n provide information on how many subsequent pixels also form the defect mask. It is possible to decode this notation into the x, y coordinate system based on equation 1.

$$[x, y] = \left[\frac{c - c\ mod(h-1)}{(h-1)}, c\ mod(h-1)\right] \quad (1)$$

where c is the pixel order counted from the upper left corner of the image and h is the height of the image in pixels. After decoding the original annotation format into the x, y coordinate system, binarized images were created where 0 indicates the background and 1 represents the area of the defect. The coordinates of the polygonal boundaries of the defects in the images were then extracted and used to generate bounding boxes. Finally, the annotations obtained in the form of bounding boxes were converted to the YOLO format. This dataset offers a large-scale and highly variable basis for evaluating the generalization capabilities of surface defect detection systems.

Together, these datasets provide a comprehensive and heterogeneous testing environment for training and validating deep learning models for defect detection, covering multiple defect types, imaging modalities, and domain-specific challenges. To further illustrate dataset variability, Figure 1 shows the distribution of object sizes relative to image dimensions, while Figure 2 presents the frequency of defect classes. Both figures are original visualizations generated from the datasets analyzed in this study.

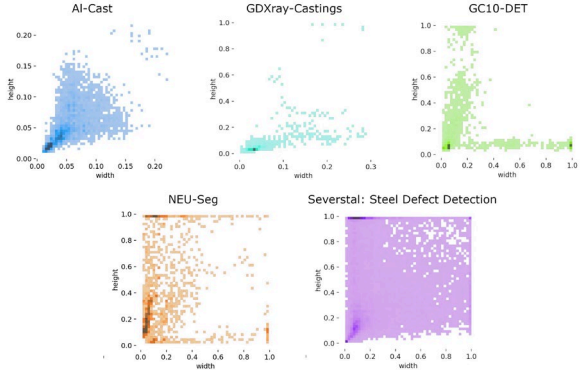

Fig. 1. Visualization of individual object sizes in the dataset relative to the overall image size

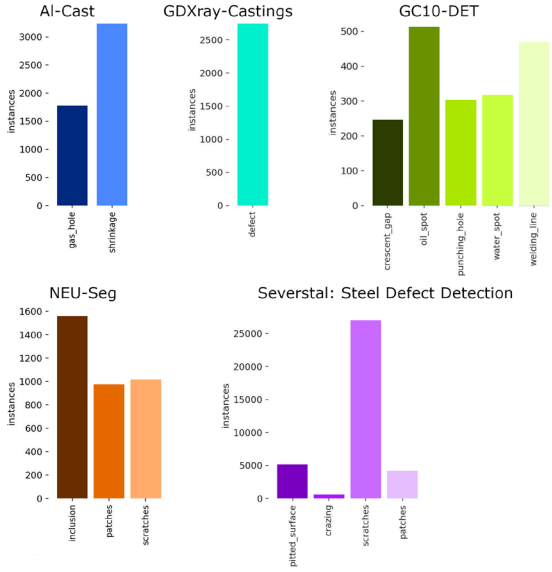

Fig. 2. Visualization of the distribution of object classes across the datasets used

2.2. Description of the training process

For the training process, convolutional neural network models of the YOLO architecture type were selected, specifically the models YOLOv8x, YOLOv9e, YOLOv10x, YOLOv11x, and YOLOv12x. The number of parameters for these networks is presented in Table 1. Training was conducted on a computational GPU server equipped with an NVIDIA A100 graphics card, using the Debian 10 operating system. All selected YOLO-type CNN models were trained for 200 epochs with the following hyperparameter settings: a learning rate of 0.01 and a batch size of 32. The input image size for the training dataset was 640 × 640 pixels. The optimizer used in the training process was SGD (Stochastic Gradient Descent). These settings were consistent across all five types of CNNs and all input datasets.

The main evaluation metrics used to compare the performance of the individual networks were Precision (P), defined by equation (2), Recall (R), calculated using equation (3), and Mean Average Precision (mAP), represented by equation (4).

$$P = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{FP_i+TP_i} \tag{2}$$

$$R = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{FN_i+TP_i} \tag{3}$$

$$mAP = \frac{\sum_{i=1}^{n}\int_0^1 P_i(R_i)dR_i}{n} \tag{4}$$

In these equations, n denotes the total number of object classes in the dataset. For each class i, TP_i represents the number of true positives (correctly detected defect instances), FP_i the number of false positives (incorrect detections where no defect exists), and FN_i the number of false negatives (missed defect instances). Precision (P) expresses the fraction of correctly detected positive samples relative to all detected samples, while Recall (R) indicates the fraction of correctly detected samples relative to the total number of ground-truth instances [58]. The term $P_i(R_i)$ denotes the precision as a function of recall for class i, and its integral corresponds to the area under the precision–recall curve, also known as the Average Precision (AP). The mean Average Precision (mAP) is the mean of AP values across all classes [59]. In this study, mAP was computed at both a fixed IoU threshold of 0.5 ($mAP50$) and across multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05 ($mAP50–95$).

Table 1.
Comparison of the number of parameters in the YOLO-type CNN model architectures.

Type of CNN	Number of parameters (mil.)	Source
YOLOv8x	68.2	[60]
YOLOv9e	58.1	[61]
YOLOv10x	29.5	[62]
YOLOv11x	56.9	[63]
YOLOv12x	59.1	[64]

3. Results and Discussion

This section presents a comprehensive evaluation of the most recent YOLO-based object detection models (YOLOv8x, YOLOv9e, YOLOv10x, YOLOv11x, and YOLOv12x) applied to five publicly available datasets relevant to surface defect detection: AlcastXray, Castings, GC10-DET, NEU-SEG, and SSDD. We assessed each model's performance based on key metrics such as Precision, Recall, mAP50, and mAP50–95, alongside practical deployment-oriented resource metrics, including inference latency, throughput (FPS), model size, GPU memory usage, and power draw.

3.1. Comparative Evaluation of YOLO Models on Surface Defect Datasets

The Al-Cast dataset contains X-ray radiographic images of aluminium castings labelled with typical internal casting defects, primarily gas porosity (gas holes) and shrinkage cavities. Among the evaluated models, YOLOv9e demonstrated the strongest overall performance on the "all" class, with a Precision of 0.971, Recall of 0.972, and mAP50 of 0.982. It also maintained the highest mAP50–95 (0.564), making it the best-performing model on this dataset. YOLOv12x and YOLOv11x followed closely, each exceeding 0.96 in Precision and Recall, and mAP50 above 0.98. The training curves (see Figure 3) consistently demonstrated smooth and stable convergence for all models.

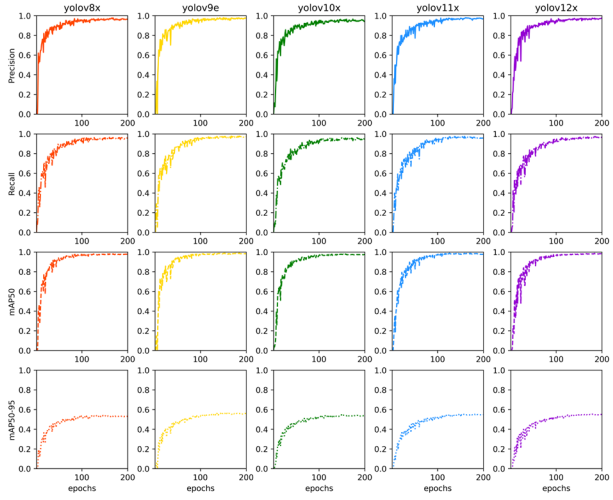

Fig. 3. Training curves on the Al-Cast dataset

When the detection task was broken down by defect type, it became evident that shrinkage defects were easier to identify than gas holes. This can be attributed to the fact that shrinkage defects tend to have larger, more distinct boundaries, whereas gas porosity often appears as small, low-contrast circular voids that are harder to distinguish from background noise. In fact, YOLOv12x achieved particularly high performance for shrinkage detection, reaching a Precision of approximately 0.984, a Recall close to 0.982, and an mAP50 of about 0.993. Conversely, detection of gas holes, while still robust, showed slightly lower performance, with the best mAP50 (0.983) observed with YOLOv11x. Figure 4 provides the class-wise detection results, highlighting the differences between shrinkage and gas porosity across models. Figure 5 illustrates example inferences on Al-Cast images. The first two examples depict shrinkage cavities, which all models successfully detected, differing mainly in confidence scores. The third example, however, contains multiple small gas porosity defects, which posed greater difficulty. None of the models identified all defects correctly: YOLOv8x, YOLOv10x, and YOLOv12x each missed only one defect, whereas YOLOv9e and YOLOv11x showed lower sensitivity to smaller porosity regions. This reflects a broader trend that small, low-contrast defects remain more challenging for YOLO-based models, even when trained on specialized datasets.

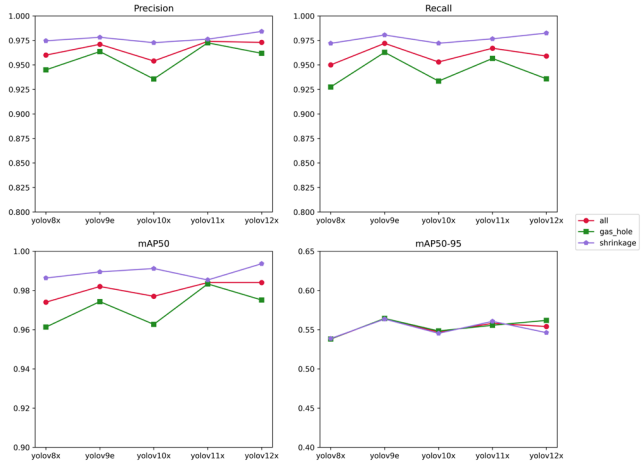

Fig. 4. Class-wise detection performance on the Al-Cast dataset

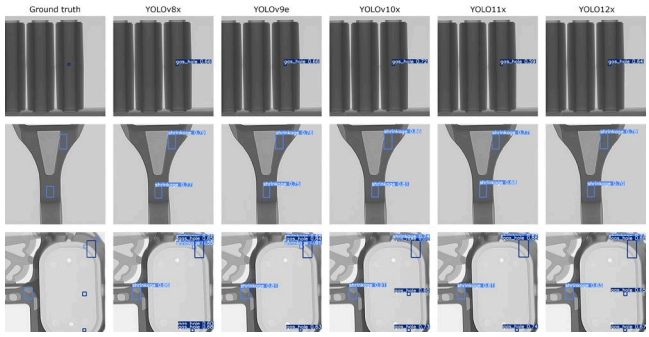

Fig. 5. Example of inference on the Al-Cast dataset

The GDXray-Castings dataset consists of X-ray images of aluminium castings labelled with surface-level defects. The dataset supports binary classification under a single "defect" class. Among the evaluated models, YOLOv10x achieved the highest mAP50 (0.914) and robust Precision (approx. 0.913), while YOLOv12x recorded the highest Precision (approx. 0.932). YOLOv9e outperformed others in Recall (0.874), suggesting better sensitivity to true defect instances. Despite variations in individual metrics, the mAP50–95 values across all models clustered within a narrow range (0.59–0.61). This consistency suggests that all the models are comparably effective at localising defect boundaries, a crucial factor in industrial applications. The training curves (see Figure 6) show consistent convergence across all models, confirming stable training behaviour.

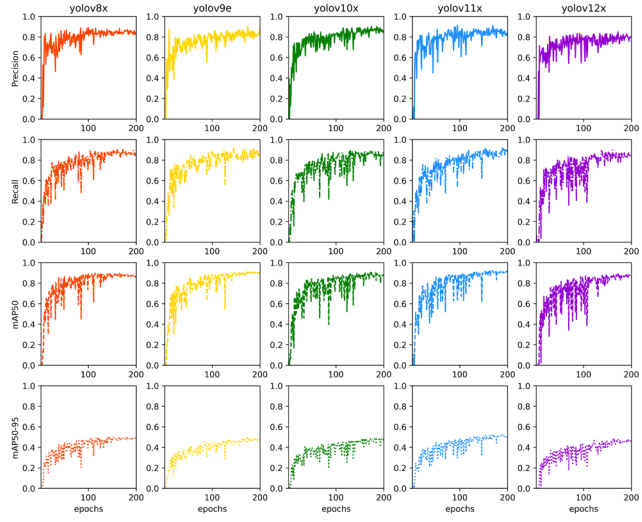

Fig. 6. Training curves on the GDXray-Castings dataset

As the dataset does not distinguish among defect categories, the evaluation focused on the binary task of detecting whether a defect is present. Figure 7 presents the benchmark comparison of detection performance across models, showing minor differences in precision–recall trade-offs. Figure 8 presents another set of inference examples from the trained models (YOLOv8x, YOLOv9e, YOLOv10x, YOLOv11x, and YOLOv12x). All models were able to correctly detect the actual positions of the defects in all three presented cases. The only variation among the models was the confidence level of their predictions. In the first and third examples, where only one object was present in each image, YOLOv10x showed the highest prediction confidence. In the second example, where four objects were to be detected, YOLOv10x again demonstrated the highest confidence for three out of four objects, but the lowest for the last one. This indicates a possible trade-off between robustness across multiple instances and confidence in isolated detections.

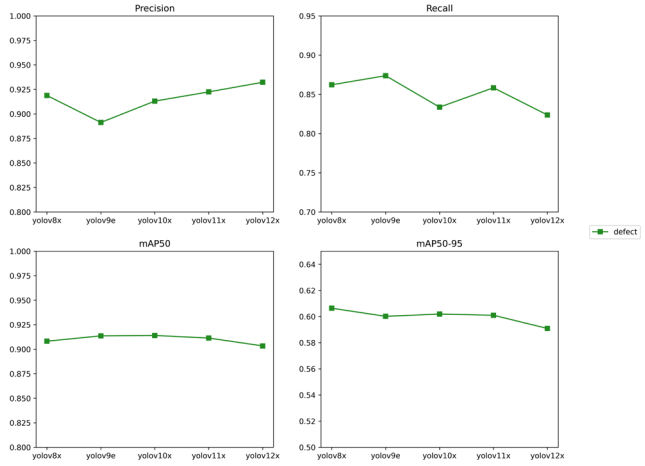

Fig. 7. Class-wise detection performance on the GDXray-Castings dataset

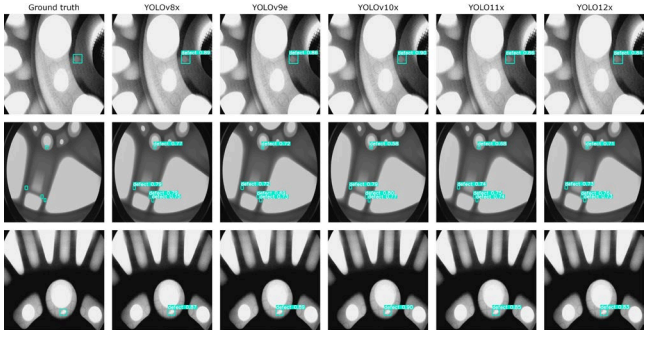

Fig. 8. Example of inference on the GDXray-Castings dataset

Overall, the results suggest that while all YOLO architectures were effective for binary defect detection in X-ray castings, YOLOv9e was more sensitive to defect presence (higher Recall), whereas YOLOv10x and YOLOv12x offered higher prediction confidence (Precision). This balance is important in industrial contexts, where false negatives may risk defective products entering production lines, while false positives may lead to unnecessary re-inspection or rejection.

The GC10-DET dataset (Grayscale Casting 10-type Defect Detection) comprises grayscale industrial images with multiple defect classes, including crescent_gap, oil_spot, punching_hole, water_spot, and welding_line. This multi-class setup increases complexity by requiring models to differentiate among several distinct defect morphologies rather than simply separating "defect" versus "no defect."

Among the tested models, YOLOv11x and YOLOv9e delivered the highest mAP50 (0.912) on the overall "all" class, and performed consistently across most individual defect types. YOLOv9e showed superiority in terms of Precision (about 0.879). The training curves (Figure 9) demonstrated that all models converged smoothly without instability.

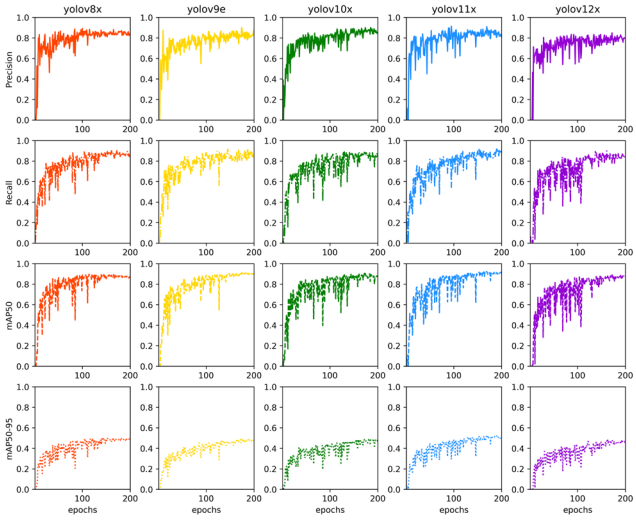

Fig. 9. Training curves on the GC10-DET dataset

Class-specific analysis revealed that punching_hole defects were consistently the easiest to detect, with all models exceeding mAP50 > 0.95. This is likely because their regular circular shape and strong contrast with the surrounding surface make them visually distinctive. Conversely, oil_spot proved more challenging: their irregular shapes and diffuse boundaries often blended into background textures, leading to lower detection confidence and occasional misclassifications. These trends are illustrated in the benchmark results shown in Figure 10. Inference examples in Figure 11 further highlight these challenges. In the first and third examples, all models were able to detect the material defects without difficulty. The second example, however, was markedly different. The best performance was achieved by YOLOv8x and YOLOv11x, which correctly detected all three objects over their full area or length. YOLOv10x and YOLOv12x also detected all three objects but failed to fully capture the welding line defect; only part of its true extent was covered by the predicted bounding box. YOLOv9e performed the worst in this case: not only did it fail to mark the full length of the welding line defect, but it also detected it twice. These differences emphasize that long, thin defects such as welding lines pose particular challenges for object detectors, as they extend across multiple receptive fields and are difficult to enclose within bounding boxes.

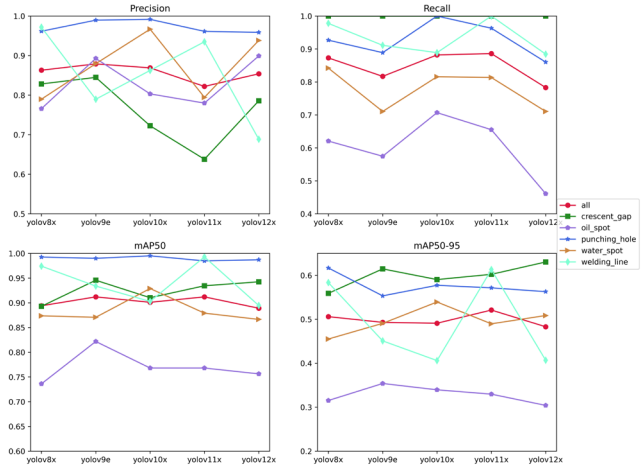

Fig. 10. Class-wise detection performance on the GC10-DET dataset

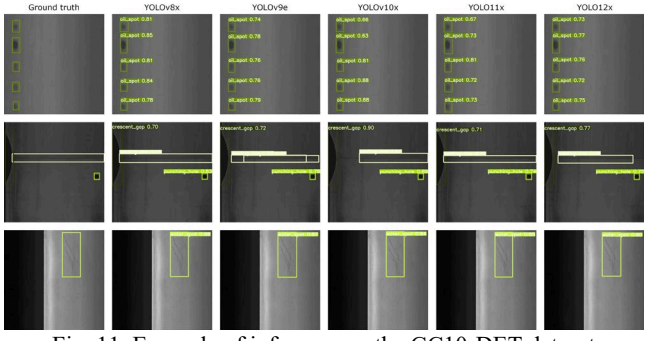

Fig. 11. Example of inference on the GC10-DET dataset

Overall, the GC10-DET results demonstrate that while YOLO architectures perform reliably on simple, highly distinguishable defects (e.g., punching_hole), they show greater variability on subtle or elongated defects (oil_spot, welding_line). These findings highlight the importance of considering defect morphology when selecting and optimizing models for deployment in industrial inspection pipelines.

The NEU-SEG dataset includes high-resolution grayscale images of metal surfaces labelled with three defect types: inclusion, patches, and scratches. The detection task on this dataset is challenging due to texture similarities between defective and non-defective areas.

Across models, YOLOv9e achieved the best overall performance on the “all” class, with the highest Precision (0.863) and mAP50 (0.880), as well as the best mAP50–95 (0.587). YOLOv11x showed strong Recall (0.828), indicating greater sensitivity in detecting defect instances, while YOLOv12x delivered consistent accuracy across all metrics. The training curves (Figure 12) confirmed stable convergence across all models.

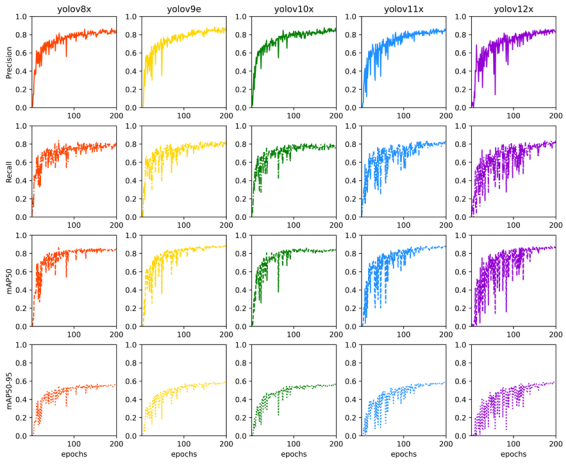

Fig. 12. Training curves on the NEU-SEG dataset

At the class level, all models showed strong balance on patches and inclusion, but more fluctuation in scratches, where mAP50 ranged from 0.834 (YOLOv8x) to 0.895 (YOLOv9e). Scratches are typically thin, elongated, and low-contrast, which makes them harder to separate from background surface textures. This variation across models is illustrated in Figure 13. Inference results (Figure 14) further demonstrate these challenges. In the first and third examples, the models had no difficulty detecting the objects or assigning the correct class, showing strong generalisation. In the second example, three models (YOLOv8x, YOLOv11x, and YOLOv12x) failed to detect the same scratch defect. In contrast, YOLOv9e and YOLOv10x were able to identify all objects and assign the correct classes. YOLOv10x showed noticeably higher confidence in class assignments. These results suggest that architectural refinements in YOLOv9 and YOLOv10 enhanced sensitivity to fine linear structures compared to earlier versions.

Overall, the NEU-SEG results confirm that YOLO-based detectors perform well on compact and clearly bounded defects (patches, inclusions) but struggle with fine, elongated features such as scratches. Models such as YOLOv9e and YOLOv10x demonstrated improved robustness to these difficult cases, making them more suitable for industrial inspection scenarios where thin surface defects may indicate critical quality issues.

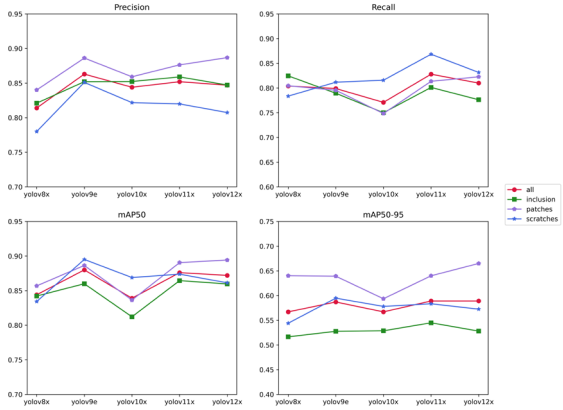

Fig. 13. Class-wise detection performance on the NEU-SEG dataset

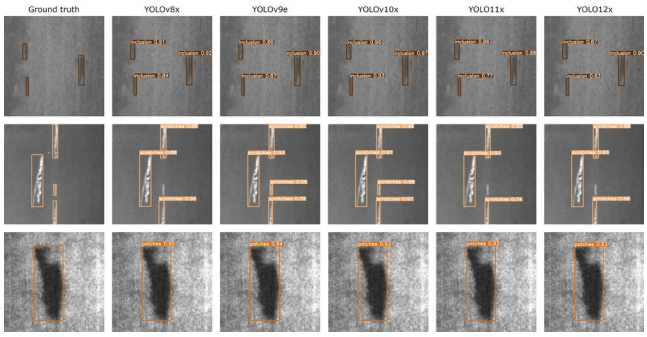

Fig. 14. Example of inference on the NEU-SEG dataset

The SSDD (Severstal Steel Defect Detection) dataset contains complex maritime imagery of ship hull surfaces with four primary defect classes: pitted_surface, crazing, scratches, and patches. This dataset is especially challenging due to lighting variations and diverse surface conditions. This dataset is especially valuable because it simulates the complexities encountered in real-world inspection of ship hulls, including variations in lighting, perspective, and surface degradation.

Across the evaluated YOLO architectures, YOLOv9e demonstrated the strongest performance on this dataset, achieving the highest mAP50 (0.918), indicating superior overall detection accuracy. YOLOv8x recorded the best Precision (0.925) and mAP50–95 (0.743), making it particularly effective for detailed spatial localization. YOLOv11x achieved the top Recall (0.853), reflecting its sensitivity in detecting defect instances. While YOLOv12x remained competitive, it did not lead in any major metric on this dataset. As shown in the training curves (Figure 15), all models converged efficiently.

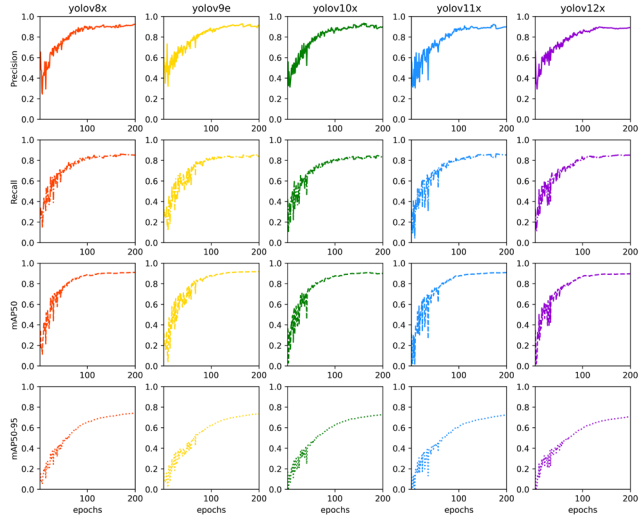

Fig. 15. Training curves on the SSDD dataset

Class-wise, all models excelled at detecting pitted_surface defects (Precision > 0.93, mAP50 > 0.92), showing that these types of defects have distinctly recognisable features. However, crazing presented more difficulty. Their irregular, fine crack-like patterns often blend into surface textures or are partially obscured by

lighting artefacts, which reduces detection confidence. The full comparative analysis is depicted in Figure 16.

Inference examples (Figure 17) illustrate these trends. In most cases, all models correctly detected the defects and assigned the correct class labels, with only minor variations in confidence scores. YOLOv9e consistently balanced detection accuracy and confidence, while YOLOv8x, although slightly less sensitive (lower Recall), delivered highly precise detections with minimal false positives. These results suggest that YOLOv9e is best suited for comprehensive inspections requiring maximum detection accuracy, while YOLOv8x is advantageous in contexts where avoiding false alarms is critical.

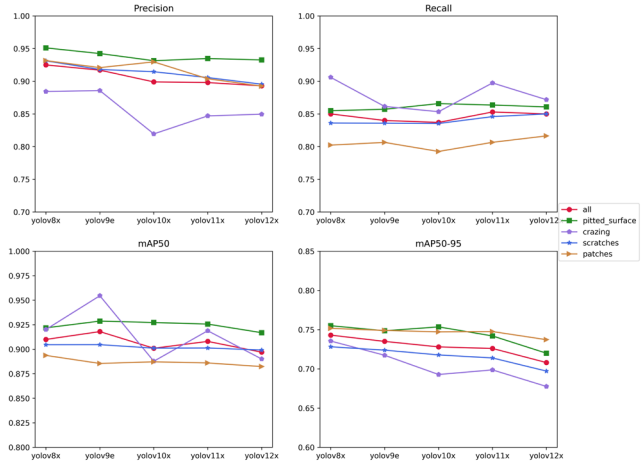

Fig. 16. Class-wise detection performance on the SSDD dataset

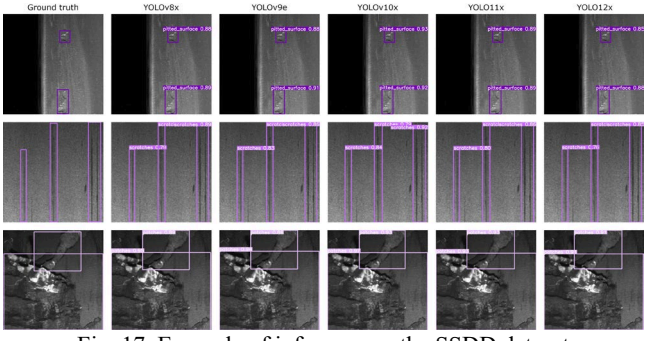

Fig. 17. Example of inference on the SSDD dataset

Overall, the SSDD results underscore that defect morphology and imaging conditions significantly affect detection reliability. While all YOLO architectures demonstrated robustness, the balance between accuracy, recall, and precision should guide model selection for practical deployments in steel inspection workflows.

3.2. Resource Efficiency Overview

In addition to detection accuracy, we conducted a comprehensive evaluation of the computational efficiency and hardware demands of each YOLO-based model. Key metrics included inference time, throughput (FPS), model size, GPU power draw, and memory consumption.

Across all datasets, YOLOv8x consistently exhibited the fastest inference times, typically ranging between 19.2 and 19.8 ms per image, and the highest throughput, frequently exceeding 47 FPS. This makes YOLOv8x a strong candidate for real-time or latency-sensitive industrial applications. YOLOv10x, the most lightweight model (~61 MB), offered an excellent trade-off between compactness and speed, achieving FPS above 46 and maintaining very low GPU memory usage (2.24–2.39 GB). These attributes make it particularly suitable for deployment on embedded or resource-constrained industrial devices. YOLOv12x, while showing slightly longer inference times (~27–30 ms) and lower throughput (~32–35 FPS), maintained high and stable accuracy across all datasets. Although it was not the top performer in any individual dataset, it consistently ranked among the top models, suggesting robust generalization. Its GPU power draw (223–254 W) and memory consumption (2.24–2.90 GB) remained within a deployable range for modern edge devices. YOLOv9e, despite being relatively compact (~111.8 MB), showed the slowest inference performance (~32–33 ms/image, ~29 FPS). However, it consistently achieved top detection metrics, outperforming other models on multiple datasets. This highlights a trade-off where computational speed is sacrificed in favor of superior accuracy and spatial localization performance.

Importantly, all models remained within practical deployment limits, using less than 3 GB of GPU memory and drawing 216–257 W in power. A comparative summary of resource efficiency metrics across models and datasets is provided in Table 2.

4. Conclusions

This study presented a thorough evaluation of five state-of-the-art YOLO-based object detection models (YOLOv8x, YOLOv9e,

YOLOv10x, YOLOv11x, and YOLOv12x) across five benchmark datasets commonly used in surface defect detection: AlcastXray, Castings, GC10-DET, NEU-SEG, and SSDD. The evaluation considered both detection performance metrics (Precision, Recall, mAP50, mAP50–95) and deployment-oriented efficiency indicators (inference time, throughput, model size, GPU power draw, and memory usage).

The results indicate that YOLOv9e consistently ranked among the top performers in detection accuracy, achieving the highest mAP scores on multiple datasets. It particularly excelled on AlcastXray, GC10-DET, NEU-SEG, and SSDD, although it showed the lowest throughput and longest inference times. Conversely, YOLOv8x offered the best computational performance, consistently delivering the fastest inference times and highest throughput, making it highly suitable for real-time applications. YOLOv10x, the smallest model in the benchmark, emerged as a strong candidate for edge deployment due to its compact size, efficient GPU utilisation, and consistently high FPS. While YOLOv12x did not dominate any individual dataset, it maintained reliably strong performance across all datasets, with moderate resource consumption, suggesting that it is a robust and generalizable model option for practical scenarios.

Table 2.
Summary of resource efficiency metrics for each model and dataset

Dataset	Model	Size (MB)	Inference time (ms/image)	Process time (ms/image)	FPS	GPU power draw (W)	GPU memory usage (GB)
Al-Cast	yolov8x	130.4	19.57	21.15	47.3	246	2.28
	yolov9e	111.8	32.12	33.83	29.6	245	2.45
	yolov10x	61.1	20.45	21.03	47.6	246	2.24
	yolo11x	109.1	21.58	23.33	42.9	245	2.03
	yolo12x	113.6	27.32	28.85	34.7	246	2.24
GDXray-Castings	yolov8x	130.4	19.26	20.87	47.9	224	2.83
	yolov9e	111.8	32.01	33.51	29.8	230	2.48
	yolov10x	61.1	19.96	20.57	48.6	224	2.39
	yolo11x	109.1	22.05	23.64	42.3	225	2.28
	yolo12x	113.6	27.21	28.72	34.8	230	2.28
GC10-DET	yolov8x	130.4	19.76	21.34	46.9	216	2.88
	yolov9e	111.8	32.21	33.79	29.6	225	2.75
	yolov10x	61.2	20.81	21.43	46.7	218	2.88
	yolo11x	109.1	20.65	22.23	45	218	2.88
	yolo12x	113.6	27.16	28.71	34.8	223	2.88
NEU-SEG	yolov8x	130.4	19.56	21.18	47.2	240	2.9
	yolov9e	111.8	32.54	34.1	29.3	238	2.75
	yolov10x	61.1	20.01	20.61	48.5	240	2.9
	yolo11x	109.1	20.62	22.14	45.2	240	2.88
	yolo12x	113.6	29.67	31.27	32	239	2.9
SSDD	yolov8x	130.4	19.36	20.89	47.9	257	2.36
	yolov9e	111.8	32.32	33.89	29.5	252	2.75
	yolov10x	61.1	20.06	20.64	48.4	256	2.36
	yolo11x	109.1	20.69	22.28	44.9	256	2.07
	yolo12x	113.6	27.04	28.57	35	254	2.33

Overall, no single model dominated across all aspects. The optimal choice depends on the specific application constraints; YOLOv9e for accuracy-critical tasks, YOLOv8x or YOLOv10x for resource-constrained or real-time deployments, and YOLOv12x as a stable all-rounder.

While this benchmark study provides valuable guidance using controlled datasets, future research should focus on assessing these models in real-world industrial settings, particularly when integrated into edge computing platforms such as the NVIDIA Jetson series or industrial-grade embedded GPUs. Such evaluations will offer deeper insights into performance under practical constraints, including thermal limitations, inconsistent lighting conditions, sensor noise, and latency introduced by hardware, ultimately facilitating more informed deployment decisions in production environments.

References

[1] Gill, R., Srivastava, D., Hooda, S., Singla, C. & Chaudhary, R. (2024). Unleashing sustainable efficiency: the integration of computer vision into industry 4.0. *Engineering Management Journal*. 37(4), 414-432. https://doi.org/10.1080/10429247.2024.2383518.

[2] Tabernik, D., Šela, S., Skvarč, J. & Skočaj, D. (2020). Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*. 31(3), 759-776. https://doi.org/10.1007/s10845-019-01476-x.

[3] Liu, W., Zhang, J., Su, Z., Zhou, Z. & Liu, L. (2021). Binary neural network for automated visual surface defect detection. *Sensors*. 21(20), 6868, 1-16. https://doi.org/10.3390/s21206868.

[4] He, Y., Song, K., Meng Q. & Yan, Y. (2020). An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Transactions on Instrumentation and Measurement*. 69(4), 1493-1504. DOI:10.1109/TIM.2019.2915404.

[5] Ashrafi, S., Teymouri, S., Etaati, S., Khoramdel, J., Borhani, Y. & Najafi, E. (2025). Steel surface defect detection and segmentation using deep neural networks. *Results in Engineering*. 25, 103972, 1-11. https://doi.org/10.1016/j.rineng.2025.103972.

[6] Saberironaghi, A., Ren, J. & El-Gindy, M. (2023). Defect detection methods for industrial products using deep learning techniques: a review. *Algorithms.* 16(2), 95, 1-30. https://doi.org/10.3390/a16020095.

[7] Chen, Y., Ding, Y., Zhao, F., Zhang, E., Wu, Z. & Shao, L. (2021). Surface defect detection methods for industrial products: a review. *Applied Sciences*. 11(16), 7657. https://doi.org/10.3390/app11167657.

[8] Ye, X., Wang, L., Huang, Ch. & Luo, X. (2024). Wind turbine blade defect detection with a semi-supervised deep learning framework. *Engineering Applications of Artificial Intelligence*. 136(A), 108908, 1-11. https://doi.org/10.1016/j.engappai.2024.108908.
[9] Pang, R., Ning, J., Leng, X., Chen, C., Zhang, P. & Liu, J. (2024). ANP-Net: attention neural processes network for semi-supervised pavement defect semantic segmentation. *International Journal of Parallel, Emergent and Distributed Systems*. 40(3), 296-311. https://doi.org/10.1080/17445760.2024.2405966.
[10] Guo, Z., Wang, C., Yang, G., Huang, Z. & Li, G. (2022). MSFT-YOLO: improved YOLOv5 based on transformer for detecting defects of steel surface. *Sensors*. 22(9), 3467, 1-15. https://doi.org/10.3390/s22093467.
[11] Carrilho, R., Hambarde, K.A. & Proença, H. (2024). A novel dataset for fabric defect detection: bridging gaps in anomaly detection. *Applied Sciences*. 14(12), 5298. https://doi.org/10.3390/app14125298.
[12] Shao, R., Zhou, M., Li, M., Han, D. & Li, G. (2024). TD-Net: tiny defect detection network for industrial products. *Complex & Intelligent Systems*. 10, 3943-3954. https://doi.org/10.1007/s40747-024-01362-x.
[13] Lang, D. & Lv, Z. (2025). SEPDNet: simple and effective PCB surface defect detection method. *Scientific Reports.* 15(1), 10919, 1-15. https://doi.org/10.1038/s41598-024-84859-2.
[14] Ma, J., Zhang, T., Yang, C., Cao, Y., Xie, L., Tian, H. & Li, X. (2023). Review of wafer surface defect detection methods. *Electronics*. 12(8), 1787, 1-17. https://doi.org/10.3390/electronics12081787.
[15] Lu, S., K. Wu, K. & Chen, J. (2023). Solar cell surface defect detection based on optimized YOLOv5. *IEEE Access*. 11, 71026-71036. DOI: 10.1109/ACCESS.2023.3294344.
[16] He, Z., Yang, W., Liu, Y., Zheng, A., Liu, J., Lou, T. & Zhang, J. (2024). Insulator defect detection based on YOLOv8s-SwinT. *Information*. 15(4), 206, 1-15. https://doi.org/10.3390/info15040206.
[17] Shan, W. & Yue, Y. (2024). Apple defect detection in complex environments. *Electronics*. 13(23), 4844, 1-17. https://doi.org/10.3390/electronics13234844.
[18] Hu, X., Hu, Y., Cai, W., Xu, Z., Zhao, P., Liu, X., She, Q., Hu, Y. & Li, J. (2021). Automatic detection of small sample apple surface defects using ASDINet. *Foods*. 12(6), 1352, 1-29. https://doi.org/10.3390/foods12061352.
[19] Wang, F., Lv, C., Pan, Y., Zhou, L. & Zhao, B. (2023). Efficient non-destructive detection for external defects of kiwifruit. *Applied Sciences*. 13(21), 11971, 1-14. https://doi.org/10.3390/app132111971.
[20] Liu, C., Shen, Y., Mu, F., Long, H., Bilal, A., Yu, X. & Dai, Q. (2025). Detection of surface defects in soybean seeds based on improved Yolov9. *Scientific Reports*. 15(1), 12631, 1-21. https://doi.org/10.1038/s41598-025-92429-3.
[21] Han, J., Cui, G., Li, Z. & Zhao, J. (2024). DBCW-YOLO: a modified YOLOv5 for the detection of steel surface defects. *Applied Sciences*. 14(11), 4594. https://doi.org/10.3390/app14114594.
[22] Zeng, K., Xia, Z., Qian, J., Du, X., Xiao, P. & Zhu, L. (2025). Steel surface defect detection technology based on YOLOv8-MGVS. *Metals*. 15(2), 109, 1-16. https://doi.org/10.3390/met15020109.
[23] Yang, Z. & Liu, Y. (2025). A steel surface defect detection method based on improved RetinaNet. *Scientific Reports*. 15, 6045, 1-17. https://doi.org/10.1038/s41598-025-88527-x.
[24] Yu, F., Zhang J. & Mu, D. (2025). Steel defect detection based on YOLO-SAFD. *IEEE Access*. 13, 77291-77304. DOI: 10.1109/ACCESS.2025.3565587.
[25] Yang, T. & Li, J. (2023). Steel surface defect detection based on SSAM-YOLO. *International Journal of Information Technologies and Systems Approach* (IJITSA). 16(3), 1-13. https://doi.org/10.4018/IJITSA.328091.
[26] Wang, J., Zhang, Z., Lin, X., Zhao, J., Chen, M. & Luo, L. (2024). YOLO-DD: Improved YOLOv5 for defect detection. *Computers, Materials & Continua*. 78(1), 759-780. https://doi.org/10.32604/cmc.2023.041600.
[27] Xin, H. & Song, J. (2024). YOLOv5-ACCOF steel surface defect detection algorithm. *IEEE Access*. 12, 157496-157506. DOI: 10.1109/ACCESS.2024.3486110.
[28] Chen, Y., He, Y. & Wu, L. (2025). Detection of welding defects using the YOLOv8-ELA algorithm. *Applied Sciences*. 15(9), 5204, 1-16. https://doi.org/10.3390/app15095204.
[29] Sui T. & Wang, J. (2024). An improved multiscale semantic enhancement network for aluminum defect detection. *IEEE Access*. 12, 138362-138371. DOI: 10.1109/ACCESS.2024.3464741.
[30] Xu, Y., Zhang, K. & Wang, L. (2021). Metal surface defect detection using modified YOLO. *Algorithms*. 14, 257. https://doi.org/10.3390/a14090257.
[31] Huang, Ch., Chen, M. & Wang, L. (2024). Semi-supervised surface defect detection of wind turbine blades with YOLOv4. *Global Energy Interconnection*. 7(3), 284-292. https://doi.org/10.1016/j.gloei.2024.06.010.
[32] Fu, X., Yang, X., Zhang, N., Zhang, R., Zhang, Z., Jin, A., Ye, R. & Zhang, H. (2023). Bearing surface defect detection based on improved convolutional neural network. *Mathematical Biosciences and Engineering*. 20(7), 12341-12359. DOI:10.3934/mbe.2023549.
[33] Szatkowski, M., Wilk-Kołodziejczyk, D., Jaśkowiec, K., Małysza, M., Bitka, A. & Głowacki, M. (2023). Analysis of the possibility of using selected tools and algorithms in the classification and recognition of type of microstructure. *Materials*. 16(21), 6837, 1-13. https://doi.org/10.3390/ma16216837.
[34] Hao, A., Zhihong, L., Mingming, Q., Yuxiang, H., Fei, X. & Guojian, Z. (2024). Wood defect detection based on the CWB-YOLOv8 algorithm. *Journal of Wood Science*. 70, 26, 1-14. https://doi.org/10.1186/s10086-024-02139-z.
[35] Xie, M., Bian, H., Jiang, C., Zheng, Z., Wang, W. (2024). An improved YOLOv5 algorithm for tyre defect detection. *Electronics*. 13(11), 2207, 1-20. https://doi.org/10.3390/electronics13112207.
[36] Guo, Y., Kang, X., Li, J. & Yang, Y. (2023). Automatic fabric defect detection method using AC-YOLOv5. *Electronics*. 12(13), 2950, 1-15. https://doi.org/10.3390/electronics12132950.
[37] Yu, J., Jiang, J., Fichera, S., Paoletti, P., Layzell, L., Mehta, D. & Luo, S. (2024). Road surface defect detection-from image-based to non-image-based: a survey. *EEE transactions*

on intelligent transportation Systems. 25(9), 10581-10603. https://doi.org/10.1109/TITS.2024.3382837.
[38] Kim, G. & Kim, S. (2024). A road defect detection system using smartphones. *Sensors.* 24(7), 2099, 1-21. https://doi.org/10.3390/s24072099.
[39] Liu, X., Yang, X., Shao, L., Wang, X., Gao, Q. & Shi, H. (2024). GM-DETR: research on a defect detection method based on improved detr. *Sensors*. 24(11), 3610, 1-24. https://doi.org/10.3390/s24113610.
[40] Parlak, I.E. & Emel, E. (2023). Deep learning-based detection of aluminum casting defects and their types. *Engineering Applications of Artificial Intelligence*. 118, 105636, 1-18. https://doi.org/10.1016/j.engappai.2022.105636.
[41] Ji, X., Yan, Q., Huang, D., Wu, B., Xu, X., Zhang, A., Liao, G., Zhou, J. & Wu, M. (2021). Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition. *Journal of Materials Processing Technology*. 292, 117064, 1-12. https://doi.org/10.1016/j.jmatprotec.2021.117064.
[42] Wang, C. & Wang, Y. (2024). SLGA-YOLO: a lightweight castings surface defect detection method based on fusion-enhanced attention mechanism and self-architecture. *Sensors*. 24(13), 4088, 1-18. https://doi.org/10.3390/s24134088.
[43] Yousef, N. & Sata, A. (2023). Parametric study of inspecting surface defects in investment casting. *Jordan Journal of Mechanical and Industrial Engineering*. 17(4). https://doi.org/10.59038/jjmie/170409.
[44] Zhang, H.-B., Zhang, C.-Y., Cheng, D.-J., Zhou, K.-L. & Sun, Z.-Y. (2024). Detection transformer with multi-scale fusion attention mechanism for aero-engine turbine blade cast defect detection considering comprehensive features. *Sensors*. 24(5), 1663, 1-25. https://doi.org/10.3390/s24051663.
[45] Wang, S., Cheng, D-J., Fang, X-F. & Zhang, Ch-Y. (2025). SDA-PVTDet: A spatial-cross dual attention pyramid vision transformer detector for casting defect detection in radiography images. *Expert Systems With Applications*. 269, 126385, 1-20. https://doi.org/10.1016/j.eswa.2025.126385.
[46] Tang, W., Vian, C. M., Tang, Z. & Yang, B. (2021). Anomaly detection of core failures in die casting X-ray inspection images using a convolutional autoencoder. *Machine Vision and Applications*. 32(4), 102. https://doi.org/10.1007/s00138-021-01226-1.
[47] Du, W., Shen H. & Fu, J. (2021). Automatic defect segmentation in X-Ray images based on deep learning. *IEEE Transactions on Industrial Electronics*. 68(12), 12912-12920. DOI: 10.1109/TIE.2020.3047060.
[48] Pu, Q. C., Zhang, H., Xu, X. R., Zhang, L., Gao, J., Rodić, A., Petrovic, P. Xu, S. & Wang, Z. X. (2024). Casting-DETR: an end-to-end network for casting surface defect detection. *International Journal of Metalcasting*. 18(4), 3152-3165. https://doi.org/10.1007/s40962-023-01212-5.
[49] Wang, C. & Ye, C. (2025). OHSW-YOLO: an aluminum casting surface defects detection model. *International Journal of Metalcasting*. 20, 506-519. https://doi.org/10.1007/s40962-025-01620-9.
[50] Wu, K., Sun, S., Sun, Y., Wang, C. & Wei, Y. (2025). RBS-YOLO: a lightweight YOLOv5-based surface defect detection model for castings. *IET Image Process*. 19(1), e70018, 1-11. https://doi.org/10.1049/ipr2.70018.
[51] Parlak, I.E. & Emel, E. (2023). Deep learning-based detection of aluminum casting defects and their types. *Engineering Applications of Artificial Intelligence*. 118, 105636, 1-18. https://doi.org/10.1016/j.engappai.2022.105636.
[52] Mery, D., Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H. & Carrasco, M.. (2015). GDXray: The database of X-ray images for nondestructive testing. *Journal of Nondestructive Evaluation*. 34, 42, 1-12. https://doi.org/10.1007/s10921-015-0315-7.
[53] Lv, X., Duan, F., Jiang, J.-J., Fu, X. & Gan, L. (2020). Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*. 20(6), 1562, 1-15. https://doi.org/10.3390/s20061562.
[54] Dong, H., Song, K., He, Y., Xu, J., Yan, Y. & Meng, Q. (2020). PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection. *IEEE Transactions on Industrial Informatics*. 16(12), 7448-7458. https://doi.org/10.1109/TII.2019.2958826.
[55] Roboflow. (2023). NEU-Seg dataset [Open Source Dataset]. Roboflow Universe. Retrieved June 2, 2025 from https://universe.roboflow.com/school-4pxkq/neu-seg-bscps.
[56] Grishkin, A. (2019). Severstal: Steel Defect Detection. Kaggle. Retrieved June 7, 2025, from https://www.kaggle.com/c/severstal-steel-defect-detection/overview.
[57] Demir, F. & Parlak, K. S. (2025). Increasing the classification achievement of steel surface defects by applying a specific deep strategy and a new image processing approach. *Applied Sciences*. 15(8), 4255, 1-29. https://doi.org/10.3390/app15084255.
[58] Padilla, R., Netto, S.L., Da Silva, E.A. (2020). A Survey on performance metrics for object-detection algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), 1–3 July 2020 (pp. 237–242). Niterói, Brazil.
[59] Henderson, P., Ferrari, V. (2017). End-to-End training of object class detectors for mean average precision. In Proceedings of the Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, 20–24 November 2016 (pp. 198-213). Taipei, Taiwan.
[60] Ultralytics. (2023). *Explore Ultralytics YOLOv8*. Retrieved June 2, 2025, from https://docs.ultralytics.com/models/yolov8/.
[61] Ultralytics. (2024). *YOLOv9: A Leap Forward in Object Detection Technology*. Retrieved June 2, 2025, from https://docs.ultralytics.com/models/yolov9/.
[62] Ultralytics. (2024). *YOLOv10: Real-Time End-to-End Object Detection.* Retrieved June 2, 2025, from https://docs.ultralytics.com/models/yolov10/.
[63] Ultralytics. (2024). *Ultralytics YOLO11*. Retrieved June 2, 2025, from https://docs.ultralytics.com/models/yolo11/.
[64] Ultralytics. (2025). *YOLO12: Attention-Centric Object Detection*. Retrieved June 2, 2025, from https://docs.ultralytics.com/models/yolo12/.