ACCSCIENCE PUBLISHING

Materials Science in Additive Manufacturing

ORIGINAL RESEARCH ARTICLE

Artificial intelligence-driven defect detection and localization in metal 3D printing using convolutional neural networks

Xinyi Yin[1]*, Jan Akmal[1,2], and Mika Salmi[1]

[1]Materials to Products Research Group, Department of Energy and Mechanical Engineering, Aalto University, Espoo, Finland

[2]EOS Metal Materials, Electro Optical Systems Finland Oy, Turku, Finland

(This article belongs to the *Special Issue: Smart Additive Manufacturing: Product and Process Qualification through Innovation in Design, Modeling, Monitoring, Machine Learning, Metrology, and Materials Science*)

Abstract

Metal additive manufacturing (AM) has attracted significant interest in high-value industries due to its ability to produce complex parts flexibly, but the reliance on costly manual monitoring remains a major burden for quality control. Artificial intelligence (AI)-driven models for automated defect detection are emerging as promising solutions. This study contributes a new annotated dataset for AI research in AM and evaluates the performance of four widely used convolutional neural network (CNN) models in detecting powder bed morphology defects, based on layer-wise images acquired by the EOSTATE PowderBed system during the metal laser-based powder bed fusion process. The models were trained through transfer learning methods with manually labeled and pre-processed data. Results demonstrated that ResNet50 and EfficientNetV2B0 achieved over 99% accuracy in defect classification, while YOLOv5 outperformed Faster region-based-CNN in defect detection and localization. However, lower average precision values in object detection tasks were attributed to variability in defect scales and annotation quality. This study confirms the potential of AI-based models for defect identification in AM, with YOLOv5 demonstrating clear advantages in managing complex, multi-scale defects. Future improvements will focus on expanding the dataset and refining annotation strategies to further enhance model robustness.

Keywords: Machine learning; Defects detection; Quality control; AI-driven models; Metal additive manufacturing; Image classification; Object detection

***Corresponding author:**
Xinyi Yin
(xinyi.yin@aalto.fi)

Citation: Yin X, Akmal J, Salmi M. Artificial intelligence-driven defect detection and localization in metal 3D printing using convolutional neural networks. *Mater Sci Add Manuf.* 2025;4(3):025150022. doi: 10.36922/MSAM025150022

Received: April 7, 2025

1st revised: May 7, 2025

2nd revised: May 20, 2025

Accepted: May 20, 2025

Published online: June 25, 2025

1. Introduction

Metal additive manufacturing (AM) has attracted significant attention in high-value industries, such as aerospace, automotive, healthcare, and nuclear energy, due to its ability to produce customized, complex parts with excellent mechanical properties while minimizing waste.[1-3] However, a significant challenge in AM is the frequent occurrence of inevitable defects. Even in metal laser-based powder bed fusion (PBF-LB) alone, around 10 typical high-frequency defects may occur, including lack of fusion, keyhole porosity, gas porosity, solidification cracking, solid-state cracking, surface-connected

porosity, and impurity-induced defects, among others.[4-6] In traditional AM processes, product quality is ensured by several hours of manual monitoring of the printing process, leading to low productivity and increased costs.[7,8] Therefore, improving monitoring methods and ensuring product quality to enhance reliability has become a key research focus.[9,10]

In recent years, with the continuous development and maturation of artificial intelligence (AI), machine learning (ML) has been widely discussed and applied in the field of quality control in AM.[1,11-15] Kadam *et al.*[16] employed a series of pre-trained models for feature extraction combined with various ML algorithms to achieve fault detection in the fused deposition modeling (FDM) process, demonstrating the effectiveness of ML algorithms in anomaly detection. In 2024, Herzog *et al.*[1] reviewed the application of various ML methods and *in situ* monitoring technologies in metal AM defect detection. They surveyed 50 independent studies published since 2017, most based on computer vision algorithms to classify defective images. Among image classification-based detection algorithms, over 75% used supervised learning methods that detect anomalous data by labeling the data as "normal" and one or a few "anomalies." These methods reported accurate favorable accuracy rates of approximately 75 – 95%.[1]

Given the effectiveness of AI-based methods, many researchers aim to build AI-based automated quality monitoring systems.[17,18] Khan *et al.*[7] developed a real-time defect detection system based on a convolutional neural network (CNN) model to improve the automation of fused filament fabrication and reduce production losses and labor costs. Although the model achieved an accuracy of 84% in identifying geometric anomalies, it struggled to detect vertical plane defects, and the insufficient dataset led to inconsistencies in the model.[7] Conversely, Cannizzaro *et al.*[19] introduced a real-time defect monitoring and detection system for metal powder bed fusion (PBF) that classifies five different types of powder bed defects and monitors the profile of each printed layer. They also explored using generative adversarial networks to generate synthetic images of powder bed defects to address the scarce labeled data for training and testing ML models.[19] Tamir *et al.*[11] proposed the integration of digital twins and parallel systems into AM for real-time process monitoring to optimize intelligent manufacturing processes through the development of virtual models.

The diversity and size of the dataset directly impact the performance of ML models, and high-quality datasets are crucial for improving prediction and detection accuracy. Nonetheless, researchers often face challenges in finding suitable target datasets.[20,21] Zhang *et al.*[22] emphasized creating a systematic AM database for ML-assisted AM research. Only a few of the six existing AM research databases contain image datasets for quality monitoring, and annotated datasets are almost non-existent.[22] In 2024, Liu *et al.*[23] also identified the scarcity of image-based datasets in the AM field, thereby warranting more open datasets to facilitate quality evaluation and defect detection in AM processes.[23]

Researchers have proposed various methods to address data imbalance and scarcity.[23,24] In the study by Westphal and Seitz,[25] they combined transfer learning with pre-trained VGG16 and Xception models to classify small datasets, with results indicating that VGG16 performed best across multiple performance metrics, validating the effectiveness of CNN-based defect detection for non-destructive quality assurance. Szymanik *et al.*[26] proposed an innovative approach that combines enhanced signal analyses to improve the accuracy and effectiveness of defect detection in materials with low thermal conductivity, enabling precise detection of defects in thermal imaging and automatic identification of their type and size. Ansari *et al.*[27] designed artificial implant defects that simulate various shapes and sizes of pores and combined a CNN model with X-ray tomography data to successfully detect defects as small as 0.2 mm with up to 97% accuracy, demonstrating the potential of CNN models in PBF pore detection. Kozhay *et al.*[28] developed a CNN algorithm for detecting and classifying defects in FDM-printed images, achieving 90% accuracy. However, the system's performance outside specific layouts was limited due to insufficient data, highlighting the importance of defect data diversity for algorithm optimization.[28]

In previous research, CNN models have been widely recognized as the most suitable AI model for defect image classification and detection tasks in the AM field.[12,29] Fu *et al.*[29] systematically summarized ML algorithms for defect detection in PBF-LB processes, noting that supervised learning is the most commonly used ML method. It performs classification and regression by learning the relationship between input and output in labeled datasets, making it widely applied in PBF-LB systems for defect detection and classification. CNNs extract features through convolution layers, reduce dimensions through pooling layers, and output through fully connected layers, showcasing excellent image processing and pattern recognition capabilities. CNNs are the most popular deep learning algorithms for image recognition, image classification, and object detection, making them well-suited for defect detection in Laser-Based AM processes.[29]

As early as 2019, Han *et al.*[30] proposed a deep CNN-based AI technique to effectively monitor surface quality and detect defects by analyzing metal fracture micrographs

and metallographic images in industrial manufacturing. The experimental results demonstrated that a learning rate of 0.01, Adam optimizer, and Inception-ResNet-v2 network achieved the best classification accuracy for image classification tasks in AM.[30] Khan *et al.*[31] used optical tomography data, monitored layer by layer, and combined random forest algorithms to detect porosity and lack of fusion defects. Although an accuracy of 99.98% was achieved, the risk of overfitting existed due to the small training dataset of only 100 images. Their study further proposed that more efficient detection could be achieved with CNNs and larger-scale datasets.[31] Abhilash and Ahmed[32] used the ResNet50 CNN model to classify AM component surface conditions to determine polishing conditions and optimized surface quality while eliminating defects, achieving a prediction accuracy of 96% and significantly reducing manual inspection and material waste. Lee *et al.*[33] proposed a local detection method based on 3D-CNN to perform defect classification and local volume fraction prediction, utilizing data from the melt pool monitoring system and micro-computed tomography. This method could classify porosities caused by the lack of fusion and locked pores in the PBF-LB process.[33] Wen *et al.*[34] combined simple CNN, YOLOv4, and Detectron2 to perform classification, object detection, and segmentation tasks on scanning electron microscopy images of AM parts, achieving excellent accuracy and defect detection capabilities in static and dynamic videos.

These studies demonstrate the significant potential of CNNs for defect detection in AM, warranting further validation of the large-scale image dataset constructed in this work. Previous approaches have primarily focused on binary classification – determining whether defects exist – without providing spatial localization, which increases the burden on manual inspection. The demand for agile, real-time monitoring also exposes the limitations of pixel-level segmentation in practical applications. This study improves detection accuracy based on classification models and further incorporates advanced object detection techniques to enable both identification and localization of defects in metal 3D printing, enhancing model interpretability and applicability. All models are trained and systematically compared on a unified, large-scale dataset to ensure generalizability and provide a comprehensive performance assessment. The workflow includes layer-wise high-resolution image acquisition, data pre-processing and annotation, and model optimization through transfer learning and model evaluation. Results demonstrate that the proposed object detection approach enables accurate and efficient defect localization, significantly reducing manual effort and improving build success rates, thus laying the foundation for smarter real-time monitoring in AM.

2. Methods

2.1. Data collection and model selection

The experiment collected three sets of images from EOSTATE PowderBed (EOS Gmbh, Germany), which consists of a high-resolution (2D bitmap data resolution of 300 – 400 μm) visual wavelength camera on the EOS M290 metal machine. The EOSTATE PowderBed captured layer images of the printing builds, including defects (Table 1). The sets contained 1383, 287, and 1016 images taken layer-by-layer before and after powder bed melting. The feedstock used for test samples was 316L stainless steel. The powder size distribution was Dv(25) = 22 μm, Dv(50) = 37 μm, and Dv(75) = 58 μm, measured using laser diffraction. The recoating speed was set at 80 mm/s.[35]

AM defects differ from typical object detection targets, as they do not have uniform or consistent contours. Instead, they are abstract, mixed, and evolve gradually throughout the printing process. For example, Figure 1 presents grooves formed by debris dragging across the surface, scattered powder, and pits or voids. In the early stages, slight surface irregularities or powder residues caused by recoater vibrations may have low contrast against the background. However, if these defects are not detected and addressed in time, they can escalate into more severe issues. The model is expected to identify these subtle defect patterns early.

In practical industrial settings, the printing of a single AM part can take up to dozens of hours, depending on its size and complexity. On a typical industrial PBF-LB system, such as the EOS M290, printing one layer alone generally takes between 10 and 60 s. Since, it is difficult to predict when and where a defect may occur during the process, operators are often required to monitor the build frequently. This manual monitoring is not only time-consuming

Table 1. Printing parameters

Set	Laser power (W)	Scan speed (mm/s)	Hatch distance (mm)	Layer thickness (μm)	Volumetric energy density (J/mm^3)
1	80 – 250	780 – 3120	0.09	40	8.89, 17.81, 26.71, 35.61
2	215	1083	0.09	40	55.15
3	195	1083	0.09	20	100.03

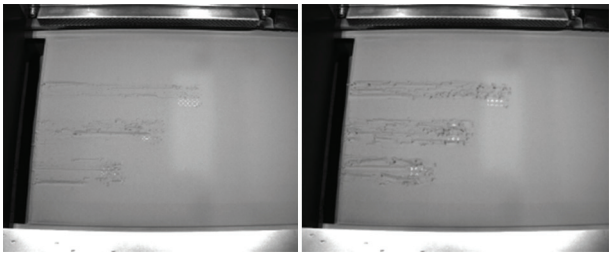

Figure 1. Example of raw powder bed data. Early-stage defects example (left); Late-stage defects example (right)

but also lacks real-time responsiveness. By integrating AI-based models into the process monitoring workflow, it becomes feasible to automatically detect surface defects and pause or stop the build when necessary. Such automated intervention can reduce unnecessary time and material loss, thereby improving overall production efficiency.

To better reflect the practical requirements of quality monitoring in AM, this study defines two task phases, aligned with the hierarchical information needs in industrial production – namely, binary defect detection and spatial localization. In the first phase, the model is designed to rapidly process layer-wise image data from the EOS M290 system and determine whether the current layer contains potential defects, classifying them as "good" or "defects." This phase simulates the industrial demand for fast screening, aiming to identify anomalous layers in real-time and enable timely intervention to minimize material and time waste. ResNet-50 and EfficientNetV2B0 are selected for comparison due to their strong trade-off between accuracy and computational efficiency. ResNet improves model depth and convergence stability through residual connections,[36] while EfficientNet balances model size, computational cost, and accuracy using compound scaling.[37] In the second phase, the model is further required to localize and highlight defect regions with bounding boxes, providing spatial information to support downstream inspection or corrective actions. This phase addresses the industrial need for defect traceability and region-specific analysis while aiming to reduce manual inspection workload. The Faster region-based-CNN (R-CNN) and YOLO Version 5 (YOLOv5; Ultralytics, United Kingdom) models are compared in this phase. Faster R-CNN uses a two-stage approach with a region proposal network, offering higher precision in detecting small or subtle defects, which benefits complex PBF-LB analysis.[38] In contrast, YOLOv5 performs single-stage detection with faster inference and better multi-scale capability, making it more suitable for real-time AM monitoring.[39,40]

All experiments were conducted on a laptop, equipped with Intel Core i7-12700H, 2.30GHz CPU, and 16GB of RAM, and using Python (version 3.10, Python Software Foundation, United Kingdom), PyTorch (version 1.12.1, Meta AI, United States), TensorFlow/Keras (version 2.13.0, Google LLC, United States), OpenCV (version 4.4.0.40, OpenCV.org, United States, Numpy (version 1.24.3, NumPy Developers, United States), and Matplotlib (version 3.7.5, Matplotlib Development Team, United States) for model training and result visualization. All experiments were run in the PyCharm Community Edition (version 2024.1.2, JetBrains s.r.o., Czech Republic) with Anaconda (version 24.3.0, Anaconda, Inc., United States) to ensure dependency isolation and version control.

Table 2. Number of "good" and "defects" powder bed images (*n*=2638)

Classification	Number of PB images, *n*
Good	1108
Defects	1530

2.2. Image classification models

ResNet and EfficientNet offer multiple versions of models, making them well-suited for transfer learning. To balance computational efficiency with performance, this study experimented with and selected ResNet-50 and EfficientNetV2B0 and implemented task-specific adaptations tailored to powder bed image characteristics.

Before training, the image data was pre-processed. First, the three datasets were merged to expand the dataset. The first 8 – 10 images from each print project, which appear overexposed due to high energy and platform reflection, were removed, along with the last few "meaningless" images from the end of the print process. This resulted in a total of 2,638 powder bed images. Data were then manually categorized into "good" and "defects" classes (Table 2). Finally, the dataset was split into training, test, and validation sets at a 70:15:15 ratio.

The default input size for the models was 224 × 224, while the dataset used in this experiment consisted of high-resolution 1280 × 1024 images. Directly inputting the original size would result in insufficient memory, and significantly downsizing the images would risk losing critical features of small defect targets. Therefore, a Python script was used to crop the "defects" images to retain only the defect region and 10% of the surrounding background. To improve the model's generalization, OpenCV was then applied for data augmentation, including normalization, rotation, horizontal flipping, scaling, and shear transformations. Finally, all images were resized to 224 × 224 pixels.

During transfer learning, multiple adjustments were made to optimize training performance. First, ResNet-50

and EfficientNetV2-B0 models pre-trained on ImageNet were imported from the TensorFlow/Keras library. Custom fully connected layers were created for comparison, and after testing Flatten and GlobalAveragePooling2D, the former was selected. Two additional dense layers were added with 256 and 128 neurons, using Rectified Linear Unit (ReLU) as the activation function. Both models included a dropout layer, where ResNet-50 randomly dropped 20% of the neurons to reduce reliance on specific neurons and enhance generalization. Given that EfficientNetV2-B0 converges quickly with a significant accuracy increase in the early epochs, the dropout rate was increased to 0.3 to prevent overfitting. The last 5 – 10 convolutional layers were unfrozen, and the model was recompiled for further training, allowing it to learn the characteristics of the new dataset.

To prevent training stagnation, a learning rate scheduling strategy was applied. The validation loss was monitored, and if no decrease was observed over five epochs, the learning rate was reduced by 20% until 1e-6, with an initial learning rate set to 1e-4. L2 regularization was introduced to improve accuracy and was adjusted from 0.02 to 0.001 to avoid excessive loss. During training, accuracy and loss steadily decreased, indicating good convergence on the training set, but validation loss remained constant, suggesting poor generalization. To address this, the training-to-validation set ratio was adjusted from 70:15:15 to 60:30:10, and the batch size was set to 32.

In addition, an EarlyStopping callback was added to prevent overfitting. EarlyStopping monitors the validation loss, and if it fails to decrease over five epochs, training is stopped, and the model is reverted to the best-performing weights. Epochs were increased to allow the accuracy to stabilize.

Given this study's binary classification problem and class imbalance, the loss function was modified from binary cross-entropy loss to focal loss ($\gamma = 2$; $\alpha = 0.25$) to optimize detection performance. Focal loss is an improved version of cross-entropy loss that introduces a weighting factor α to give weights to the positive and negative classes, focusing on harder-to-classify samples. This method is more effective for handling class imbalance. In this dataset, "good" was treated as the positive class (1) and "defects" as the negative class (0). Although the classes were relatively balanced, due to the nature of defect detection, where missing a defect is undesirable, α was set to 0.25 to encourage the model to focus more on the "defects" class.

These adaptations were designed specifically for the characteristics of AM image data and helped improve model convergence and detection accuracy. The experiment involved training with various parameter combinations, and the final parameter settings are displayed in Table 3.

2.3. Object detection models

In the second phase of the task, manual annotation of the dataset is required to achieve precise localization. In actual powder bed images, 3 – 8 overlapping defects are commonly present, along with interference and noise, making it difficult to distinguish and label specific defect types accurately. Since, in practical scenarios, the goal is only to detect the presence and location of defects to stop printing in time and reduce losses, a simple and general bounding box annotation method is sufficient to meet the requirements. For further research into the causes of defects, pixel-level semantic segmentation methods could be considered. In this study, LabelImg was used to annotate the defect locations in all images as "defects," and the model learns from these annotations to predict potential defect locations in the test images, highlighting them with bounding boxes for inspection.

Before training, approximately 10 – 20% of the images' regions outside the build platform were cropped to maintain a consistent image size, allowing for more detailed information when inputting the images at 600 × 600. The dataset was split using a commonly applied ratio, as displayed in Table 4. In addition to classification loss, Faster R-CNN also uses bounding box regression loss, while the YOLOv5 model incorporates confidence loss.

In this experiment, the YOLOv5 model was selected due to its availability in multiple sizes to meet different task requirements. It is based on PyTorch and features an optimized architecture with reasonable model complexity,

Table 3. Parameter settings of ResNet-50 and EfficientNetV2B0 models

Parameters	Specification	
	ResNet-50	**EfficientNetV2B0**
Batch size	32	32
Learning rate	1e-5	1e-4
Input image size	224×224	224×224
Loss function	Focal loss (γ=2, α=0.25)	Focal loss (γ=2, α=0.25)
Regularization	L2 (0.001)	L2 (0.001)
Layer types	Flatten, Dense (256), Dense (128), Dropout	Flatten, Dense (256), Dense (128), Dropout
Dropout rate	0.2	0.3
Epoch	50	50
Data augmentation settings	• *Normalization:* Rescale=1./255 • *Rotation:* Rotation range=40 • *Horizontal flip:* Horizontal flip=True • *Width shift:* Width shift range=0.2 • *Height shift:* Height shift range=0.2 • *Shear:* Shear range=0.2 • *Zoom:* Zoom range=0.2	

Table 4. Parameter settings for the faster R-CNN model

Parameter	Value	Description
(Training+Validation): Test	9:1	Split of training, validation, and test sets
Training: Validation	9:1	Split of training and validation data
input_shape	(600, 600)	Each input image is resized to 600×600
Backbone	resnet50	Used for feature extraction
anchors_size	(4, 16, 32)	Sets anchor box sizes during training, generating anchor boxes at three different scales to detect objects of varying sizes
Freeze_Epoch	50	End epoch for the frozen stage
Freeze_batch_size	4	Batch size during the frozen stage
Freeze_lr	1e-4	Learning rate during the frozen stage
nFreeze_Epoch	100	End epoch for the unfrozen stage
Unfreeze_batch_size	4	Batch size during the unfrozen stage
Unfreeze_lr	1e-5	Learning rate during the unfrozen stage
Confidence threshold	0.5	Predicted boxes with confidence scores above 0.5 are considered valid detections during object detection
nms_iou threshold	0.3	Intersection Over Union threshold for non-maximum suppression; when the overlap of two predicted boxes exceeds 0.3, the model retains the higher-scoring box and suppresses the other to reduce redundancy
Test anchors_size	(4, 8, 16)	Anchor box sizes used during model evaluation

R-CNN: region-based-Convolutional neural network

faster inference speed, and sound detection performance. YOLOv5 employs mosaic data augmentation, which combines four images into one input image, enhancing the detection of small objects. The model also provides an automated hyperparameter optimization system that searches for the best training configuration for hyperparameters, such as learning rate, weight decay, and mosaic probability.[39]

The model training consists of two stages: Freezing and unfreezing. First, the dataset and pre-trained weights are loaded (to accelerate training and enhance feature extraction capabilities). In the freezing stage, only the detection head parameters are trained, while the backbone network weights remain unchanged to reduce memory usage. In the unfreezing stage, the backbone network parameters are unlocked, allowing the entire model to participate in training and improve performance. During training, loss values are recorded, and model weights are saved after each epoch. A training log is generated to evaluate the model's performance.

The main settings and parameters used for the Faster R-CNN and YOLOv5 models are presented in Tables 4 and 5, respectively.

3. Results

3.1. Loss and accuracy

The loss and accuracy changes for the image classification models, ResNet50 and EfficientNetV2B0, are featured in Figures 2 and 3, respectively. Both training and validation losses for ResNet-50 decrease rapidly in the initial stages and stabilize, with closely aligned trends, indicating good model fitting and effective learning without overfitting. However, its validation accuracy fluctuates, suggesting weaker generalization. In addition, the training accuracy of EfficientNetV2B0 surpasses 90% within the first few batches and remains stable at around 100%, demonstrating superior learning capability and stability compared to ResNet-50. To ensure a fair comparison, both models were trained for 50 epochs, but EfficientNetV2B0 exhibited slight overfitting in the final stages, as indicated by declining training loss, increasing validation loss, and fluctuations in validation accuracy.

The training and validation loss changes during the Faster R-CNN training process are displayed in Figure 4. In the first 50 epochs of the freezing phase, the training and validation losses rapidly decrease in the initial few epochs and then gradually stabilize. The smoothness of the training and validation losses, with a difference of <0.1, indicates that the model did not experience significant overfitting or underfitting. During the following 50 epochs, when the unfreezing phase begins and the model undergoes full parameter updates, fluctuations in the loss are normal. However, after that, the loss decreases but does not fully converge.

One limitation of the Faster R-CNN model is that it selects 256 mini-batches of anchor boxes from the same

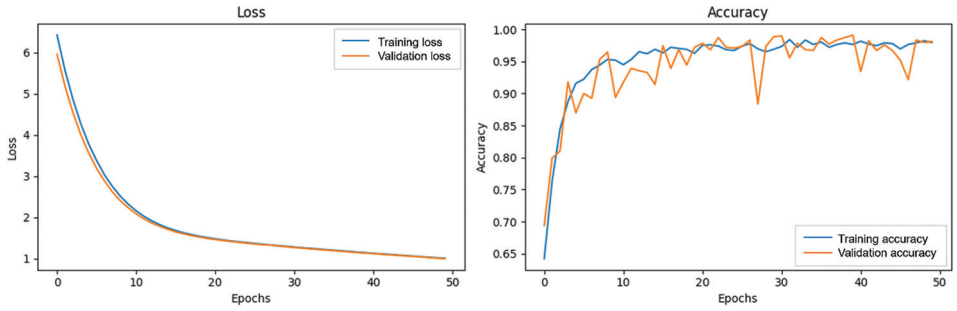

Figure 2. Loss (left) and accuracy plot (right) of the ResNet50 model

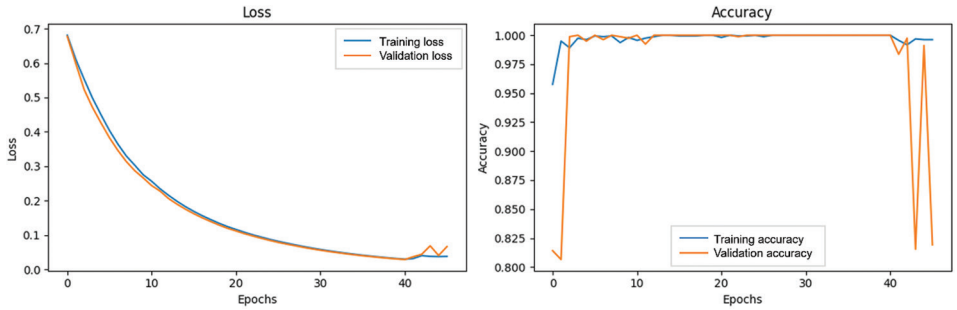

Figure 3. Loss (left) and accuracy plot (right) of the EfficientNetV2B0 model

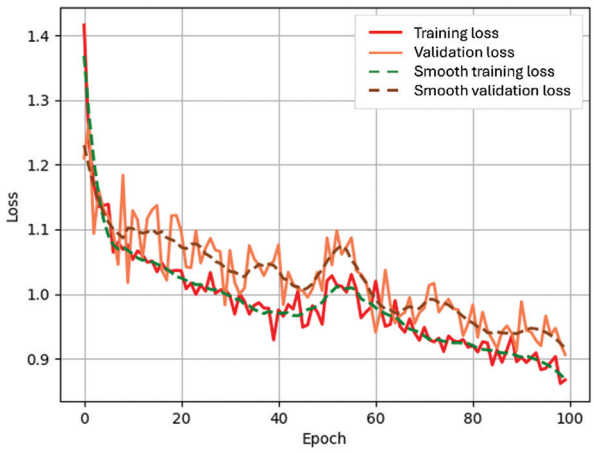

Figure 4. Loss plot of the faster region-based-convolutional neural network model

image for training, and the features of these anchors may be similar, which limits the variety of information the network receives. As a result, the network requires more iteration to learn generalized features, which can lead to longer convergence times.

Figure 5 displays the training and validation losses of the YOLO model over 100 epochs, including box loss, object loss, and classification loss, as well as key metrics, such as precision, recall, and mean average precision (mAP). The bounding box loss (train/box_loss and val/box_loss) and object loss (train/obj_loss and val/obj_loss) both gradually decrease in the training and validation sets, indicating that the model is continuously optimizing and learning. The model demonstrates good adaptability in distinguishing between foreground objects and the background. While the training loss is relatively smooth, the validation loss exhibits some fluctuation, which may be attributed to variations in the validation data. The classification loss remains zero throughout the training process, as the dataset consists of a single-class object detection task. As training progresses, precision and recall steadily increase, with the precision-recall curve stabilizing at a high level of around 0.8. This suggests that most predicted positive samples are correct, with a low false positive rate. The recall increases rapidly in the early stages and stabilizes around 0.6 – 0.7. The loss and evaluation metrics for the validation set fluctuate more significantly, likely due to the dataset's abstract and complex nature of the target features.

3.2. Image classification models comparison

Model validation is a key step in assessing performance, where the trained model is tested on unseen data to evaluate its ability to generalize. This ensures its effectiveness in practical applications. A confusion matrix is commonly used to evaluate model performance in image classification. The diagonal values indicate correct

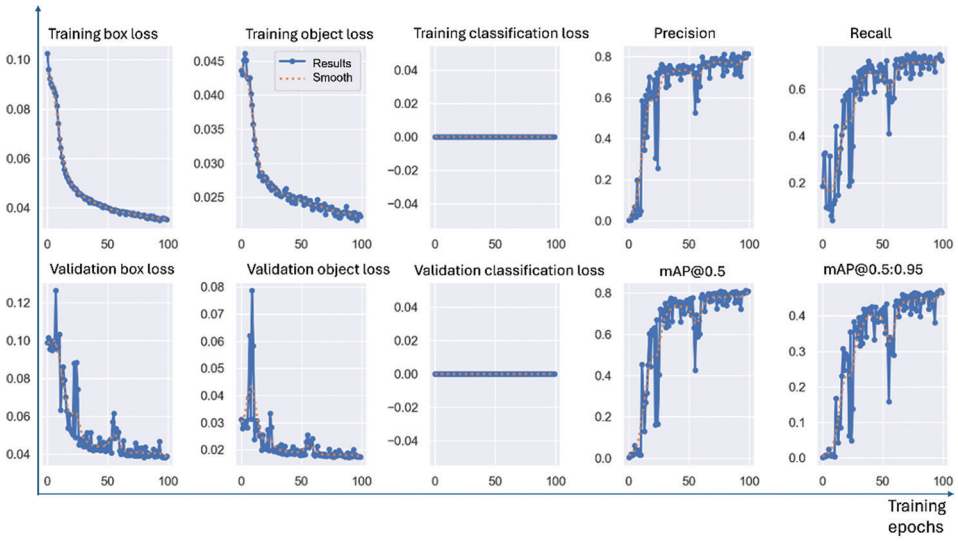

Figure 5. Performance of the YOLOv5 model

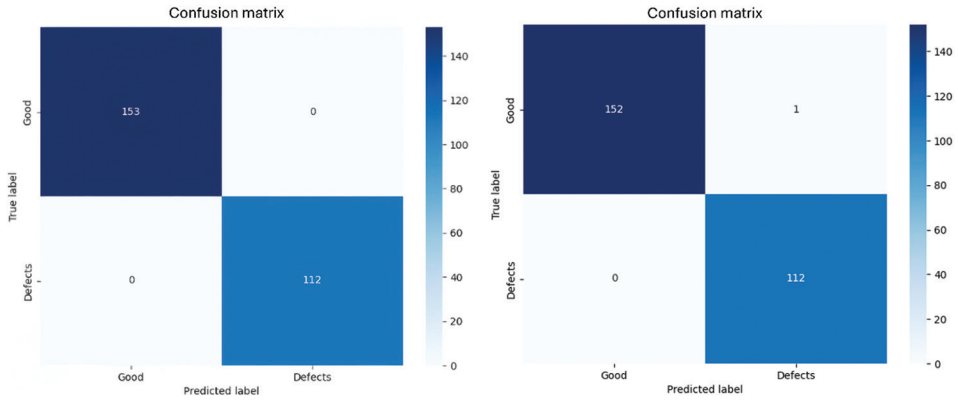

Figure 6. Confusion matrix from the ResNet50 (left) and EfficientNetV2B0 (right) models

predictions, while metrics, including accuracy, precision, and recall, are derived from the matrix to provide a comprehensive assessment (Figure 6).

In the 265 test image samples (Table 6), ResNet50 achieved perfect classification, while EfficientNetV2B0 performed equally well, with only one "good" image misclassified as "defects" and all other samples correctly classified.

After confirming that there was no information leakage, the results (Table 6) demonstrated that the image classification model could always complete the classification task perfectly, with almost no missed or incorrect detections. The choice of different loss functions made little difference since both achieved nearly 100% accuracy. On the test set, both models achieved fast detection, with the EfficientNetV2B0 model demonstrating a faster detection speed. This indicates that these two models found it relatively easy to learn the small- to medium-sized image datasets generated by 3D printers, effectively capturing the key features. This demonstrates good potential for future development and suggests that these models could handle more complex AM image datasets.

For the EfficientNetV2B0 model with deeper networks, the defect classification task appears relatively simple, reaching over 90% accuracy within five iterations and remaining at nearly 100% thereafter. If not overtrained, it outperforms ResNet50 by learning faster and demonstrating excellent stability.

3.3. Object detection models comparison

As displayed in Figure 7, Faster R-CNN achieved an accuracy of 46.25%, which is measured by the area under the precision and recall curves, reflecting the overall detection performance. A higher recall suggests that more

Table 5. Parameter settings for the Yolo-v5 model

Parameter	Value	Description
lr0	0.01	Initial learning rate
lrf	0.01	The learning rate follows a cosine decay schedule, defined as: $lr(x)=\left(1-\cos\left(\frac{x*\pi}{2*step}\right)\right)*(1-lrf)+lrf$ The final learning rate is: $lrfinal=lr0+lrf$
Momentum	0.937	Used in the optimizer to control gradient updates, helping to stabilize convergence
Weight decay	0.0005	Weight decay for regularization helps prevent overfitting by limiting the model's complexity
Warmup epochs	3.0	Several warmup epochs, where the learning rate gradually increases to avoid unstable gradients at the start
Warmup momentum	0.8	During the warmup phase, momentum values gradually increase to the value set by the momentum
Warmup bias lr	0.1	The learning rate for bias terms during the warmup phase
Box	0.05	Weight for the bounding box loss, controlling the contribution of the box regression loss (using CIOU Loss)
cls	0.5	Classification loss weight
cls pw	1.0	Weight coefficient for classification loss
obj	1.0	Weight for object loss uses BCE Loss to measure confidence in object presence
obj pw	1.0	Weight coefficient for object loss
you t	0.25	Intersection Over Union threshold; this defines whether a predicted box matches a ground truth box
anchor t	4.0	Anchor threshold, determining if an anchor box needs adjustment

Abbreviations: BCE: Binary cross-entropy; CIOU: Complete intersection over union.

Table 6. Results of the Resnet50 and EfficientNetV2B0 models

Model	Loss	Accuracy (%)	Precision	Recall	AUC	Time/image (s)
Resnet-50	0.9916	100	1.0000	1.0000	1	0.0272
EfficientNetV2B0	0.0475	99.62	0.9912	1.0000	0.9937	0.0182

Abbreviation: AUC: Area under the ROC curve.

defects are detected but also increases false positives, which lowers precision. The broad coverage of the curve in the figure suggests that the model can maintain a reasonable level of precision at a higher recall, demonstrating better generalization capability in identifying defects.

Figure 8 presents the performance metrics for the YOLOv5 model on the test set. "Images" represents the number of images, while "Instances" denotes the total number of target instances in those images. P (or precision) measures how accurately the model identifies abnormal regions, while R (or recall) indicates the percentage of actual defects detected by the model. The mAP reflects the overall effectiveness of the model. *mAP@0.5* is *mAP at IoU threshold = 0.5*, and *mAP@0.5:0.95* is *mAP averaged over IoU thresholds from 0.5 to 0.95 with a step size of 0.05*. YOLOv5 outperforms Faster R-CNN at mAP50, but its mAP50 – 95 (47.3%) suggests room for improvement in detecting small targets or complex backgrounds.

From Table 7, comparing the two object detection models reveals that YOLOv5 significantly outperforms Faster R-CNN. The YOLOv5 model achieved higher average precision (AP), precision, and recall rates on both datasets. The ground truth objects and detected objects in the Faster R-CNN model display many overlapping detection boxes, which can be adjusted by modifying the threshold. In terms of inference time, YOLOv5 is significantly faster than Faster R-CNN. By directly processing high-resolution images on standard CPU resources, both models achieve detection speeds under 2 s, which is much shorter than the printing time of a single layer on the EOS M290 (typically 10 – 60 s). This ensures that each layer can be monitored in real-time. With more powerful computing hardware, even faster batch processing could be achieved.

As displayed in Figures 9 and 10, although mAP is not exceptionally high, the models can still effectively identify and mark prominent image defects. When the

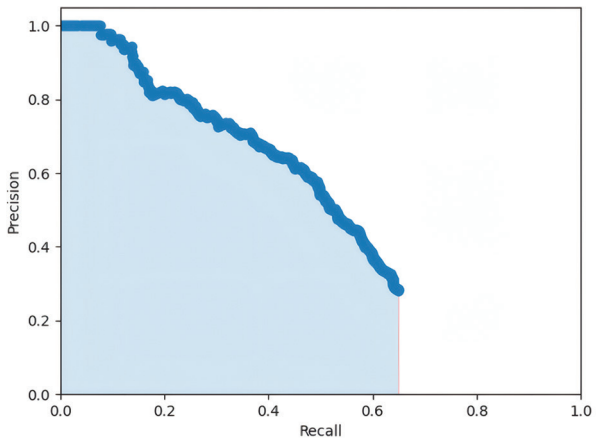

Figure 7. Precision-recall curve of the Faster R-convolutional neural network model. Average precision for the "defects" class is 46.25%

Images	Instances	P	R	mAP50	mAP50-95
153	496	0.797	0.75	0.815	0.473

Figure 8. Metrics of the YOLOv5 model

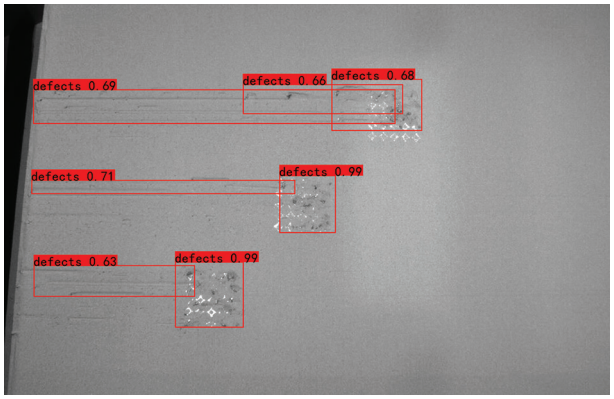

Figure 9. Test result (sample 1) of the object detection model

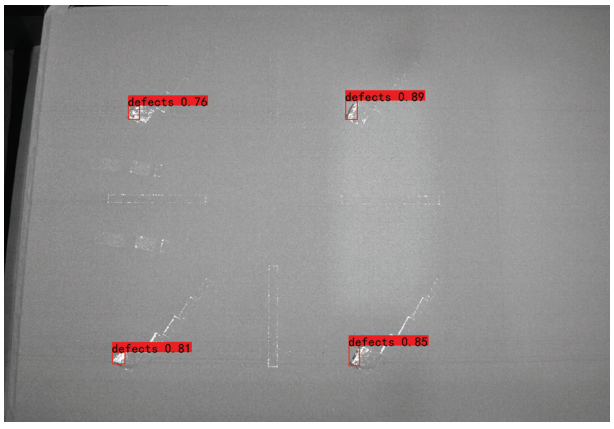

Figure 10. Test result (sample 2) of the object detection model

Table 7. Evaluation results of Faster R-CNN and YOLOv5 models

Parameter	Specification	
	Faster R-CNN	YOLOv5
mAP (%)	46.25	81.5
Test precision (%)	47.68	79.7
Test recall (%)	53.52	75.0
Ground truth objects	482	-
Detected objects	1112	-
Inference time/image (s)	1 – 2	0.5 – 1

Abbreviation: mAP: Mean average precision,
R-CNN: region-based-Convolutional neural network.

defect regions tend to have a relatively uniform shape and size, the model's confidence in predicting defects is high, approaching 1. When the defects to be detected are small, the model struggles to capture the defect's boundaries accurately. Due to the complex and abstract shapes of the defects with varying sizes, the model often produces overlapping detection boxes, which reduces confidence scores. However, the actual detection performance is already satisfactory for supporting manual inspection needs.

4. Discussion

4.1. Analysis of outcome

Based on the experimental results, the ResNet50 and EfficientNetV2B0 models used for image classification performed exceptionally well in distinguishing defective images after transfer learning, with test set accuracy of nearly 100%. However, before training the model, it is crucial to pre-process the raw images by segmenting the defect areas. Without this step, potential subtle defects may be lost during image downscaling, leading to an inability to detect printing issues promptly. Processing high-resolution images requires significant computational power, which can be challenging to access in real-world production settings. Directly feeding raw, unprocessed images into the model may result in suboptimal detection outcomes.

This experiment used three image datasets from defective printing processes. Although the overall data volume is relatively large and the distribution between normal and defective samples is fairly balanced, the nature of AM leads to minimal variation between layer-wise images, and many defects are highly similar and repetitive. This may limit the model's learning capacity. To improve generalization on new data, we applied various data augmentation techniques to increase diversity, aiming

for the model to perform well not only on specific print samples but also on new types of defects caused by different materials in future applications. In future work, we hope to expand the dataset with more experimental data or conduct more detailed classification and detection studies based on different materials and printing builds.

In the object detection phase, the YOLOv5 model achieved an AP of 81.5%, significantly higher than the AP of the Faster R-CNN model (i.e., 46.25%). This suggests that YOLOv5 may be better suited for PBF-LB defect detection tasks, offering stronger adaptability to complex, multi-scale defects with a faster inference speed of under 1 s. Nevertheless, the Faster R-CNN model also demonstrated satisfactory detection results, effectively identifying potential defects and their locations within 2 s, which ensures timely detection of layer-wise changes and confirms its feasibility for real-time monitoring and assisting manual inspection.

The generally lower precision of object detection models compared to image classification models can be attributed to several limitations, including complex defect types, variable object scales, low-contrast backgrounds, and suboptimal annotation quality. These factors often result in false positives or missed detections, reducing precision, recall, and AP. Nevertheless, this approach paves the way for real-time compensation of bounding box-based defective regions. Once the defective regions are detected, it is possible to repair them by recoating the powder bed and/or exposing the defective region to a laser with standard or customized volumetric energy density. More precise mask annotations can be used to further improve model performance. Pixel-level labeling is particularly effective for detecting complex shapes that require accurate localization, as it helps reduce boundary errors and enhances detection accuracy.

4.2. Comparison to previous studies

Most previous studies have adopted image classification methods, and this study followed a similar approach to evaluate model performance (Table 8). The reported accuracies in the literature vary widely (70 – 100%),

Table 8. Previous studies on image classification models

Author, year	Research object	Model (s) used	Accuracy (%)	References
Yin *et al*., 2025	Defect detection in PBF-LB	Resnet50 and EfficientNetV2B0	Resnet 50: 100 EfficientNetV2B0: 99.62	This work
Han *et al*., 2019	Microscopic metal images for AM defect detection	Inception-ResNet-v2	87.5	30
Khan *et al*., 2021	FFF 3D printing defect detection	CNN	84	7
Kadam *et al*., 2021	FDM defect detection	AlexNet+SVM	99.7	16
Westphal and Seitz, 2021	PBF defects in the selective laser sintering process	VGG16 and Xception CNN	VGG16: 95.8 Xception CNN: Not specified	25
Ansari *et al*., 2022	Porosity detection in PBF-LB	Custom CNN	CAD labels: 90 XCT labels: 97	27
Fu *et al*., 2022	Overview of ML-based defect detection in PBF-LB	CNN, SVM, and other ML models	~75 – 95 across studies	29
Abhilash and Ahmed, 2023	Surface quality improvement in metal AM	ResNet-50 CNN	96	32
Akmal *et al*., 2023	Defect detection in PBF-LB	CNN, ANN, MLR	CNN: 100 ANN and MLR: ~99	41
Khan *et al*., 2023	Anomaly detection in PBF-LB using OT imaging	Random forest	99.98	31
Lee *et al*., 2023	Defect detection in PBF-LB	3D-CNN	Recall: 70.47	33
Schmitt *et al*., 2023	Powder bed monitoring in metal AM	Xception-style neural network	~99.15 for large patches	42
Kozhay *et al*., 2024	Defect detection in FDM	Custom CNN with ResNet backbone	97	28
Kuriachen *et al*., 2025	Defect detection in FDM	Custom lightweight CNN	97.77	43

Abbreviations: AM: Additive manufacturing; ANN: Artificial neural network; CAD: Computer-aided design; CNN: Convolutional neural network; FDM: Fused deposition modeling; FFF: Fused filament fabrication; ML: Machine learning; MLR: Multinomial logistic regression; OT: Optical tomography; PBF-LB: Laser-based powder bed fusion; SVM: Support vector machine; XCT: X-ray computed tomography.

Table 9. Previous studies on object detection models

Reference	Research object	Model (s) used	Average precision (%)	References
Yin *et al*., 2025	Defect localization in PBF-LB	Faster R-CNN and YOLOv5	Faster R-CNN: 46.25 YOLOv5: 81.5	This work
Paraskevoudis *et al*., 2020	Detection of stringing defects in FFF 3D printing	SSD with VGG16	44	46
Scime *et al*., 2020	Powder bed anomaly detection	Dynamic Segmentation CNN	Pixel-wise accuracy: >90	49
Cannizzaro *et al*., 2021	PBF defect detection	Computer Vision+U-Net	≥75	19
Wen *et al*., 2021	Detection of cracks and pores in PBF-LB	YOLOv4 (detection) and Detectron2 (segmentation)	~50	34
Wang *et al*., 2024	Small defect detection in metallic AM based on CT images	DC-RCNN	73.3	47
Dong *et al*., 2025	Internal defect detection in AM 6061 aluminum alloy using laser ultrasound	YOLOv5	93.1%	48

Abbreviations: AM: Additive manufacturing; CNN: Convolutional neural network; CT: Computed tomography; DC-R-CNN: Depth-connected region-based convolutional neural network; FFF: Fused filament fabrication; PBF-LB: Laser-based powder bed fusion; SSD: Single shot detector.

depending on factors such as dataset size, defect types, and image quality. Although direct comparisons are not conclusive due to these differences, the cited results provide a general context for interpreting our findings.

Previous studies employed traditional ML algorithms, such as Support Vector Machine (SVM) and random forest,[16,29,31] or simpler deep learning models, such as basic CNNs for image classification tasks.[7,25] In contrast, this study adopts more sophisticated architectures, ResNet50 and EfficientNetV2B0. This study achieved near-perfect accuracy by integrating transfer learning methods to enhance model performance, outperforming many earlier studies. In addition, it was observed that EfficientNetV2B0 not only maintained a very high accuracy rate but also converged faster and demonstrated better stability.

Unlike most existing works that focus solely on image classification, this study systematically evaluated both classification and object detection models using a unified, real-world AM dataset and a consistent training pipeline for the 1st time.[44,45] By integrating recent architectures such as EfficientNetV2B0 and YOLOv5, which offer both accuracy and computational efficiency, the proposed dual-task framework addresses the practical demands of AM process monitoring and provides a valuable reference for future model selection and deployment in industrial defect detection.

Compared to image classification tasks, the application of object detection models for defect localization in AM remains highly underexplored, as illustrated in Table 9. Existing studies demonstrate that models trained with conventional annotations typically achieve AP values in the range of 40 – 50%.[34,46] A recent study by Wang *et al*.[47] proposed a depth-connected region-based (DC-RCNN) model for small defect detection on computed tomography images, but its performance was limited by a small dataset of only C-shaped components (AP up to 73.3%). Dong *et al*.[48] further validated the high accuracy of YOLOv5 (93.1%) on laser ultrasonic data, though the dataset was purely simulated through COMSOL and limited comparisons beyond the YOLO family. In contrast, this study used a larger and more diverse real-world powder bed dataset, demonstrating that optimized YOLOv5 achieves higher AP across multiple defect types. This highlights the broader potential of object detection models for industrial AM applications.

Notably, some studies employed pixel-level annotations and semantic segmentation models, achieving higher AP values (≥75% and ≥90%, respectively).[19,49] These findings suggest that pixel-level annotations can be considered to enhance label quality and improve object detection models' accuracy.[50] Pixel-level annotation in AM requires domain-specific expertise and is prohibitively time-consuming for thousands of images. Consequently, most existing segmentation studies are conducted on relatively small datasets and result in highly task-specific models with limited generalizability. Moreover, segmentation models are computationally intensive, requiring greater computing resources and longer training times, which limits their real-time applicability in edge or online inspection scenarios. Nonetheless, segmentation remains important for deeper analysis of defect formation. Future work will explore advanced segmentation techniques to support root cause investigation and closed-loop quality control in AM.

5. Conclusion

This study comparatively evaluated two image classification models and two object detection models for defect identification and localization on a PBF-LB image dataset. The key findings are summarized below:

- ResNet50 and EfficientNetV2B0 achieved over 99% accuracy in classifying recoating defects with minimal

false predictions. Transfer learning proved effective in accelerating convergence and boosting performance under limited data diversity and computational resources, and appropriate image pre-processing improved the detection of small-scale defects.

- Given the limited exploration of object detection models in AM, this study investigates their potential and demonstrates that both Faster R-CNN and YOLOv5 can effectively localize defect regions to support human inspection. YOLOv5 displays greater robustness to scale variation and complex shapes, significantly outperforming Faster R-CNN.
- To address data imbalance and the scarcity of high-quality AM datasets, this study contributes an annotated image dataset. The high similarity among layer-wise images often limits model generalization across defect types. Public release of the dataset aims to increase data diversity, improve adaptability to novel defects, and support the development of intelligent AM quality monitoring.
- The limited precision of present object detection models is mainly due to the abstract nature of defects, scale variation, low contrast with backgrounds, noise interference, and overlapping bounding boxes. Future improvements may include dataset expansion, refined annotations (e.g., mask labeling), and adoption of advanced detection frameworks to enhance accuracy and generalizability for industrial applications.

Overall, this work provides a comprehensive, task-aligned evaluation of CNN models for AM defect monitoring, supported by a realistic dataset and performance benchmarks. The findings serve as a valuable reference for future research on model selection, deployment strategies, and data standardization in AM quality control.

Acknowledgments

The authors would like to thank Björkstrand Roy, the laboratory manager of ADDLAB at Aalto University, for providing the raw data used in this study.

Funding

This research was financially supported by the Finnish Doctoral Program Network in Artificial Intelligence (AI-DOC, decision number VN/3137/2024-OKM-6) and the Tandem Industry Academia funding from the Finnish Research Impact Foundation.

Conflict of interest

Mika Salmi serves as the Editorial Board Member of the journal but was not in any way involved in the editorial and peer-review process conducted for this paper, directly or indirectly. Other authors declare they have no competing interests.

Author contributions

Conceptualization: Mika Salmi and Xinyi Yin
Data curation: Xinyi Yin
Formal analysis: Xinyi Yin
Methodology: Xinyi Yin and Jan Akmal
Software: Xinyi Yin
Supervision: Mika Salmi and Jan Akmal
Visualization: Xinyi Yin
Writing – original draft: Xinyi Yin
Writing – review and editing: Mika Salmi and Jan Akmal

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

The dataset used in this study has been deposited in Zenodo and is publicly available at: https://doi.org/10.5281/zenodo.14996806.

References

1. Herzog T, Brandt M, Trinchi A, Sola A, Molotnikov A. Process monitoring and machine learning for defect detection in laser-based metal additive manufacturing. *J Intell Manuf*. 2024;35(4):1407-1437.

   doi: 10.1007/s10845-023-02119-y

2. Kim H, Lin Y, Tseng TLB. A review on quality control in additive manufacturing. *Rapid Prototyp J*. 2018;24(3):645-669.

   doi: 10.1108/RPJ-03-2017-0048

3. Vasques CMA, Cavadas AMS, Abrantes JCC. Technology overview and investigation of the quality of a 3D-printed maraging steel demonstration part. *MSAM*. 2025;4(2):025040002.

   doi: 10.36922/msam025040002

4. DebRoy T, Wei HL, Zuback JS, *et al*. Additive manufacturing of metallic components-process, structure and properties. *Prog Mater Sci*. 2018;92:112-224.

   doi: 10.1016/j.pmatsci.2017.10.001

5. Tan C, Li R, Su J, *et al*. Review on field assisted metal additive manufacturing. *Int J Mach Tools and Manuf*. 2023;189:104032.

   doi: 10.1016/j.ijmachtools.2023.104032

6. Brennan MC, Keist JS, Palmer TA. Defects in metal additive manufacturing processes. *J Mater Eng Perform*. 2021;30(7):4808-4818.

doi: 10.1007/s11665-021-05919-6

7. Khan MF, Alam A, Siddiqui MA, *et al*. Real-time defect detection in 3D printing using machine learning. *Mater Today Proc*. 2021;42:521-528.

   doi: 10.1016/j.matpr.2020.10.482

8. Gunasegaram DR, Barnard AS, Matthews MJ, *et al*. Machine learning-assisted *in-situ* adaptive strategies for the control of defects and anomalies in metal additive manufacturing. *Addit Manuf*. 2024;81:104013.

   doi: 10.1016/j.addma.2024.104013

9. Cataldo SD, Vinco S, Urgese G, *et al*. Optimizing quality inspection and control in powder bed metal additive manufacturing: Challenges and research directions. *Proc IEEE*. 2021;109(4):326-346.

   doi: 10.1109/JPROC.2021.3054628

10. Xiao Y, Wang X, Yang W, *et al*. Data-driven prediction of future melt pool from built parts during metal additive manufacturing. *Addit Manuf*. 2024;93:104438.

    doi: 10.1016/j.addma.2024.104438

11. Tamir TS, Xiong G, Shen Z, *et al*. 3D printing in materials manufacturing industry: A realm of Industry 4.0. *Heliyon*. 2023;9(9):e19689.

    doi: 10.1016/j.heliyon.2023.e19689

12. Goh GD, Sing SL, Yeong WY. A review on machine learning in 3D printing: Applications, potential, and challenges. *Artif Intell Rev*. 2021;54(1):63-94.

    doi: 10.1007/s10462-020-09876-9

13. Liu Q, Chen W, Yakubov V, Kruzic JJ, Wang CH, Li X. Interpretable machine learning approach for exploring process-structure-property relationships in metal additive manufacturing. *Addit Manuf*. 2024;85:104187.

    doi: 10.1016/j.addma.2024.104187

14. Ng WL, Goh GL, Goh GD, Ten JSJ, Yeong WY. Progress and opportunities for machine learning in materials and processes of additive manufacturing. *Adv Mater*. 2024;36(34):2310006.

    doi: 10.1002/adma.202310006

15. Ukwaththa J, Herath S, Meddage DPP. A review of machine learning (ML) and explainable artificial intelligence (XAI) methods in additive manufacturing (3D printing). *Mater Today Commun*. 2024;41:110294.

    doi: 10.1016/j.mtcomm.2024.110294

16. Kadam V, Kumar S, Bongale A, Wazarkar S, Kamat P, Patil S. Enhancing surface fault detection using machine learning for 3D printed products. *Appl Syst Innov*. 2021;4(2):34.

    doi: 10.3390/asi4020034

17. Chua ZY, Ahn IH, Moon SK. Process monitoring and inspection systems in metal additive manufacturing: Status and applications. *Int J Precis Eng Manuf Green Technol*. 2017;4(2):235-245.

    doi: 10.1007/s40684-017-0029-7

18. Chang LK, Chen RS, Tsai MC, *et al*. Machine learning applied to property prediction of metal additive manufacturing products with textural features extraction. *Int J Adv Manuf Technol*. 2024;132(1):83-98.

    doi: 10.1007/s00170-024-13165-y

19. Cannizzaro D, Varrella AG, Paradiso S, *et al*. *Image Analytics and Machine Learning for In-Situ Defects Detection in Additive Manufacturing*. United States: IEEE; 2021. p. 603-608.

20. Akbari P, Zamani M, Mostafaei A. Machine learning prediction of mechanical properties in metal additive manufacturing. *Addit Manuf*. 2024;91:104320.

    doi: 10.1016/j.addma.2024.104320

21. Tang Y, Rahmani Dehaghani M, Sajadi P, Wang GG. Selecting subsets of source data for transfer learning with applications in metal additive manufacturing. *J Intell Manuf*. 2024;36:3185-3206.

    doi: 10.1007/s10845-024-02402-6

22. Zhang Y, Safdar M, Xie J, Li J, Sage M, Zhao YF. A systematic review on data of additive manufacturing for machine learning applications: The data quality, type, preprocessing, and management. *J Intell Manuf*. 2023;34(8):3305-3340.

    doi: 10.1007/s10845-022-02017-9

23. Liu X, Mileo A, Smeaton AF. *A Systematic Review of Available Datasets in Additive Manufacturing*. [arXiv Preprint]; 2024.

24. Djenouri Y, Srivastava G, Lin JCW. Applied AI in defect detection for additive manufacturing: Current literature, metrics, datasets, and open challenges. *IEEE Instrum Meas Mag*. 2024;27(4):46-53.

    doi: 10.1109/MIM.2024.10540405

25. Westphal E, Seitz H. A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. *Addit Manuf*. 2021;41:101965.

    doi: 10.1016/j.addma.2021.101965

26. Szymanik B, Psuj G, Hashemi M, Lopato P. Detection and identification of defects in 3D-printed dielectric structures via thermographic inspection and deep neural networks. *Materials*. 2021;14:4168.

    doi: 10.3390/ma14154168

27. Ansari MA, Crampton A, Garrard R, Cai B, Attallah M. A convolutional neural network (CNN) classification to identify the presence of pores in powder bed fusion images. *Int J Adv Manuf Technol*. 2022;120(7):5133-5150.

    doi: 10.1007/s00170-022-08995-7

28. Kozhay K, Turarbek S, Asselbekova T, Ali MH, Shehab E. Convolutional neural network-based defect

detection technique in FDM technology. *Procedia Comput Sci*. 2024;231:119-128.

doi: 10.1016/j.procs.2023.12.183

29. Fu Y, Downey ARJ, Yuan L, Zhang T, Pratt A, Balogun Y. Machine learning algorithms for defect detection in metal laser-based additive manufacturing: A review. *J Manuf Process*. 2022;75:693-710.

doi: 10.1016/j.jmapro.2021.12.061

30. Han F, Zou J, Ai Y, Xu C, Liu S, Liu S. *Image Classification and Analysis During the Additive Manufacturing Process Based on Deep Convolutional Neural Networks*. United States: IEEE; 2019. p. 1-4.

31. Khan IA, Birkhofer H, Kunz D, Lukas D, Ploshikhin V. A random forest classifier for anomaly detection in laser-powder bed fusion using optical monitoring. *Materials (Basel)*. 2023;16(19):6470.

doi: 10.3390/ma16196470

32. Abhilash PM, Ahmed A. Convolutional neural network-based classification for improving the surface quality of metal additive manufactured components. *Int J Adv Manuf Technol*. 2023;126(9):3873-3885.

doi: 10.1007/s00170-023-11388-z

33. Lee KH, Lee HW, Yun GJ. A defect detection framework using three-dimensional convolutional neural network (3D-CNN) with *in-situ* monitoring data in laser powder bed fusion process. *Optics Laser Technol*. 2023;165:109571.

doi: 10.1016/j.optlastec.2023.109571

34. Wen H, Huang C, Guo S. The application of convolutional neural networks (CNNs) to recognize defects in 3D-printed parts. *Materials (Basel)*. 2021;14(10):2575.

doi: 10.3390/ma14102575

35. Yin X, Akmal JS, Salmi M, Björkstrand R. *Data From: Annotated Image Dataset for Defects Detection in Laser Powder Bed Fusion*. Geneva: Zenodo; 2025.

doi: 10.5281/zenodo.14996806

36. He K, Zhang X, Ren S, Sun J. *Deep Residual Learning for Image Recognition*. United States: IEEE; 2016. p. 770-778.

37. Tan M. *Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks*. [arXiv Preprint]; 2019.

38. Ren S. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. [arXiv Preprint]; 2015.

39. Redmon J. *You Only Look Once: Unified, Real-Time Object Detection*. United States: IEEE; 2016.

40. Chen K, Zhang P, Yan H, *et al*. A review of machine learning in additive manufacturing: Design and process. *Int J Adv Manuf Technol*. 2024;135(3):1051-1087.

doi: 10.1007/s00170-024-14543-2

41. Akmal J, Macarie M, Björkstrand R, Minet K, Salmi M. Defect detection in laser-based powder bed fusion process using machine learning classification methods. *IOP Conf Ser Mater Sci Eng*. 2023;1296(1):012013.

doi: 10.1088/1757-899X/1296/1/012013

42. Schmitt AM, Sauer C, Höfflin D, Schiffler A. Powder bed monitoring using semantic image segmentation to detect failures during 3D metal printing. *Sensors (Basel)*. 2023;23(9):4183.

doi: 10.3390/s23094183

43. Kuriachen B, Jeyaraj R, Raphael D, Ashok P, Sundari PS, Paul A. Defect detection in fused deposition modelling using lightweight convolutional neural networks. *Eng Appl Artif Intell*. 2025;141:109802.

doi: 10.1016/j.engappai.2024.109802

44. Sousa J, Brandau B, Darabi R, *et al*. Artificial intelligence for control in laser-based additive manufacturing: A systematic review. *IEEE Access*. 2025;13:30845-30860.

doi: 10.1109/ACCESS.2025.3537859

45. Soori M, Jough FKG, Dastres R, Arezoo B. Additive manufacturing modification by artificial intelligence, machine learning, and deep learning: A review. *Addit Manuf Front*. 2025;4(2):200198.

doi: 10.1016/j.amf.2025.200198

46. Paraskevoudis K, Karayannis P, Koumoulos EP. Real-time 3D printing remote defect detection (stringing) with computer vision and artificial intelligence. *Processes*. 2020;8(11):1464.

doi: 10.3390/pr8111464

47. Wang Y, Wang Z, Liu W, *et al*. A novel depth-connected region-based convolutional neural network for small defect detection in additive manufacturing. *Cognit Comput*. 2024;17(1):36.

doi: 10.1007/s12559-024-10397-8

48. Dong K, Ni M, Liang C, *et al*. Automatic detection and localization of internal defects in additively manufactured aluminum alloy based on deep learning. *Measurement*. 2025;244:116383.

doi: 10.1016/j.measurement.2024.116383

49. Scime L, Siddel D, Baird S, Paquit V. Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation. *Addit Manuf*. 2020;36:101453.

doi: 10.1016/j.addma.2020.101453

50. Ferguson M, Ak R, Lee YTT, Law KH. Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart Sustain Manuf Syst*. 2018;2(1):137-164.

doi: 10.1520/SSMS20180033