The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-026-19034-0

ORIGINAL ARTICLE

Open source deep learning models for accurate fastener defect inspection

Andrea Santomauro[1] · Luigi Portinale[1] · Stefania Montani[1] · Giorgio Leonardi[1] · Mario Brumini[2] · Guido Noce[2] · Enrico Ottaviano[2]

Received: 17 April 2026 / Accepted: 25 August 2026

Abstract
In fastener production, ensuring consistent product quality is essential to protect the safety and performance of components used in demanding applications. Although mechanical sorting equipment and early generations of machine-vision technology have successfully handled dimensional and structural checks, the reliable identification of surface flaws remains difficult. Recent progress in inspection technologies, especially the combination of Artificial Intelligence (AI), Deep Learning (DL), and advanced vision systems, has opened the door to highly accurate, cost-effective surface-defect detection that can outperform human inspectors. Despite these advances, most AI/DL models are designed for general use and still require substantial application-specific training and tuning, which slows large-scale deployment and leads to high costs. The present work investigates the possibility of adopting open source DL models, specifically Transformer-based architectures and Convolutional Neural Networks (CNNs), to classify fastener component images into defective or non-defective, and to segment defective ones, in order to enhance detection localization and to support continuous, high-volume quality-control processes in a cheaper, flexible and fully modifiable way. Our findings indicate that the built-in inductive biases of CNNs continue to offer a significant edge. As a result, CNNs, being generally simpler to train, are well-suited for practical industrial applications where computational resources and latency are key concerns. At the same time, Transformer-based models can serve as complementary approaches, with their effectiveness varying based on the dataset's properties and the type of defects involved.

Keywords Artificial vision systems · Deep learning · Fasteners quality control · Industrial research

1 Introduction

In the fastener manufacturing industry – which encompasses the production of pieces such as screws, gears, bolts, rivets, and threaded rods – maintaining high product quality is essential to guarantee the safety and reliability of end-use applications [1–3]. Even small defects, such as cracks, burrs, or irregularities in the threading, can undermine a component's structural soundness and lead to serious risks, particularly in sectors like automotive, aerospace, construction and railways [4, 5]. Traditionally, quality assurance in this field strongly depends on manual inspections [6]. Although effective to some extent, such inspections face limitations regarding speed, consistency, and objectivity. With the rise of mass production and increasingly stringent quality requirements – both for safety and to ensure uninterrupted operation of automated assembly lines – the shift toward automated defect detection and rejection systems

Andrea Santomauro, Luigi Portinale, Stefania Montani, Giorgio Leonardi, Mario Brumini, Guido Noce, Enrico Ottaviano contributed equally to this work.

✉ Andrea Santomauro
andrea.santomauro@uniupo.it

✉ Luigi Portinale
luigi.portinale@uniupo.it

✉ Stefania Montani
stefania.montani@uniupo.it

✉ Giorgio Leonardi
giorgio.leonardi@uniupo.it

[1] Computer Science Institute, DISIT, University of Piemonte Orientale, Alessandria, Italy

[2] DIMAC srl, Novara, Italy

Published online: 02 September 2026

has become indispensable [7]. Early solutions relied on mechanical sorting devices followed by the use of equipment incorporating machine vision and non-destructive testing techniques. These technologies can check dimensional accuracy and assess the structural integrity of materials. However, identifying surface flaws such as lines, scratches, stains, oxidation, and other blemishes has long been hindered by the limitations of conventional image-processing methods. Only in recent years has technological progress enabled the integration of machine vision with Artificial Intelligence (AI) and Deep Learning (DL) [8]. This development has made it possible to detect surface defects with accuracy and reliability comparable or exceeding those of human inspectors, while significantly reducing costs.

Current AI/DL tools available to system integrators and machine manufacturers are designed for general-purpose use and must accommodate inspection requirements across diverse industries. As a result, they often require careful, scenario-specific training, which hampers widespread adoption of AI-enabled inspection systems by necessitating specialized expertise and case-by-case validation. Moreover, industrial software incurs licensing costs and may not offer the flexibility of a fully modifiable system.

The main objective of this study is to identify, evaluate, and test open source DL models, with particular attention to the comparison between Transformers and Convolutional Neural Networks (CNNs), to verify the possibility of obtaining high-performance results, while offering significant advantages, such as the capability of adapting the model to specific scenarios, intervening in every phase of the pipeline, and having full transparency on the architecture and on the parameters used. Indeed, identifying minuscule, localized, and often ambiguous defects, appearing in unpredictable positions and with highly variable morphologies, requires effectively capturing both local visual patterns and broader spatial context. While convolutional architectures are well-suited for modeling fine-grained local features, Transformer-based models offer a complementary strength, by enabling direct modeling of long-range dependencies across the image. Therefore, rather than assuming the superiority of one paradigm over the other, we adopt a comparative perspective.

It is worth noting that the proposed investigation goes beyond a localized empirical evaluation by addressing a fundamental, open question in contemporary computer vision: how do distinct deep learning architectural paradigms behave when processing ultra-localized, fine-grained, and low-contrast surface anomalies under strict low-data industrial regimes? Fastener defect inspection serves as an ideal proxy for this inquiry. Identifying anomalies like superficial cracks or micro-burrs requires an architecture to balance local spatial micro-features with global structural context.

By benchmarking Convolutional Neural Networks (CNNs) against Vision Transformers (ViTs), this study maps the operational boundaries of structural inductive biases versus unrestricted global self-attention.

In more detail, the study focuses on the identification of failures in fasteners exploiting the control machinery developed by DIMAC S.R.L.[1], an Italian company recognized as a leader in fastener quality-control technologies, and comparison has been structured around two macro-objectives:

- **binary classification** of component images in correct (OK) *vs* defective (NOK);
- **segmentation and defect localization** in defective images, allowing for a detailed analysis of each item and providing further useful information.

The details of the approach and the experiments will be discussed in the next sections, organized as follows. In Section 2 we present related work. Section 3 presents the architectures to be compared. Section 4 describes the available datasets, and the classification and segmentation results. Finally, Section 5 is devoted to conclusions.

2 Related work

For many years, convolutional neural networks (CNNs) have represented the dominant paradigm for computer vision tasks [9–11]. Their effectiveness stems from the ability to capture local spatial patterns through convolutional filters, which operate on restricted receptive fields to detect low-level features such as edges, textures, and shapes. These features are progressively combined across layers, enabling hierarchical representations that encode increasingly abstract semantic information. This inductive bias toward locality and translation invariance allows CNNs to achieve strong performance even with relatively limited training data. However, modeling long-range dependencies remains less direct, often requiring deeper architectures or specific design choices such as dilated convolutions or global pooling mechanisms [12, 13].

More recently, Transformer-based models, originally introduced in natural language processing [14], have been adapted to vision tasks, offering an alternative approach to explicit visual feature extraction [15, 16]. The core component of these models is the self-attention mechanism, which enables each element of the input sequence to directly interact with all others, allowing the model to capture global contextual relationships from early stages. A standard Transformer block consists of a Multi-Head Self-Attention layer, which models pairwise interactions between tokens,

[1] https://www.dimacsrl.com

followed by a position-wise feed-forward network that applies non-linear transformations independently to each token. The Vision Transformer (`ViT`) [17] extends this framework to images by representing them as sequences of fixed-size patches, effectively treating visual data similarly to tokenized text. This formulation removes the strong locality bias of CNNs and allows global relationships between image regions to be modeled directly through attention. As a result, ViTs have demonstrated competitive performance in several large-scale benchmarks, particularly when trained on sufficiently large datasets. However, unlike CNNs, ViTs lack built-in inductive biases such as locality and translation equivariance, making them generally more data-hungry and potentially less efficient in low-data regimes.

From an interpretability perspective, attention mechanisms provide direct access to attention weights, which can offer insights into how information is aggregated across the image. Nevertheless, recent studies have highlighted that attention maps do not always correspond to faithful explanations of model decisions, similarly to post-hoc methods used for CNNs (e.g., gradient-based attribution) [18, 19]. Therefore, both families of models present advantages and limitations in terms of explainability.

Several works have explored hybrid or hierarchical Transformer variants to bridge the gap between CNNs and pure ViTs. Among these, the Swin Transformer [20–22] introduces a hierarchical architecture with shifted local windows, combining the efficiency of localized processing with the ability to model cross-region interactions. This design enables better scalability and has shown strong performance in dense prediction tasks such as semantic segmentation [23, 24].

A further relevant line of research concerns lightweight architectures explicitly designed for industrial inspection, where deployment constraints such as parameter count, inference speed, and memory footprint are as important as raw accuracy. Zhang et al. [25] propose LARD-YOLOv8, a lightweight bearing surface defect detection model built on the YOLOv8n architecture, which combines an efficient detection head with a reparameterized feature extraction module to reduce parameter count and computational load while sustaining real-time inference speed. Tu et al. [26] instead address industrial defect image classification under a few-shot learning regime, employing a memory-efficient DenseNet backbone enhanced with an attention mechanism to achieve a substantially smaller parameter count than comparable architectures such as ResNet12, while requiring only a handful of labeled examples per defect category. These works represent a relevant point of comparison from a deployment standpoint. However, it is worth noting that the models discussed above are designed and evaluated for object detection and few-shot classification tasks respectively, whereas our study focuses on fully supervised binary classification (OK/NOK) and semantic segmentation. These tasks differ in their outputs, loss functions, and evaluation metrics, which makes a direct quantitative comparison both methodologically ambiguous and potentially misleading. We therefore discuss these approaches here to position our work with respect to the broader lightweight-model literature, rather than including them as direct experimental baselines.

Overall, existing literature suggests that CNNs and Transformer-based models exhibit complementary strengths: CNNs provide strong inductive biases and data efficiency, while Transformers offer greater flexibility in modeling global context. In this work, we do not hypothesize the superiority of one paradigm over the other; instead, we systematically compare CNN-based and Transformer-based architectures in the context of fastener classification. Building upon our preliminary study on defective fastener identification [27], we aim to assess under which conditions each family of models is most effective, as detailed in the following section.

3 The approach

In this work, we have adopted representative CNN-based and Transformer-based architectures for both classification and semantic segmentation tasks, with the goal of systematically comparing their performance and behavior. The next subsections will provide details on the selected open source implementations.

3.1 Classification

For the classification task, we have evaluated two widely adopted CNN models, namely `EfficientNet` and `ResNet`, alongside the Vision Transformer (`ViT`).

`ResNet` [11] is characterized by the introduction of residual connections, which enable the training of very deep networks by mitigating the vanishing gradient problem and facilitating feature reuse across layers. `EfficientNet` [28] instead adopts a compound scaling strategy, jointly optimizing depth, width, and resolution to achieve a favorable trade-off between accuracy and computational efficiency.

In contrast, as already mentioned, the Vision Transformer (`ViT`) [17] processes images as sequences of patches and models their relationships through self-attention mechanisms, enabling the capture of global dependencies from early stages of the network.

To ensure a fair comparison and improve data efficiency, all models were initialized using weights pretrained on large-scale datasets (such as ImageNet [9]), and

subsequently fine-tuned on the target dataset [29, 30]. This transfer learning strategy allowed us to leverage previously learned representations while adapting the models to the specific characteristics of the defect detection task. Specifically, we adopted a phased fine-tuning approach where the backbone parameters were initially frozen to stabilize the classification head, followed by unfreezing the final stage (the last residual or mobile inverted bottleneck blocks) to refine high-level feature extraction for defect patterns.

Table 1 summarizes the characteristics of the mentioned architectures, in the specific versions we chose, and provides details on their parameters.

3.2 Segmentation

For the semantic segmentation task, we have compared the widely used `U-Net` architecture with the more recent `Mask2Former` architecture. The choice of `U-Net` is motivated by its strong performance in industrial segmentation [31, 32], while `Mask2Former` represents a state-of-the-art Transformer-based alternative.

The original `U-Net` [33] is a convolutional neural network originally designed for biomedical image segmentation. Its encoder–decoder structure, combined with skip connections, allows the model to capture both contextual information and fine spatial details, making it particularly effective in scenarios with limited training data and small structures.

On the other hand, `Mask2Former` [34] is a Transformer-based architecture that formulates segmentation as a mask classification problem. In our implementation, `Mask2Former` employs a Swin Transformer backbone, which enables hierarchical feature extraction through shifted window-based self-attention. It leverages attention mechanisms to model global relationships across the image and employs a unified framework capable of handling semantic, instance, and extensive (e.g., panoptic) segmentation tasks. This design enables flexible and context-aware predictions, potentially improving performance in complex scenes with ambiguous or heterogeneous regions.

Similarly to the classification setting, both models [35, 36] were initialized with pre-trained weights (on the ADE20K dataset [37]), and fine-tuned on the target dataset with custom labels corresponding to background and defect. For these architectures, the decoder and task-specific predictors remained fully trainable throughout the process, while the encoder backbone was selectively unfrozen in its deepest layers to preserve spatial hierarchies while optimizing semantic boundary detection for the mask predictions.

Table 2 summarizes the characteristics of the chosen architectures, and provides details on their parameters.

Table 1 Comparison of classification architectures used in this study

Model	Type	Parameters (M)	Input Size
`ResNet-50`	CNN	∼23.5M	224×224
`EfficientNet-B0`	CNN	∼4M	224×224
`ViT-L/16`	Transformer	∼303.3M	224×224

3.3 Training recipe

To ensure reproducibility, this subsection details the optimization protocol, loss functions, fine-tuning strategy, data augmentation pipeline, and hardware configuration adopted for all five architectures. The single 70/15/15 split and the 10-fold cross-validation protocol share identical hyperparameters, except where explicitly noted below.

Optimizer and learning rate schedule All classification models (`ResNet-50`, `EfficientNet-B0`, `ViT-L/16`) were trained with the AdamW optimizer, an initial learning rate of 5×10^{-4}, and a cosine annealing schedule with linear warm-up over the first 10% of training steps, updated at every optimization step. A weight decay of 1×10^{-4} was applied to `ResNet-50` and `EfficientNet-B0`; `ViT-L/16` used the standard AdamW weight decay of 0.01. `Mask2Former` was likewise trained with AdamW, a learning rate of 5×10^{-4}, and a weight decay of 1×10^{-4}, while `U-Net` was trained with the Adam optimizer at the same learning rate. Both segmentation models used a plateau-based schedule (`ReduceLROnPlateau`, reduction factor of 0.5 after 3 epochs without improvement in the validation loss), rather than the cosine schedule adopted for the classification models. All models were trained for a maximum of 50 epochs, with a batch size of 32.

Loss functions and class imbalance handling For the classification task, a weighted cross-entropy loss was used, with class weights computed independently for each split (or for each fold, in the cross-validation setting) using a balanced class-weighting scheme based on the inverse class frequency of the corresponding training partition only. For segmentation, `U-Net` was trained with a combined loss consisting of the Binary Cross-Entropy loss and the Dice loss in equal proportion (0.5 each), while `Mask2Former` used its native mask classification loss, combining classification, mask, and Dice terms as defined in the original architecture.

Partial fine-tuning strategy As anticipated in Section 3, all models were fine-tuned starting from pre-trained weights,

Table 2 Comparison of segmentation architectures used in this study

Model	Type	Parameters (M)
`U-Net`	CNN	∼31M
`Mask2Former`	Transformer	∼216M

with a partial unfreezing strategy applied to reduce training time and mitigate overfitting on the relatively small target datasets. For `ResNet-50`, the final classification layer together with the last two residual stages (`layer3` and `layer4`) were unfrozen. For `EfficientNet-B0`, the classification head and the last two mobile inverted bottleneck blocks were unfrozen. For `ViT-L/16`, the classification head, all layer normalization parameters, and the last 6 of the 24 encoder transformer blocks were unfrozen. For both segmentation architectures, the decoder and task-specific predictors remained fully trainable, while the encoder backbone was selectively unfrozen in its deepest layers, consistently with the strategy already described in Section 3.

Data augmentation Two complementary augmentation strategies were adopted. First, an online augmentation pipeline was applied to every training image at each epoch, consisting of a random resized crop to 224×224 pixels followed by a random horizontal flip; validation and test images were instead resized to 256 pixels on the shorter side and center cropped to 224×224, with both splits normalized using the standard ImageNet channel statistics. Second, as already described in Section 4, the NOK class was further augmented offline through horizontal flipping, vertical flipping, and random angular rotation, in order to counteract the class imbalance highlighted in Table 3. Crucially, for the 10-fold cross-validation protocol, this offline augmentation was generated exclusively within the training portion of each fold, and only after the fold split had already been determined. This ensures a strict separation between training and validation data, ruling out any risk of data leakage or near-duplicate contamination of the validation set, since no synthetic variant of a validation image can appear in the corresponding training partition, or vice versa. The same principle was applied to the single 70/15/15 split, where synthetic images were included exclusively in the training portion and were never generated from, nor included in, the validation or test partitions.

Cross-validation, data splitting, and random seeds The 10-fold cross-validation protocol used a stratified splitting strategy for the classification task, in order to preserve the OK/NOK class ratio across folds, and a standard, non-stratified splitting strategy for the segmentation task, since only defective images are involved in that setting. In both cases, the fold partitioning was performed with a fixed random seed (42), ensuring that the same partitions can be reproduced across independent runs. Early stopping was applied uniformly across all models and both evaluation protocols, monitoring the validation loss with a patience of 15 epochs, except for the *cups* dataset in the segmentation task, for which a patience of 5 epochs was adopted after preliminary experiments showed earlier convergence and a tendency to overfit beyond that point.

Numerical precision and hardware All models were trained using automatic mixed precision, with gradient norm clipping at a maximum value of 1.0 to stabilize optimization. Training was carried out on the same computing infrastructure described in Section 4.3 (NVIDIA L40 GPU, CUDA 11.7, PyTorch 1.13.1).

4 Experimental results

In this section, we discuss our experimental results, detailing data preparation, training and evaluation, both for the classification and the segmentation task on three real-world datasets of different kinds of fasteners.

4.1 Classification experiments

4.1.1 Datasets

We have collected three distinct datasets, each one related to a different mechanical part, namely: *gears*, *cups* and *screw heads*. All the images were acquired under real operating conditions, using the existing industrial infrastructure of Dimac S.r.l.. The control machinery involved is equipped with automatic vision systems with different types of cameras: top-view, which are cameras positioned vertically with respect to the plane of transit of the parts, capable of capturing high-resolution images from above (useful for screw heads); side cameras, positioned laterally with respect to the transit of the parts (useful for gears and cups). Furthermore, the machine for acquiring images of screw heads was configured with a metallic disk (as shown in Fig. 1 on the left side) where the screws were positioned in the appropriate seat; in the case of gears and cups, instead, the machine was equipped with a glass disk (as shown in Fig. 1 on the right side) to facilitate focus on the part.

Table 3 Number of OK and NOK images collected for each dataset, prior to augmentation, and corresponding class ratio.

Dataset	OK	NOK	Total	OK (%)	NOK (%)	OK:NOK
Cups	1327	576	1903	69.7	30.3	2.30:1
Gears	941	793	1734	54.3	45.7	1.19:1
Screw heads	1474	621	2095	70.4	29.6	2.37:1
Total	**3742**	**1990**	**5732**	**65.3**	**34.7**	**1.88:1**

Fig. 1 Example configurations of machines with metallic disk (left) and glass disk (right)

(a) Cup (b) Gear (c) Screw head

Fig. 2 Example input images from the three datasets used in this study

All images are captured under frontal illumination, in some cases combined with backlight illumination used to suppress background clutter and increase the contrast between the component and its surroundings. For each application, the illumination conditions, the optics, and the camera positioning are empirically selected by the industrial partner to best highlight the defects relevant to that specific component. Once this operating configuration has been established for a given application, training images are acquired under these fixed conditions and are not subsequently modified.

Figure 2 shows an example input image for each of the three datasets considered in this study.

To prepare the binary classification dataset, we split the images into two distinct folders, OK and NOK, corresponding respectively to the positive samples (labeled as conforming to specification) and negative samples (labeled as defective).

Table 3 reports the number of OK and NOK images collected for each dataset prior to augmentation, together with the corresponding class ratio. The three datasets differ noticeably in their degree of class imbalance. The *gears* dataset is the most balanced, with an OK to NOK ratio of approximately 1.2 to 1, whereas *cups* and *screw heads* exhibit a comparatively higher imbalance, at approximately 2.3 to 1 and 2.4 to 1 respectively. Across all three datasets combined, OK images account for 65.3% of the 5,732 collected samples, confirming that defective instances consistently represent the minority class, which motivates the augmentation strategy for the NOK class described below.

Since NOK cases were underrepresented, a data augmentation strategy was implemented; in particular, for each image three synthetic versions were produced through horizontal flipping, vertical flipping and random angular rotation. Synthetic images were included exclusively in the training set, accounting for 70% of the total data, while being intentionally excluded from the validation (15%) and test (15%) sets. To ensure the robustness of the experimental evaluation, we conducted experiments using both a standard 70/15/15 data split and a stratified 10-fold cross-validation scheme. In fact, while the single split provides an immediate snapshot of the model behavior on one partitioning of the data, cross-validation offers a more robust estimate of generalization performance and of the variability across different folds. For all datasets and models, early stopping on the validation loss was employed.

4.1.2 Classification results

Table 4 summarizes the results obtained with a single 70/15/15 train/validation/test split, considering only the test partition for the final evaluation; columns ROC and PR

Table 4 Classification results — single 70/15/15 train/validation/test split (test set only). No confidence interval is provided for single-split evaluation. Best values per metric within each dataset are highlighted in bold

Dataset	Model	Arch.	Accuracy	ROC	PR	F1(NOK)	F1(OK)	Macro F1
			[%]	[%]	[%]	[%]	[%]	[%]
Cups	ResNet-50	*CNN*	**86.0**	**90.1**	**84.0**	**78.0**	**90.0**	**84.0**
	EfficientNet-B0	*CNN*	66.0	76.6	54.9	59.0	70.0	65.0
	ViT_L/16	*Transformer*	65.2	82.4	67.7	61.8	68.0	64.9
Gears	ResNet-50	*CNN*	95.0	**98.9**	**99.0**	94.0	**96.0**	95.0
	EfficientNet-B0	*CNN*	84.0	93.3	92.1	82.0	86.0	84.0
	ViT_L/16	*Transformer*	**95.4**	98.7	98.3	**94.9**	95.8	**95.4**
Screw heads	ResNet-50	*CNN*	95.0	98.6	97.8	91.0	96.0	94.0
	EfficientNet-B0	*CNN*	93.0	98.4	97.2	90.0	95.0	92.0
	ViT_L/16	*Transformer*	**95.3**	**99.2**	**98.4**	**91.8**	**96.7**	**94.2**

Table 5 Classification results — 10-fold stratified cross-validation. Values are reported as mean ± 95% confidence interval. Best values per metric within each dataset are highlighted in bold. NOK = defective class; OK = conforming class

Dataset	Model	Accuracy	ROC	PR	F1(NOK)	F1(OK)
		[%]	[%]	[%]	[%]	[%]
Cups	ResNet-50	**87.1 ± 1.7**	**92.5 ± 1.7**	**87.9 ± 2.6**	**78.4 ± 3.0**	**90.8 ± 1.2**
	EfficientNet-B0	76.2 ± 2.6	82.1 ± 1.9	68.6 ± 2.7	64.4 ± 3.1	82.1 ± 2.3
	ViT_L/16	83.3 ± 1.5	89.0 ± 1.2	83.1 ± 2.2	71.4 ± 4.9	88.1 ± 1.0
Gears	ResNet-50	94.5 ± 1.4	99.1 ± 0.2	99.0 ± 0.3	94.0 ± 1.6	94.9 ± 1.3
	EfficientNet-B0	92.1 ± 1.4	97.8 ± 0.5	97.5 ± 0.6	91.3 ± 1.5	92.8 ± 1.3
	ViT_L/16	**95.3 ± 1.2**	**99.2 ± 0.3**	**99.1 ± 0.3**	**94.9 ± 1.3**	**95.6 ± 1.2**
Screw heads	ResNet-50	**95.5 ± 1.0**	98.7 ± 0.5	97.9 ± 0.7	**92.6 ± 1.6**	**96.8 ± 0.7**
	EfficientNet-B0	94.6 ± 1.2	98.6 ± 0.7	97.7 ± 0.9	91.1 ± 2.0	96.1 ± 0.9
	ViT_L/16	94.8 ± 1.2	**98.8 ± 0.6**	**97.9 ± 0.9**	91.6 ± 1.8	96.3 ± 0.9

represent the AUC measures of the ROC curve and of the Precision/Recall curve respectively.

In addition, Table 5 reports the results concerning the same performance measures, obtained with 10-fold stratified cross-validation, expressed as mean performance together with the corresponding 95% confidence interval.

Overall, the results show that classification performance strongly depends on the dataset. The *gears* and *screw heads* datasets appear to be more amenable to binary discrimination, whereas *cups* emerges as the most challenging scenario, with the lowest performance scores. This trend is consistent across both evaluation protocols and across all considered architectures.

Concerning the single train/validation/test split (Table 4), the best performance on *cups* is achieved by ResNet-50, which reaches an accuracy of 86.0%, a ROC-AUC of 90.1% , a PR-AUC of 84.0%, and a macro F1-score of 84.0%. On this dataset, both EfficientNet-B0 and ViT_L/16 exhibit a marked performance drop, showing that the cup classification problem is substantially more difficult. In particular, the lower F1 values suggest that the visual differences between defective and non-defective parts are subtle and difficult to capture reliably by a global image-level classifier.

On *gears*, all models achieve strong performance, with ViT_L/16 and ResNet-50 showing complementary strengths across metrics. ViT_L/16 attains the best accuracy (**95.4%**), F1-score on the NOK class (**94.9%**), and macro F1-score (**95.4%**), indicating a more balanced performance across classes and a slight advantage in overall classification. In contrast, ResNet-50 achieves the highest ROC-AUC (**98.9%**), PR-AUC (**99.0%**), and F1-score on the OK class (**96.0%**), suggesting a stronger separability of the classes and particularly robust performance on the majority/OK class. Overall, while the Transformer-based model provides a more balanced classification, the CNN-based model remains highly competitive and even superior in terms of ranking-based metrics.

The *screw heads* dataset yields the strongest single-split performance overall. ViT_L/16 achieves the best values for every considered metric, including 95.3% accuracy, 99.2% ROC-AUC, and 98.4% PR-AUC, while maintaining balanced F1-scores for both defective and conforming classes. The two CNN baselines also perform very well, with only limited differences from the Transformer. This confirms that screw head defects are generally easier to distinguish at image level than those characterizing the cup dataset.

The results obtained through 10-fold stratified cross-validation (Table 5) largely confirm the trends observed in the single-split evaluation, while also providing a more stable estimate of model behavior. On *cups*, ResNet-50

again proves to be the best-performing model, with an accuracy of $87.1 \pm 1.7\%$, ROC-AUC of $92.5 \pm 1.7\%$, PR-AUC of $87.9 \pm 2.6\%$, F1-score on the defective class of $78.4 \pm 3.0\%$, and F1-score on the conforming class of $90.8 \pm 1.2\%$. The gap with `EfficientNet-B0` remains substantial, while `ViT_L/16` improves over its single-split result but still does not surpass `ResNet-50`. This further supports the interpretation that, in the cup dataset, a convolutional representation is more suitable for capturing the subtle and localized appearance differences associated with defects.

For *gears*, cross-validation confirms the strong performance of all models and identifies `ViT_L/16` as the best overall architecture. It achieves $95.3 \pm 1.2\%$ accuracy, $99.2 \pm 0.3\%$ ROC-AUC, $99.1 \pm 0.3\%$ PR-AUC, $94.9 \pm 1.3\%$ F1-score for NOK, and $95.6 \pm 1.2\%$ F1-score for OK. The narrow confidence intervals indicate stable behavior across folds and suggest that the model generalizes reliably on this dataset.

On *screw heads*, the results are again very good for all three architectures. `ResNet-50` yields the best accuracy ($95.5 \pm 1.0\%$) and the highest F1-scores for both NOK and OK, whereas `ViT_L/16` achieves the best ROC-AUC ($98.8 \pm 0.6\%$) and shares the best PR-AUC value ($97.9 \pm 0.9\%$). The difference between `ResNet-50` and `ViT_L/16` is therefore limited and metric-dependent, suggesting that both architectures are highly effective for screw head classification. `EfficientNet-B0` is only slightly behind, which further indicates that this dataset is comparatively less challenging than *cups*.

From an application perspective, the most critical result represents the one observed on *cups*. Although acceptable global scores can still be obtained, the lower F1-score on the defective class indicates that classification at image level may be insufficient when anomalies are very small, localized, or weakly contrasted with respect to the surrounding surface. This motivates the transition to a segmentation-based strategy, discussed in the next section, where the objective is not only to assign a class label but also to localize the defect region explicitly.

4.2 Segmentation experiments

4.2.1 Datasets

The three augmented real-world datasets used for classification have also been used for evaluating segmentation, with the aim of localizing the defect in the considered fastener components. Datasets have been manually annotated through an asymmetrical strategy, with the focus of highlighting the specific region of the image where the defect was present, explicitly omitting the annotation of the background and the correct area of the component image.

Therefore, the background has been implicitly considered as a residual class, deducible from the absence of annotations, according to a common convention in segmentation tasks for rare or sporadic objects. Since the goal was to visually identify and locate the defects, only images belonging to the NOK class, i.e., actually containing visual anomalies, were considered. The bounding polygons in the annotation file have been expressed as a percentage relative to the image width and height. These values have been converted into absolute pixel coordinates before drawing the polygon on the mask.

The final result of data processing is a binary mask for each image, where the pixels belonging to the defective areas are set to 255 (white), while the rest of the image to 0 (black). An example is shown in Fig. 3.

Defect size characterization To quantify the morphological variability of defects across datasets, we performed a systematic analysis of their spatial properties using the binary segmentation masks. For each connected component in a mask, two metrics were computed: the *major axis*, defined as the longest side of the tight bounding box (in pixels), which captures the actual extent of a defect regardless of

Original image with defect (NOK). Obtained mask.

Fig. 3 Gear with its generated binary mask

its shape (particularly meaningful for elongated anomalies such as scratches); and the *defect coverage*, expressed as the percentage of the total image area occupied by the component. Components smaller than 10 pixels in area were discarded as annotation noise; in particular, 284 such single-pixel artifacts were removed from the *screw heads* dataset, where they appeared alongside genuine defects of hundreds to thousands of pixels.

Table 6 summarizes the key statistics per dataset. The *gears* dataset exhibits low size variability (coefficient of variation CV = 36% on major axis), with all defects falling within a narrow range of 26–148 px and covering at most 1.14% of the image. By contrast, the *cups* and *screw heads* datasets show substantially higher variability: the major axis spans 9–257 px for *cups* (CV = 73%) and 30–413 px for *screw heads* (CV = 48%). The mean aspect ratio (AR = major / minor bounding-box side, always ≥ 1) reveals that *cups* defects are predominantly elongated (mean AR = 2.5, max = 12.9), consistent with the presence of linear scratches, whereas *gears* and *screw heads* defects tend to be more compact (mean AR ≤ 1.8). It is important to note that, despite these morphological differences, all defects are intentionally treated as a single, uniform class in our framework. In a real industrial quality-control setting, the same imaging machine and the same algorithmic pipeline must handle every type of anomaly in the same way, without prior knowledge of its shape or extent; the goal is therefore to build a model that generalizes across the full spectrum of defect appearances rather than specializing on a particular morphology.

Given the high morphological variability observed in *cups* and *screw heads*, defects were further stratified into three size-based subsets (Small: $\leq$P33, Medium: P33–P66, Large: >P66) according to their major axis. Figure 4 illustrates the resulting distributions for these two datasets via log-scale histograms and box plots. Both datasets contain a broad spectrum of defect sizes spanning approximately one order of magnitude, which poses additional challenges for segmentation models that must generalize across heterogeneous anomaly scales.

Experiments were first performed using a standard 70/15/15 data split; subsequently, a stratified cross-validation scheme was adopted to ensure more reliable and statistically robust evaluation results. Early stopping on the validation loss was adopted in order to reduce overfitting.

4.2.2 Segmentation results

Table 7 summarizes the performance obtained with a single 70/15/15 train/validation/test split, considering the defect class only. In this setting, results are expressed in terms of `IoU`, `Dice`, precision, and recall. Table 8, instead, reports the results obtained with 10-fold stratified cross-validation, expressed as mean performance with the corresponding 95% confidence interval.

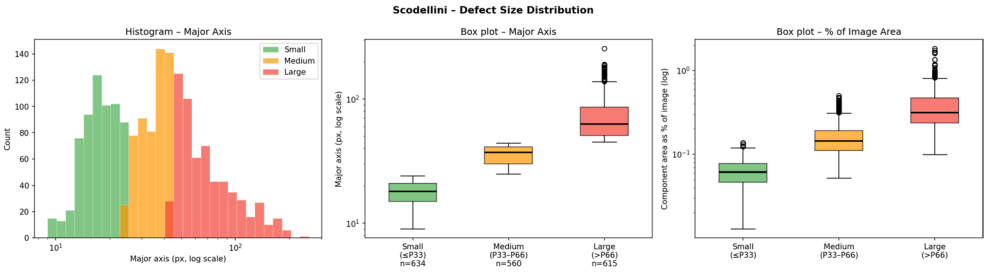
Overall, the segmentation results confirm that localizing defects at pixel level is a more informative and appropriate strategy than global image classification, especially in scenarios where anomalies are small, sparse, or weakly contrasted with the surrounding material. Across the considered datasets, both models achieve very high performance on background segmentation, while the real challenge lies in correctly identifying defect regions. In this regard, `U-Net` generally emerges as the most robust architecture, although `Mask2Former` remains competitive on some datasets and, in specific cases, attains higher recall.

Concerning the single train/validation/test split (Table 7), the *cups* dataset emerges to be the most challenging one. `U-Net` achieves the best `IoU` (0.425), `Dice` (0.590), and recall (0.510), whereas `Mask2Former` yields the highest precision (0.680). This suggests that `U-Net` provides a more balanced segmentation of cup defects, while `Mask2Former` tends to produce more conservative predictions, identifying fewer defective pixels but with a lower proportion of false positives. The result is consistent with the difficulty already observed in the classification setting: defects in cups are highly localized and often subtle, which makes dense prediction particularly challenging.

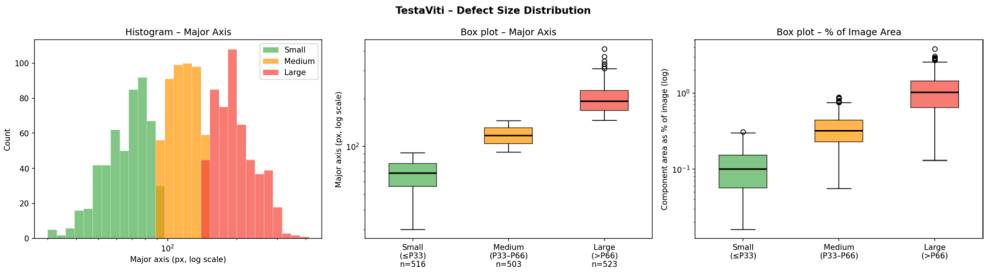
On *gears*, both architectures yield markedly stronger results. `U-Net` and `Mask2Former` obtain the same `IoU` (0.700) and `Dice` score (0.820), indicating a very similar overall overlap with the ground truth masks. However, the two models differ in their error profile: `U-Net` achieves higher precision (0.830), while `Mask2Former` reaches the highest recall (0.900). This means that the Transformer-based model is more aggressive in detecting defect pixels, reducing missed defective regions at the cost of a slightly

Table 6 Defect size statistics per dataset (per connected component; annotation noise < 10 px filtered). Major axis = longest bounding-box side (px); AR = aspect ratio (major / minor, ≥ 1); % img = component area as a fraction of total image area. CV = coefficient of variation of the major axis

Dataset	Defective	Comp.	Major axis (px)			AR	% of image		**CV**
	images		Min	Mean	Max	Mean	Mean	Max	
Cups	1091	1809	9	43	257	2.5	0.21	1.84	73%
Gears	258	336	26	53	148	1.8	0.43	1.14	36%
Screw heads	620	1542	30	129	413	1.6	0.52	3.84	48%



(a) Cups



(b) Screw heads

Fig. 4 Defect size distribution for the two high-variability datasets, split into Small (≤P33), Medium (P33–P66), and Large (>P66) subsets by major axis. Left: log-scale histogram; center: box plot of major axis; right: box plot of defect coverage (% of image area)

Table 7 Segmentation results – single 70/15/15 train/validation/test split (test set only, defect class). Best per metric within each dataset is highlighted in bold

Dataset	Model	IoU	Dice	Precision	Recall
Cups	`U-Net`	**0.425**	**0.590**	0.580	**0.510**
	`Mask2Former`	0.400	0.550	**0.680**	0.460
Gears	`U-Net`	0.700	0.820	**0.830**	0.825
	`Mask2Former`	0.700	0.820	0.760	**0.900**
Screw heads	`U-Net`	**0.600**	**0.750**	0.750	**0.760**
	`Mask2Former`	0.570	0.720	0.750	0.700

larger number of false positives. From an application perspective, such a behavior may be particularly relevant in industrial inspection scenarios where missing a defect is more critical than over-segmenting it slightly.

The *screw heads* dataset also shows good segmentation performance, with `U-Net` outperforming `Mask2Former` on most of the considered metrics. In particular, `U-Net` reaches an `IoU` of 0.600, a `Dice` score of 0.750, and a recall of 0.760, whereas `Mask2Former` obtains slightly lower values for `IoU` and `Dice` while matching `U-Net` in precision (0.750). This indicates that both models are able to localize screw defects reasonably well, but `U-Net` provides more complete masks and more stable overlap with the ground truth.

The 10-fold stratified cross-validation results reported in Table 8 confirm the trends observed in the single-split evaluation and provide a more robust estimate of generalization. As expected, background metrics are almost perfect

Table 8 Segmentation results – 10-fold stratified cross-validation, defect class only. Values reported as mean ± 95% confidence interval. Best value per metric and dataset is highlighted in bold

Dataset	Model	Precision [%]	Recall [%]	F1 / Dice [%]	IoU [%]
Cups	`U-Net`	**68.6 ± 1.9**	**46.2 ± 2.5**	**55.1 ± 1.8**	**38.1 ± 1.7**
	`Mask2Former`	57.8 ± 4.4	41.6 ± 3.7	48.2 ± 2.9	31.8 ± 2.5
Gears	`U-Net`	**85.9 ± 2.5**	**76.9 ± 3.5**	**81.0 ± 2.1**	**68.2 ± 2.8**
	`Mask2Former`	84.9 ± 2.6	70.5 ± 4.2	76.9 ± 3.2	62.7 ± 4.1
Screw heads	`U-Net`	**75.0 ± 2.7**	**73.1 ± 2.5**	**73.9 ± 1.3**	**58.7 ± 1.6**
	`Mask2Former`	75.0 ± 2.9	70.7 ± 2.5	72.6 ± 1.3	57.1 ± 1.6

for all datasets and both architectures, owing to the severe class imbalance. Therefore, the most informative part of the analysis concerns the defect rows.

For *cups*, `U-Net` clearly outperforms `Mask2Former` on all defect-related metrics, obtaining $68.6 \pm 1.9\%$ precision, $46.2 \pm 2.5\%$ recall, $55.1 \pm 1.8\%$ `Dice`, and $38.1 \pm 1.7\%$ `IoU`. The corresponding values for `Mask2Former` are consistently lower, namely $57.8 \pm 4.4\%$ precision, $41.6 \pm 3.7\%$ recall, $48.2 \pm 2.9\%$ `Dice`, and $31.8 \pm 2.5\%$ `IoU`. These results indicate that the cup dataset represents the most difficult also in the segmentation

setting, but they also show that pixel-wise prediction is still more informative than classification alone, since the model can often identify at least part of the defective region even when global discrimination is difficult.

On *gears*, cross-validation again reveals strong performance for both architectures, with `U-Net` maintaining a consistent advantage. In particular, it achieves $85.9 \pm 2.5\%$ precision, $76.9 \pm 3.5\%$ recall, $81.0 \pm 2.1\%$ `Dice`, and $68.2 \pm 2.8\%$ `IoU` on the defect class, compared with $84.9 \pm 2.6\%$, $70.5 \pm 4.2\%$, $76.9 \pm 3.2\%$, and $62.7 \pm 4.1\%$, respectively, for `Mask2Former`. Although the gap is not dramatic, it is systematic and suggests that the convolutional inductive bias of `U-Net` is particularly well suited to this kind of localized surface anomaly.

For *screw heads*, the same pattern is observed. `U-Net` reaches the best defect-class scores, with $75.0 \pm 2.7\%$ precision, $73.1 \pm 2.5\%$ recall, $73.9 \pm 1.3\%$ `Dice`, and $58.7 \pm 1.6\%$ `IoU`. `Mask2Former` remains close in terms of precision ($75.0 \pm 2.9\%$), but its recall, `Dice`, and `IoU` are slightly lower. This indicates that both models are effective on screw head defects, but `U-Net` provides a more favorable balance between correct localization and spatial overlap with the annotated masks.

4.3 Inference efficiency

Beyond predictive accuracy, a critical dimension of industrial deployability is inference efficiency. To quantify the computational overhead of each evaluated architecture, we measured per-image inference latency (mean and standard deviation), throughput, and peak GPU memory consumption under realistic single-item inspection conditions (batch size of 1). All measurements were conducted using CUDA event timing, with 50 warm-up iterations followed by 200 measured passes.

Evaluation hardware. All benchmarks were executed on a server equipped with an NVIDIA L40 GPU (47.7 GB GDDR6, 142 streaming multiprocessors, 864 GB/s memory bandwidth), a 24-core CPU, 135 GB of RAM, running CUDA 11.7 and PyTorch 1.13.1. Input images were fed at their native inspection resolution, padded to the nearest multiple of 32 where required by encoder-decoder architectures: 544×736 pixels for *cups* and *gears*, and 1088× 1440 pixels for *screw heads*. Classification models always receive 224×224 images after resizing.

The results, summarized in Table 9, confirm a substantial efficiency advantage for convolutional architectures. `ResNet-50` and `EfficientNet-B0` achieve sub-4 ms latency per image, sustaining over 250 and 280 images per second respectively, with a negligible memory footprint (235 MB and 154 MB). `ViT-L/16`, despite its considerably larger parameter count, remains within 8.5 ms per image, a throughput level compatible with the requirements of many industrial inspection lines. On the segmentation side, `U-Net` processes *cups* and *gears* images at 106 images per second; the screw-head variant, which handles images four times larger in pixel area, still reaches 38 images per second. `Mask2Former` is considerably slower, particularly at high resolution: on the *screw heads* dataset it requires 288 ms per image (approximately 3.5 images per second), a consequence of its architectural complexity and the quadratic attention cost associated with large feature maps.

Table 9 Inference efficiency for all model and dataset combinations. Latency is reported as mean ± standard deviation over 200 measured passes at batch size 1 on an NVIDIA L40 GPU. Classification models always receive 224×224 inputs; segmentation models use the native inspection resolution padded to the nearest multiple of 32

Model	Dataset	Latency (ms/img)	Tput. (img/s)	VRAM (MB)
`ResNet-50`	Cups	3.87 ± 0.06	258.3	234.9
	Gears	3.88 ± 0.05	257.9	234.9
	Screw heads	3.88 ± 0.06	258.0	234.9
`EfficientNet-B0`	Cups	3.58 ± 0.04	279.6	154.0
	Gears	3.49 ± 0.08	286.4	154.0
	Screw heads	3.41 ± 0.06	292.9	154.0
`ViT-L/16`	Cups	8.56 ± 0.16	116.8	1,226.3
	Gears	8.41 ± 0.16	119.0	1,226.3
	Screw heads	8.46 ± 0.13	118.2	1,226.3
`U-Net`	Cups	9.45 ± 0.05	105.8	257.4
	Gears	9.41 ± 0.07	106.3	257.4
	Screw heads	26.46 ± 0.47	37.8	718.6
`Mask2Former`	Cups	71.80 ± 0.84	13.9	1,330.7
	Gears	77.22 ± 16.25	12.9	1,330.7
	Screw heads	288.25 ± 3.58	3.5	2,668.4

Applicability to the target deployment hardware The industrial partner Dimac S.r.l. equips their quality-control workstations with the following configuration: Intel Core i7-12700 processor (8 performance cores at 2.1 GHz, 4 efficiency cores at 1.6 GHz), 32 GB DDR5-4800 RAM, 512 GB SSD, and an NVIDIA RTX 5070 GPU with 12 GB GDDR7 memory [38]. Although direct testing on Dimac's production machines was not possible, a conservative latency estimate can be derived from the published hardware specifications of the two GPUs. The NVIDIA L40 delivers approximately 90.5 TFLOPS of FP32 throughput at a memory bandwidth of 864 GB/s [39], whereas the RTX 5070 provides approximately 30.9 TFLOPS at 672 GB/s [38]. These figures yield a compute throughput ratio of approximately 2.9× and a memory bandwidth ratio of approximately 1.3×. The actual latency slowdown

depends on the arithmetic intensity of each model: at batch size 1, CNN architectures (`ResNet-50`, `EfficientNet-B0`, `U-Net`) exhibit low arithmetic intensity and operate in the memory-bandwidth-bound regime, so the expected slowdown is closer to the bandwidth ratio (approximately 1.3–1.5×). Transformer-based models (`ViT-L/16`, `Mask2Former`), which rely on large matrix multiplications with higher arithmetic intensity, are expected to be more compute-bound and therefore closer to the throughput ratio (approximately 2.5–3×). As a conservative upper bound, we estimate the latency on the RTX 5070 to be at most 2 to 3 times higher than the values reported in Table 9 for the most demanding architectures. Under this projection, CNN-based classifiers would still execute in under 10–12 ms per image, well within the real-time budget of typical inspection lines. `Mask2Former` on the highest-resolution dataset could approach or exceed 500 ms per image, which may require targeted optimization (e.g., input resolution reduction or model quantization) for time-critical deployments. Importantly, the highest peak VRAM observed across all benchmarks is 2,668 MB (`Mask2Former` at screw-head resolution), which fits comfortably within the 12 GB capacity of the RTX 5070, confirming that all evaluated models are deployable on the target hardware without memory-related constraints.

4.4 Defect tolerance and deployment considerations

Operating point and defect tolerance In the deployment setting provided by the industrial partner Dimac S.r.l., the classification and segmentation pipelines expose a set of adjustable parameters that determine the operating threshold used to separate OK from NOK items. The choice of these parameters depends on the sensitivity required by the specific application and on production constraints, and is ultimately set by the machine operator. A hard requirement of the deployment is that false negatives, namely defective items misclassified as OK, must be minimized as strongly as possible, since a defective part reaching the client is not an acceptable outcome. To satisfy this requirement, the operating thresholds are deliberately configured in a restrictive manner, which in turn increases the false positive rate, namely conforming items misclassified as NOK and consequently discarded. This trade-off is explicitly accepted by Dimac's clients, since an elevated scrap rate, although economically undesirable, is preferable to shipping defective components. No quantitative scrap rate figures were made available by the industrial partner for the datasets considered in this study, so this discussion should be interpreted as a qualitative account of the operating philosophy underlying threshold selection, rather than as a quantitative bound on acceptable error rates.

Robustness to acquisition condition variability As described in Section 4, all training and inspection images are acquired under a combination of frontal and, in some configurations, backlight illumination, with illumination, optics, and camera positioning empirically tuned for each application and kept fixed once the operating configuration has been established. According to the industrial partner, the deployed models have shown a limited robustness to variations in illumination or camera positioning with respect to the conditions used during training. Since this aspect has not been systematically investigated, either by the industrial partner or in the present study, it remains unclear whether this sensitivity stems from an intrinsic limitation of the evaluated architectures or from a training protocol that was not specifically optimized for robustness to acquisition variability. We consider a systematic assessment of this aspect, for instance through targeted data augmentation or domain randomization strategies applied to illumination and camera pose, a relevant direction for future work.

5 Conclusions

Commercial tools for defect detection in fastener manufacturing, although high-performing and widely adopted in industrial environments, typically require significant economic investment and are often tightly integrated into proprietary ecosystems, thus limiting flexibility, customization, and experimentation.

In this work, we have shown that open source architectures can provide relevant and significant performance both in defect classification and in defect localization, offering a viable and flexible alternative to commercial solutions. In particular, we have systematically compared convolutional neural networks and Transformer-based models across multiple datasets and evaluation protocols.

While previous literature has widely explored this architectural dichotomy on massive general-purpose benchmarks (e.g., ImageNet, COCO), there remains a critical gap in understanding how these paradigms scale down to fine-grained industrial surface metrology. Dense manufacturing defects present unique challenges: they are often non-linearly distributed, heavily disguised by metallic glare, and occupy a tiny fraction of the total image pixels.

Therefore, this work establishes a rigorous comparative framework. By assessing both global image classification and localized dense pixel-wise masks across single-split and 10-fold cross-validation protocols, we isolate the exact conditions under which a modeler should favor the strict,

computationally lightweight spatial constraints of a CNN over the flexible, data-hungry global representation capacity of an industrial Transformer.

The results highlight that classification performance is strongly dataset-dependent. While high accuracy can be achieved on datasets such as gears and screw heads, classification becomes significantly more challenging when defects are small, subtle, or weakly contrasted, as in the case of cups. In such scenarios, global image-level classification may not be sufficient to capture the relevant visual cues.

Segmentation experiments further confirm this observation. Taken together, the results lead to three main conclusions:

- the segmentation task is strongly dataset-dependent, with *cups* emerging the most challenging benchmark and *gears* the one on which defect localization is most reliable,
- `U-Net` emerges as the most robust architecture across all datasets, consistently achieving the best or tied-best performance in the most relevant metrics,
- `Mask2Former` remains competitive and, in some cases, achieves higher recall, which may be desirable in applications where missing a defect is particularly critical.

Most importantly, segmentation proves to be a more appropriate framework than classification when dealing with small and spatially localized defects. This is particularly evident for the *cups* dataset, where classification performance is limited, while segmentation still allows the models to identify defective regions with a non-negligible degree of accuracy. Beyond improving interpretability, segmentation provides a more suitable paradigm for industrial defect analysis, as it not only determines whether a component is defective, but also explicitly indicates *where* the anomaly is located.

From an architectural perspective, our results suggest that the adoption of Transformer-based models does not necessarily guarantee superior performance in this context. While Vision Transformers and `Mask2Former` demonstrate strong capabilities, especially in capturing global relationships, convolutional architectures such as `ResNet` and `U-Net` remain highly competitive and, in many cases, outperform their Transformer counterparts. This is particularly relevant when considering computational efficiency: CNN-based models require significantly fewer parameters (see Tables 1 and 2) and are generally easier to train, making them more suitable for real-world industrial deployment where resources and latency constraints are critical.

Therefore, rather than representing a strict replacement of convolutional approaches, Transformer-based models should be considered as complementary tools, whose benefits depend on the specific characteristics of the dataset and the nature of the defects. In many practical scenarios, especially those involving localized anomalies, the inductive biases of CNNs still provide a strong advantage.

It is important to explicitly delimit the scope of the conclusions drawn above. All experimental findings reported in this work refer specifically to the three fastener types examined (cups, gears, and screw heads), acquired under the particular acquisition conditions, camera configurations, and imaging equipment of the industrial partner Dimac S.r.l., and evaluated using the specific model versions and configurations detailed in Section 3. Different fastener geometries, defect morphologies, acquisition setups, or production environments may lead to a different relative performance between CNN based and Transformer based architectures, and the results presented here should not be interpreted as a general statement of superiority of one paradigm over the other across industrial visual inspection tasks at large. Further validation on additional component types, defect categories, and production lines would be required before extending the present conclusions beyond the specific experimental conditions considered in this study.

Overall, within the scope of the conditions evaluated in this study, and unlike proprietary software, the open source approach explored in this work offers:

- full traceability, enabling complete control over the processing pipeline and model parameters;
- explainability, through segmentation outputs that provide visual insight into the model's decisions;
- no vendor lock-in, eliminating recurring licensing costs and dependency on specific providers.

Finally, techniques such as partial fine-tuning allow for significantly reducing training time while maintaining high performance, making advanced AI solutions more accessible. Within the specific setting studied here, the combination of efficient architectures, flexible training strategies, and dedicated hardware suggests a viable path toward scalable and customizable inspection systems for the fastener production lines considered in this work. Extending this conclusion to other industrial sectors, such as automotive or aerospace component manufacturing, would require dedicated validation on the corresponding defect types and production conditions.

Funding Open access funding provided by Università degli Studi del Piemonte Orientale Amedeo Avogrado within the CRUI-CARE Agreement. The present research has been funded by Regione Piemonte (Italy) under the EU framework FESR 2021/2027 (SWITCH research plan).

Data Availability The datasets used and/or analyzed during the current study are not publicly available due to non-disclosure agreements (NDA). The code developed for this study, comprising the data preprocessing, model training, evaluation, and cross-validation pipeline for all five architectures, is publicly available at https://github.com/andreasenz/fastener-defect-inspection.

Declarations

Conflicts of interest The authors declare no competing interests.

References

1. Hudgins A, James B (2014) Fatigue of threaded fasteners. AM&P Technical Articles
2. Croccolo D, De Agostinis M, Fini S, Mele M, Olmi G, Scapecchi C, Tariq MHB (2023) Failure of threaded connections: A literature review. Machines 11(2):212
3. Shang Z, Li L, Zheng S, Mao Y, Shi R (2024) Fiq: A fastener inspection and quantization method based on mask frcn. Appl Sci 14(12):5267
4. Cundiff CH, Crites RW (1995) Structural integrity of fasteners. Technical report. https://tinyurl.com/2d2pnkyr
5. Gao X, Feng Q, Liu X, Wang Z, Gao S (2025) Study on failure analysis and reliability optimization of high-speed railway fastener system. Eng Fail Anal 170:109279
6. ASTM International (2024) Standard Practice for Fastener Sampling for Specified Mechanical Properties and Performance Inspection (ASTM F1470-24). https://tinyurl.com/2zrmv74c
7. Fasteners — Quality assurance system. ISO 16426 (2002)
8. Verdhan V (2021) Computer Vision Using Deep Learning. Apress, Berkeley, CA
9. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: NIPS 12, Advances in Neural Information Processing Systems, pp. 1097–1105
10. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Proc. of ICLR 2015, San Diego, CA, pp. 1–14. https://arxiv.org/pdf/1409.1556.pdf
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 770–778
12. Strubell E, Verga P, Belanger D, McCallum A (2017) Fast and accurate entity recognition with iterated dilated convolutions. In: 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp. 2670–2680. https://doi.org/10.18653/v1/D17-1283
13. Salehi A, Balasubramanian M (2023) Ddcnet: Deep dilated convolutional neural network for dense prediction. Neurocomputing 523:116–129
14. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention Is All You Need. https://arxiv.org/abs/1706.03762
15. Khan A, Rauf Z, Sohail A, Khan AR, Asif H, Asif A, Farooq U (2023) A survey of the vision transformers and their CNN-transformer based variants. Artif Intell Rev 56(3):2917–2970
16. Wang A (2025) Vision transformers (vits): A new era in computer vision – a review. Journal of Computing and Electronic Information Management 18(3):23–30
17. Dosovitskiy A, Beyer L, Kolesnikov A (2021) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929
18. Tursunalieva A, Alexander DLJ, Dunne R, Li J, Riera L, Zhao Y (2024) Making sense of machine learning: A review of interpretation techniques and their applications. Appl Sci 14(2):496. https://doi.org/10.3390/app14020496
19. Di Marino A, Bevilacqua V, Ciaramella A, De Falco I, Sannino G (2025) Ante-hoc methods for interpretable deep models: A survey. ACM Computing Survey 57(10):1–36. https://doi.org/10.1145/3728637
20. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. https://arxiv.org/abs/2103.14030
21. Liu Z, Hu H, Lin Y, Yao Z, Xie Z, Wei Y, Ning J, Cao Y, Zhang Z, Dong L, Wei F, Guo B (2022) Swin transformer v2: Scaling up capacity and resolution. In: International Conference on Computer Vision and Pattern Recognition (CVPR)
22. Arain TA, Zhang P, Meng Q, Muhammad AU (2025) Swin transformer with attention mechanism: a novel framework for person re-identification. Pattern Anal Appl 28:95
23. Csurka G, Volpi R, Chidlovskii B (2023) Semantic Image Segmentation: Two Decades of Research. https://arxiv.org/abs/2302.06378
24. Jorge P, Volpi R, Torr PHS, Rogez G (2023) Reliability in semantic segmentation: Are we on the right track? In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 7173–7182
25. Zhang B, Xun R, Xu J (2026) A lightweight bearing defect detection model suitable for industrial scenarios. Measurement 258:119239. https://doi.org/10.1016/j.measurement.2025.119239
26. Tu M, Qi Z, Yu L, Zhang L (2023) Few-shot industrial defect image classification based on lightweight model with attention mechanism. In: Proceedings of the 2023 4th Asia Service Sciences and Software Engineering Conference (ASSE 2023), pp. 202–209. https://doi.org/10.1145/3634814.3634841
27. Portinale L, Leonardi G, Montani L, Garau S, Noce G, Brumini M, Ottaviano E (2025) Efficient transformer-based identification of defects in fasteners. In: Thematic Workshops at Ital-IA 2025. CEUR Proceeding, vol. 4121. https://ceur-ws.org/Vol-4121/
28. Tan M, Le QV (2020) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. https://arxiv.org/abs/1905.11946
29. PyTorch Team (2024) ResNet50 — Torchvision 0.17 Documentation. https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html. Accessed: 2024-05-22
30. PyTorch Team (2024) EfficientNet B0 — Torchvision 0.17 Documentation. https://docs.pytorch.org/vision/main/models/generated/torchvision.models.efficientnet_b0.html. Accessed: 2024-05-22
31. Neven R, Goedemé T (2021) A multi-branch u-net for steel surface defect type and severity segmentation. Metals 11(6):870. https://doi.org/10.3390/met11060870
32. Amin A, Ma H, Shazzad Hossain M, Roni NA, Haque E, Asaduzzaman SM, Abedin R, Ekram AB, Farzana Akter R (2022) Industrial product defect detection using custom u-net. In: 2022 25th

International Conference on Computer and Information Technology (ICCIT), pp 442–447. https://doi.org/10.1109/ICCIT57492.2022.10055987
33. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. https://arxiv.org/abs/1505.04597
34. Cheng G, Misra I, Schwing AG, Kirillov A, Girdhar R (2021) Masked-attention mask transformer for universal image segmentation. arXiv:2112.01527
35. Research FA (2022) Mask2Former Swin-Large ADE Semantic Segmentation Model. Hugging Face, Model Card
36. Iakubovskii P. Segmentation Models.Pytorch. https://github.com/qubvel-org/segmentation_models.pytorch
37. Zhou B, Zhao H, Puig Fernandez X, Xiao T, Fidler S, Barriuso A (2019) Adela andTorralba: Semantic understanding of scenes through the ADE20K dataset. Int J Comput Vision 127:302–321. https://doi.org/10.1007/s11263-018-1140-0
38. Corporation NVIDIA (2025) NVIDIA GeForce RTX 5070 Product Specifications. Santa Clara, NVIDIA Corporation, Product specifications
39. NVIDIA Corporation (2023) NVIDIA L40 GPU Datasheet. NVIDIA Corporation,(Technical datasheet)