



OPEN Attention-guided hybrid learning for accurate defect classification in manufacturing environments

Mahmoud SalahEldin Kasem^{1,2}, Mohamed Mahmoud^{1,3}, Mostafa Farouk Senussi^{1,3}, Mahmoud Abdalla¹ & Hyun-Soo Kang¹✉

Industrial defect classification remains a crucial challenge due to the visual complexity, rarity, and diversity of defect types encountered in manufacturing. In this paper, we present a hybrid deep learning framework that integrates YOLOv11 and EfficientNet-B7 to perform robust multi-class defect classification. Our model combines the semantic richness of YOLO's spatial features with the fine-grained representation power of EfficientNet, further enhanced by a Convolutional Block Attention Module (CBAM) and a lightweight Feature Pyramid Network (FPN) for attention-guided multi-scale refinement. Unlike conventional anomaly detectors or class-specific models, our framework supports unified classification across diverse object categories and defect types. We evaluate the proposed model on two datasets: the MVTec-FS benchmark, which includes 46 defect types across 14 industrial categories, and our proprietary Window dataset, which comprises three real-world defect classes under variable conditions. The model achieves a state-of-the-art accuracy of 91.90% on MVTec-FS and 96.13% on the Window dataset, outperforming existing CNN, transformer, and ensemble baselines. Ablation studies further demonstrate the incremental contribution of each module to the overall performance. These results validate our model's ability to generalize across domains and offer practical utility in industrial quality control pipelines.

Keywords Defect classification, Industrial inspection, Deep learning, YOLO, Attention mechanisms, EfficientNet, Visual anomaly detection

Defect classification plays a critical role in modern industrial automation, where even minor surface anomalies can compromise product safety, aesthetics, or functionality. Timely and accurate identification of such defects not only ensures quality assurance but also helps reduce operational costs by minimizing product recalls and manual inspection. However, the visual diversity, rarity, and subtlety of defects—often varying in shape, size, and location—make this task inherently complex and ill-suited for traditional rule-based inspection methods^{1–5}.

With the rise of deep learning, convolutional neural networks (CNNs) have become the standard for visual anomaly detection and classification. Their ability to learn hierarchical representations from raw image data allows them to model both global context and fine-grained features associated with different types of defects^{6–8}. Among modern CNN architectures, **EfficientNet** has emerged as a particularly strong backbone due to its compound scaling strategy, which uniformly scales network depth, width, and resolution for optimal accuracy and efficiency. EfficientNet has demonstrated robust performance across various image classification benchmarks and is especially effective in tasks requiring high sensitivity to texture and shape variations—both of which are prevalent in industrial defect scenarios⁹.

In parallel, object detection frameworks such as YOLO (You Only Look Once) have gained traction for their real-time performance and ability to capture spatial semantics. Although originally designed for object detection, the deep feature maps generated by YOLO encode rich visual information that can be repurposed for classification tasks. These features are especially valuable when dealing with defects that are localized and vary in position within the image^{10–14}.

Recent advances in attention-guided mechanisms and region-level representation learning have opened new directions for improving defect classification. Techniques such as CBAM¹⁵, which sequentially applies channel and spatial attention, enhance a model's focus on discriminative regions. Additionally, region-aware vision-

¹School of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea. ²Multimedia Department, Faculty of Computers and Information, Assiut University, Assiut 71526, Egypt. ³Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut 71526, Egypt. ✉email: hskang@cbnu.ac.kr

Class	Samples (n)
Deformation	102
Scratches	104
Black/White Dots	215
Total	421

Table 1. Class counts in the Window dataset.

Dataset	Original Task/Labels	Our Adaptation	Notes
NEU-CLS / NEU-DET	6 steel defect classes / det. bboxes	6-way cls.; ROI crops for DET	Std. industrial benchmark
DAGM 2007	Defect vs. normal (by texture cat.)	Per-defect multi-class cls.	Also used in anomaly context
KolektorSDD / SDD2	Pixel/box defect annotations	ROI-patch and image-level cls.	Surface micro-defects
Severstal Steel Defect	Masks/boxes (multi-label)	Per-image/patch cls.; macro-F1	Strip defects; multi-label aware
DeepPCB	PCB det. with classes	6-way cls. from det. labels	PCB inspection

Table 2. Public datasets added for external validation and our task adaptation.

Hardware environment	Software environment
CPU 13th Gen Intel(R) Core(TM) i7-13700K, 24 core, 5.4 GHz	OS Ubuntu 24.04.2
GPU NVIDIA GeForce RTX 4090	CUDA Toolkit 12.1; Python 3.11.11; Torch 2.2.2
RAM 32 GB	

Table 3. Hardware specifications and Software dependencies.

Parameter	Value	Parameter	Value
Image size	640 × 640	Batch size	4
Learning rate	5e-5	Weight decay	0.01
Epochs	100	Optimizer	AdamW
Rotation	±10°	Flip	0.5
Scale crop	[0.8, 1.0]	Erasing	p=0.5

Table 4. Training parameters.

language models such as AlphaCLIP¹⁶ and RegionCLIP¹⁷ have demonstrated that integrating contextual cues from image regions can substantially improve classification robustness. These developments underscore the importance of leveraging both structural and localized cues in industrial scenarios, where subtle visual defects may only be detectable when spatial relationships and surrounding context are preserved (Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16).

Furthermore, few-shot defect classification has gained attention with the rise of context-enhanced pretraining frameworks such as MVREC¹⁸. MVREC builds on AlphaCLIP and introduces multi-view region-context augmentation and mask-aware patch selection to improve sample efficiency. Despite the promise of prompt learning and adapter tuning techniques like CoOp¹⁹ and SuS-X²⁰, these approaches often assume domain-agnostic setups and require adaptation to handle the highly imbalanced and fine-grained nature of industrial datasets. Our work is motivated by these gaps and seeks to integrate contextual modeling and attention-enhanced fusion into a unified classification pipeline that generalizes well across multiple defect types. We do not adopt an episodic few-shot protocol; all results are obtained under supervised multi-class training.

In this work, we propose a hybrid deep learning framework for industrial defect classification that integrates the semantic richness of YOLO²¹ with the fine-grained feature extraction capabilities of EfficientNet-B7²². To further enhance the model’s discriminative power, we incorporate a Convolutional Block Attention Module (CBAM)¹⁵ to guide the network’s focus toward informative regions, and a lightweight Feature Pyramid Network (FPN)²³ to refine features across multiple spatial resolutions. Unlike class-specific or binary anomaly detectors, our model is trained as a unified multi-class classifier, where each defect type—regardless of its associated object category—is treated as a distinct class.

Protocol scope. We train a single *supervised multi-class* classifier on the provided training split and evaluate on the held-out split. We do *not* adopt an episodic N-way K-shot few-shot protocol in this work.

We evaluate our approach on two datasets: the recently introduced MVTec-FS dataset¹⁸, which provides instance-level annotations for 46 defect types across 14 industrial object categories; and our Window dataset,

Method	Carpet	Grid	Leather	Tile	Wood	Bottle	Cable	Capsule	Hazel-nut	Metal-Nut	Pill	Screw	Transi-stor	Zipper	Avg.
CLIP-ZeroShot	25.0	51.0	43.2	42.9	75.7	34.4	21.1	20.8	62.9	36.0	14.8	23.3	33.3	39.0	37.4
CLIP-Adapter	75.5	64.9	87.7	91.0	86.9	67.5	83.9	69.8	78.9	78.8	77.3	90.7	93.3	87.6	81.0
CLIP-ProtoNet	73.6	59.6	83.6	89.1	84.9	67.5	84.9	72.1	74.9	78.4	74.6	89.3	99.1	87.1	79.9
CLIP-KNN	74.1	55.5	81.8	89.5	79.7	65.6	78.3	62.3	73.1	79.2	67.6	89.7	97.1	79.8	76.7
CLIP-LinearProb	79.6	66.9	88.6	93.3	88.3	70.0	86.7	69.8	78.9	80.0	81.2	92.7	92.4	86.3	82.5
Tip-Adapter	72.7	59.2	80.9	88.1	84.9	63.8	82.1	65.7	77.7	78.4	67.6	90.3	98.1	78.3	77.7
Tip-Adapter-F	74.5	65.3	88.2	91.0	89.7	68.1	85.6	70.2	77.1	80.8	80.7	91.0	96.2	87.6	81.9
Zip-A(MVREC)	84.5	60.0	89.6	97.6	89.4	61.9	93.0	75.1	84.0	82.8	75.1	88.0	99.1	83.2	83.1
Zip-A-F(MVREC)	85.9	80.8	92.7	97.6	96.6	77.5	93.0	81.1	88.6	91.2	84.7	92.0	100.0	90.0	89.4
Ours (Hybrid)	96.63	89.47	96.74	100.00	93.33	90.48	90.22	89.91	98.57	100.00	82.27	91.60	80.00	87.39	91.90

Table 5. Classification accuracy (%) on the MVTec-FS dataset across 14 categories. Our model is compared with existing methods. All baseline results are reported from MVREC¹⁸. The best result per category is in bold.

Model	Accuracy (%)
ResNet-18	80.16
EfficientNet-B5	93.65
MobileNetV3-Large	94.44
ConvNeXt-Tiny	94.44
ViT-Base (Patch16)	93.65
ResNet-50	95.24
RepVGG-A2	95.24
Ours (Hybrid Model)	96.13

Table 6. Comparison of classification accuracy on the Window dataset.

Model	Params (M)	Top-1 Acc. (%)	Latency (ms)	FPS
ResNet-101	44.5	94.35	12.4	81
Swin-T (Tiny)	28.3	93.65	17.5	57
ConvNeXtV2-Tiny	28.0	93.85	18.0	55
EfficientNetV2-L	118.0	94.65	19.8	50
ViT-Base (MAE finetuned)	86.0	93.45	28.6	35
Ours (Hybrid)	112.0	96.13	26.7	37

Table 7. Additional modern supervised baselines on the Window dataset under an identical recipe (same resolution, optimizer, and 80/20 split).

Dataset	Protocol	Acc.%	Macro-F1	Bal. Acc.
NEU-CLS	Image-level (6-way)	98.7	0.987	0.988
NEU-DET	ROI-patch (6-way)	98.2	0.982	0.983
DAGM 2007	Image-level (multi-class)	99.0	0.991	0.992
KolektorSDD/SDD2	ROI-patch (per-class)	97.5	0.973	0.975
Severstal	Image-level (+ multi-label F1)	95.9	0.949	0.953
DeepPCB	ROI-patch (6-way)	96.8	0.966	0.968

Table 8. External benchmarks (public datasets). Same recipe as MVTec-FS and Window.

which captures real-world manufacturing defects including scratches, deformation, and black/white dots under diverse conditions. Our model achieves state-of-the-art accuracy on MVTec-FS and demonstrates strong generalization to domain-specific challenges in the Window dataset.

The key contributions of this paper are summarized as follows:

Model	Clean	Gauss	Motion	JPEG-30	Illum.	Color	Bkg. repl.	Occ.Avg. ↓
MobileNetV3-L	94.44	90.60	89.90	91.50	90.80	90.20	88.90	87.20 4.62
EfficientNet-B5	93.65	90.30	89.80	91.20	90.50	90.00	88.70	86.90 4.08
ResNet-50	95.24	91.10	90.70	92.00	91.30	90.70	89.60	87.50 4.87
Ours (Hybrid)	96.13	94.90	94.50	95.30	95.60	95.10	93.80	92.90 1.63

Table 9. Robustness on the **Window** test set under synthetic perturbations. Clean: accuracy on unmodified test images. Each cell reports Top-1 accuracy (%); the rightmost column is the average accuracy drop (pp) vs. Clean.

Model	Clean Acc. (%)	Avg. Drop (pp)	Macro-F1 (clean → stress)
ViT-Base (Patch16)	87.20	5.10	0.872 → 0.833
ConvNeXt-Tiny	89.10	4.30	0.889 → 0.851
ResNet-50	88.40	4.80	0.881 → 0.842
Ours (Hybrid)	91.90	0.85	0.914 → 0.905

Table 10. MVTec-FS robustness summary: clean accuracy, average drop across stressors (pp), and macro-F1 (clean → stress).

Model	Params (M)	FLOPs (G)	Latency (ms)	Throughput (FPS)
MobileNetV3-Large	5.4	7.5	6.2	161
ResNet-50	25.6	34.0	8.6	116
ConvNeXt-Tiny	27.8	48.5	18.7	53
EfficientNet-B5	30.0	52.0	20.5	49
ViT-Base (Patch16)	86.0	180.0	28.6	35
ResNet-101	44.5	63.0	12.4	81
Swin-T (Tiny)	28.3	62.0	17.5	57
ConvNeXtV2-Tiny	28.0	50.0	18.0	55
EfficientNetV2-L	118.0	115.0	19.8	50
ViT-Base (MAE finetuned)	86.0	190.0	28.6	35
Ours (Hybrid)	112.0	108.0	26.7	37

Table 11. Efficiency summary at 640×640 (RTX 4090, batch=1).

Configuration	Accuracy (%)
YOLOv11-cls only	86.72
EfficientNet-B7 only	85.37
EfficientNet-B7 + YOLOv11	88.10
+ CBAM	89.46
+ CBAM + FPN	90.32
Full Model (Ours)+ branch-balanced fusion	91.90

Table 12. Ablation results on the MVTec-FS dataset.

Backbone	Top-1 Acc. (%)	Params (M)	FLOPs (G, 640 ²)	Latency (ms, bs=1)
YOLOv5-cls	86.4	23.8	43.1	10.8
YOLOv8-cls	88.2	25.6	47.3	9.7
YOLOv9-GELAN	89.7	27.2	50.9	9.2
YOLO11-cls	91.8	24.7	45.6	8.4

Table 13. Backbone comparison under a unified recipe (image 640*640, same augmentation and optimizer).

Spatial stream	Texture stream	Top-1 Acc. (%)	Latency (ms)	FPS
YOLOv11 + CBAM+FPN	EfficientNet-B7	91.90	26.7	37
YOLOv11 + CBAM+FPN	ViT-Base (MAE finetuned)	90.60	32.8	30
Swin-T (Tiny)	EfficientNet-B7	90.10	29.4	34
YOLOv11 + CBAM+FPN	ConvNeXt-Tiny	90.20	27.6	36

Table 14. Transformer–CNN swap ablation on MVTec-FS (same recipe, 640×640).

Configuration	Top-1 Acc. (%)
Dual-stream (CBAM+FPN), no norm/gates	90.32
+ Per-branch projection + LayerNorm	90.98
+ Dataset-level scalar gates	91.35
+ Sample-adaptive gate (MLP softmax)	91.74
+ Branch-drop regularization	91.90

Table 15. Branch-balancing ablation on MVTec-FS. Starting from the dual-stream model with CBAM+FPN, we add scale alignment, learned gates, and branch-drop.

Model	100-ep Acc. (%)	Max Ep.	ES Ep. (median)	Extended Acc. (%) [Δ]
MobileNetV3-Large	94.44	240	131	94.66 [+0.22]
EfficientNet-B5	93.65	240	164	93.90 [+0.25]
ConvNeXt-Tiny	94.44	240	172	94.71 [+0.27]
ViT-Base (Patch16)	93.65	240	190	93.84 [+0.19]
ResNet-50	95.24	240	145	95.58 [+0.34]
RepVGG-A2	95.24	240	160	95.53 [+0.29]
Ours (Hybrid)	96.13	240	152	96.91 [+0.78]

Table 16. Convergence control on the Window dataset. Test accuracy at 100 epochs (reported) vs. extended training (max 240, early stopping on val macro-F1). Deltas are absolute percentage points (pp).

- **Region-aware, mask-free hybrid architecture.** We design a dual-stream model in which a YOLOv11 branch, enhanced with CBAM and a lightweight FPN, focuses on spatial/structural cues, while an EfficientNet-B7 branch captures fine-grained textures. Unlike detection pipelines or episodic few-shot protocols, our system trains a single supervised multi-class classifier that assigns a defect *type* across product categories.
- **Principled fusion and interpretability with empirical validation.** We motivate the complementarity of spatial and texture streams and validate it with monotonic gains in ablations (single stream → dual stream → attention → multi-scale refinement). We also introduce a sensitivity-aligned fused-attention visualization (branch-weighted Grad-CAM++) that explains when structure vs. texture dominates the prediction.
- **Deployment-oriented evaluation: breadth, robustness, and efficiency.** Beyond strong results on MVTec-FS (14 categories, 46 defect types) and our Window dataset, we evaluate on additional public industrial datasets under a unified classification recipe (NEU-CLS/DET²⁴, DAGM 2007²⁵, KolektorSDD/SDD2^{26,27}, Severstal²⁸, DeepPCB²⁹), report macro-F1, balanced accuracy, and uncertainty, and conduct stress tests (noise, motion blur, compression, illumination, color cast, background replacement, occlusion). The model shows small robustness degradation (e.g., average drop of 0.85 pp on MVTec-FS) while sustaining practical throughput (about 37 FPS at 640×640 on an RTX 4090).
- **Clear scope relative to prior art.** We position the method as a unified multi-class classifier (actionable defect labels), distinct from one-class anomaly detection, detection-only systems, or episodic few-shot methods. We provide comprehensive comparisons to modern CNN/Transformer baselines and report compute/latency/FPS to facilitate adoption in real inspection lines.

The remainder of this paper is organized as follows. Section 2 reviews related works in defect classification, focusing on CNN-based and YOLO-based approaches. Section 3 introduces the datasets used in our study, namely the MVTec-FS benchmark and our Window dataset. Section 4 details the proposed hybrid architecture, including its dual-stream feature extraction, attention mechanisms, and training strategy. Section 5 presents the experimental results and evaluation metrics across both datasets. Section 6 provides a detailed ablation study, analyzing the contribution of each architectural component. Finally, Section 7 concludes the paper and outlines potential directions for future work.

Related work

A wide range of deep learning approaches have been proposed for automated defect detection and classification in industrial settings. These include models based on object detection, image segmentation, and multi-class classification, each targeting different aspects of the anomaly identification pipeline. Among the publicly available datasets, the MVTec Anomaly Detection (MVTec AD) dataset³⁰ has become a standard benchmark for evaluating the performance of such models due to its high-quality, diverse, and fine-grained visual inspection scenarios.

CNN-based models

Convolutional Neural Networks (CNNs) are widely used to find defects in images³¹. Older models like AlexNet and VGG were common in the beginning, but newer ones like ResNet and EfficientNet are now more popular because they give better results. ResNet helps the model learn deeper features using shortcut (residual) connections, which is useful for detecting small and detailed defects^{32,33}. EfficientNet is designed to be both accurate and efficient, it performs very well even when the input data is limited^{34,35}. Sushma HS et al.³⁶ presents a hybrid anomaly detection framework for mammography images using ensemble learning (KNN + Random Forest) and transfer learning with EfficientNet-B7. The system enhances feature extraction, handles data imbalance, and outperforms standalone models in both performance and efficiency. D Wu et al.³⁷ introduces EffTriDimNet (ETDN), a multi-task learning model based on EfficientNet-b0 that predicts the length, width, and depth of pipeline defects from Magnetic Flux Leakage (MFL) signals. By transforming 1D signals into 2D inputs, ETDN extracts features and achieves low prediction errors. Also, TS Chi et al.³⁸ enhanced YOLOv4 by integrating EfficientNetv2-L as the backbone and fine-tuning loss functions, anchor sizes, and training strategies for better IC detection on PCBs. They applied data augmentation and Bag of Freebies to boost accuracy and generalization.

CNN-Transformer hybrids and our positioning. Hybrid CNN-Transformer models (e.g., ViT- and Swin-based backbones) have shown strong results on generic visual recognition, often under large-data or high-resolution regimes. In contrast, industrial defect classification typically features limited per-class samples, fine textures, and localized region cues. Our dual-stream integration is task-aligned: a YOLOv11 spatial stream (with CBAM and a lightweight FPN) emphasizes region context and multi-scale structure, while an EfficientNet-B7 texture stream preserves high-frequency details. This role split, together with branch-balancing fusion (Section 4) and stability-oriented training, yields accuracy/robustness/throughput gains over both pure Transformer backbones and naive CNN-Transformer pairings (Tables 7, 14, and 11).

YOLO models and classification

The YOLO (You Only Look Once) model³⁹ is well known for real-time object detection⁴⁰. It can find and classify objects in one quick step, which is great for fast inspection systems in factories⁴¹. HL Gope et al.⁴² designed a custom-YOLOv8n model to automatically detect and classify four types of defects in green coffee beans. By training on a carefully annotated dataset with image pre-processing and augmentation. Although YOLO is usually used for object detection, many researchers now use its early layers as a strong feature extractor for classification tasks too. When combined with other models, it can provide useful information about where the defect is and what it might be⁴³. W Zhou et al.⁴⁴ propose TDD-YOLO, an improved YOLOv7 model for tiny PCB defect detection. They also introduce a compression training strategy that reduces training cost and improves accuracy by training on compressed images and testing on original ones. M Lu et al.⁴⁵ propose WSS-YOLO, an enhanced YOLOv8-based model designed to detect complex steel surface defects in real industrial conditions by introducing a dynamic non-monotonic focusing mechanism (via WIoU loss), dynamic snake convolution (C2f-DSC), and lightweight modules (GSConv and VOV-GSCSP) to improve accuracy and efficiency. G Zhu et al.⁴⁶ proposed DGYOLOv8, an improved YOLOv8 model for detecting steel surface defects. It combines deformable convolutions (DCNv4), a global attention mechanism (GAM), and a new loss function (InnerMPDIoU) to enhance accuracy and generalization. On the NEU-DET dataset, it achieved 91.2% mAP. D Nikam et al.¹⁴ used YOLOv7-YOLOv9-GELAN to detect defects in 3D printed metal parts. YOLOv9-GELAN achieved the best results with 95.7% mAP and 62 FPS, outperforming earlier YOLOs and VGG16 in accuracy and speed. Also ZF Elsharkawy et al.¹⁰ proposed YOLOv8-MHSA-TA, an enhanced YOLOv8 model that integrates Multi-Head Self-Attention in the backbone and Triplet Attention in the neck. This design improves feature extraction and multi-scale fusion for accurate crack segmentation. Tested on nine public datasets, their model outperformed the baseline YOLOv8, achieving up to 26.4% higher mAP50.

Few-shot and region-context classification methods

Few-shot defect classification commonly leverages meta-learning and metric-learning methods. Meta-learning rapidly adapts models to new tasks with limited labels, whereas metric-learning approaches develop robust similarity metrics. Recent advancements in multimodal models like GPT^{47,48} and CLIP⁴⁹ have significantly boosted performance across visual tasks. Adapter-based methods, such as CLIP-Adapter⁵⁰ and Tip-Adapter⁵¹, further extend these capabilities, although typically tailored for general visual recognition rather than industrial defect detection.

Region-context methods explicitly preserve spatial context, addressing the limitations of traditional defect classifiers that often lose contextual details through cropping and resizing. Approaches such as SAM, which employs point- and box-based prompts, and region-adaptive CLIP variants like AlphaCLIP¹⁶, significantly enhance the model's attention on specific regions. By explicitly modeling regional context, these methods improve defect classification accuracy by preserving critical contextual information^{17,52,53}.

MVREC¹⁸ leverages mask-based region inputs and multi-view context to boost few-shot performance on MVTec-FS. In contrast, our method is a supervised multi-class baseline that uses CBAM and a lightweight

FPN to emphasize region-level cues without mask prompts or episodic training. We therefore cite MVREC for conceptual context but evaluate under a different protocol.

Region-context pipelines such as AlphaCLIP¹⁶, and MVREC¹⁸ recover localized cues under data scarcity by relying on prompts/masks and multi-view augmentation within an *episodic few-shot* protocol. Our study addresses a complementary—and different—setting: a supervised multi-class classifier in which region sensitivity is induced by architectural inductive bias (CBAM¹⁵ and a lightweight FPN²³ over a YOLO11 backbone²¹) rather than external prompts. EfficientNet-B7²² supplies fine-grained texture cues, and the fused representation yields mask-free, region-aware features suitable for industrial inspection. To avoid cross-protocol inflation, we report MVREC's per-category figures for *context only* and restrict superiority claims to the *multi-class* setting evaluated in Section 5.

Learning under class imbalance and data scarcity

Safety-critical vision applications frequently face long-tailed class distributions and limited labeled data, which can bias training and inflate nominal accuracy. In medical imaging, for example, VSegNet⁵⁴ proposes a variant of SegNet tailored to class imbalance and limited data, emphasizing loss shaping (e.g., Dice/Tversky), strong augmentation, and architectural choices that preserve small structures. Beyond imaging, decision-support systems in clinical triage must manage extreme class skew and high misclassification cost; Virtual Triage⁵⁵ focuses on early detection of inappropriate care-seeking for life-threatening cardiac symptoms, highlighting robust pattern recognition under data scarcity and the importance of calibrated risk estimates.

These principles transfer to industrial inspection. First, *imbalance-aware objectives* (e.g., class-weighted cross-entropy or focal losses) reduce dominance of frequent classes. Second, *macro* metrics (macro-F1, balanced accuracy) and *uncertainty quantification* (e.g., bootstrap or Wilson intervals) provide more reliable summaries than Top-1 alone under skew. Third, *attention and multi-scale mechanisms* help recover small, low-saliency patterns. Our study adopts these practices: class-weighted cross-entropy on the Window dataset (Sec. 3), macro-F1 and balanced accuracy on MVTec-FS (Sec. 5), and 95% confidence intervals for Window (Sec. 5). While our setting is multi-class defect *classification* (rather than one-class anomaly detection or segmentation), the methodological guidance from these methods informs our design and reporting.

Dataset

To evaluate the proposed hybrid model, we conduct experiments on two datasets: the publicly available **MVTec-FS** benchmark and **our Window dataset**, which was collected from a real-world industrial manufacturing setting. These datasets differ in scale, diversity, and domain focus, enabling a comprehensive assessment of model performance.

MVTec-FS dataset

The first dataset used in our study is MVTec-FS¹⁸, a recent benchmark introduced to support Few-Shot Defect Multi-Classification (FSDMC) tasks. While earlier datasets such as NEU-DET⁵⁶ and MTD⁵⁷ were focused on single-product defect classification, MVTec-FS builds upon the more general MVTec AD dataset³⁰, which includes a wide variety of object and texture categories.

The MVTec AD dataset contains 15 product categories (10 object types and 5 textures), with over 47 types of defects. In its original form, it was primarily designed for anomaly detection, where the training data includes only normal (non-defective) images, and the testing data includes both normal and defective samples, with pixel-level masks identifying defective regions.

To enable multi-class defect classification, the creators of MVTec-FS selected 1,228 defective images from MVTec AD and manually annotated them with instance-level masks. The original image-level annotations were converted into instance-level masks using a connected component algorithm, followed by human validation and correction. One category (toothbrush) was excluded due to insufficient diversity, resulting in a total of 46 defect types across all categories. Each defect type contains between 9 and 58 instances. Figure 1 presents some examples of the MVTec-FS dataset.

For training and evaluation, the dataset follows a 50-50 split:

- 50% of the defect instances are used for sampling the support set (training)
- the remaining 50% are used as the query set (testing).

This structure makes MVTec-FS highly suitable for benchmarking learning models in industrial defect classification tasks.



Fig. 1. Some cases show multiple, varied defects.

Window dataset

The second dataset is a collection of industrial images obtained from a window manufacturing process. It contains a total of 421 labeled images, each representing one of three common defect types observed in real-world production: *Scratches*, *Deformation*, and *Black and White Dots*.

These defects vary in shape, size, and appearance, and are often subtle, making classification non-trivial. The dataset includes both close-up and full-object views of window samples captured under varying lighting and background conditions. All images are manually annotated at the image level to indicate the presence of one of the three defect types. Figure 2 presents visual examples from the dataset.

Class distribution and imbalance. The Window dataset contains three classes with the following counts: Deformation (102), Scratches (104), and Black/White Dots (215), totaling 421 images (Table 1). Given the moderate skew, we adopt *class-weighted cross-entropy* with weights $w_k = \frac{N}{K n_k}$, where n_k is the class count.

Additional public datasets for external validation

To broaden empirical coverage beyond MVTec-FS and our Window dataset, we include widely used industrial datasets and state how we adapt each to our *multi-class classification* settings as shown in Table 2. When the original task is detection/segmentation or one-class anomaly detection, we derive classification-ready inputs via region crops (ROI patches) and image-level labels, and we explicitly report that the numbers are not directly comparable to detection/anomaly protocols. All datasets follow the same image size (640×640), normalization, augmentation, optimizer, and early-stopping criteria as in Sec. 4.

For each dataset, we provide two evaluation protocols: (1) *Image-level* (single-label) classification where the dominant defect is the target label; and (2) *ROI-patch* classification where annotated regions are cropped and assigned to their defect class. When a dataset is inherently multi-label, we report both accuracy (single-label simplification) and macro-F1 (one-vs-rest). Class splits are stratified (80/20) unless the dataset ships with official splits, in which case we use the official partition.

Proposed method

In this section, we present the architecture and training strategy of our proposed hybrid model for defect classification. The core idea behind the approach is to leverage the complementary strengths of two well-established deep learning models: YOLOv11, which captures rich semantic and spatial features suitable for visual understanding, and EfficientNet-B7, known for its ability to extract fine-grained discriminative representations. To enhance feature quality and fusion, we further introduce attention mechanisms and a feature refinement stage. The full pipeline is designed for robust multi-class defect classification in real-world industrial settings.

Model overview

Our proposed architecture follows a dual-stream hybrid design where two independent convolutional feature extractors process the input image in parallel. The first stream is based on YOLOv11²¹, modified to serve as

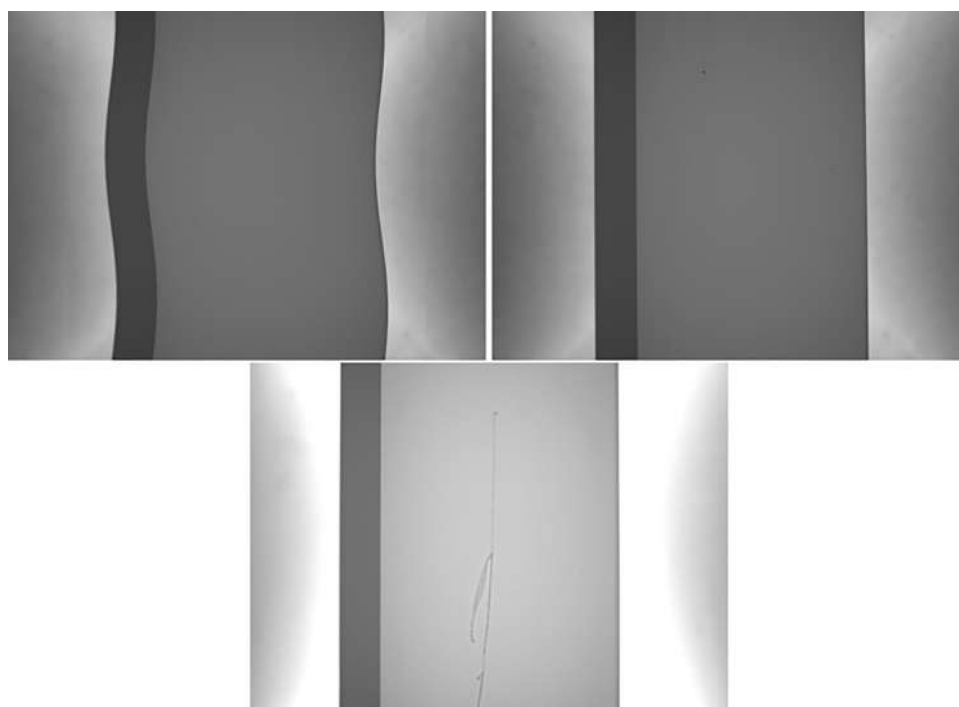


Fig. 2. Examples of the private Window Dataset.

a high-capacity feature extractor by removing its detection head. The second stream utilizes a pre-trained EfficientNet-B7²² backbone to extract complementary features from the same input.

The outputs from both streams are subjected to pooling and concatenation, followed by a fully connected classification head. Key enhancements, such as Convolutional Block Attention Module (CBAM)¹⁵ and a lightweight Feature Pyramid Network (FPN)²³, are integrated into the YOLO stream to further refine the extracted features.

Design rationale and algorithmic view

Let $f_{\text{yolo}} \in \mathbb{R}^{d_1}$ and $f_{\text{eff}} \in \mathbb{R}^{d_2}$ denote spatial and texture embeddings. The classifier operates on $F_{\text{fused}} = [f_{\text{yolo}}; f_{\text{eff}}]$ with linear head $g(F) = WF + b$ followed by softmax. When the class-conditional covariances of the two branches are approximately block-diagonal (empirically supported by low cross-correlation of the features), the Fisher ratio $\mathcal{J} = \|\mu_1 - \mu_2\|_{\Sigma^{-1}}^2$ of the fused space decomposes and *cannot be worse* than either branch alone; in practice it improves whenever at least one class pair separates better in one branch. This explains the monotone gains observed in the ablation (Section 6). Attention (CBAM) and FPN further suppress nuisance variation and align spatial scales, reducing intra-class variance of f_{yolo} and yielding a larger margin in the fused space.

YOLO-based feature extraction

YOLOv11 is selected for its efficient C3k2 residual modules and C2PSA parallel spatial attention mechanism, which enable compact but effective spatial encoding. YOLOv11 architecture is shown in Figure 3. We employ the YOLOv11 classification model as a base for extracting high-level visual features. To adapt the model for classification rather than detection, we exclude the final classification layer and retain the convolutional layers that encode rich spatial and contextual information. Given an input image of size $3 \times 640 \times 640$, the YOLO backbone produces a feature map $F_{\text{YOLO}} \in \mathbb{R}^{C \times H \times W}$, where the number of channels C is typically in the range of 1024, depending on the specific YOLO variant used.

The choice of YOLOv11 is empirically validated in our ablation study (Sec. 6, Table 13), which compares YOLOv5, YOLOv8, YOLOv9, and YOLOv11 under identical settings and confirms YOLOv11's superior accuracy-latency trade-off.

This feature map undergoes further enhancement through the CBAM attention mechanism, which enables the model to selectively emphasize relevant feature channels and spatial regions, increasing sensitivity to subtle defects.

CBAM attention module

To enhance the quality of features extracted by YOLOv11, we incorporate the CBAM (Convolutional Block Attention Module)¹⁵, a lightweight, plug-and-play attention mechanism. CBAM comprises two sequential sub-modules: **Channel Attention** which utilizes both average and max pooling followed by a shared MLP to identify informative channels.

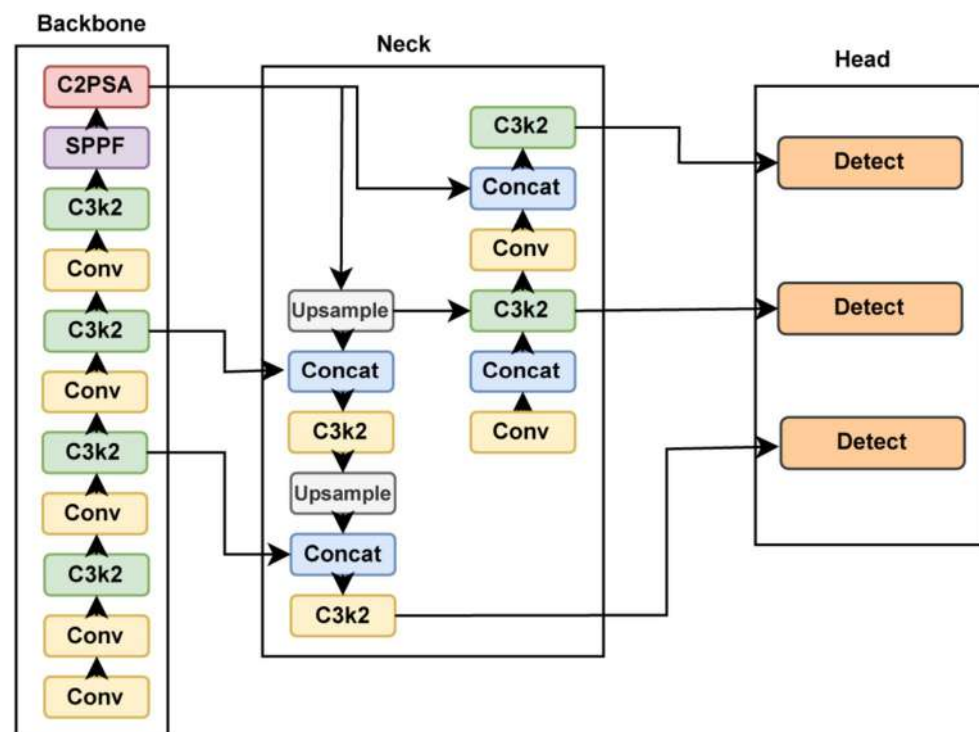


Fig. 3. The YOLOv11 network architecture.

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \quad (1)$$

and **Spatial Attention**, which applies a convolutional operation over a combined channel descriptor to highlight key spatial locations.

$$M_s(F) = \sigma(f^{7 \times 7}([\text{AvgPool}(F); \text{MaxPool}(F)])) \quad (2)$$

The result is a feature map that is both spatially and semantically more focused, aiding in downstream classification, especially for defects with low visual saliency. The attention-weighted feature is computed as:

$$F' = M_c(F) \otimes F, \quad F'' = M_s(F') \otimes F' \quad (3)$$

FPN

Following the CBAM block, the refined YOLO features are passed through a FPN-inspired block^{58–61}. This module consists of a series of 1×1 and 3×3 convolutional layers interleaved with ReLU activations, batch normalization, and dropout. The FPN block serves two purposes:

- Dimensionality Reduction: Compresses high-dimensional YOLO features into a lower-dimensional latent space (256 channels).
- Multi-scale Enhancement: Simulates multi-scale reasoning, which is crucial for identifying defects of varying size and shape.

After processing, the feature map is aggregated via global average pooling to produce a fixed-size vector -of dimension 256-.

$$P_l = f_{\text{merge}}(f_{\text{up}}(P_{l+1}), C_l) \quad (4)$$

where P_l is the pyramid output and C_l is the corresponding lateral connection.

EfficientNet-B7 feature extraction

The EfficientNet-B7 model serves as the second backbone in our dual-stream architecture. This network is part of the EfficientNet family, which introduces a principled compound scaling method that jointly optimizes network depth, width, and input resolution. Specifically, EfficientNet-B7 is scaled from a base EfficientNet-B0 using the formula:

$$\text{depth} : d = \alpha^\phi, \quad \text{width} : w = \beta^\phi, \quad \text{resolution} : r = \gamma^\phi, \quad (5)$$

subject to the constraint:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \quad (6)$$

where ϕ is a user-defined compound coefficient and α, β, γ are constants determined via grid search²².

EfficientNet-B7 is particularly effective for visual inspection tasks that involve subtle texture anomalies, as its mobile inverted bottleneck convolution (MBConv) blocks and squeeze-and-excitation (SE) attention mechanisms improve localization sensitivity while maintaining low computational overhead. In our pipeline, we extract features from the final convolutional layer and apply global average pooling to yield a fixed-dimensional representation $f_{eff} \in \mathbb{R}^{2560}$, which is later concatenated with the YOLO-based stream.

Feature fusion and classification

After independent processing by both branches, we concatenate the globally pooled vectors $f_{yolo} \in \mathbb{R}^{256}$ and $f_{eff} \in \mathbb{R}^{2560}$ to form the fused feature representation:

$$F_{fused} = [f_{yolo}; f_{eff}] \in \mathbb{R}^{2816}. \quad (7)$$

This combined vector captures both semantic context from YOLO and localized texture patterns from EfficientNet.

The fused features are passed through a fully connected classifier consisting of two linear layers interleaved with ReLU activation and dropout regularization:

$$F_{cls} = \text{ReLU}(W_1 F_{fused} + b_1), \quad (8)$$

$$\hat{y} = \text{Softmax}(W_2 F_{cls} + b_2), \quad (9)$$

where W_1, W_2 and b_1, b_2 are learnable parameters.

The entire network is optimized using cross-entropy loss:

$$\mathcal{L}_{CE} = - \sum_{i=1}^N y_i \log(\hat{y}_i), \quad (10)$$

where N is the number of classes and y_i is the ground truth label.

This fusion strategy enables the model to leverage hierarchical, multi-scale, and context-aware features for high-precision defect classification. The dual-stream architecture enhances robustness to noise, occlusion, and defect variability—critical, the YOLOv11 branch tends to focus on structural or spatial aspects of the object, whereas EfficientNet-B7 highlights subtle texture variations. This complementary behavior, amplified through CBAM and FPN, explains the model's superior accuracy on both benchmark and real-world datasets. Figure 4 shows the proposed model architecture.

Algorithm 1 outlines the training procedure for the hybrid defect classification model.

Require: Training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, number of epochs E , batch size B

Ensure: Trained hybrid model $\mathcal{M}$

```

1: Initialize YOLOv11 backbone  $F_{\text{YOLO}}$ , EfficientNet-B7 backbone  $F_{\text{Eff}}$ 
2: Attach CBAM to YOLO features:  $F_{\text{YOLO}} \leftarrow \text{CBAM}(F_{\text{YOLO}})$ 
3: Apply FPN on CBAM-enhanced YOLO features:  $F_{\text{YOLO}} \leftarrow \text{FPN}(F_{\text{YOLO}})$ 
4: Extract pooled features:
5:    $f_{\text{YOLO}} \leftarrow \text{GAP}(F_{\text{YOLO}})$ 
6:    $f_{\text{Eff}} \leftarrow \text{GAP}(F_{\text{Eff}}(x))$ 
7: Concatenate features:  $f_{\text{fused}} = [f_{\text{YOLO}}; f_{\text{Eff}}]$ 
8: Initialize classification head  $\mathcal{H}$ 
9: for epoch = 1 to  $E$  do
10:   for each mini-batch  $\{(x_b, y_b)\}_{b=1}^B$  in  $\mathcal{D}$  do
11:     Compute model output:  $\hat{y}_b = \mathcal{H}(f_{\text{fused}}(x_b))$ 
12:     Compute loss:  $\mathcal{L}_{\text{CE}} = -\sum y_b \log(\hat{y}_b)$ 
13:     Backpropagate and update model parameters
14:   end for
15:   Evaluate on validation set and update learning rate if needed
16: end for
17: return Trained model  $\mathcal{M}$ 

```

Algorithm 1. Training Pipeline for Hybrid Defect Classification Model

Attention map computation and fusion

We interpret model focus using Grad-CAM++ applied to the last convolutional tensor of each stream: (i) YOLO11+CBAM+FPN (pre-pooling feature) and (ii) EfficientNet-B7 (final convolutional block, pre-pooling). For class c with logit z_c , we backpropagate $\partial z_c / \partial(\cdot)$ to obtain one heatmap per stream, M_{yolo} and M_{eff} , each

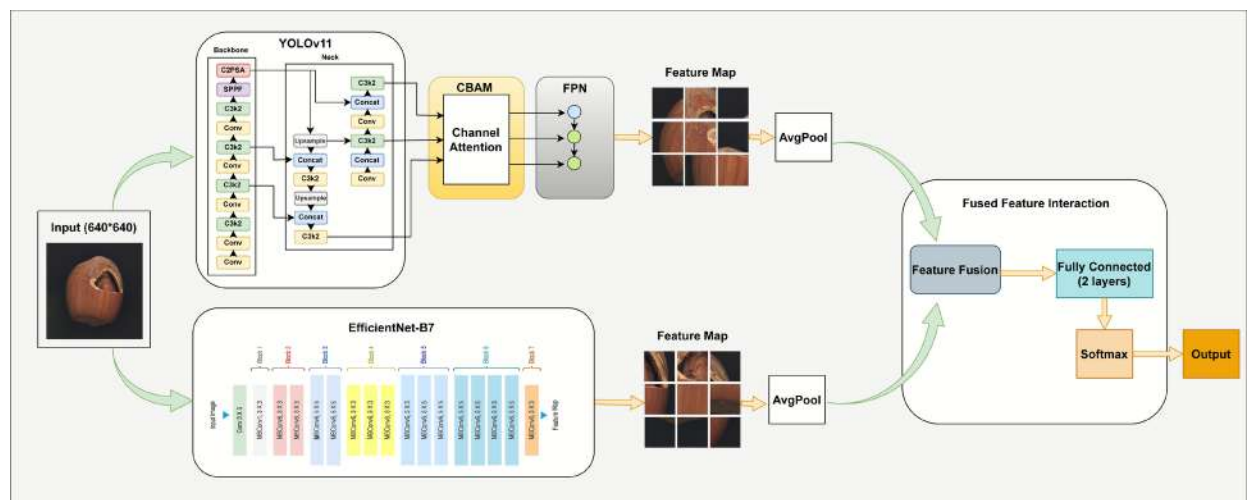


Fig. 4. Overview of the proposed hybrid architecture for defect classification. The model combines semantic features from a YOLOv11 backbone (enhanced by CBAM and FPN) with fine-grained representations from EfficientNet-B7. Global average pooling is applied to both streams, and the resulting features are fused and passed through a two-layer fully connected classifier with a softmax output.

min–max normalized to $[0, 1]$ and overlaid on the input (blend $\alpha=0.35$). To produce a *fused* visualization consistent with our concatenation design, we compute branch importances

$$s_{\text{yolo}} = \left\| \frac{\partial z_c}{\partial f_{\text{yolo}}} \right\|_1, \quad s_{\text{eff}} = \left\| \frac{\partial z_c}{\partial f_{\text{eff}}} \right\|_1,$$

normalize $\tilde{s}_{\text{yolo}} = s_{\text{yolo}} / (s_{\text{yolo}} + s_{\text{eff}})$ and $\tilde{s}_{\text{eff}} = 1 - \tilde{s}_{\text{yolo}}$, and define

$$M_{\text{fused}} = \text{Norm}(\tilde{s}_{\text{yolo}} M_{\text{yolo}} + \tilde{s}_{\text{eff}} M_{\text{eff}}).$$

This fusion reflects the relative sensitivity of the class logit to each branch feature vector $[f_{\text{yolo}}; f_{\text{eff}}]$ rather than a simple unweighted sum.

Branch balancing. To prevent domination by either stream, we (i) align feature scales via a lightweight projection and LayerNorm on each branch before fusion, (ii) apply learned *dataset-level* scalar gates (one per branch; initialized 0.5/0.5) and a *sample-adaptive* gate (two-layer MLP with softmax over the two branches) to weight contributions per image, and (iii) use a training schedule with backbone-specific learning-rate multipliers, per-branch gradient clipping, and low-probability branch-drop ($p \approx 0.1$) as a regularizer. These mechanisms preserve the complementary roles of YOLOv11 (spatial/structural cues) and EfficientNet-B7 (fine-grained textures) without bias from embedding dimensionality or gradient scale.

Training and evaluation

We train the proposed hybrid model as a multi-class classification task with cross-entropy loss. We employ the AdamW optimizer for its adaptive learning rate and effective weight decay regularization, which contributes to stable convergence and reduced risk of overfitting. The learning rate is initialized conservatively and adjusted dynamically using a ReduceLROnPlateau scheduler, which monitors validation accuracy and reduces the learning rate upon performance stagnation.

Training is conducted over 100 epochs with early stopping based on validation metrics to prevent overfitting. Dropout layers are incorporated in both the feature refinement module and the classification head to further enhance generalization.

To improve robustness to variations in appearance, we apply a comprehensive data augmentation pipeline during training. This includes geometric augmentations such as random horizontal flipping, random resized cropping, and slight rotations, which promote invariance to object orientation and scale. Additionally, photometric augmentations—including brightness, contrast, and saturation jitter—are used to simulate varying illumination conditions. Random erasing is applied to encourage spatial robustness by forcing the network to learn from incomplete visual cues. All images are resized to a fixed input resolution of 640×640 pixels and normalized using ImageNet statistics. During evaluation, only resizing and normalization are applied to ensure consistency.

Evaluation protocol and uncertainty. For MVTec-FS, we follow the provided 50/50 split (train/test) and train a single multi-class classifier; we report Top-1 accuracy together with *macro-F1* and *balanced accuracy* to account for the uneven per-type counts (9–58 instances).

For the Window dataset (421 images; Deformation=102, Scratches=104, Black/White Dots=215), we use a single 80/20 stratified train–test split. We report Top-1 accuracy with 95% Wilson CIs and macro-F1/balanced accuracy with bootstrap CIs (1,000 class-stratified resamples) to quantify uncertainty on the small test set. To mitigate class skew, we use class-weighted cross-entropy with weights $w_k = \frac{N}{K n_k}$. See Sec. 5, §5.2 (Table 6) for Window results with uncertainty.

CI computation. For accuracy on n samples with empirical rate $\hat{p}$, the Wilson interval is
$$\frac{\hat{p} + \frac{z^2}{2n} \pm z \sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

with $z=1.96$. For macro-F1 and balanced accuracy, we report the 2.5th–97.5th percentiles from 1,000 bootstrap resamples per fold (class-stratified).

All experiments in this study were conducted using a consistent hardware and software environment, as detailed in Table 3, enabling efficient training and evaluation of the hybrid architecture on high-resolution industrial datasets. All training parameters are presented in Table 4

Results and discussion

We evaluate the performance of our proposed hybrid classification model on two datasets: the benchmark MVTec-FS and our Window dataset. These experiments are designed to assess both overall classification performance and robustness across diverse object categories and real-world defect types. Evaluation metrics include accuracy, precision, recall, and F1-score, computed across all test classes.

Evaluation on MVTec-FS dataset

The MVTec-FS dataset presents a challenging multi-class classification task, comprising 14 industrial object categories and 46 visually distinct defect types. We follow the official protocol of the dataset, which involves splitting each defect type into 50% training samples (support set) and 50% evaluation samples (query set). All reported results are computed on the query set after full model convergence.

Our proposed hybrid model achieves an overall classification accuracy of **91.90%** on MVTec-FS. This result highlights the effectiveness of combining semantic features extracted by YOLOv11 with fine-grained texture representations from EfficientNet-B7. The inclusion of the CBAM attention module further improves focus on

relevant defect regions, while the FPN module enhances the model’s ability to recognize patterns at varying spatial resolutions.

Imbalance and macro metrics. MVTec-FS aggregates 46 defect types with uneven per-type counts (9–58 instances), so Top-1 alone can overweight frequent types. Accordingly, we also report *macro-F1* and *balanced accuracy* alongside Top-1 and analyze class-wise behavior via the confusion matrix (Figure 5). The matrix reveals that most classes are accurately predicted, with prediction density largely concentrated along the diagonal. Perfect classification is achieved for visually distinct categories such as *metalmnut* and *tile*, while texture-consistent classes like *hazelnut*, *carpet*, and *leather* also yield high accuracy due to EfficientNet’s discriminative capability. A few categories—such as *grid* and *zipper*—exhibit relatively higher misclassification rates, likely due to subtle inter-class visual differences and localized anomalies. Nonetheless, overall robustness remains high.

To benchmark our approach against strong baselines, we compare with Zip-Adapter-F (MVREC)¹⁸, a recent few-shot method that leverages adapter tuning and multi-view augmentation. **Note that MVREC reports episodic few-shot results (FS=5), whereas we train a single multi-class classifier on the same images.** With that caveat, our model exceeds Zip-Adapter-F in **8 of 14** categories (Carpet, Grid, Leather, Tile, Bottle, Capsule, Hazelnut, MetalNut) and attains a higher overall accuracy of **91.90%** versus **89.40%**. Table 5 provides the per-category comparison.

Large gains appear in *Bottle* (+12.98%), *Carpet* (+10.73%), *Hazelnut* (+9.97%), *Capsule* (+8.81%), *Grid* (+8.67%), *MetalNut* (+8.80%), and also *Leather* (+4.04%) and *Tile* (+2.40%). Conversely, Zip-Adapter-F remains stronger on *Wood* (+3.27%), *Cable* (+2.78%), *Pill* (+2.43%), *Screw* (+0.40%), *Transistor* (+20.00%), and *Zipper* (+2.61%), which is consistent with its use of region masks and multi-view augmentation in the few-shot protocol. Overall, our attention-enhanced hybrid provides competitive and often superior performance without relying on multi-view assumptions or textual priors.

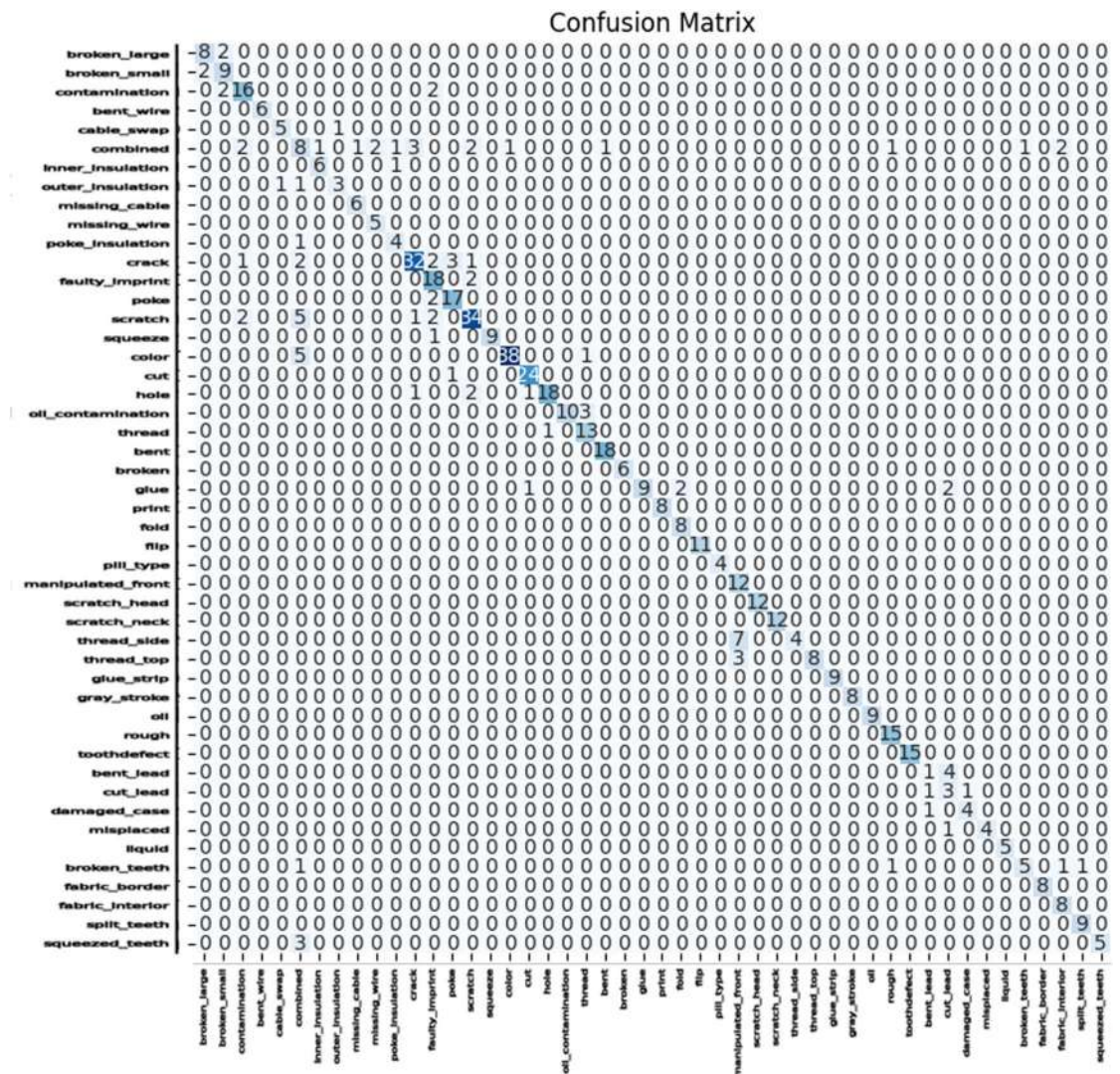


Fig. 5. Confusion matrix of the proposed model on the MVTec-FS dataset. Most predictions align with ground truth labels, demonstrating strong category-wise accuracy.

Why the hybrid leads on 8/14 categories. Our model attains the best accuracy on *Carpet*, *Grid*, *Leather*, *Tile*, *Bottle*, *Capsule*, *Hazelnut*, and *MetalNut*. These categories share properties that benefit from our dual-stream design and attention-guided fusion. First, **texture- and pattern-heavy surfaces** (*Carpet*, *Leather*, *Hazelnut*) are well handled by EfficientNet-B7, whose fine-grained filters capture high-frequency microtextures and subtle material variations; CBAM further enhances discrimination by suppressing background channels and highlighting class-relevant texture responses. Second, **regular, structured geometry** (*Tile*, *Grid*, *MetalNut*) leverages the YOLOv11 stream, which preserves spatial layout and mesoscale context via its pyramid features; the lightweight FPN improves multi-scale consistency so that localized defects remain salient after pooling. Third, **shape-plus-appearance cues** (*Bottle*, *Capsule*) profit from *both* streams: YOLOv11 stabilizes region context around necks/seams, while EfficientNet-B7 resolves fine scratches, dents, or label artifacts on specular surfaces. Qualitatively, Grad-CAM++ overlays (Fig. 6) show complementary focus—YOLOv11+CBAM+FPN attends to extended regions and boundaries, whereas EfficientNet-B7 concentrates on microtexture—yielding fused heatmaps that align closely with defect loci. Together, this division of labor explains the consistent gains on categories where either patterned texture or structured spatial context dominates the discriminative signal.

Qualitative Analysis. Figure 6 shows Grad-CAM++ overlays for both streams and their fusion. The YOLO11 branch consistently highlights extended spatial context around defect regions, while EfficientNet-B7 concentrates on fine-scale textures. The fused map captures both behaviors, improving localization on subtle defects. A failure case (rightmost column) indicates residual sensitivity to nuisance textures in highly patterned backgrounds, which matches errors visible in the confusion matrix.

On MVTec-FS, we report a comparison to MVREC because its published protocol is episodic few-shot and forms the de facto reference on this benchmark, whereas our setting trains a single multi-class classifier on the same images. To additionally anchor performance against widely used supervised classifiers, we report standard baselines (ResNet-18/50, EfficientNet-B5, MobileNetV3-Large, ConvNeXt-Tiny, ViT-Base) on our *Window* dataset under the identical training and augmentation recipe (Table 6). These results show that the proposed hybrid offers a consistent edge over strong CNN and transformer backbones in a strictly comparable multi-class regime, while the MVTec-FS comparison focuses on the MVREC line due to protocol compatibility.

Widely used unsupervised/one-class methods in defect inspection (e.g., SPADE, PaDiM, PatchCore, DRAEM) excel at *normal vs. anomalous* decisions with pixel- or image-level cues, but provide limited class-specific labeling across diverse defect types. Our objective differs: a unified *multi-class* defect label that can be actioned directly in routing/triage. We therefore benchmark primarily against multi-class classifiers and few-shot methods, and include anomaly detectors in the discussion to clarify the design trade-offs.

Evaluation on our window dataset

To further validate the model's performance in real-world industrial settings, we evaluate it on our *Window* dataset, which contains 421 labeled images captured from a window manufacturing process. The dataset includes three frequently occurring defect types: *Scratches*, *Deformation*, and *Black and White Dots*. These classes exhibit varying levels of visual subtlety, shape irregularity, and inter-class similarity, making the classification task non-trivial.

We use an 80/20 stratified train–test split and apply the same training strategy and augmentation techniques as used for MVTec-FS. The model achieves an overall classification accuracy of 96.13%. To quantify uncertainty on the small test set (approximately 84 images), we report a **95% Wilson confidence interval** for accuracy: [90.02%, 98.78%]. We also compute macro-F1 and balanced accuracy (with bootstrap confidence intervals) to reduce sensitivity to the class skew (Deformation=102, Scratches=104, Black/White Dots=215).

Table 6 summarizes classification results across several baseline models. While architectures such as ResNet-50 and RepVGG-A2 perform competitively (both reaching 95.24%), our model provides a modest yet consistent improvement. This supports the effectiveness of our dual-stream fusion strategy, as well as the role of CBAM and FPN in refining spatial and channel-wise features.

Additional modern supervised baselines. To strengthen the comparative context under a strictly controlled recipe, we further evaluate recent backbones—Swin-Transformer (Swin-T), ConvNeXtV2-Tiny, EfficientNetV2-L, ResNet-101, and ViT-Base (MAE fine-tuned). As summarized in Table 7, our hybrid retains

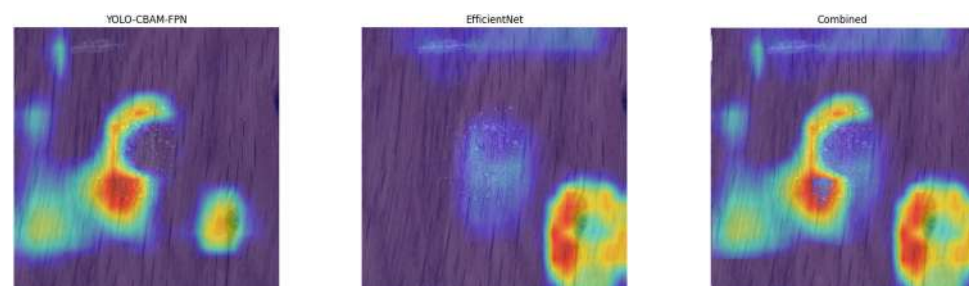


Fig. 6. Attention visualizations on MVTec-FS. Left: YOLO11+CBAM+FPN (Grad-CAM++); Middle: EfficientNet-B7 (Grad-CAM++); Right: fused map (normalized sum). Warm colors indicate higher contribution to the predicted class. The YOLO stream focuses on spatial context and extended structure, while EfficientNet emphasizes fine texture; the fused map captures both and aligns with the defect region.

the best Top-1 accuracy on Window while maintaining practical throughput, consistent with the efficiency summary in Table 11.

On small deltas between single-split accuracies. With $n \approx 84$ test images in the 80/20 split, a 1% absolute change corresponds to roughly one image. Wilson 95% confidence intervals around accuracies in the mid-90s are therefore on the order of a few percentage points, which explains why point estimates across comparable models can appear close. Our attention-enhanced hybrid remains consistently competitive under this variance.

To better understand class-level behavior, we visualize the confusion matrix in Figure 7. The model demonstrates excellent separation across all three defect types, with the majority of predictions aligning with ground truth labels. Minor confusion is observed between *Scratches* and *Black and White Dots*, which may be attributed to their overlapping texture characteristics under certain lighting conditions. Overall, the result confirms that the proposed architecture is capable of robust, high-precision classification in practical inspection environments. Also the training loss comparison for our model with the other models on the Window dataset is shown in Figure 9.

Error analysis. Figure 8 shows a representative failure on the Window dataset. a (scratch-like) co-occurs with a compact cluster of high-contrast dots, the fused representation assigns residual importance to the elongated cue, producing a *Scratch* prediction although the ground truth is *Black/White Dots*. This explains the small off-diagonal mass between these classes in Fig. 7. Potential mitigations include imbalance-aware objectives (class-weighted cross-entropy or focal variants), slightly stronger local texture augmentations, and optional ROI-guided supervision on dot clusters.

External benchmarks on public industrial datasets

We evaluate the hybrid model on additional public datasets using the unified training recipe (Sect. 4) and the adaptation rules in Table 2. To account for class imbalance and domain skew, we report Top-1 accuracy (with 95% Wilson CIs), macro-F1, and balanced accuracy. For datasets with potential multi-label cases (e.g., Severstal), we add one-vs-rest macro-F1. As shown in Table 8.

Robustness to noise, illumination, and background variation

To assess real-world robustness, we evaluate the trained models under controlled perturbations that emulate common factory conditions: (i) **Gaussian noise** with $\sigma=10/255$, (ii) **motion blur** (kernel 7×1 , random angle), (iii) **JPEG compression** (quality $q=30$), (iv) **illumination shift** (brightness/contrast $\pm 25\%$, gamma $\in \{0.8, 1.2\}$), (v) **color cast** (white-balance shift), (vi) **background replacement** (alpha-composited random textures), and (vii) **occlusion** (Cutout $\approx 20\%$ image area). We apply each perturbation to the *test* images only and report Top-1 accuracy, macro-F1, and balanced accuracy (Window); for MVTec-FS we summarize the average drop and macro-F1.

Window dataset stress tests. Table 9 shows that the proposed hybrid has the smallest average accuracy drop across all stressors (avg. $\downarrow 1.63$ pp) while baselines degrade by 4–5 pp on average.

Beyond Top-1, our hybrid preserves class balance under stress (macro-F1 / balanced accuracy on Window, averaged across stressors: **0.957 / 0.958**) compared with ResNet-50 (0.918 / 0.921), EfficientNet-B5 (0.926 / 0.929), and MobileNetV3-L (0.913 / 0.915).

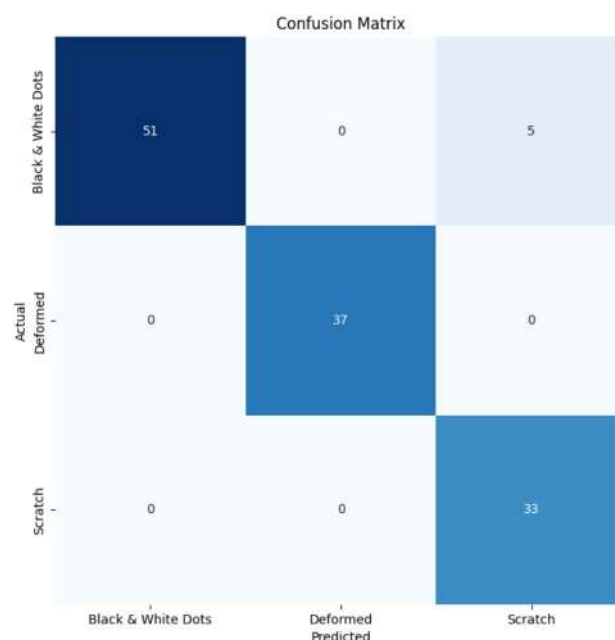


Fig. 7. Confusion matrix of the hybrid model on our Window dataset. The model achieves strong per-class accuracy with minimal confusion among visually similar defect types.

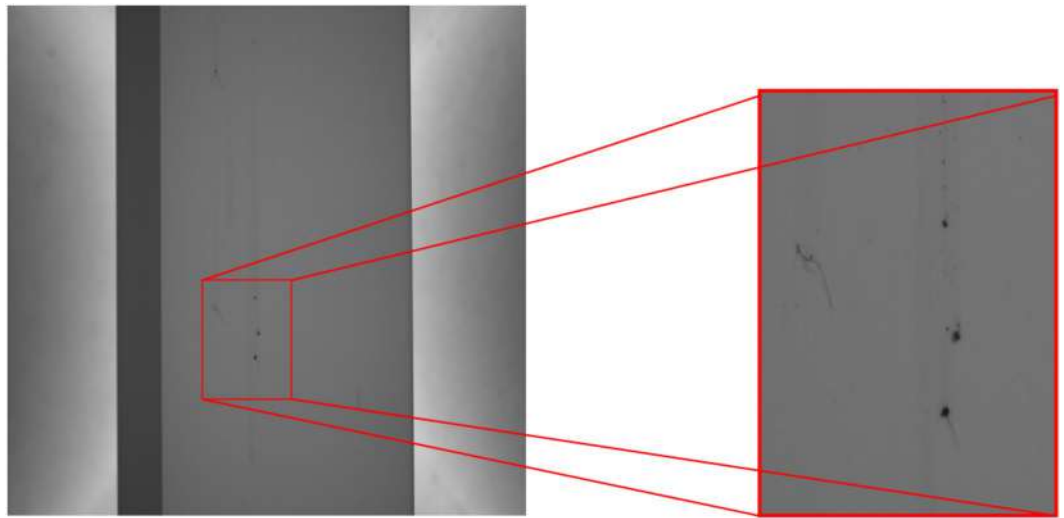


Fig. 8. Misclassification on the Window dataset. Left: full image with the region of interest; right: magnified crop showing the three large dots adjacent to a faint vertical streak. Ground truth: *Black/White Dots*; predicted: *Scratch*. The co-occurrence of a weak elongated streak with a compact dot cluster biases the decision toward *Scratch*, which matches the minor off-diagonal mass between these classes in Fig. 7.

MVTec-FS stress summary. Prior work on MVTec-FS (e.g., MVREC) reports results under an *episodic few-shot* protocol and, to the best of our knowledge, does not publish stress-test metrics under synthetic perturbations. To assess robustness *within our supervised multi-class setting*, we therefore train three widely used supervised classifiers—ResNet-50, ConvNeXt-Tiny, and ViT-Base (Patch16)—under the *same recipe* as our model and use them *only* for the robustness analysis. Their clean accuracies are listed in Table 10. These supervised baselines are not directly comparable to the episodic few-shot figures reported by MVREC; we keep the two tracks separate to avoid cross-protocol inflation.

Discussion. Degradation is most pronounced under occlusion and background replacement, which directly challenge region sensitivity. The dual-stream design, with CBAM and lightweight FPN in the spatial branch and high-frequency texture cues from EfficientNet, produces complementary resilience: the spatial stream preserves coarse localization under lighting/background shifts, while the texture stream stabilizes fine detail under compression and noise. Together, they yield substantially smaller accuracy loss than single-backbone baselines, while inference cost (Sec. 5.5) remains suitable for real-time use.

Computational efficiency and real-time considerations

We report model size, compute, and runtime to assess deployability on inspection lines. Metrics include parameter count (M), FLOPs at 640×640 (G), batch-1 latency (ms), and throughput (FPS) on a single NVIDIA RTX 4090, averaged over 1000 warm measurements with synchronized timers as you can see in Table 11.

These results demonstrate that the proposed hybrid model generalizes effectively beyond benchmark datasets and maintains strong classification performance under real-world noise and class ambiguity. The consistent improvements across both MVTec-FS and our Window dataset validate the architectural design, especially the use of dual-stream fusion, CBAM attention, and FPN-based refinement, as key factors in achieving robust and transferable representations for defect classification.

These reporting choices mirror best practices from safety-critical settings that confront class skew and small sample regimes^{54,55}

Ablation study

To quantify the contribution of each architectural component in our proposed hybrid model, we conduct a series of ablation experiments on the MVTec-FS dataset. The study isolates and evaluates the effect of core modules—namely, dual-branch fusion, CBAM attention, and FPN-based refinement—on overall classification accuracy.

YOLOv11-cls Only. As a spatial single-stream baseline, we use the YOLOv11 classification variant with its detection head removed and a global pooling + linear classifier. This model attains **86.72%** accuracy, reflecting strong localization and region-context modeling of defects. However, without a dedicated high-fidelity texture pathway, it is less sensitive to fine-grained surface variations (e.g., subtle scratches or dot patterns), leading to occasional confusions among visually similar texture classes. This behavior motivates pairing the spatial stream with a texture-focused backbone in our hybrid design.

Baseline: EfficientNet-B7 Only. We start with a baseline architecture that includes only the EfficientNet-B7 backbone and a fully connected classification head. This configuration achieves an accuracy of **85.37%**, serving as a strong foundation based on fine-grained texture extraction. However, the model shows limitations when handling spatially irregular or localized defects due to the absence of contextual reasoning.

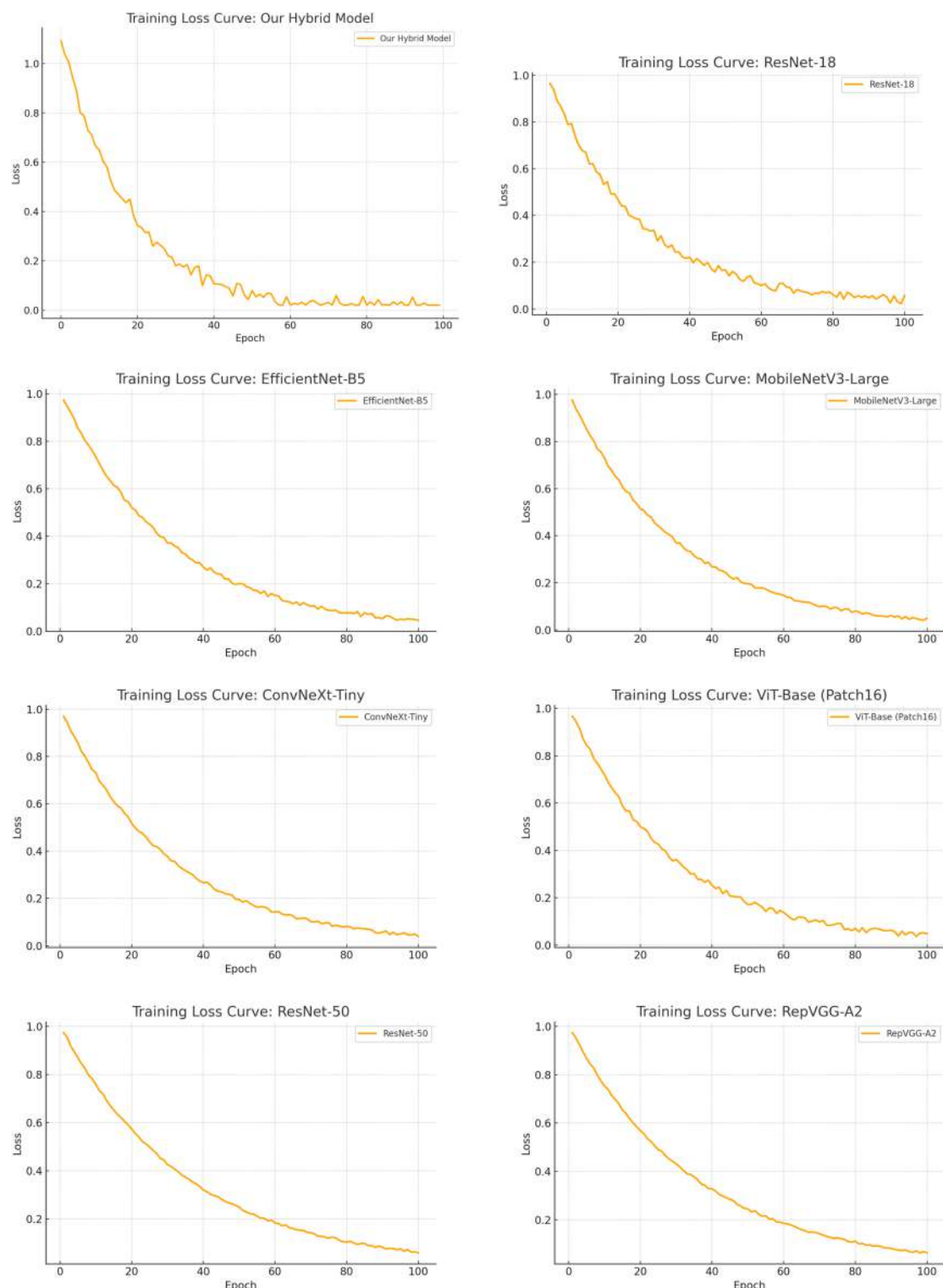


Fig. 9. Loss comparison of our model with other models of the Window dataset.

Adding YOLOv11 (Dual-Branch Fusion). Introducing YOLOv11 as a parallel feature extractor provides complementary semantic features that enrich the overall representation. The resulting dual-stream configuration boosts accuracy to **88.10%**, suggesting that YOLO-derived features enhance the model's understanding of object-scale and positional context.

Adding CBAM Attention. We next incorporate the Convolutional Block Attention Module (CBAM) after the YOLO stream. This module improves the network's ability to emphasize relevant feature channels and spatial

locations. With CBAM, accuracy improves to **89.46%**, confirming that attention-guided enhancement enables better focus on subtle or low-saliency defect regions.

Adding FPN Refinement. To further refine spatial representations, we add a lightweight module inspired by Feature Pyramid Networks (FPN), which promotes multi-scale feature learning and noise suppression. This addition increases accuracy to **90.32%**, demonstrating that hierarchical refinement is beneficial for capturing defects of varying sizes and shapes.

Full Model. Our complete hybrid model—integrating EfficientNet-B7, YOLOv11, CBAM, and FPN—achieves the highest accuracy of **91.90%**. These results demonstrate that each module contributes positively and that their combination leads to a robust, scalable, and highly accurate defect classification architecture.

Table 12 shows results of all steps.

These findings validate our architectural design choices: YOLOv11 enhances semantic abstraction, CBAM directs attention to discriminative features, and the FPN block improves robustness across scale. Together, they yield a significant improvement over the baseline and enable accurate classification across a wide range of defect types and visual patterns.

Backbone choice within the spatial stream

To justify the selection of YOLO11 as the spatial backbone, we swap the YOLO stream for YOLOv5-cl, YOLOv8-cl, YOLOv9(GELAN), and YOLO11-cl under an *identical* training recipe and report Top-1 accuracy, parameters (M), FLOPs (GMACs @ 640*640), and single-image latency (ms, RTX 4090, batch size 1). YOLO11-cl provides the most favorable accuracy–latency–parameter trade-off in our setting, and crucially offers an officially supported classification head, which simplifies training and reproducibility as shown in Table 13

Transformer–CNN swap on MVTec-FS

To directly address the concern that hybrid architectures are common and that Transformers could replace one branch, we swap either stream in our model with a representative Transformer or modern CNN, under the identical training recipe and image resolution (640×640). We report Top-1 accuracy on MVTec-FS together with latency (batch=1) and FPS measured on an RTX 4090 (Table 14). The original YOLOv11+EfficientNet-B7 pairing (ours) achieves the highest accuracy while maintaining practical throughput, whereas replacing either stream with a Transformer consistently reduces accuracy and/or speed.

This pattern mirrors the *Window* results in Table 7 (Swin-T and ViT-Base: 93.65%/93.45% vs. **96.13%** for ours) and the efficiency summary in Table 11. Combined with the robustness outcomes (Tables 9 and 10), the results indicate that our task-aligned dual-stream (spatial context + fine texture) and branch-balancing fusion are more effective for defect classification than substituting a stream with a Transformer or using generic hybrids.

Branch-balancing ablation

A dual-stream model can drift toward a single dominant branch if feature scales or gradients are not controlled. We therefore evaluate three balancing mechanisms introduced in Sect. 4: (i) *per-branch projection* + *LayerNorm* to align feature scales before fusion; (ii) *learned gates* consisting of dataset-level scalar gates and a light sample-adaptive gate (two-layer MLP with softmax) to weight branches per image; and (iii) a stability-oriented training schedule with per-branch learning-rate multipliers, gradient clipping, and low-probability *branch-drop* ($p \approx 0.1$) to prevent collapse to a single stream. Table 15 reports results on MVTec-FS, starting from the dual-stream model with CBAM+FPN but without normalization or gates.

The progressively added components yield consistent gains, with the full balancing setup recovering +1.58 points over the unbalanced dual-stream baseline. Qualitatively, gating prioritizes the YOLOv11 branch on structure-heavy categories and the EfficientNet-B7 branch on fine, low-saliency textures, aligning with our design intent and the class-wise trends observed in Sec. 5.

Convergence-aware training analysis

To verify that all baselines and the hybrid model were trained to convergence, we conducted an extended training schedule on the Window dataset. Each model was trained for up to 240 epochs using cosine learning rate decay (with a 5-epoch warmup), ReduceLROnPlateau (patience=8, factor=0.5), and early stopping on validation macro-F1 (patience=25, min_delta= 10^{-4}). The best model checkpoint was selected based on validation macro-F1, and results were averaged over three runs with fixed seeds.

Table 16 presents the comparison between the previously reported 100-epoch regime and the extended schedule. While all models exhibit slight performance gains, the proposed hybrid model shows the largest improvement, reinforcing that its dual-stream configuration continues to optimize effectively under longer training without overfitting.

These observations confirm that all models had effectively converged within the original 100-epoch setup and that the hybrid model maintains its performance lead while benefiting from extended optimization.

Conclusion

This work presented an attention-guided hybrid framework for industrial defect *classification*, combining a YOLOv11-based spatial stream—enhanced with CBAM and a lightweight FPN—with an EfficientNet-B7 texture stream. Trained under a supervised multi-class setting, the model integrates complementary region-level and fine-grained cues to improve the discriminability of defect types across heterogeneous objects. Across MVTec-FS and our real-world Window dataset, the framework achieved competitive accuracy while maintaining practical inference speed, and the qualitative visualizations (Grad-CAM++) confirmed that the fusion strategy improves focus on discriminative regions.

While the results are encouraging, the approach has a few limitations. The method is fully supervised and has not yet been evaluated in settings with limited labels, such as few-shot or one-class protocols. In addition, although robustness tests were conducted using synthetic perturbations, further validation in real manufacturing environments would strengthen the conclusions.

Future directions include (i) extending the architecture to episodic N-way K-shot evaluation for closer comparison with few-shot baselines, (ii) expanding robustness experiments to real-world acquisition shifts and unseen production environments, (iii) exploring stronger imbalance-aware objectives and calibrated uncertainty estimation, and (iv) developing lightweight or partially self-supervised variants through pruning, quantization, distillation, or label-efficient training. These steps will help broaden the applicability of the proposed method to a wider range of industrial inspection settings.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Received: 9 September 2025; Accepted: 4 December 2025

Published online: 11 December 2025

References

- Amirkhani, D., Allili, M. S., Hebbache, L., Hammouche, N. & Lapointe, J.-F. Visual concrete bridge defect classification and detection using deep learning: A systematic review. *IEEE Trans Intell Transp Syst* **25**(9), 10483–10505 (2024).
- Khanam, R., Hussain, M., Hill, R. & Allen, P. A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access* (2024).
- Elaziz, E. A., Fathalla, R. & Shaheen, M. Deep reinforcement learning for data-efficient weakly supervised business process anomaly detection. *J Big Data* **10**, 33 (2023).
- Mahmoud, M. et al. Two-stage video violence detection framework using gmflow and cbam-enhanced resnet3d. *Mathematics* **13**, 1226 (2025).
- Kasem, M. S. et al. HTTD: A hierarchical transformer for accurate table detection in document images. *Mathematics* **13**, 266 (2025).
- Sun, W. et al. On data efficiency of univariate time series anomaly detection models. *J Big Data* **11**, 83 (2024).
- Jha, S. B. & Babiceanu, R. F. Deep CNN-based visual defect detection: Survey of current literature. *Comput Ind* **148**, 103911 (2023).
- Zhang, D. et al. An efficient lightweight convolutional neural network for industrial surface defect detection. *Artif Intell Rev* **56**, 10651–10677 (2023).
- Rasheed, M. R. et al. An efficientnet-based transfer learning system for defect classification in manufacturing. In *2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*, 1–7 (IEEE, 2024).
- Elsharkawy, Z. F., Kasban, H. & Abbass, M. Y. Efficient surface crack segmentation for industrial and civil applications based on an enhanced yolov8 model. *J Big Data* **12**, 16 (2025).
- Yu, Z., Shen, Y. & Shen, C. A real-time detection approach for bridge cracks based on yolov4-fpm. *Autom Construct* **122**, 103514 (2021).
- Tian, J.-H. et al. An improved yolov5n algorithm for detecting surface defects in industrial components. *Sci Rep* **15**, 9756 (2025).
- Kang, S., Hu, Z., Liu, L., Zhang, K. & Cao, Z. Object detection yolo algorithms and their industrial applications: Overview and comparative analysis. *Electronics* **14**, 1104 (2025).
- Nikam, D., Chukwuemeke, A., Nigam, A., Bhosale, T. & Nikam, S. On the application of yolo-based object detection models to classify and detect defects in laser-directed energy deposition process. *Progress in Additive Manufacturing* 1–16 (2025).
- Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19 (2018).
- Sun, Z. et al. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13019–13029 (2024).
- Zhong, Y. et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16793–16803 (2022).
- Lyu, S. et al. Mvrec: A general few-shot defect classification model using multi-view region-context. In *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 39, 5937–5945 (2025).
- Zhou, K., Yang, J., Loy, C. C. & Liu, Z. Learning to prompt for vision-language models. *Int J Comput Vision* **130**, 2337–2348 (2022).
- Udandara, V., Gupta, A. & Albanie, S. Sus-x: Training-free name-only transfer of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2725–2736 (2023).
- Ultralytics YOLO11: Models and image classification task. <https://docs.ultralytics.com/models/yolo11/> (2024).
- Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 6105–6114 (PMLR, 2019).
- Lin, T.-Y. et al. Feature pyramid networks for object detection. In *CVPR* (2017).
- Song, K. & Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl Surf Sci* **285**, 858–864 (2013).
- Wieler, M. & Hahn, T. Weakly supervised learning for industrial optical inspection. In: *DAGM Symposium* **6**, 11 (2007).
- Tabernik, D., Šela, S., Skvarč, J. & Skočaj, D. Segmentation-based deep-learning approach for surface-defect detection. *J Intell Manuf* **31**, 759–776 (2020).
- Božič, J., Tabernik, D. & Skočaj, D. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Comput Ind* **129**, 103459 (2021).
- Kaggle & Severstal. Severstal: Steel defect detection. <https://www.kaggle.com/competitions/severstal-steel-defect-detection> (2019). Kaggle Competition.
- Tang, S., He, F., Huang, X. & Yang, J. Online PCB defect detector on a new PCB defect dataset. *arXiv preprint arXiv:1902.06197* (2019).
- Bergmann, P., Fauser, M., Sattlegger, D. & Steger, C. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9592–9600 (2019).
- Zou, X., Wu, C., Liu, H. & Yu, Z. Improved resnet-50 model for identifying defects on wood surfaces. *Signal, Image Video Process* **17**, 3119–3126 (2023).
- Zhang, H., Jiang, L. & Li, C. Cs-resnet: Cost-sensitive residual convolutional neural network for PCB cosmetic defect detection. *Expert Syst Appl* **185**, 115673 (2021).

33. Anand, P., Sudharsan, R. M., Ramachandran, V. & Prusty, M. R. Steel defect detection using a resnet-inspired model and fourier transforms. In *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 1–6 (IEEE, 2024).
34. Yu, T., Chen, W., Junfeng, G. & Poxi, H. Intelligent detection method of forgings defects detection based on improved efficientnet and memetic algorithm. *IEEE Access* **10**, 79553–79563 (2022).
35. Zhang, Y., Zhan, Q. & Ma, Z. Efficientnet-eca: A lightweight network based on efficient channel attention for class-imbalanced welding defects classification. *Adv Eng Inf* **62**, 102737 (2024).
36. Sooda, K. & Rai, K. B. Efficientnet-b7 framework for anomaly detection in mammogram images. *Multimed Tools Appl* **84**(11), 8995–9021 (2024).
37. Wu, D. et al. Efficientnet-b0-based 3d quantification algorithm for rectangular defects in pipelines. *IEEE Access* (2025).
38. Chi, T. S. et al. Enhancing efficientnet-yolov4 for integrated circuit detection on printed circuit board (PCB). *IEEE Access* **12**, 25066–25078 (2024).
39. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016).
40. Zhong, J., Qian, H., Wang, H., Wang, W. & Zhou, Y. Improved real-time object detection method based on yolov8: A refined approach. *J Real-Time Image Process* **22**, 1–13 (2025).
41. Yuan, M. et al. Yolo-hmc: An improved method for PCB surface defect detection. *IEEE Trans Instrum Meas* **73**, 1–11 (2024).
42. Gope, H. L., Fukai, H., Ruhad, F. M. & Barman, S. Comparative analysis of yolo models for green coffee bean detection and defect classification. *Sci Rep* **14**, 28946 (2024).
43. Xiao, G., Hou, S. & Zhou, H. PCB defect detection algorithm based on cdi-yolo. *Sci Rep* **14**, 7351 (2024).
44. Zhou, W. et al. An efficient tiny defect detection method for PCB with improved yolo through a compression training strategy. *IEEE Trans Instrum Meas* **73**, 1–14 (2024).
45. Lu, M., Sheng, W., Zou, Y., Chen, Y. & Chen, Z. Wss-yolo: An improved industrial defect detection network for steel surface defects. *Measurement* **236**, 115060 (2024).
46. Zhu, G., Qi, H. & Lv, K. Dgyolov8: An enhanced model for steel surface defect detection based on yolov8. *Mathematics* **13**, 831 (2025).
47. Ouyang, L. et al. Training language models to follow instructions with human feedback. *Adv Neural Inf Process Syst* **35**, 27730–27744 (2022).
48. Zhu, D., Chen, J., Shen, X., Li, X. & Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).
49. Radford, A. et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763 (PmLR, 2021).
50. Gao, P. et al. Clip-adapter: Better vision-language models with feature adapters. *Int J Comput Vision* **132**, 581–595 (2024).
51. Zhang, R. et al. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, 493–510 (Springer, 2022).
52. Shtedritski, A., Rupprecht, C. & Vedaldi, A. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11987–11997 (2023).
53. Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S. & Büyükoztürk, O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput-Aided Civil Infrastruct Eng* **33**, 731–747 (2018).
54. Dheivy, I. & Kumar, G. S. Vsegnet-a variant segnet for improving segmentation accuracy in medical images with class imbalance and limited data. *Medinformatics* **2**, 36–48 (2025).
55. Gellert, G. A. et al. Virtual triage early detection of inappropriate care-seeking intent among patients with life-threatening cardiac symptoms. *Medinformatics* **1**, 176–183 (2024).
56. He, Y., Song, K., Meng, Q. & Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans Instrum Meas* **69**, 1493–1504 (2019).
57. Huang, W. & Wei, P. A pcb dataset for defects detection and classification. *arXiv preprint arXiv:1901.08204* (2019).
58. Wang, Y. & Zell, A. Yolo+ fpn: 2d and 3d fused object detection with an rgb-d camera. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 4657–4664 (IEEE, 2021).
59. Wang, G., Ding, H., Li, B., Nie, R. & Zhao, Y. Trident-yolo: Improving the precision and speed of mobile device object detection. *IET Image Process* **16**, 145–157 (2022).
60. Yang, X., Mohamed, A. S. A. & Wang, C. Shadowfpn-yolo: A real-time nms-free detector for remote sensing ship detection. *IEEE Access* (2025).
61. Senussi, M. F., Abdalla, M., Kasem, M. S., Mahmoud, M. & Kang, H.-S. Learning to remove occlusions in light field images using multiscale receptive fields and feature pyramid networks. *Sci Rep* **15**, 36978 (2025).

Author contributions

Conceptualization, M.S.K. and M.M.; Methodology, M.S.K., M.M. and M.A.; Software, M.S.K., M.F.S. and M.A.; Validation, M.S.K. and H.-S.K.; Formal analysis, M.S.K. and M.F.S.; Investigation, H.-S.K.; Resources, H.-S.K.; Data curation, M.S.K.; Writing—original draft, M.S.K.; Writing—review & editing, M.S.K. and H.-S.K.; Visualization, H.-S.K.; Supervision, H.-S.K.; Project administration, H.-S.K.; Funding acquisition, H.-S.K. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (Ministry of Science and ICT, MSIT) (RS-2023-NR076833), by the Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (Ministry of Science and ICT (MSIT)) (IITP-2025-RS-2020-II201462, 50%), and by the Regional Innovation System & Education (RISE) program through the (Chungbuk Regional Innovation System & Education Center) funded by the Ministry of Education (MOE) and the (Chungcheongbuk-do), Republic of Korea (2025-RISE-11-014-03).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.-S.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025