2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA) | 979-8-3315-5383-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/ETFA65518.2025.11205693

A Comparative Study of Attention-Augmented YOLO Architectures for Defect Detection in Fused Deposition Modelling

Hasan Cezayirli
İzmir Institute of Technology
Mechanical Engineering
İzmir, Türkiye
hasancezayirli@iyte.edu.tr

Halil Tetik
İzmir Institute of Technology
Mechanical Engineering
İzmir, Türkiye
haliltetik@iyte.edu.tr

Mehmet İsmet Can Dede
İzmir Institute of Technology
Mechanical Engineering
İzmir, Türkiye
candede@iyte.edu.tr

Wai Lwin Phone
London South Bank University
Computer Science and Digital Technologies
London, UK
s4220393@lsbu.ac.uk

Bugra Alkan
London South Bank University
Computer Science and Digital Technologies
London, UK
alkanb@lsbu.ac.uk

***Abstract*—Additive manufacturing (AM), particularly fused deposition modelling (FDM), facilitates the fabrication of complex geometries with increasing flexibility and efficiency. Ensuring consistent print quality in FDM processes necessitates the development of accurate defect detection mechanisms. Attention-augmented YOLO (You Only Look Once) models have emerged as a promising solution for addressing this challenge. In this study, we systematically benchmark and evaluate the performance of YOLO architectures enhanced with attention mechanisms within the context of FDM 3D printing applications. The models were trained and evaluated using representative defect datasets. The attention-augmented models demonstrate improved detection performance.**

***Index Terms*—Fused Deposition Modeling, Additive Manufacturing, Defect Detection, Artificial Intelligence, Machine Vision, YOLO, Attention Mechanisms**

I. Introduction

Fused Deposition Modelling (FDM), which is one of the most versatile Additive Manufacturing (AM) methods for 3D printing of thermoplastic polymers such as Acrylonitrile Butadiene Styrene (ABS) and Polylactic Acid (PLA), enables the creation of complex geometries from digital designs in a cost-effective manner with shorter cycle times [1, 2]. This approach employs heat to soften thermoplastic filaments extruded by the printhead which follows the path generated to create the cross-sectional geometry of the part to be fabricated so that 3D parts can be created from CAD models in a layer-by-layer approach [3]. Despite its many advantages such as fast production, cost efficiency, ease of access, broad material adaptation, and the ability to produce complex components, FDM is susceptible to defects such as warping, stringing, layer shifting, and extrusion inconsistencies, which compromise print quality and increase material waste [4, 5]. Traditional defect detection methods, including manual inspection and basic sensor systems, are inefficient and lack the precision required for large-scale production [6]. This has prompted the development of automated, vision-based solutions leveraging machine vision and deep learning, particularly Convolutional Neural Networks (CNNs) such as You Look Only Once (YOLO) [7].

The evolution from YOLOv1 to YOLOv11 reflects significant performance enhancements [7]. For instance, Sani *et al.* improved YOLOv4 for flow defect detection in FDM and compared YOLOv3 and YOLOv4 variants for anomaly detection, emphasizing the balance between speed and precision [8]. They also evaluated advanced YOLOv11 models for defect detection in FDM workflows, benchmarking their performance against other YOLO variants. The YOLOv11s model achieved a mean Average Precision (mAP) of 0.8308 at IoU threshold 0.5 (mAP@0.5) and 0.5361 across IoU thresholds ranging from 0.5 to 0.95 (mAP@0.5:0.95).

Attention mechanisms enhance YOLO models by improving feature focus and reducing false positives. An attention-enhanced YOLOv8 model integrating Multi-Headed Self-Attention (MHSA) and Convolutional Block Attention Module (CBAM) achieved a 92.1% mAP in extrusion defect detection for large-scale FDM printing [8]. YOLOv11 further integrates advanced attention mechanisms to enhance feature representation and detection accuracy, primarily through its C2PSA (Convolutional block with Parallel Spatial Attention) module and Partial Self-Attention (PSA) [9]. The C2PSA mechanism combines parallel convolutional pathways with spatial attention, allowing the model to dynamically focus

The present work was conducted in the framework of the TWIN-IT-ROMANS project (https://twin-it-romans.iyte.edu.tr/), titled "Twinning IzTech in Robotics Manufacturing Systems," which has received funding from the European Union's Horizon Europe Framework Programme HORIZON-WIDERA-2023-ACCESS-02 (Twinning Bottom-Up) under Grant Agreement No 101160215.

on critical regions of an image, such as small or occluded objects, by weighting feature maps based on spatial importance. Additionally, YOLOv11 employs Partial Self-Attention to selectively apply self-attention to specific regions [10].

Although attention mechanisms have been explored in previous studies, the systematic benchmarking of their performance on real-world industrial datasets has received limited attention. In response, this study aims to analyse the effectiveness of attention-augmented YOLO models through a structured benchmarking approach. Five distinct attention mechanisms were carefully integrated into the YOLOv11 architecture and evaluated against a benchmark dataset, with YOLOv11 and YOLOv8 models employed as baselines. Several key performance indicators were used to assess their efficacy. The defect types analysed include warping, spaghetti, cracking, stringing, layer shifting, and curling. The results demonstrated that attention-augmented YOLO models achieved improved performance across all selected defect types, thereby enhancing object detection outcomes without considerable degradation in computational efficiency.

II. Background

A. *Common FDM Defects and Detection Technologies*

Some commonly observed issues in the FDM process, such as thermal inconsistencies, inferior gear feeding, filament winding, nozzle blockage, or poor bed leveling, result in certain defects in the parts being 3D printed and/or even unsuccessful prints at all [1]. Frequently observed defects during FDM printing were recently listed and summarized by He *et al.* [11]. Among those, warping, cracking, stringing, layer-shifting and off-platform types of defects are most commonly observed during FDM process [12]. In warping the contact surface of the part to the build plate forms a curve due to internal stress created by uneven cooling of layers [13]. Cracking usually occurs due to weak inter-layer bonding and results in 3D printed parts with irreversible damage [11]. Stringing is the defect where an excess amount of filament is deposited and thin strands between the different sections of prints are created [14]. Layer-shifting is the incorrect positioning of a layer in x-y plane relative to the previously printed layers [15]. Off-platform occurs due to failed bonding between the first layer of the print and the build plate [12].

B. *YOLO*

YOLO, unlike traditional object detection models, employs a single-stage detection approach, enabling faster and more efficient real-time defect detection, thereby transforming the field of computer vision. On the other hand, like many other object detection algorithms, YOLO models can struggle with detecting relatively smaller objects in an image. YOLO models sometimes produce bounding boxes that include background pixels, affecting the accuracy of object localization [16]. YOLO divides an image into a grid and predicts bounding boxes and class probabilities directly for each grid cell in one pass through the network. It combines object localization and classification in one step [17]. YOLO architecture consists of three stages as shown in **Figure 1**. The backbone in YOLO architecture is responsible for feature extraction from the input image. It consists of convolutional layers that process the image to generate feature maps, capturing essential spatial and semantic information. The neck acts as an intermediary between the backbone and the head. Its function is to refine and enhance the features extracted by the backbone. The head is where YOLO performs its actual object detection tasks [16].

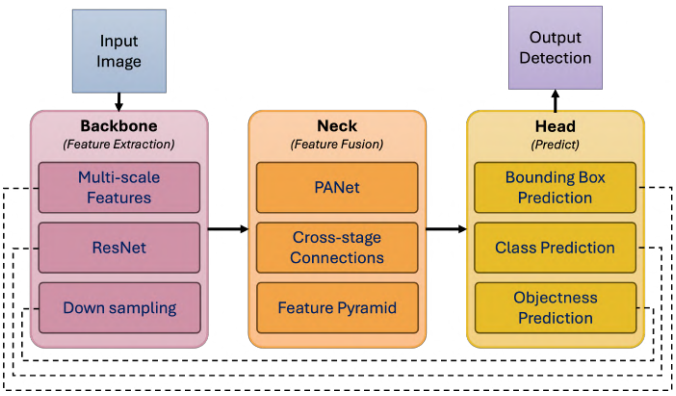

Fig. 1. YOLO architecture with labelled components: backbone for feature extraction, neck for feature refinement, and head for object detection tasks.

C. *Attention Mechanisms*

Attention mechanisms in CNNs are techniques that help the network focus on more informative features in the spatial dimension. They improve CNN performance by strengthening channel-wise, spatial-wise, and domain attention [18].

*1) **Convolutional Block Attention Module**:* CBAM mechanism combines channel and spatial attention. Channel attention is applied through average-pooling and max-pooling followed by a shared MLP to generate channel-wise weights, while spatial attention applies pooling across the channel axis to create a spatial map highlighting key regions. CBAM enhances feature representations by suppressing background noise and emphasizing diagnostically significant structures [19].

*2) **Squeeze-and-Excitation**:* SE mechanism refines feature maps by sequentially applying squeeze (global average pooling), excitation (two-layer gating to model channel dependencies), and re-weighting (emphasizing discriminative features). This approach dynamically optimizes channel-wise feature importance [20].

*3) **Efficient Channel Attention**:* ECA mechanism enables efficient cross-channel interaction without reducing the data. This improves the model's ability to focus on the most relevant features in an image, such as distinguishing small or similar-looking pests for real-time detection tasks [21].

*4) **Normalisation-based Attention Module**:* NAM is generated by applying a spatial attention mechanism to a feature map. The NAM highlights which spatial regions in the feature map are considered most important by the model [22].

*5) **Coordinate Attention**:* CA embeds precise positional information into the attention process, allowing the network to better capture both "what" and "where" aspects of objects within an image [23].

III. Research Methodology

The primary aim of this study is to evaluate the performance advantages of attention-augmented YOLO models for detecting printing defects in industry-ready settings. To achieve this, a systematic methodology is adopted to assess the efficacy of integrating attention mechanisms into YOLOv11, with YOLOv11 and YOLOv8 serving as baseline models for comparison.

A. Image Annotation

The dataset, sourced from [12], comprises 1,912 images of various printing defects, categorised into five initial defect classes including warping, stringing, and layer shifting. As the original dataset was unannotated, all images were manually annotated using the MATLAB Image Labeler toolbox. Several refinements were introduced during the annotation process. Firstly, the 'off-platform' class was relabelled as 'spaghetti' to more accurately reflect the visual characteristics of that defect. Additionally, a new class, 'curling,' was incorporated into the taxonomy. Notably, many images exhibited multiple co-occurring defects. While each image retained its original class-based assignment, additional defects were also annotated. Bounding boxes were drawn to encompass all visible defects within each image, thereby enabling multi-label object detection.

To ensure the accuracy and consistency of the annotations, both the classification of defect types and the delineation of bounding boxes were independently reviewed and validated by domain experts. The annotated final dataset comprises six classes of 3D printing defects: warping, spaghetti, cracking, stringing, layer shifting, and curling as shown in **Figure 2**.

B. Data Preprocessing

Images were split into subsets using an 80/20 ratio for training and test, respectively. Then, to balance the class distribution among the training subset, targeted data augmentation was performed using geometric and photometric transformations such as flipping, rotation, scaling, affine shifts, and brightness/contrast adjustments [24]. Each class was augmented until it reached a label count similar to that of the most frequent class. However, since the 'warping' class also appeared in images primarily assigned to other defects, perfect balancing was not possible. In **Table I**, the defects and their counts before and after augmentation are presented.

TABLE I
Label Counts

Defect Label	Before Augmentation	After Augmentation
warping	780	1211
spaghetti	155	830
cracking	417	921
stringing	474	911
layer shifting	380	937
curling	301	898

Fig. 2. Final annotated images illustrating various defect types. From top to bottom, the primary class (as the original dataset) for each row is: cracking, layer shifting, warping, stringing, and spaghetti (off-platform).

C. Implemented Attention-Augmented YOLOv11 Models

In this study, five distinct attention-augmented YOLOv11 models were benchmarked. For all models, the input tensor was of the shape $[B, C, H, W]$. Where B is the batch size, representing the number of input samples processed simultaneously. C is the number of channels, corresponding to different feature maps. H is the height of each feature map. W is the width of each feature map.

All the attention blocks were inserted directly after their respective blocks without altering the subsequent tensor dimensions or disrupting the original YOLOv11 structure. Inserting attention blocks into the backbone affects the feeding of the head block because it alters the size and appearance of the backbone. Thus, all the attention-augmented architectures were modified to maintain the same structure as the baseline.

The architecture of the implemented YOLO-CBAM model, illustrated in **Figure 3**, follows the baseline head structure described above, with attention modules integrated into the backbone without altering tensor dimensions.

*1) **YOLO with Convolutional Block Attention Module (YOLO-CBAM)**:* The implemented YOLO-CBAM consists of two modules: a channel attention module and a spatial attention module. The channel attention module uses global average pooling to squeeze the spatial dimensions, reducing the input tensor to $[B, C, 1, 1]$. This pooled output is flattened and passed through two linear layers with a ReLU activation in between. The first linear layer reduces the channel dimensionality by a given reduction ratio, and the second restores it.

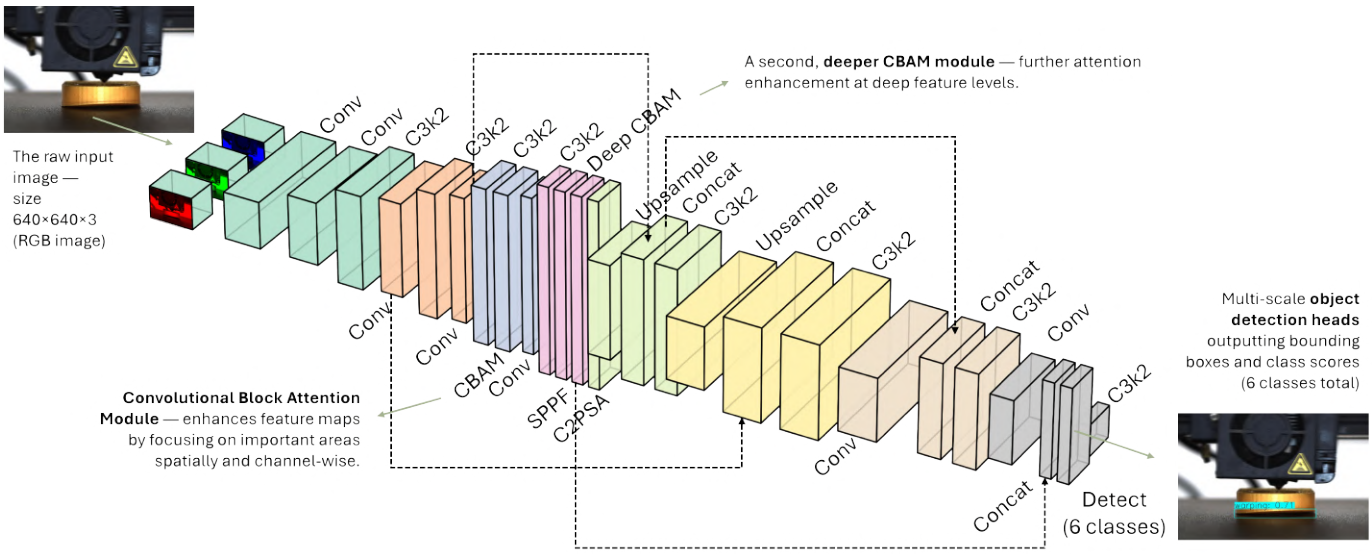

Fig. 3. YOLO-CBAM: Yolov11 with integrated CBAM architecture

The result is passed through a sigmoid function and reshaped back to $[B, C, 1, 1]$ to form a set of channel-wise attention weights. These weights are applied to the original input via element-wise multiplication, modulating the importance of each channel.

Next, the spatial attention module computes two spatial descriptors by performing average and max pooling operations along the channel axis, each producing a tensor of shape $[B, 1, H, W]$. These are concatenated along the channel dimension into a $[B, 2, H, W]$ tensor and passed through a convolutional layer, followed by padding. The result is passed through a sigmoid function to produce a spatial attention map of shape $[B, 1, H, W]$, which is then element-wise multiplied with the input from the channel attention stage. The final output has the same shape as the input.

CBAM blocks were integrated at two specific points within the backbone. A CBAM block was inserted after the second C3k2 block. Another CBAM block was inserted after the final C3k2 block before the SPPF block.

CBAM is introduced with the objective of enhancing feature representations by sequentially applying channel and spatial attention mechanisms. Channel attention aims to identify and amplify the most informative feature channels, while spatial attention aims to emphasize critical spatial regions within feature maps. By embedding these blocks after mid- and deep-stage C3k2 blocks, the design was intended to guide the network to prioritize both important features and important spatial locations before multi-scale aggregation.

*2) **YOLO with Squeeze-and-Excitation (SE) Block (YOLO-SE)**:* The SE block first applies a squeeze operation using global average pooling to the input tensor, reducing each channel to a scalar and producing a tensor of shape $[B, C, 1, 1]$. This is followed by an excitation operation, where the tensor passes through a lightweight two-layer bottleneck formed by two 1×1 convolutional layers. The first convolution reduces the number of channels by a given reduction ratio, followed by a ReLU activation, and the second convolution restores the number of channels to the original count. Then, the result is passed through a sigmoid function. Finally, the resulting attention tensor is multiplied element-wise with the original input tensor, producing an output tensor of the same shape as the input.

SE blocks were integrated at three points within the backbone. An SE block was inserted after the first C3k2 block. The second SE block was inserted after the second C3k2 block. The final SE block was inserted after the SPPF block.

SE is integrated with the aim of promoting channel-wise recalibration at different stages of feature extraction. By adaptively reweighting channel responses, the model is expected to suppress less informative features and highlight the most discriminative ones. This mechanism helps the network focus on useful channels across early, intermediate, and deep feature maps, thereby enriching representational capacity without modifying the spatial dimensions.

*3) **YOLO with Efficient Channel Attention (ECA) Block (YOLO-ECA)**:* The ECA block first applies a squeeze operation using global average pooling, producing a tensor of shape $[B, C, 1, 1]$. The tensor is then reshaped to $[B, 1, C]$ and passed through a one-dimensional convolution with a specified kernel size. The result is passed through a sigmoid function. Finally, the resulting attention tensor is reshaped back to $[B, C, 1, 1]$ and multiplied element-wise with the original input tensor, resulting in an output tensor of shape $[B, C, H, W]$.

The ECA blocks were integrated at three points within the backbone. The first ECA block was inserted after the first C3k2 block. The second ECA block was inserted after the second C3k2 block. The final ECA block was inserted after the C2PSA block.

ECA is integrated to encourage channel attention recalibration. Instead of introducing fully connected layers like SE, ECA employs a one-dimensional convolution to capture local cross-channel interactions efficiently. By embedding ECA at

multiple stages — early, mid, and after the C2PSA block — the design aims to progressively refine channel importance across different depths of the network.

*4) **YOLO with Coordinate Attention (CA) Block (YOLO-CA)**:* The CA block first applies a coordinate pooling operation: global average pooling is performed separately along the width and height axes, resulting in two tensors of shapes $[B, C, H, 1]$ and $[B, C, 1, W]$, respectively. The height-pooled tensor is concatenated with a transposed version of the width-pooled tensor to form a combined tensor of shape $[B, C, H + W, 1]$.

This combined tensor is passed through a shared bottleneck transformation consisting of a 1×1 convolution, batch normalisation, and a ReLU activation. The output is then split back into two branches corresponding to the height and width attention maps. Each branch passes through a separate 1×1 convolution to restore the original channel dimensions. Each branch is then passed through a sigmoid activation function. Finally, the original input tensor is element-wise multiplied with both the height and width attention maps. The resulting output tensor has the same shape as the input tensor.

CA blocks were integrated at two points within the backbone. The first CA block was integrated after the first C3k2 block. The second CA block was integrated after the SPPF block.

CA is integrated to encode positional information into the attention process, allowing the model to better localize salient features along spatial dimensions. By separately capturing width and height dependencies, the network aims to preserve fine-grained location information while performing channel recalibration. The integration after early and deep backbone modules was intended to enhance spatial sensitivity both at low- and high-level feature representations.

*5) **YOLO with normalisation-based Attention Module (NAM) (YOLO-NAM)**:* The NAM block first applies instance normalisation to normalise each feature map individually across spatial dimensions. The normalised tensor is then passed through a 1×1 convolution that reduces the number of channels from C to 1, producing a spatial attention map. The result is passed through a sigmoid function. Finally, the original input tensor is element-wise multiplied with the generated spatial attention mask, resulting in an output tensor of the same shape $[B, C, H, W]$.

NAM blocks were inserted at three points within the backbone. The first NAM block was integrated after the third C3k2 block. The second NAM block was integrated after the fourth C3k2 block. The final NAM block was integrated after the SPPF block.

NAM is implemented with the goal of promoting spatial feature normalisation and selective emphasis. By applying instance normalisation followed by a learned spatial mask, NAM was intended to filter and enhance spatially significant regions. Their sequential placement after mid and deep C3k2 modules and after the SPPF block was intended to progressively guide spatial focus as feature maps advanced toward the detection heads.

D. Key Performance Indicators

The trained model is evaluated using the test split of the dataset, which contains labelled data with predefined ground truth bounding boxes. After making predictions on the test set, the model's outputs are compared against the ground truth annotations. A prediction is considered a True Positive (TP) if it correctly matches a ground truth object. If the model makes an incorrect prediction—such as detecting an object that doesn't exist or misclassifying it—it is counted as a False Positive (FP). A True Negative (TN) occurs when the model correctly identifies that no object is present where there is no ground truth annotation. If the model fails to detect an object that is present in the ground truth, it is recorded as a False Negative (FN).

The proportion between TP detections and all detections (TP and FP) is defined as precision (P). Hence, the higher the precision, the better the prediction result. The proportion between TP detections and all defects (TP and FN) is defined as recall (R). Hence, the higher the recall, the better the prediction result.

$$P = \frac{TP}{TP + FP} \quad \text{and} \quad R = \frac{TP}{TP + FN} \tag{1}$$

F1-Score is defined as the harmonic mean of the precision and recall.

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2 \times P \times R}{P + R} \tag{2}$$

Intersection over Union (IoU) is an evaluation metric that measures the overlap between the prediction and the ground truth. mAP@0.5 is the mean Average Precision (mAP) across N classes when the IoU threshold is 0.5.

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^{N} AP(i) \tag{3}$$

mAP@0.5:0.95 is the mAP across N classes for IoU thresholds ranging from 0.5 to 0.95 in 0.05 increments.

$$mAP@0.5 : 0.95 = \frac{1}{10} \sum_{i=0.5}^{0.95} mAP@(i) \tag{4}$$

Specificity (S) is defined as the proportion of true negative predictions among all negative ground truth instances (TN + FP). Hence, the higher the specificity, the better the model is at avoiding false detections.

$$S = \frac{TN}{TN + FP} \tag{5}$$

IV. Benchmark Results and Discussions

The selected models were trained using the Ultralytics framework with custom YAML configurations [25]. Each model was trained individually within a consistent environment, applying fixed parameters across all experiments. Specifically, all models, both with and without attention mechanisms, were trained for 50 epochs with a batch size of 64,

TABLE II
EVALUATION RESULTS FOR BASELINES AND YOLOV11 MODELS INTEGRATED WITH ATTENTION MECHANISMS

	YOLOv8 Baseline	YOLOv11 Baseline	YOLO-CBAM	YOLO-CA	YOLO-SE	YOLO-ECA	YOLO-NAM
Precision (%)	93.29	93.68	95.54	96.01	97.03	**97.28**	96.22
Recall (%)	89.74	87.82	91.55	90.32	90.63	**92.03**	90.65
F1 Score (%)	91.42	90.57	93.41	93.04	93.60	**94.67**	93.31
mAP@0.5 (%)	93.15	93.13	94.44	94.01	94.73	**95.64**	94.60
mAP@0.5:0.95 (%)	**61.20**	59.14	60.47	60.05	60.16	60.34	59.56
Specificity (%)	**99.09**	98.86	99.03	98.94	98.84	98.92	98.81

using an input image size of 640 × 640 pixels. The AdamW optimizer was employed throughout the training process [26].

To minimize the impact of architectural variations and enable a fair comparison between attention-integrated models, 37 different architectures were created by systematically modifying the implementation locations of the attention blocks. In addition, targeted hyperparameter tuning was conducted, focusing solely on adjusting the kernel sizes and reduction ratios within the CBAM, SE, ECA, and CA modules. All other hyperparameters remained fixed across all experiments to ensure consistency.

The best results for ECA were achieved with a kernel size of 5. The best results for CA and SE were achieved with a reduction ratio of 16, and the best results for CBAM were achieved with a kernel size of 5 and a reduction ratio of 8. Following this process, the best-performing configuration for each attention mechanism was selected and subsequently used for benchmarking, thereby ensuring that performance comparisons reflected the optimized capabilities of each model rather than incidental architectural differences. In Section 3.C, all given implementation architectures represent the best variations identified during these iterations.

The models were trained on a workstation equipped with an Intel i7-14700K CPU, 64 GB of RAM, and an NVIDIA RTX 4070 Ti GPU. To ensure consistency across results, evaluation was performed using the default tools provided by the Ultralytics framework. Validation metrics, including precision, recall, F1 score, mean Average Precision at IoU threshold 0.5 (mAP@0.5), and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95), were reported individually for each class. Additionally, specificity per class was calculated from the confusion matrix, and Receiver Operating Characteristic (ROC) curves were generated based on the top-1 detection confidences for each class. The average detection time per image was also recorded using Python's timing libraries, providing an indication of the computational overhead introduced by each attention variant.

The training time for the baseline YOLOv8 was 12 minutes 52 seconds, and for the baseline YOLOv11 it was 13 minutes 20 seconds. The training times for CBAM, ECA, NAM, SE, and CA were 14 minutes 1 second, 14 minutes 6 seconds, 13 minutes 53 seconds, 14 minutes 15 seconds, and 14 minutes 7 seconds, respectively.

The average detection time per image for the baseline YOLOv8 was 0.0652 seconds, and for the baseline YOLOv11 it was 0.0654 seconds. The average detection times per image for CBAM, ECA, NAM, SE, and CA were 0.0664 second, 0.0663 seconds, 0.0666 seconds, 0.0678 seconds, and 0.0679 seconds, respectively.

The evaluation results presented in Table II demonstrate a clear performance improvement when attention mechanisms are integrated into the YOLOv11 architecture. Compared to the V8 and V11 baselines, all attention-augmented models achieved higher precision, recall, F1 scores, and mAP@0.5 values, indicating superior detection capability across classes. Notably, the ECA-enhanced model exhibited the highest precision (97.28%), recall (92.03%), and F1 score (94.67%), outperforming both the baseline models and other attention configurations. Similarly, the ECA-integrated model attained the highest mAP@0.5 value (95.64%), confirming its efficacy in accurately localizing objects across varying scales. Although the V8 baseline retained the highest mAP@0.5:0.95 score (61.20%), the marginal differences observed suggest that attention modules, while enhancing overall detection accuracy, may slightly compromise very fine-grained localization metrics. Specificity scores remained consistently high across all models, with minor variations, indicating that the introduction of attention mechanisms did not substantially affect the models' ability to correctly identify negative instances.

YOLOv11 introduces architectural enhancements over YOLOv8, including the use of C3k2 blocks and an attention-based C2PSA module. However, baseline V11 has slightly underperformed compared to baseline V8. This is likely due to reduced feature richness in intermediate layers. V8 uses consistent C2f blocks with balanced expansion ratios, maintaining higher internal feature width (e.g., 128 hidden channels at 256 output), while V11 uses a mix of C3k2 configurations—some relying on narrower bottlenecks (e.g., 64 hidden channels with 0.25 expansion). This may explain its slightly better performance in precision-sensitive and fine-grained localization tasks.

In examining the relative performance of the different attention mechanisms, it is evident that lightweight modules such as ECA and SE delivered notable gains with minimal computational overhead, compared to more complex modules like CBAM and CA. The ECA module, in particular, emerged as the most effective, striking an optimal balance between accuracy and model complexity. Conversely, the NAM-integrated model, while improving upon the baselines, exhibited slightly lower precision and mAP scores compared to ECA and SE, suggesting that the choice of attention mechanism significantly influences the detection performance. Overall, these results substantiate the hypothesis that incorporating attention mechanisms into the YOLOv11 backbone enhances feature

TABLE III
F1 SCORE COMPARISON ACROSS INDIVIDUAL DEFECT TYPES

Defect Type	YOLOv8 Baseline	YOLOv11 Baseline	YOLO-CBAM	YOLO-CA	YOLO-SE	YOLO-ECA	YOLO-NAM
Warping	91.37	91.10	91.03	88.99	90.20	**92.00**	91.05
Spaghetti	73.12	71.40	86.92	84.07	89.59	**92.88**	85.67
Cracking	**99.95**	99.96	99.18	99.90	99.80	98.88	99.85
Stringing	94.41	93.63	93.03	93.62	94.48	**94.42**	93.09
Layer Shifting	99.31	98.69	**99.41**	98.08	98.28	98.44	99.31
Curling	90.34	88.64	90.87	**93.58**	89.28	92.44	90.87
Mean	91.42	90.57	93.41	93.04	93.60	**94.67**	93.31

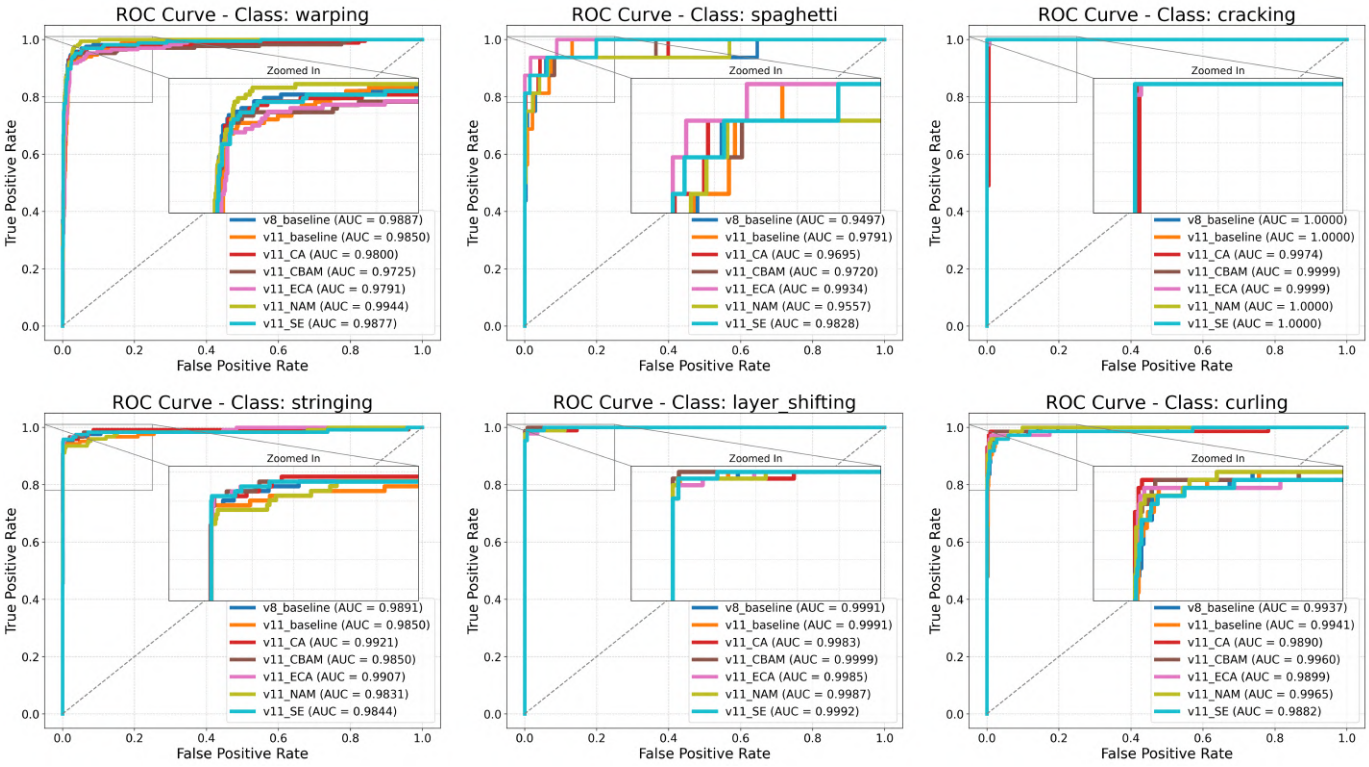

Fig. 4. ROC curves per defect class for baseline and attention-augmented YOLOv11 models.

representation, thereby leading to improved object detection outcomes without considerable degradation in computational efficiency.

Table III presents the comparative F1 scores achieved by the baseline models and attention-augmented YOLOv11 variants across individual defect types. The results demonstrate that the integration of attention mechanisms consistently improves detection performance, particularly for more challenging classes such as spaghetti and curling. For instance, the YOLOv11 model enhanced with ECA achieved the highest F1 score of 92.88% for the spaghetti class, indicating a substantial improvement over the baseline models. Similarly, improvements were observed across other defect types, with the CBAM-integrated model achieving the best performance for layer shifting, and the CA-augmented model excelling in the detection of curling defects. Although the cracking class already exhibited near-perfect scores across all models, attention modules further stabilised detection performance, ensuring consistently high F1 values.

The ROC-AUC curves presented in **Figure 4** provide a comparative evaluation of the models' discriminative capabilities across the individual defect classes. In general, all models exhibit high overall performance, with area under the curve (AUC) values exceeding 0.95 for the majority of classes, indicating excellent separability between defective and non-defective samples. Notably, the cracking and layer shifting classes consistently achieve near-perfect AUC scores (close to or exactly 1.0000) across all models, reflecting the relative ease of detecting these defects due to their distinct visual characteristics. Among the attention-augmented models, the YOLO-ECA variant demonstrated superior and consistently high AUC scores across all classes, particularly achieving an AUC of 0.9999 for the cracking class and 0.9997 for stringing, indicating exceptionally reliable classification performance. The NAM-integrated model also performed robustly, achieving the highest AUC for warping (0.9944) and maintaining strong scores across other classes. Importantly, the models integrating attention mechanisms notably improved the AUC for the more

challenging spaghetti and curling classes when compared to the baseline models.

Based on the results, a sample detection batch from the ECA-integrated YOLO model, which achieved the best performance, is presented in **Figure 5**. The images were taken from the test batch.

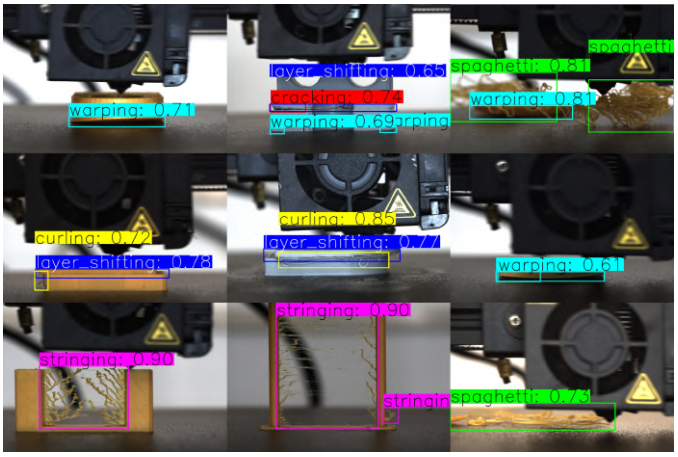

Fig. 5. Detected defects by YOLO-ECA model

V. Conclusion

This study presents a comprehensive framework for evaluating attention-augmented YOLO architectures in FDM 3D printing. The main contributions include benchmarking two baseline YOLO models (YOLOv8 and YOLOv11) and five attention-integrated YOLOv11 models (CBAM, SE, ECA, CA, NAM).

While attention mechanisms significantly improve overall detection accuracy, they can introduce additional computational overhead and may not uniformly benefit all defect types, particularly those requiring fine-grained localization. Future work could investigate hybrid attention mechanisms that combine the strengths of different modules, aiming to balance detection performance across diverse defect types in FDM 3D printing.

References

[1] M. Baechle-Clayton, E. Loos, M. Taheri, and H. Taheri, "Failures and flaws in fused deposition modeling (fdm) additively manufactured polymers and composites," *Journal of Composites Science*, vol. 6, no. 7, p. 202, 2022.

[2] M. F. Khan, A. Alam, M. A. Siddiqui, M. S. Alam, Y. Rafat, N. Salik, and I. Al-Saidan, "Real-time defect detection in 3d printing using machine learning," *Materials Today: Proceedings*, vol. 42, pp. 521–528, 2021.

[3] O. A. Mohamed, S. H. Masood, and J. L. Bhowmik, "Optimization of fused deposition modeling process parameters: a review of current research and future prospects," *Advances in manufacturing*, vol. 3, pp. 42–53, 2015.

[4] K. Rajan, M. Samykano, K. Kadirgama, W. S. W. Harun, and M. M. Rahman, "Fused deposition modeling: process, materials, parameters, properties, and applications," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 3, pp. 1531–1570, 2022.

[5] A.-M. Tălîngă, A. Hadăr, M.-V. Drăgoi, I. Nisipeanu, H. A. Ali, and C. P. Suciu, "Yolo-v8 in capturing imperfections generated by changing 3d printer parameters," *U.P.B. Sci. Bull. Series C*, vol. 86, no. 4, 2024.

[6] U. Delli and S. Chang, "Automated process monitoring in 3d printing using supervised machine learning," *Procedia Manufacturing*, vol. 26, pp. 865–870, 2018, 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.

[7] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023.

[8] A. R. Sani, A. Zolfagharian, and A. Z. Kouzani, "Automated defects detection in extrusion 3d printing using yolo models," *Journal of Intelligent Manufacturing*, pp. 1–21, 2024.

[9] A. F. Rasheed and M. Zarkoosh, "Yolov11 optimization for efficient resource utilization," *arXiv preprint arXiv:2412.14790*, 2024.

[10] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors," *arXiv preprint arXiv:2411.00201*, 2024.

[11] H. He, Z. Zhu, Y. Zhang, Z. Zhang, T. Famakinwa, and R. Yang, "Machine condition monitoring for defect detection in fused deposition modelling process: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 132, no. 7, pp. 3149–3178, 2024.

[12] W. Hu, C. Chen, S. Su, J. Zhang, and A. Zhu, "Real-time defect detection for fff 3d printing using lightweight model deployment," *The International Journal of Advanced Manufacturing Technology*, vol. 134, no. 9, pp. 4871–4885, 2024.

[13] K. Singh, "Experimental study to prevent the warping of 3d models in fused deposition modeling," *International Journal of Plastics Technology*, vol. 22, no. 1, pp. 177–184, 2018.

[14] O. Charia, H. Rajani, I. Ferrer Real, M. Domingo-Espin, and N. Gracias, "Real-time stringing detection for additive manufacturing," *Journal of Manufacturing and Materials Processing*, vol. 9, no. 3, p. 74, 2025.

[15] K. Erokhin, S. Naumov, and V. Ananikov, "Defects in 3d printing and strategies to enhance quality of fff additive manufacturing. a review," *ChemRxiv*, 2023.

[16] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using yolo: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023.

[17] A. Vijayakumar and S. Vairavasundaram, "Yolo-based object detection models: A review and its applications," *Multimedia Tools and Applications*, vol. 83, no. 35, pp. 83 535–83 574, 2024.

[18] W. Li, K. Liu, L. Zhang, and F. Cheng, "Object detection based on an adaptive attention mechanism," *Scientific Reports*, vol. 10, no. 1, p. 11307, 2020.

[19] J. Yan, Y. Zeng, J. Lin, Z. Pei, J. Fan, C. Fang, and Y. Cai, "Enhanced object detection in pediatric bronchoscopy images using yolo-based algorithms with cbam attention mechanism," *Heliyon*, vol. 10, no. 12, 2024.

[20] K. A. Gladis, J. B. Madavarapu, R. R. Kumar, and T. Sugashini, "In-out yolo glass: Indoor-outdoor object detection using adaptive spatial pooling squeeze and attention yolo network," *Biomedical Signal Processing and Control*, vol. 91, p. 105925, 2024.

[21] Z. Tang, J. Lu, Z. Chen, F. Qi, and L. Zhang, "Improved pest-yolo: Real-time pest detection based on efficient channel attention mechanism and transformer encoder," *Ecological Informatics*, vol. 78, p. 102340, 2023.

[22] H. Ahn, S. Son, J. Roh, H. Baek, S. Lee, Y. Chung, and D. Park, "Safp-yolo: Enhanced object detection speed using spatial attention-based filter pruning," *Applied Sciences*, vol. 13, no. 20, p. 11237, 2023.

[23] Y. Li, M. Zhang, C. Zhang, H. Liang, P. Li, and W. Zhang, "Yolo-ccs: Vehicle detection algorithm based on coordinate attention mechanism," *Digital Signal Processing*, vol. 153, p. 104632, 2024.

[24] C. Santos, M. Aguiar, D. Welfer, and B. Belloni, "A new approach for detecting fundus lesions using image processing and deep neural network architecture based on yolo model," *Sensors*, vol. 22, no. 17, p. 6441, 2022.

[25] R. G. Baldovino, A. J. P. Vidad, R. P. B. Abastillas, N. T. Bugtai, E. P. Dadios, R. R. P. Vicerra, A. A. Bandala, A. R. See, and N. R. Roxas Jr, "Comprehensive analysis on ultralytics-supported yolo models for detection and recognition of large office objects for indoor navigation," *Procedia Computer Science*, vol. 246, pp. 3851–3858, 2024.

[26] R. Llugsi, S. El Yacoubi, A. Fontaine, and P. Lupera, "Comparison between adam, adamax and adam w optimizers to implement a weather forecast based on neural networks for the andean city of quito," in *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*. IEEE, 2021, pp. 1–6.