Multi-Scale Performance Benchmarking of YOLO Models for Effervescent Tablet Defect Detection

Mustafa Yurdakul * and Ahmet Çakmak

Posted Date: 21 April 2026

doi: 10.20944/preprints202604.1475.v1

Keywords: effervescent tablets; defect detection; object detection; YOLO; deep learning; computer vision; pharmaceutical quality control

 doi:10.20944/preprints202604.1475.v1

Article

Multi-Scale Performance Benchmarking of YOLO Models for Effervescent Tablet Defect Detection

Mustafa Yurdakul * and Ahmet Çakmak

Kırıkkale University, Computer Engineering Dept, Kırıkkale, Türkiye

* Correspondence: mustafayurdakul@kku.edu.tr

Abstract

Effervescent tablets are highly hygroscopic solid dosage forms in which even minor surface defects can compromise product stability, dose uniformity, and patient safety. Reliable, high-throughput defect detection is therefore essential, yet the existing literature overwhelmingly focuses on compressed or film-coated tablets and rarely offers a systematic comparison across recent YOLO families and scales. This study presents a multi-scale performance benchmarking of three recent YOLO families—YOLO11, YOLO12, and YOLO26—on a newly constructed effervescent tablet defect dataset. The dataset comprises 251 high-resolution images acquired under controlled illumination, each containing 12 tablets, and is manually annotated in YOLO format across six physical-condition classes (intact, damaged, cracked, broken, moist, and stained), yielding 3,012 bounding-box instances. All five standard scale variants (n, s, m, l, x) of each family were trained for 100 epochs under identical hyper-parameter settings, producing fifteen model variants that are compared in terms of mAP@0.5, mAP@0.5:0.95, precision, recall, inference speed (FPS), parameter count, and FLOPs. Experimental results show that YOLO11l achieves the best overall accuracy, with 96.8% mAP@0.5 and 91.7% mAP@0.5:0.95, while YOLO11n offers the most attractive real-time trade-off at 345.9 FPS with 95.6% mAP@0.5 and only 2.5M parameters. YOLO12 variants deliver competitive accuracy but at markedly lower inference speeds for the larger scales, whereas YOLO26 scales lag in the nano regime (88.0% mAP@0.5) but close the gap at l/x scales. Class-wise analysis of YOLO11l shows consistently high performance across all six defect categories, with mAP@0.5 ranging from 0.940 (damaged) to 0.994 (stained). The results provide practical guidance for selecting a YOLO configuration for real-time effervescent tablet inspection lines and demonstrate that modern nano- and small-scale detectors are already sufficient for high-throughput pharmaceutical quality control.

Keywords: effervescent tablets; defect detection; object detection; YOLO; deep learning; computer vision; pharmaceutical quality control

1. Introduction

Pharmaceutical manufacturing is a strictly regulated sector in which any deviation from quality specifications can compromise both patient safety and therapeutic efficacy [1]. Solid oral dosage forms—including compressed tablets, film-coated tablets, and effervescent tablets—constitute the most widely used drug delivery format due to their ease of administration, stable dosing, and cost-effective large-scale production [2]. However, during compression, coating, packaging, and transport, tablets are exposed to mechanical stress, humidity, and environmental factors that may cause a wide range of surface and structural defects such as cracks, chipping, breakage, moisture absorption, and staining. Such defects not only reduce the product's visual and commercial quality but may also alter disintegration behavior, dose uniformity, and ultimately the bioavailability of the active pharmaceutical ingredient [3,4]. Therefore, accurate, consistent, and high-throughput defect detection is a critical requirement of modern pharmaceutical quality assurance programs.

Effervescent tablets represent a particularly sensitive subclass of solid oral dosage forms. Because they are designed to release carbon dioxide upon dissolution in water, they are formulated

 doi:10.20944/preprints202604.1475.v1

with highly hygroscopic components (typically citric or tartaric acid combined with carbonate or bicarbonate salts). As a result, even minor cracks or surface defects can trigger premature moisture absorption and initiate effervescence before the product reaches the patient [5,6]. In practice, this makes effervescent products substantially more vulnerable to quality deviations than conventional compressed tablets, underscoring the operational importance of reliable, in-line visual inspection.

Traditional quality control in pharmaceutical manufacturing relies heavily on manual visual inspection by trained operators [7]. Although this approach is still common, it has several well-documented limitations: it is subjective, operator-dependent, fatigue-sensitive, and cannot scale to the throughput of modern high-speed production lines. Classical rule-based image processing methods (thresholding, edge detection, template matching) offer partial automation, but they typically depend on hand-crafted features, fixed illumination conditions, and a narrow set of assumptions about tablet appearance [8]. They tend to perform poorly under varying lighting conditions, on reflective surfaces, with heterogeneous defect morphologies, or when tablets whose backgrounds blend into the defect region [9,10].

In recent years, deep-learning-based object detection has emerged as a powerful alternative for industrial visual inspection [11,12]. Among the available frameworks, the You Only Look Once (YOLO) family is especially attractive because it formulates detection as a single-stage regression task, predicting bounding-box coordinates and class labels in a single forward pass, thereby achieving real-time throughput [13]. Since YOLOv1, successive versions have introduced important architectural innovations—anchor-free heads, CSP-based backbones, decoupled heads, and attention-centric modules—that have progressively improved the accuracy–speed trade-off required for in-line industrial deployment [14–16]. In the context of pharmaceutical inspection, YOLO variants have been successfully applied to pill identification [17], blister-pack defect detection [18], film-coating defect recognition [10], tableting defect detection [19], and crystal morphology analysis in microfluidic systems [20].

Despite these advances, several important gaps remain in the literature. First, the overwhelming majority of YOLO-based pharmaceutical defect-detection studies focus on compressed or film-coated tablets in blister packages, while effervescent tablets—whose surface properties, defect types, and hygroscopic behavior differ meaningfully—have received comparatively little attention. Second, most studies benchmark only a single YOLO variant (e.g., YOLOv5 or YOLOv8) at a single scale, without systematically comparing the multi-scale (nano to extra-large) trade-offs that actually govern industrial deployment decisions [21,22]. Third, the most recent members of the YOLO family—YOLO11 (with C3k2 blocks and C2PSA spatial attention) [23], YOLO12 (area attention and R-ELAN) [24], and YOLO26—have not yet been jointly evaluated on a pharmaceutical defect-detection task, even though their architectural differences are directly relevant to the small, low-contrast, morphologically diverse defects encountered in effervescent tablets.

Motivated by these gaps, this study presents a multi-scale benchmarking of three recent YOLO families (YOLO11, YOLO12, and YOLO26) across all five standard scale variants (n, s, m, l, x) on a newly constructed effervescent tablet defect dataset. The dataset comprises 251 high-resolution images, each containing 12 effervescent tablets, manually annotated into six physical-condition classes (intact, damaged, cracked, broken, moist, stained). The principal contributions of this work can be summarised as follows:

- A new multi-class effervescent tablet defect dataset is introduced, covering six realistic defect categories captured under controlled imaging conditions, with per-tablet bounding-box annotations in YOLO format.
- A systematic multi-scale benchmark of three recent YOLO families (YOLO11, YOLO12, YOLO26) is provided, evaluating fifteen model variants in total under identical training and hyper-parameter conditions.
- A comprehensive trade-off analysis is presented across accuracy (mAP@0.5, mAP@0.5:0.95, precision, recall), speed (FPS), and computational cost (parameters, FLOPs), together with class-wise performance on the six defect categories.

 doi:10.20944/preprints202604.1475.v1

- Practical deployment guidance is provided on which YOLO configuration is most suitable for real-time effervescent tablet inspection lines based on the observed accuracy–efficiency trade-offs.

The remainder of this paper is organized as follows. Section 2 reviews related work on pharmaceutical defect detection and recent YOLO architectures. Section 3 describes the dataset, imaging setup, annotation strategy, and evaluated YOLO variants. Section 4 details the experimental setup and evaluation metrics. Section 5 reports the quantitative and class-wise results. Section 6 discusses the findings, limitations, and avenues for future work. Section 7 concludes the paper.

2. Related Work

The problem of automated tablet defect detection sits at the intersection of pharmaceutical quality control and industrial computer vision. Early work relied almost exclusively on classical image processing—thresholding, morphological operations, edge extraction, and template-matching against a reference "golden" tablet [8,25]. Such approaches are attractive from a cost perspective but become brittle whenever lighting, tablet orientation, or defect morphology deviates from the calibrated conditions. As high-resolution cameras and GPU-accelerated deep learning have become more affordable, convolutional neural networks (CNNs) and, more recently, single-stage object detectors have become the dominant paradigms [11,12].

2.1. CNN-Based Pharmaceutical Inspection

Ma et al. [9] developed one of the first fully automated deep-learning pipelines for pharmaceutical tablets, applying a CNN to X-ray computed tomography slices in order to detect internal cracks that are invisible to surface inspection. Their system dramatically reduced the analysis time of X-ray computed tomography (XRCT) data while matching the accuracy of trained analysts. Pathak et al. [26] collected 6,000 factory-scale tablet images labeled as GOOD or NOT-GOOD and trained transfer-learning CNNs (ResNet and DenseNet variants) with data augmentation, reporting classification accuracies above 94% and substantially outperforming SVM and KNN baselines on the same data. Complementary work by Zhang et al. [27] applied CNN-based models to capsule surface defects with comparable gains over classical pipelines. These studies established that deep learning can deliver both the accuracy and robustness required for industrial pharmaceutical inspection—but they were limited to whole-image classification, not localized multi-defect detection.

2.2. YOLO-Based Tablet and Pill Inspection

The shift toward single-stage object detection began with the work of Ficzere et al. [10], who coupled a YOLOv5 detector with classical image analysis to simultaneously classify film-coating defects and measure coating thickness in real time. Their system achieved a classification accuracy of 98.2% across five defect classes and was explicitly designed to match the throughput of continuous film-coating lines. Diószegi et al. [19] extended this line of work to uncoated compressed tablets, reporting 99.2% real-time recognition accuracy with YOLOv5 across five tableting-defect categories and further demonstrating that the same pipeline can predict disintegration time and crushing strength from surface texture alone.

More recent studies have moved toward YOLOv8-based architectures with custom enhancements. Rajappa et al. [17] proposed CBS-YOLOv8, which replaces the SPPF module with SimSPPF and introduces coordinate attention together with BiFPN-style feature fusion. On their custom blister-pack dataset, CBS-YOLOv8 reached an mAP@0.5 of 97.4% at 79.25 FPS, outperforming Faster R-CNN (89.3%), SSD (86.5%), YOLOv5s (96.6%), YOLOv7 (96.0%), and all native YOLOv8 variants (95.4–96.8%). Bandyopadhyay et al. [18] extended YOLOv8 with adaptive-confidence tracking for multi-scale defect counting on high-speed pharmaceutical production lines, explicitly addressing identity switches and temporal inconsistencies. Yan et al. [28] enhanced YOLOv8n with a Mamba-like linear attention module for irregular film-coated tablets, showing that lightweight

 doi:10.20944/preprints202604.1475.v1

attention can improve sensitivity to small coating defects without compromising real-time performance.

Pill identification—a closely related task—also illustrates YOLO's maturity for pharmaceutical imaging. Chen and Wang [29] benchmarked YOLOv5 against YOLOv8 on a multi-class pharmacy pill dataset, reporting 94.83% and 96.95% mAP, respectively, with YOLOv8 offering a markedly better speed–accuracy trade-off. Wei et al. [20] proposed YOLO-PBESW, a lightweight YOLO variant for identifying indomethacin crystal morphologies in microfluidic droplets, demonstrating that the YOLO framework generalizes well to microscopy-scale pharmaceutical targets. Kim et al. [30] applied a lightweight YOLOv7 variant to capsule color and shape verification on a conveyor line, reporting a further reduction in memory footprint without measurable loss in accuracy.

2.3. Multi-Scale and Multi-Version YOLO Benchmarks

A growing body of work has begun to systematically compare multiple YOLO versions and scales across industrial defect domains. Khanam and Hussain [23] analyzed the architectural innovations of YOLO11—the C3k2 cross-stage partial block, SPPF, and the C2PSA spatial-attention module—and reported that YOLO11m reaches 95.0% mAP on MS COCO with approximately 22% fewer parameters than YOLOv8m, setting a new efficient baseline. Tian et al. [24] subsequently introduced YOLO12, which replaces CNN-centric backbones with an attention-centric design based on area attention, R-ELAN, and FlashAttention; YOLO12-N achieves 40.6% mAP on COCO with a latency of 1.64 ms, outperforming YOLO11-N by 1.2 mAP at comparable speed. Ultralytics [15] continues to release updated family members (including the YOLO26 line evaluated here) through their open-source codebase.

In the industrial defect domain, a comparative study on solar photovoltaic panels [22] evaluated YOLOv5, YOLOv8, and YOLOv11 on a multi-class defect dataset: YOLOv11 delivered the highest mAP@0.5 (93.4%), YOLOv5 achieved the fastest inference (7.1 ms), while YOLOv8 showed the strongest recall for rare defect classes. A similar multi-scale evaluation of YOLO11 variants (n, s, m, l, x) for PCB defect detection [31] reported consistently higher mAP for larger variants, at the cost of inference speed, echoing the general trend observed across industrial domains. Vijayakumar and Vairavasundaram [32] provide a broader review of YOLO-based object detection across industrial and scientific applications, reinforcing the observation that the accuracy–speed balance shifts noticeably with scale. Other recent surveys on industrial surface-defect detection [33,34] arrive at the same conclusion: single-scale benchmarks tend to understate deployment trade-offs.

2.4. Summary of Prior Work and Positioning of This Study

Table 1 summarises the representative studies discussed above by application, method, dataset, classes, and reported numerical results. Three consistent patterns emerge. (i) The pharmaceutical-inspection literature is dominated by YOLOv5 and YOLOv8, with almost no systematic evaluation of the very recent YOLO11/YOLO12/YOLO26 families. (ii) Most studies fix the model scale a priori (typically n or s) rather than characterizing the full nano–extra-large accuracy/cost curve. (iii) Effervescent tablets, despite their hygroscopic sensitivity and commercial relevance, have not been systematically targeted—existing work focuses on compressed, film-coated, or capsule formulations in blister packages. The present study addresses all three gaps by benchmarking three recent YOLO families across five scales on a dedicated multi-class effervescent tablet defect dataset.

Table 1. ***Summary of representative studies on defect detection in pharmaceutical and related industrial domains, highlighting methods, datasets, defect classes, and reported numerical results.***

Study	Application	Method	Dataset	Classes	Results

 doi:10.20944/preprints202604.1475.v1

Ma et al. [9]	Internal tablet crack detection	CNN on X-ray CT slices	XRCT slices of oral tablets	Crack / no-crack	High accuracy; significantly faster than manual XRCT analysis
Ficzere et al. [10]	Film-coated tablet defect recognition	YOLOv5 + classical image processing	Custom film-coated tablet images	5 defect classes	Classification accuracy 98.2%; real-time coating thickness measurement
Diószegi et al. [19]	Tableting defect detection and disintegration prediction	YOLOv5 on surface images	Custom tableting defect dataset	5 defect classes	Real-time recognition accuracy 99.2%; matches press throughput
Rajappa et al. [17]	Blister-pack defect detection	CBS-YOLOv8 (CA + BiFPN + SimSPPF)	Custom blister + SESOVER A-ST	Broken, empty, cracked, foreign, colour mismatch	mAP@0.5 = 97.4% at 79.25 FPS; outperforms YOLOv5s/YOLOv7/YOLOv8
Bandyopadhyay et al. [18]	Tablet defect tracking and counting	YOLOv8 + adaptive-confidence tracking	Production-line video frames	Multiple tablet defect classes	Reduced identity switches; improved counting consistency
Yan et al. [28]	Irregular film-coated tablet inspection	YOLOv8n + Mamba-like linear attention	Irregular film-coated tablet images	Multiple coating defect classes	Improved small-defect sensitivity; low-overhead real-time deployment
Lin and Xiao [25]	Tablet defect detection (bottling line)	Biaxial-plane discrete scanning	Front and side tablet views	Defective / non-defective	Up to 20%/100% similarity gap (front/side) between defective and standard tablets

 doi:10.20944/preprints202604.1475.v1

Pathak et al. [26]	Pill defect classification	Improved CNN + Gaussian smoothing	6,000 factory-scale tablet images	GOOD / NOT-GOOD	High classification accuracy on a factory-scale dataset
Chen and Wang [29]	Pill / capsule identification	YOLOv5 vs YOLOv8 comparison	Multi-class pharmacy pill dataset	Multiple pill classes	YOLOv8 96.95% mAP; YOLOv5 94.83% mAP; YOLOv8 better speed–accuracy
Wei et al. [20]	Indomethacin crystal morphology identification	YOLO-PBESW (lightweight YOLO)	Microfluidic droplet microscopy	Multiple crystal morphologies	Lightweight design; high identification accuracy
Solar-PV study [22]	PV panel defect detection	YOLOv5 vs YOLOv8 vs YOLOv11	PV panel defect dataset	Multiple defect categories	YOLOv11 highest mAP@0.5 (93.4%); YOLOv5 fastest; YOLOv8 best recall for rare classes
Khanam and Hussain [23]	YOLO11 architectural analysis	YOLO11 family (n, s, m, l, x)	MS COCO	80 object classes	YOLO11m 95.0% mAP with 22% fewer parameters than YOLOv8m
Tian et al. [24]	YOLO12 (attention-centric)	A2 attention + R-ELAN + FlashAttention	MS COCO	80 object classes	YOLO12-N 40.6% mAP @ 1.64 ms; +1.2% over YOLO11-N

3. Materials and Methods

3.1. Dataset

An image dataset was constructed to analyze the most common physical deformations and surface defects that may occur during the production and packaging of effervescent tablets. In this context, photographs of both undamaged and damaged tablets were captured using a Samsung NX-model camera under fixed-focus, controlled shooting conditions. A fixed focus ensured a consistent level of sharpness across all images, thereby preserving the dataset's homogeneity. Tablets were placed on a neutral grey background to prevent color leakage between the defect regions and the background, and the distance between the camera and the shooting table was kept constant throughout all sessions. The data-collection setup and imaging environment are illustrated in Figure 1.

 doi:10.20944/preprints202604.1475.v1

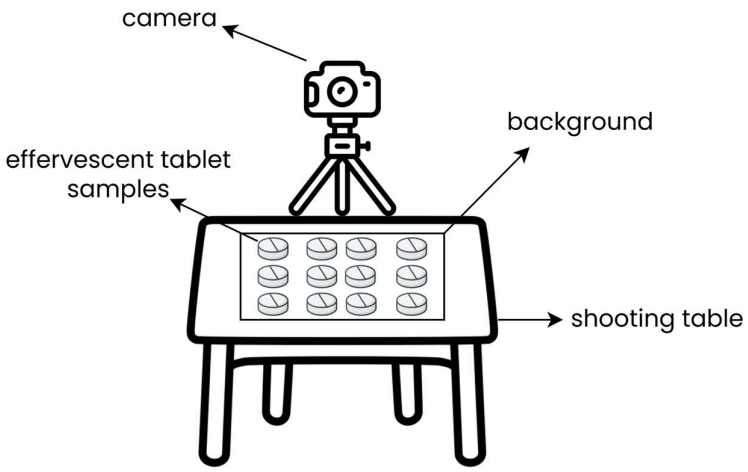
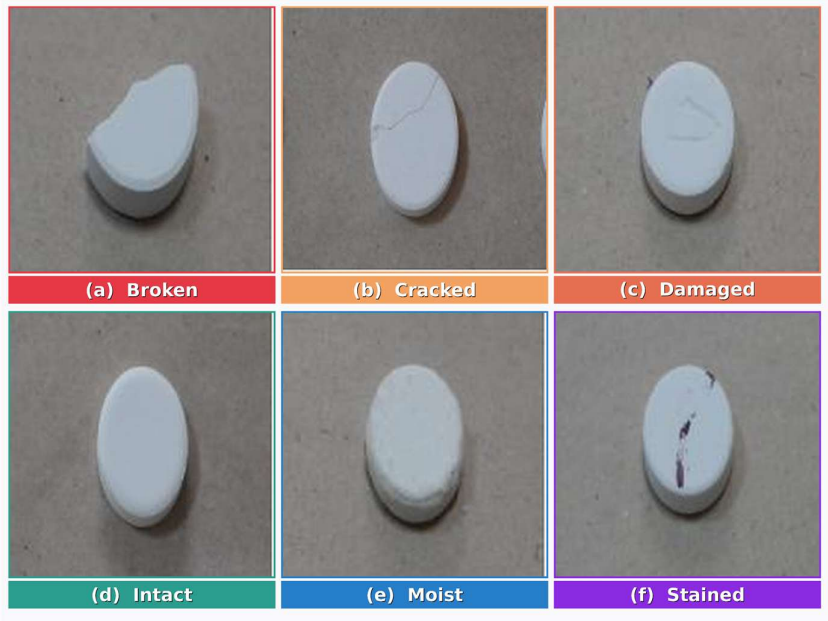


Figure 1. *Data-acquisition setup and controlled imaging environment used for capturing effervescent tablet images. A tripod-mounted Samsung NX camera captured images from a fixed height above a grey shooting table populated with twelve effervescent tablets per frame.*

The dataset comprises 251 high-resolution images, each containing twelve effervescent tablets in various physical conditions. This structure not only increases the sample diversity but also enables multi-object detection within a single image, making the dataset well-suited to object-detection tasks.

The tablets were classified into six categories based on their physical condition and surface features: intact (no visible deformation), damaged (structural deterioration), cracked (fine surface fissures), broken (fragmented or structurally compromised), moist (surface moisture present), and stained (discolored or contaminated surface). This classification reflects the defects that are commonly observed in pharmaceutical quality-control practice.

For model training and evaluation, all tablets in the images were manually annotated with bounding boxes in the YOLO format, with each object labeled by its class and normalized spatial coordinates. This annotation strategy enables the use of YOLO-based object-detection models to simultaneously localize and classify multiple tablets in each image. Representative examples of each class are shown in Figure 2.



 doi:10.20944/preprints202604.1475.v1

Figure 2. *Representative examples of effervescent tablet conditions in the proposed dataset: (a) broken, (b) cracked, (c) damaged, (d) intact, (e) moist, and (f) stained. The six classes capture the typical surface and structural deviations encountered during effervescent-tablet quality control.*

The total number of annotated bounding-box instances is 3,012. An inspection of Table 2 reveals that the dataset exhibits moderate class imbalance: certain classes, such as "intact" and "damaged," are overrepresented, while other classes, such as "moist" and "broken," are underrepresented. This imbalance reflects the natural frequency with which each defect type appears in practice and is handled through the data-augmentation strategy described in Section 3.3.

Table 2. ***Distribution of effervescent tablet samples across the six defect classes in the proposed dataset.***

Class	Intact	Damaged	Cracked	Broken	Moist	Stained	Total
Count	617	609	445	400	353	588	3,012

Beyond class frequency, the spatial extent of the annotated defect regions also varies noticeably across categories. Figure 3 reports the per-class distribution of normalized bounding-box areas.

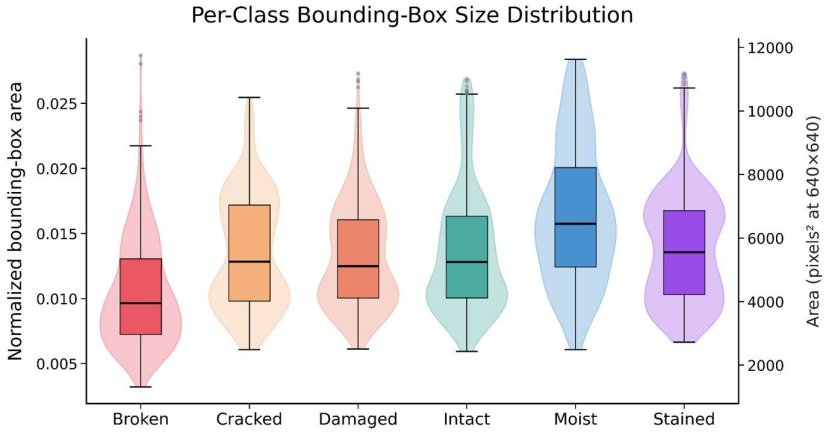


Figure 3. *Per-class distribution of normalised bounding-box areas, illustrating size variability and scale differences among tablet-condition categories. The violin plots show the full probability density of bounding-box areas, while the internal box plots summarise the median and inter-quartile range for each class.*

The chart in Figure 3 shows that the distribution of bounding-box areas varies across classes, indicating that object sizes differ by class. In particular, the "moist" and "stained" classes exhibit larger median areas with wider tails, whereas the "broken" class is distributed more tightly around smaller values. This demonstrates the presence of objects at different scales within the dataset and motivates the use of multi-scale object-detection models.

3.2. Data Pre-Processing and Splitting

Prior to model training, all captured images were processed through a consistent pre-processing pipeline to ensure uniformity across the dataset and to meet the input requirements of the YOLO detectors. All images were first inspected visually to remove out-of-focus frames, over- or under-exposed samples, and images in which tablets were occluded by foreign objects. The remaining images were then resized to 640×640 pixels using bilinear interpolation, preserving the aspect ratio by letterbox padding where required. Pixel intensities were normalized to the [0, 1] range to stabilize the optimization process during training.

The final dataset, consisting of 251 annotated images, was partitioned into three disjoint subsets for training, validation, and testing using a stratified random split of 70%, 20%, and 10%, respectively. Stratification was performed at the class level to ensure that all six defect categories were

 doi:10.20944/preprints202604.1475.v1

proportionally represented across subsets. This strategy avoids distributional bias and provides a reliable estimate of generalization performance. The same subset indices were used for all YOLO variants evaluated in this study, thereby ensuring a fair and reproducible comparison among models.

3.3. Data Augmentation

Because the dataset contains a limited number of images and exhibits moderate class imbalance, data augmentation was employed during training to improve model robustness and generalization. All augmentations were applied on the fly through the default Ultralytics augmentation pipeline [15], which ensures that each epoch sees a slightly different version of every training sample while the validation and test subsets remain unaltered.

The augmentation strategy consisted of geometric and photometric transformations selected to reflect realistic variations observed on an inspection line. Geometric augmentations included random horizontal and vertical flips, small-angle rotations, random scaling, and random translations, enabling the models to become invariant to tablet orientation and position within the field of view. Photometric augmentations included HSV-based adjustments of hue, saturation, and value, which simulate lighting fluctuations and minor color variability that may arise from variations in ambient illumination or tablet surface reflectivity. In addition, mosaic augmentation was employed to combine four training images into a single composite image, improving the detectors' ability to learn contextual cues and to handle small objects. Bounding-box coordinates were updated accordingly after each transformation to maintain annotation consistency.

3.4. YOLO Models

You Only Look Once (YOLO) is a widely used object-detection framework that performs object localization and classification simultaneously in a single forward pass, enabling high-speed, real-time detection [13]. Unlike traditional two-stage detectors, YOLO treats object detection as a regression problem, directly predicting bounding-box coordinates and class probabilities from the input image.

Since its introduction, the YOLO family has undergone significant architectural evolution [32,35]. The original YOLOv1 [13] laid the foundation for real-time object detection, while subsequent versions—YOLOv2 [14] and YOLOv3 [36]—introduced substantial improvements in accuracy and speed. Later versions (YOLOv4 through YOLOv7) further enhanced performance through more advanced backbone networks, feature-aggregation strategies, and optimized training techniques [37–39]. More recent versions, such as YOLOv8 [15] and beyond, adopt anchor-free detection and more flexible architectures, improving efficiency and generalization [40,41]. The overall evolution of the YOLO family is illustrated in Figure 4.

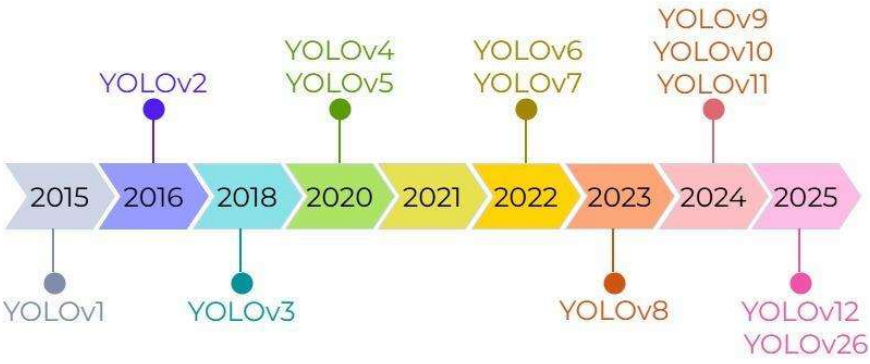


Figure 4. *Evolution of the YOLO object-detection family, from YOLOv1 (2015) through the most recent YOLO12 and YOLO26 releases (2025). Each release has introduced architectural refinements in the backbone, neck, and detection head, with progressive improvements in the accuracy–speed trade-off.*

 doi:10.20944/preprints202604.1475.v1

In the present study, the most recent members of the YOLO family—YOLO11, YOLO12, and YOLO26—were compared. These families stand out for their architectural improvements, scalability, and optimization techniques, and they represent the current state of the art in object-detection performance. Accordingly, the study aims to comprehensively analyze the effects of structural and algorithmic advances in modern YOLO architectures on a multi-class, defect-focused dataset.

YOLO11

YOLO11 [23] is organised as a backbone–neck–head detection pipeline. Starting from the input image, the model first applies a series of Conv layers and C3k2 blocks to extract increasingly abstract visual features. At the end of the backbone, an SPPF block enlarges the contextual field of the extracted representation, while a C2PSA block further refines feature relationships before the features are transferred to the neck. In the neck, repeated Upsample and Concat operations merge deeper semantic features with earlier intermediate feature maps. After each fusion stage, C3k2 blocks reprocess the combined features to strengthen the representation before forwarding them to the next stage. Finally, the Detect block produces object predictions from the refined multi-scale features. Overall, the diagram shown in Figure 5 describes YOLO11 as a structured process of hierarchical feature extraction, contextual enhancement, cross-stage fusion, and final detection.

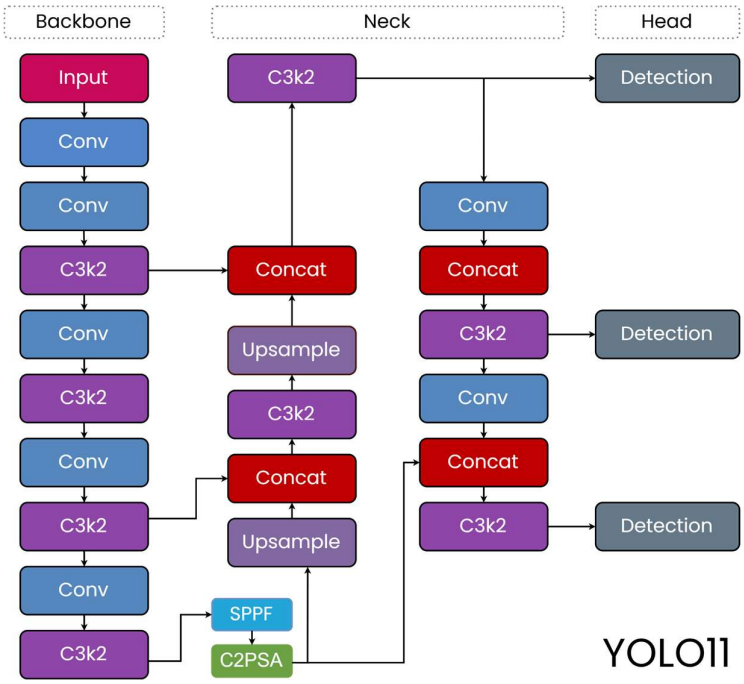


Figure 5. *Overall architecture of YOLO11, illustrating the backbone, neck, and head sections. The backbone stacks Conv and C3k2 blocks followed by an SPPF block and a C2PSA spatial-attention module; the neck performs top-down and bottom-up feature fusion; and the head produces predictions at three scales.*

YOLO12

YOLO12 [24], introduced by Tian et al., follows a backbone–neck–head detection architecture similar to that of YOLO11, but places greater emphasis on internal feature refinement by adopting an attention-centric design. Starting from the input image, the model first processes the visual data through Conv and C3k2 blocks to construct the initial feature hierarchy. In the deeper stages of the backbone, A2C2f blocks are incorporated to enable more advanced feature transformation and aggregation through area-attention before the extracted features are transferred to the neck. Within the neck, repeated Upsample and Concat operations combine deeper semantic features with earlier intermediate outputs, thereby integrating spatial and semantic information more effectively. After

 doi:10.20944/preprints202604.1475.v1

each fusion step, the aggregated features are further refined before being passed to the next stage. Finally, the Detect block produces bounding-box and class predictions from the refined multi-scale feature maps. Overall, the diagram in Figure 6 presents YOLO12 as a framework that preserves the general YOLO detection flow while placing greater emphasis on attention-based post-fusion refinement and feature modeling.

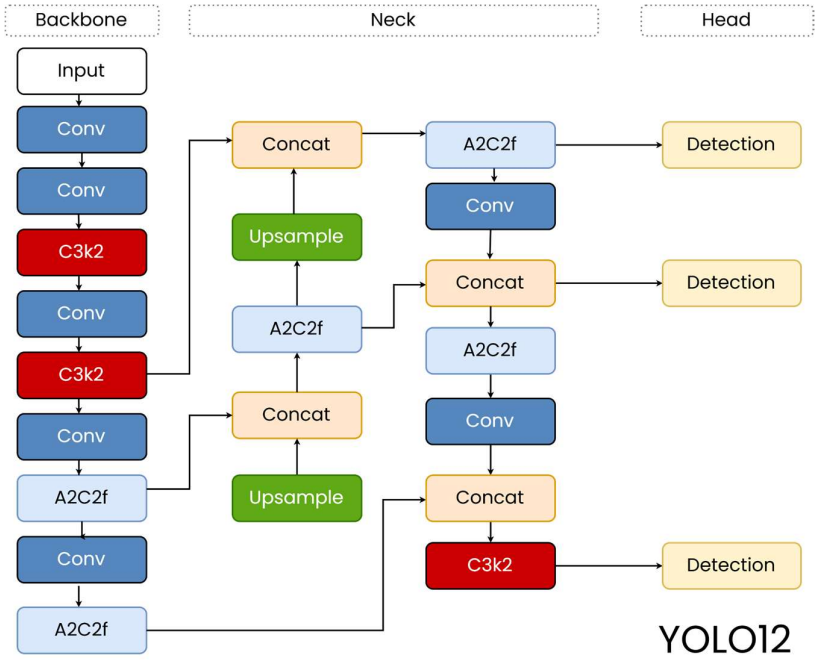


Figure 6. *Overall architecture of YOLO12, illustrating the backbone, neck, and head sections. Compared with YOLO11, YOLO12 replaces selected C3k2 blocks in the deeper stages with A2C2f area-attention blocks, providing attention-based feature refinement prior to and after neck fusion.*

YOLO26

YOLO26 [15] is presented as a multi-stage object-detection architecture composed of backbone, neck, and detection-head modules. Beginning with the input image, the model applies a sequence of Conv and C3k2 blocks to progressively transform the raw visual data into deeper, more abstract feature representations. At the end of the backbone, an SPPF block enhances contextual information, while a C2PSA block further refines features before they are transferred to the neck. In the neck, several Upsample and Concat operations repeatedly combine deep feature maps with intermediate representations from earlier stages. These fused features are then reorganized and refined through additional Conv and C3k2 blocks before being delivered to the final Detect block. As a result, the diagram shown in Figure 7 highlights YOLO26 as a model that emphasizes repeated multi-stage fusion and refinement prior to final object prediction.

 doi:10.20944/preprints202604.1475.v1

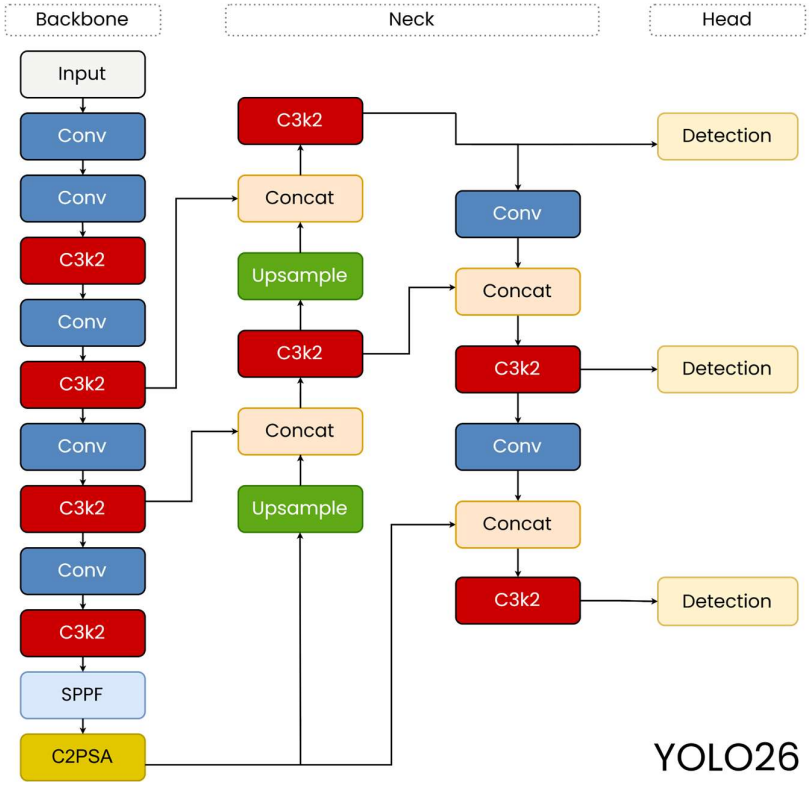


Figure 7. *Overall architecture of YOLO26, illustrating the backbone, neck, and head sections. YOLO26 retains the SPPF and C2PSA blocks of YOLO11 while extending the neck with deeper cross-stage fusion, which is intended to provide stronger feature refinement at larger scales.*

The YOLO11, YOLO12, and YOLO26 models each represent distinct architectural enhancements and are available in the following standard scales: nano (n), small (s), medium (m), large (l), and extra-large (x). This scaling approach enables systematic examination of the trade-off among model complexity, accuracy, and speed. The reference performance and computational metrics published by the authors and Ultralytics for the different model families and scales are summarised in Table 3.

Table 3. ***Reference performance and computational characteristics of the YOLO11, YOLO12, and YOLO26 model families across scales on MS COCO, adapted from the corresponding source repositories [15,23,24].***

Model	Input	mAP@50-95	CPU ONNX (ms)	Params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	56.9	194.9
YOLO12n	640	40.6	1.64	2.6	6.5
YOLO12s	640	48.0	2.61	9.3	21.4

 doi:10.20944/preprints202604.1475.v1

Model	Input	mAP@50-95	CPU ONNX (ms)	Params (M)	FLOPs (B)
YOLO12m	640	52.5	4.86	20.2	67.5
YOLO12l	640	53.7	6.77	26.4	88.9
YOLO12x	640	55.2	11.79	59.1	199.0
YOLO26n	640	40.9	38.9 ± 0.7	2.4	5.4
YOLO26s	640	48.6	87.2 ± 0.9	9.5	20.7
YOLO26m	640	53.1	220.0 ± 1.4	20.4	68.2
YOLO26l	640	55.0	286.2 ± 2.0	24.8	86.4
YOLO26x	640	57.5	525.8 ± 4.0	55.7	193.9

Upon examining Table 3, it is observed that as the model size increases, the mAP values rise steadily, while the computational cost (FLOPs) and inference time increase significantly. Notably, the highest accuracy on MS COCO is achieved by the large-scale variants (l and x) of the YOLO26 family. In contrast, YOLO11 models offer more balanced performance at a lower computational cost. YOLO12 models generally demonstrate moderate performance in both accuracy and speed, serving as a balance point between the other two model families.

4. Experimental Setup

All models in this study were trained in an identical hardware and software environment. The experiments were conducted on a workstation equipped with an NVIDIA RTX A5000 GPU (24 GB GDDR6, 384-bit memory bus). The Ultralytics-based deep-learning library [15] was used for training and inference. All models were trained with an input size of 640×640 pixels, and the same hyperparameters were applied throughout training to ensure a fair comparison. Each model was trained for 100 epochs using the recommended default YOLO parameters, including the SGD optimizer with an initial learning rate of 0.01, a momentum of 0.937, and a weight decay of 5×10^{-4}. The batch size was set to 16, and mixed-precision training was enabled. Data-augmentation transformations (geometric and photometric, as described in Section 3.3) were applied during training to improve the generalization of the models.

4.1. Evaluation Metrics

In this study, the performance of the YOLO models was comprehensively evaluated in terms of accuracy, speed, and computational cost. To this end, the following metrics were used: mean Average Precision at an Intersection-over-Union threshold of 0.5 (mAP@0.5), mean Average Precision over the 0.5:0.95 threshold range (mAP@0.5:0.95), Precision, Recall, Frames Per Second (FPS), number of parameters (Params), and floating-point operations (FLOPs).

Precision is the proportion of predicted positive samples that are actually positive, while Recall is the proportion of actual positive samples that are successfully detected by the model. They are defined as follows:

$$\text{Precision} = TP / (TP + FP) \quad (1)$$

$$\text{Recall} = TP / (TP + FN) \quad (2)$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively. The Average Precision (AP) is a fundamental indicator of object-detection performance and is computed as the area under the Precision–Recall curve:

$$\mathbf{AP} = \int_0^1 \mathbf{P(R)}\ dR \quad \mathbf{(3)}$$

 doi:10.20944/preprints202604.1475.v1

The mean Average Precision (mAP) is obtained by averaging AP over all object classes:

$$mAP = (1 / N) \cdot \Sigma_{i=1}^{N} AP_i \quad (4)$$

where N is the total number of classes. The mAP@0.5 metric used in this study was calculated with an Intersection-over-Union (IoU) threshold of 0.5, while mAP@0.5:0.95 averages the mAP over IoU thresholds from 0.5 to 0.95 in steps of 0.05. The IoU between a predicted box B_p and a ground-truth box B_{gt} is defined as:

$$IoU = |B_p \cap B_{gt}| / |B_p \cup B_{gt}| \quad (5)$$

FPS (Frames Per Second) refers to the number of images that a model can process per second and is an indicator of real-time performance:

$$FPS = 1 / t_{inference} \quad (6)$$

where $t_{inference}$ is the average forward-pass latency per image. The number of parameters (Params) reflects the total count of learnable weights in the model, while FLOPs (Floating-Point Operations) represent the total number of floating-point operations performed during a single forward pass at 640×640 input resolution. Assessing all these metrics together provides a balanced view of the models with respect to accuracy, speed, and computational cost.

5. Results

5.1. Quantitative Results

This section reports the full quantitative comparison of the fifteen YOLO model variants (three families × five scales) trained and evaluated under identical conditions on the proposed effervescent tablet dataset. Each variant is evaluated with respect to detection accuracy (mAP@0.5, mAP@0.5:0.95, Precision, Recall), real-time performance (FPS), and computational cost (Params, FLOPs). The full set of results is reported in Table 4.

Table 4. ***Comprehensive performance comparison of YOLO11, YOLO12, and YOLO26 model families and their scale variants (n, s, m, l, x) on the proposed effervescent tablet dataset.***

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	FPS	Params (M)	FLOPs (G)
YOLO11n	95.6	90.6	94.0	91.4	345.9	2.5	6.4
YOLO11s	96.0	91.2	92.0	93.6	224.2	9.4	21.6
YOLO11m	96.7	91.6	94.8	93.1	185.2	20.0	68.2
YOLO11l	96.8	91.7	91.5	93.3	142.9	25.3	87.3
YOLO11x	96.6	91.4	94.3	92.6	88.5	56.8	195.5
YOLO12n	95.6	90.6	92.9	91.0	303.0	2.5	6.5
YOLO12s	96.7	90.8	90.0	92.3	243.9	9.2	21.5
YOLO12m	96.3	91.2	92.6	93.1	169.5	20.1	67.8
YOLO12l	96.7	91.8	93.9	93.2	69.4	26.3	89.4
YOLO12x	96.3	91.4	90.1	93.1	37.0	59.1	199.9
YOLO26n	88.0	83.0	85.7	74.8	298.1	2.5	5.8
YOLO26s	94.6	89.0	88.9	91.1	254.5	9.9	22.5

 doi:10.20944/preprints202604.1475.v1

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	FPS	Params (M)	FLOPs (G)
YOLO26m	95.6	90.3	93.4	88.9	66.7	21.7	74.8
YOLO26l	96.4	90.8	92.4	91.4	63.3	26.1	93.2
YOLO26x	96.5	91.5	93.7	92.6	52.0	58.8	208.6

An examination of Table 4 reveals several consistent patterns. Within the YOLO11 family, the mAP@0.5 metric rises smoothly from 95.6% at the nano scale to a maximum of 96.8% at the large scale, then slightly decreases to 96.6% at the extra-large scale—indicating that scaling beyond YOLO11l does not bring further accuracy gains on this dataset. The mAP@0.5:0.95 metric follows a similar trend, peaking at 91.7% for YOLO11l. At the opposite end of the efficiency spectrum, YOLO11n achieves the highest inference speed in the benchmark (345.9 FPS) while still reaching 95.6% mAP@0.5 with only 2.5 M parameters and 6.4 GFLOPs, making it the strongest candidate for resource-constrained deployment.

The YOLO12 family achieves accuracy levels very close to YOLO11 (mAP@0.5 between 95.6% and 96.7%), but with a noticeably different cost profile. While the nano and small variants preserve real-time throughput (303.0 FPS and 243.9 FPS, respectively), the large and extra-large variants drop to 69.4 FPS and 37.0 FPS. This latency penalty is a direct consequence of the area-attention and R-ELAN blocks introduced in YOLO12 [24], which improve feature representation but incur additional memory traffic and computation at larger scales.

The YOLO26 family exhibits the most pronounced scale dependence. YOLO26n underperforms all other models, achieving only 88.0% mAP@0.5 and 74.8% Recall, which are 7.6 percentage points and 16.2 percentage points lower than YOLO11n, respectively. However, as the scale increases, YOLO26 rapidly closes the gap: YOLO26 reaches 94.6% mAP@0.5, YOLO26m reaches 95.6%, and the l and x variants reach 96.4% and 96.5%, respectively, becoming essentially competitive with YOLO11 and YOLO12 at the same scales. In terms of FPS, however, the YOLO26 family remains slower than YOLO11 across all scales, owing to its deeper neck design.

Overall, the combined accuracy–speed comparison identifies two practical operating points on the proposed dataset. For maximum accuracy, YOLO11l is preferable (96.8% mAP@0.5, 91.7% mAP@0.5:0.95, 142.9 FPS). For real-time edge deployment where inference speed is critical, YOLO11n is the most attractive choice (95.6% mAP@0.5, 345.9 FPS, 2.5 M parameters, 6.4 G FLOPs). These two configurations jointly span the trade-off envelope observed in Table 4 and constitute the focus of the class-wise analysis reported in Section 5.2.

5.2. Analysis of the Best-Performing Model

This subsection presents a detailed analysis of the best-performing model identified in Table 4, namely YOLO11l. The analysis focuses on its aggregated performance metrics, convergence behavior, class-wise performance, and qualitative detection results.

A summary of the six headline metrics reported by YOLO11l on the held-out test set is shown in Figure 8. The model reaches 96.8% mAP@0.5, 91.7% mAP@0.5:0.95, 91.5% Precision, 93.3% Recall, 142.9 FPS, and 87.3 GFLOPs, confirming that YOLO11l simultaneously delivers high localization accuracy, balanced precision and recall, and real-time throughput.

 doi:10.20944/preprints202604.1475.v1

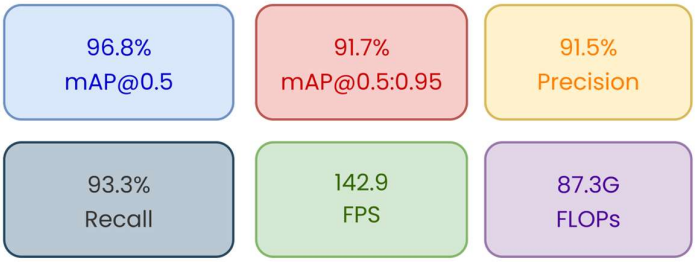


Figure 8. *Summary of the six headline evaluation metrics obtained by the best-performing YOLO11l model on the effervescent tablet test set: mAP@0.5, mAP@0.5:0.95, Precision, Recall, inference speed (FPS), and computational cost (FLOPs).*

The convergence behaviour of YOLO11l during training is shown in Figure 9. The top row reports the three training-loss components (box regression loss, classification loss, and distribution focal loss) together with precision and recall curves on the training set. The bottom row reports the corresponding validation-loss components and the evolution of mAP@0.5 and mAP@0.5:0.95.

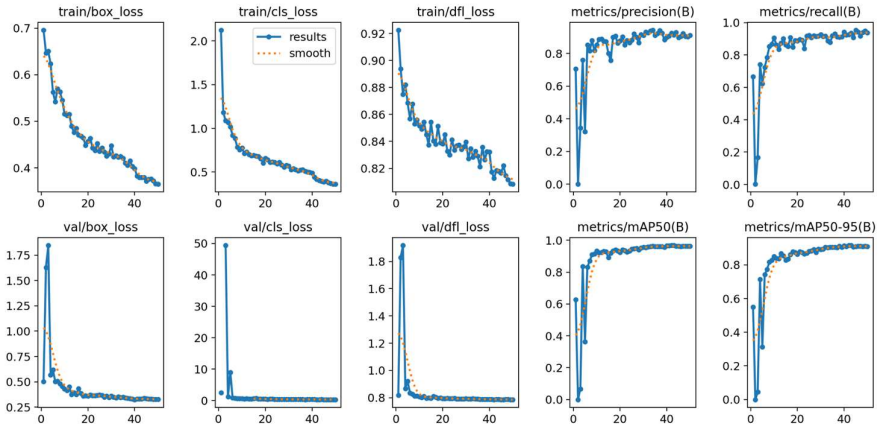


Figure 9. *Training and validation curves of the YOLO11l model on the effervescent tablet dataset. Top row: training box loss, classification loss, DFL loss, precision, and recall. Bottom row: validation box loss, classification loss, DFL loss, mAP@0.5, and mAP@0.5:0.95. All curves are displayed together with a smoothed trend line.*

As illustrated in Figure 9, all three training-loss components decrease monotonically and smoothly throughout the 100-epoch schedule, indicating stable optimization. The validation losses display a short initial spike during the first two to three epochs—caused by the randomly initialized detection head producing high-loss predictions on unseen validation images—and then converge rapidly to a low and stable plateau. Validation precision, recall, mAP@0.5, and mAP@0.5:0.95 all saturate around epochs 40–50 and remain close to their final values until the end of training, supporting the claim that 100 epochs are sufficient for convergence on this dataset and that no overfitting is observed.

The Precision–Recall (PR) curve and the corresponding per-class Average Precision values are shown in Figure 10. The overall model reaches an mAP@0.5 of 0.968 across all six classes. The per-class ordering is consistent with the qualitative difficulty of the classes: “stained” achieves the highest AP (0.994), followed by “broken” (0.988), “cracked” (0.971), “intact” (0.966), “moist” (0.951), and “damaged” (0.940). This ordering indicates that defect categories characterized by strong color or geometric contrast (stained, broken) are the easiest to detect, while the “damaged” class—which semantically overlaps with both “cracked” and “broken”—remains the most challenging.

 doi:10.20944/preprints202604.1475.v1

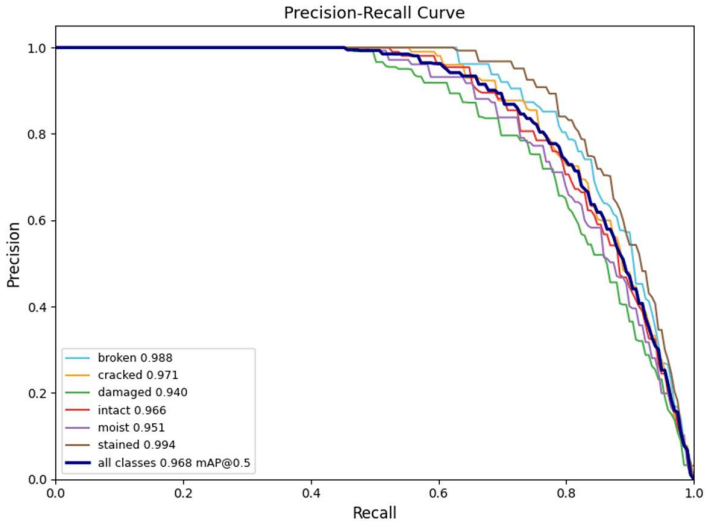


Figure 10. *Precision–Recall curves of the YOLO11l model on the effervescent tablet test set. The overall curve (thick navy line) corresponds to an mAP@0.5 of 0.968, and per-class AP values are reported in the legend for each of the six defect categories (broken, cracked, damaged, intact, moist, stained).*

To quantify the pattern of misclassifications, the confusion matrix of YOLO11l is reported in Figure 11. Diagonal entries are consistently large across all classes, confirming the model's high overall accuracy. Off-diagonal errors are concentrated in the "damaged/cracked/broken" cluster, where semantic overlap between defect definitions is highest, and in the "background" column, which reflects a small number of false positives where background texture was classified as a defect.

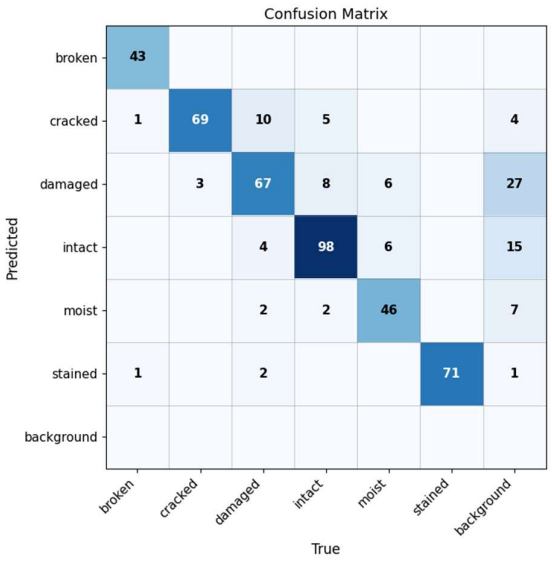


Figure 11. *Confusion matrix of the YOLO11l model on the effervescent tablet test set. Diagonal entries represent correct detections per class; off-diagonal entries indicate inter-class confusions and background-related false positives.*

The class-wise evaluation metrics for YOLO11l are reported in Table 5 and visualized as a grouped bar chart in Figure 12. Together, they characterize the model's fine-grained behavior for each defect category.

 doi:10.20944/preprints202604.1475.v1

Table 5. *Class-wise performance of the best-performing YOLO11l model on the effervescent tablet test set. Performance is reported in terms of Precision, Recall, mAP@0.5, and mAP@0.5:0.95 (%).*

Class	Instances	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
broken	45	99.0	95.6	98.8	91.2
cracked	72	86.3	96.4	97.1	92.5
damaged	85	87.3	87.1	94.0	88.5
intact	113	88.8	91.4	96.6	92.4
moist	58	92.8	89.1	95.1	91.0
stained	71	95.0	100.0	99.4	94.4
All classes	**444**	**91.5**	**93.3**	**96.8**	**91.7**

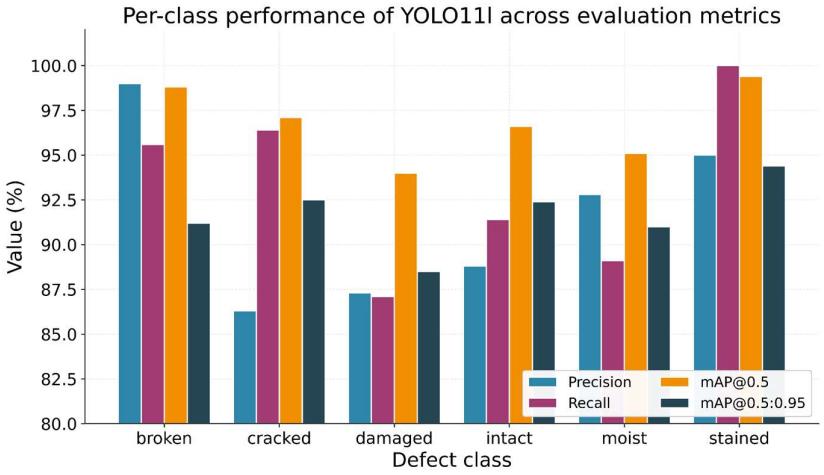


Figure 12. *Per-class performance of the YOLO11l model across the four evaluation metrics (Precision, Recall, mAP@0.5, and mAP@0.5:0.95) on the six defect categories of the effervescent tablet dataset.*

Table 5 and Figure 12 together show that YOLO11l achieves consistently high performance across all six defect categories. The "stained" class yields the highest overall performance, with 100% Recall, 99.4% mAP@0.5, and 94.4% mAP@0.5:0.95, confirming that the strong chromatic contrast of stained regions makes them the easiest to detect. The "broken" class also reaches very high scores (99.0% Precision, 95.6% Recall, 98.8% mAP@0.5), benefiting from the pronounced geometric discontinuities of broken tablets. Conversely, the "damaged" class records the lowest mAP@0.5 (94.0%) and the lowest Recall (87.1%), consistent with its visual similarity to the "cracked" and "broken" classes. Moderately challenging categories such as "moist" (95.1% mAP@0.5) and "intact" (96.6% mAP@0.5) lie in between, with the "moist" class displaying a slightly lower Recall (89.1%) due to subtle surface texture cues that can be confused with other conditions.

A qualitative comparison between the ground-truth annotations and YOLO11l's predictions on held-out test images is provided in Figure 13. The left half of the figure shows the manually annotated bounding boxes, while the right half shows the corresponding predictions with confidence scores. The predictions closely match the ground-truth boxes both in position and in class labels, with high confidence scores (typically in the range 0.8–1.0) on easy classes such as "intact", "stained", and "broken", and slightly lower confidence on the harder "damaged" and "cracked" examples.

 doi:10.20944/preprints202604.1475.v1

Figure 13. *Qualitative comparison between ground-truth annotations (top) and YOLO11l predictions (bottom) on representative effervescent tablet test images. Each tablet is enclosed in a coloured bounding box with its class label; predicted boxes additionally show the model's confidence score.*

Finally, a gallery of the most representative misclassification cases produced by YOLO11l is shown in Figure 14. The majority of errors occur between the semantically close classes "cracked", "damaged", and "broken", as already suggested by the confusion matrix in Figure 11. In several cases, visually subtle cues (such as a hairline fissure versus a thicker crack, or a small chipped edge versus a broader broken region) are responsible for the confusion. A smaller number of errors involve confusion between "intact"/"moist" and between "stained"/"damaged", reflecting situations in which a faint moisture film or a small discolored spot is present without a clear geometric defect.

 doi:10.20944/preprints202604.1475.v1

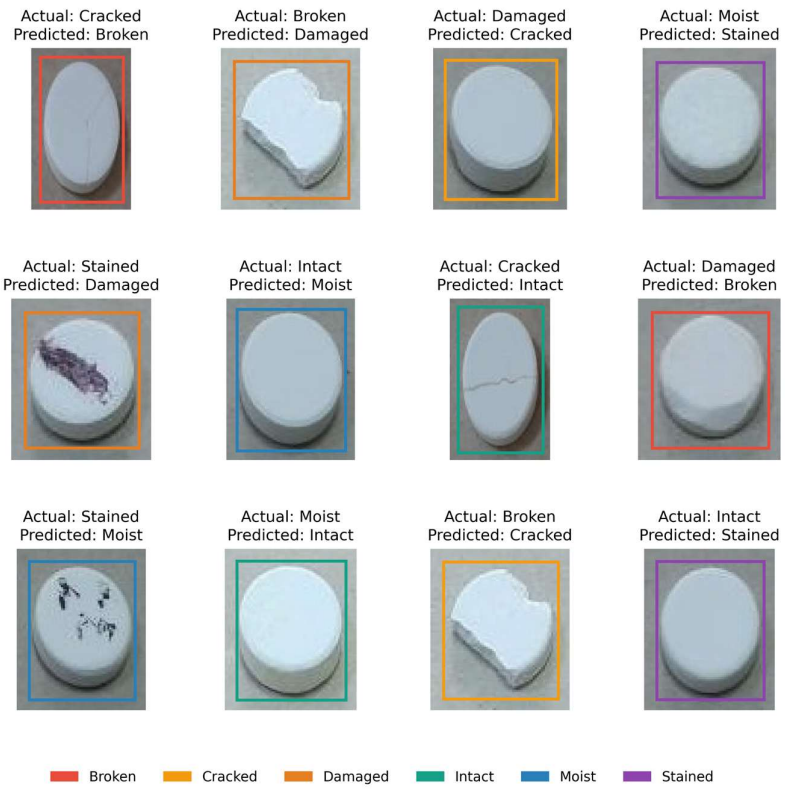


Figure 14. *Gallery of misclassified examples produced by YOLO11l on the effervescent tablet test set. Each panel shows a tablet along with its true label ("Actual") and the model's prediction ("Predicted"). Most errors occur between the semantically close categories "cracked", "damaged", and "broken".*

6. Discussion

The experimental results reported in Section 5 provide a comprehensive view of how recent YOLO families (YOLO11, YOLO12, and YOLO26) perform on a multi-class effervescent tablet defect dataset across all five standard scale variants. Several consistent and practically relevant observations emerge from the quantitative comparison (Table 4), the class-wise analysis (Table 5 and Figure 12), and the qualitative inspection (Figure 13 and Figure 14).

First, the YOLO11 family delivers the strongest overall accuracy–efficiency trade-off on the proposed dataset. YOLO11l achieves the highest mAP@0.5 (96.8%) and mAP@0.5:0.95 (91.7%), closely followed by YOLO11m and YOLO11x. Interestingly, further scaling from l to x does not yield a measurable accuracy gain, suggesting that YOLO11l represents the practical saturation point on this dataset. At the opposite end of the spectrum, YOLO11n achieves 95.6% mAP@0.5 at 345.9 FPS with only 2.5 M parameters and 6.4 GFLOPs, which is particularly attractive for embedded or edge-deployed inspection systems with constrained computational budgets. This observation is consistent with the findings of Khanam and Hussain [23], who also report that YOLO11 provides an improved efficiency frontier compared with earlier YOLO versions.

Second, the YOLO12 family achieves accuracy levels competitive with YOLO11 but incurs a markedly higher inference cost at larger scales. While YOLO12s and YOLO12m reach mAP@0.5 values of 96.7% and 96.3% at reasonable throughput, YOLO12l and YOLO12x drop to 69.4 FPS and 37.0 FPS, respectively. This behavior can be attributed to the attention-centric R-ELAN and area-attention modules introduced in YOLO12 [24], which improve feature representation but also raise the memory and latency footprint at large scales. For industrial applications where sustained line speed is critical, this trade-off is unfavorable compared with the YOLO11 variants of equivalent accuracy.

 doi:10.20944/preprints202604.1475.v1

Third, the YOLO26 family exhibits a more pronounced scale dependence than the other two families. YOLO26n substantially underperforms its YOLO11 and YOLO12 counterparts (88.0% mAP@0.5 vs 95.6%), with notably lower recall (74.8%), indicating a higher miss rate on small or low-contrast defects. However, as the scale increases, YOLO26 closes the gap rapidly: YOLO26x reaches 96.5% mAP@0.5, essentially matching YOLO11 and YOLO12 at the x scale. This pattern suggests that the newer YOLO26 design benefits disproportionately from additional capacity and may require larger training datasets to unlock its full potential in the nano regime.

Fourth, the class-wise analysis of the best-performing model (YOLO11l) reveals that the six defect categories are not equally difficult. As shown in Figure 12 and the per-class PR curves of Figure 10, the "stained" class achieves the highest mAP@0.5 (0.994) and perfect recall (1.000), likely because stained regions exhibit strong color contrast against the tablet surface. The "broken" class is also detected with high reliability (mAP@0.5 = 0.988) due to the pronounced geometric discontinuities that characterize broken fragments. In contrast, the "damaged" class is the most challenging category (mAP@0.5 = 0.940) because it encompasses heterogeneous structural deterioration that visually overlaps with both "cracked" and "broken" samples. Misclassifications between these three semantically related classes account for the majority of errors, as illustrated in the confusion matrix (Figure 11) and in the qualitative misclassification gallery (Figure 14). This observation highlights the inherent ambiguity in defining mutually exclusive defect categories and suggests that a refined annotation protocol—potentially with severity grading rather than discrete labels—could further improve performance.

Compared with prior pharmaceutical defect-detection studies summarised in Table 1, the present work contributes in three key ways. Whereas Ficzere et al. [10] and Diószegi et al. [19] reported strong results for film-coated and compressed tablets with earlier YOLO versions, and Rajappa et al. [17] focused on blister-pack defects, the proposed study is the first to systematically benchmark the recent YOLO11, YOLO12, and YOLO26 families on effervescent tablets—a product class for which even minor surface defects can trigger premature effervescent reactions due to hygroscopicity. Additionally, by evaluating all five scale variants under identical training conditions, this work offers the first comprehensive multi-scale accuracy–cost curve specifically tailored to pharmaceutical inspection, rather than reporting a single a priori chosen variant. This aligns with the multi-version benchmarking direction advocated by recent surveys on YOLO-based detection [32,33].

Despite these contributions, several limitations should be acknowledged. The dataset, while sufficient to support statistically meaningful comparisons, is still relatively small (251 images, 3,012 instances) and was acquired under fixed, controlled imaging conditions. Consequently, generalization to real production lines—where illumination, camera angle, and tablet orientation may vary significantly—cannot be guaranteed without further validation. The annotation protocol also distinguishes among semantically related defect classes (broken/cracked/damaged) that occasionally overlap, thereby introducing an upper bound on achievable accuracy. Finally, the study benchmarks three YOLO families on a single hardware configuration (RTX A5000); deployment on edge-class devices (e.g., NVIDIA Jetson Orin) may yield different relative rankings due to differences in memory bandwidth and tensor-core utilization [34].

Future work will focus on expanding the dataset with additional effervescent tablet formulations and imaging conditions, including variable lighting and production-line backgrounds, to assess cross-domain robustness. Further directions include exploring semi-supervised and self-supervised learning [41] to reduce annotation effort, integrating severity-aware multi-task heads that jointly predict defect class and severity, and deploying the best-performing YOLO variants on embedded edge devices for real-time in-line inspection.

7. Conclusion

This study presented a systematic multi-scale benchmark of three recent YOLO families (YOLO11, YOLO12, and YOLO26) for the detection of surface defects in effervescent tablets. A new

 doi:10.20944/preprints202604.1475.v1

dataset of 251 high-resolution images containing 3,012 manually annotated tablet instances across six defect categories (intact, damaged, cracked, broken, moist, and stained) was introduced and used to train and evaluate fifteen YOLO model variants under identical training and hyper-parameter conditions.

The experimental results demonstrate that YOLO11l provides the highest detection accuracy, with an mAP@0.5 of 96.8% and an mAP@0.5:0.95 of 91.7%, while YOLO11n offers the most favorable real-time trade-off, reaching 95.6% mAP@0.5 at 345.9 FPS with only 2.5 M parameters. YOLO12 variants deliver comparable accuracy to YOLO11 but incur a substantial latency penalty for larger scales due to their attention-centric design. YOLO26 variants perform competitively at the l and x scales but show a notable drop in accuracy at the nano scale, indicating a stronger dependence on model capacity. Class-wise analysis of YOLO11l confirms consistently high performance across all six defect categories, with mAP@0.5 values ranging from 0.940 (damaged) to 0.994 (stained).

From a practical standpoint, these findings provide concrete guidance for pharmaceutical manufacturers seeking to integrate deep learning into their quality-control pipelines: YOLO11l is recommended when maximum accuracy is prioritized, whereas YOLO11n is recommended when computational resources or line throughput are the primary constraints. Overall, the proposed multi-scale benchmark provides a reproducible baseline for future work on effervescent tablet inspection and, more broadly, demonstrates that modern YOLO variants are already mature enough to meet the stringent accuracy and speed requirements of in-line pharmaceutical quality assurance.

Author Contributions: Mustafa Yurdakul and Ahmet Melih Çakmak contributed equally to this work. Both authors were involved in the conceptualization, methodology, analysis, and writing of the manuscript. All authors read and approved the final manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study does not involve human participants or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study consists of effervescent tablet images collected under controlled laboratory conditions. The dataset is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Code Availability: All the codes of this study can be accessed by following the link: https://github.com/cakmakahmet/Multi-Scale-Performance-Benchmarking-of-YOLO-Models-for-Effervescent-Tablet-Defect-Detection.

References

1. U.S. Food and Drug Administration, “Guidance for Industry: PAT — A Framework for Innovative Pharmaceutical Development, Manufacturing, and Quality Assurance,” FDA, Rockville, MD, USA, 2004.
2. L. X. Liu, I. Marziano, A. C. Bentham, J. D. Litster, E. T. White, and T. Howes, “Effect of particle properties on the flowability of ibuprofen powders,” International Journal of Pharmaceutics, vol. 362, no. 1–2, pp. 109–117, 2008.
3. A. H. Sabri, C. N. Hallam, N. A. Baker, D. S. Murphy, and I. P. Gabbott, “Understanding tablet defects in commercial manufacture and transfer,” Journal of Drug Delivery Science and Technology, vol. 46, pp. 1–6, 2018.
4. E. Yost, P. Chalus, S. Zhang, S. Peter, and A. S. Narang, “Quantitative X-ray microcomputed tomography assessment of internal tablet defects,” Journal of Pharmaceutical Sciences, vol. 108, no. 5, pp. 1818–1830, 2019.

 doi:10.20944/preprints202604.1475.v1

5. D. L. Galata, O. Péterfi, M. Ficzere, B. Szabó-Szőcs, E. Szabó, and Z. K. Nagy, "The current state-of-the-art in pharmaceutical continuous film coating — A review," International Journal of Pharmaceutics, vol. 669, 125052, 2025.
6. M. Aslani and H. Jouyban-Gharamaleki, "Formulation development and stability assessment of effervescent tablets: A critical review," Drug Development and Industrial Pharmacy, vol. 49, no. 4, pp. 259–274, 2023.
7. M. Možina, D. Tomaževič, F. Pernuš, and B. Likar, "Automated visual inspection of imprint quality of pharmaceutical tablets," Machine Vision and Applications, vol. 24, no. 1, pp. 63–73, 2013.
8. D. Forcinio, "Machine-vision inspection: ensuring pharmaceutical product quality," Pharmaceutical Technology, vol. 42, no. 9, pp. 40–43, 2018.
9. X. Ma, N. Kittikunakorn, B. Sorman, H. Xi, A. Chen, M. Marsh, A. Mongeau, N. Piché, R. O. Williams III, and D. Skomski, "Application of deep learning convolutional neural networks for internal tablet defect detection: high accuracy, throughput, and adaptability," Journal of Pharmaceutical Sciences, vol. 109, no. 4, pp. 1547–1557, 2020.
10. M. Ficzere, L. A. Mészáros, N. Kállai-Szabó, A. Kovács, I. Antal, Z. K. Nagy, and D. L. Galata, "Real-time coating thickness measurement and defect recognition of film-coated tablets with machine vision and deep learning," International Journal of Pharmaceutics, vol. 623, 121957, 2022.
11. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
12. X. Zhao, W. Li, Y. Zhang, T. A. Gulliver, S. Chang, and Z. Feng, "A deep learning–based approach for visual inspection of industrial products," IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7654–7664, 2021.
13. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.
14. J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 7263–7271.
15. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
16. J. Terven, D.-M. Córdova-Esparza, and J. Romero-González, "A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS," Machine Learning and Knowledge Extraction, vol. 5, no. 4, pp. 1680–1716, 2023.
17. M. Rajappa, K. Kotecha, and A. Kulkarni, "Real-time visual intelligence for defect detection in pharmaceutical packaging," Scientific Reports, vol. 14, 18811, 2024.
18. S. Bandyopadhyay et al., "Multi-scale trajectory tracking of pharmaceutical defects with adaptive learning and precision counting," Journal of Pharmaceutical Innovation, 2025, doi: 10.1007/s12247-025-10359-z.
19. A. Diószegi, M. Ficzere, L. A. Mészáros, O. Péterfi, A. Farkas, D. L. Galata, and Z. K. Nagy, "Automated tablet defect detection and the prediction of disintegration time and crushing strength with deep learning based on tablet surface images," International Journal of Pharmaceutics, vol. 657, 124127, 2024.
20. J. Wei, J. Liang, J. Song, and P. Zhou, "YOLO-PBESW: A lightweight deep-learning model for the efficient identification of indomethacin crystal morphologies in microfluidic droplets," Micromachines, vol. 15, no. 9, 1136, 2024.
21. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 2023, pp. 7464–7475.
22. J. R. Arjoune, S. Bazi, and A. Alharbi, "Comparative performance evaluation of YOLOv5, YOLOv8, and YOLOv11 for solar panel defect detection," Preprints.org, 202501.0788, 2025.
23. R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," arXiv preprint arXiv:2410.17725, 2024.
24. Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-centric real-time object detectors," arXiv preprint arXiv:2502.12524, 2025.

 doi:10.20944/preprints202604.1475.v1

25. H.-C. Lin and S.-X. Xiao, "Development of a tablet defect detection model using a biaxial-plane discrete scanning algorithm," The International Journal of Advanced Manufacturing Technology, vol. 128, pp. 3265–3279, 2023.
26. A. Pathak, R. Chaudhary, and P. Singh, "Deep learning-based automatic detection of defective tablets in pharmaceutical manufacturing," in Proc. International Conference on Image Processing and Machine Intelligence, 2022, pp. 231–240.
27. Y. Zhang, H. Li, and F. Wang, "Automatic capsule surface defect detection based on deep convolutional neural networks," Journal of Intelligent Manufacturing, vol. 32, no. 6, pp. 1669–1681, 2021.
28. J. Yan, H. Liu, Y. Shen, and Q. Zhao, "Enhanced YOLOv8n with Mamba-like linear attention for irregular film-coated tablet inspection," Measurement, vol. 236, 115072, 2024.
29. J. Chen and Y. Wang, "Pill recognition via deep-learning approaches: a comparison of YOLOv3, YOLOv5, and YOLOv8," Mekatronika — Journal of Intelligent Manufacturing and Mechatronics, vol. 6, no. 1, pp. 21–33, 2024.
30. H. Kim, S. Park, and J. Lee, "Lightweight YOLOv7-based real-time capsule inspection on conveyor lines," Applied Sciences, vol. 13, no. 18, 10245, 2023.
31. Z. Wang, C. Zhang, and L. Chen, "Multi-scale YOLOv11 for printed-circuit-board defect detection," Electronics, vol. 14, no. 3, 512, 2025.
32. A. Vijayakumar and S. Vairavasundaram, "YOLO-based object-detection models: A review and its applications," Multimedia Tools and Applications, vol. 83, no. 35, pp. 83535–83574, 2024.
33. K. Gao, S. Chen, and W. Zhou, "A survey on deep-learning-based industrial surface-defect detection," IEEE Transactions on Industrial Informatics, vol. 20, no. 4, pp. 4832–4851, 2024.
34. S. Liu, Y. Chen, and X. Wang, "Edge-deployment of deep object detectors on NVIDIA Jetson devices: a benchmark study," Sensors, vol. 23, no. 9, 4512, 2023.
35. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.
36. J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.
37. C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," arXiv preprint arXiv:2402.13616, 2024.
38. A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-time end-to-end object detection," arXiv preprint arXiv:2405.14458, 2024.
39. C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object-detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022.
40. Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," arXiv preprint arXiv:2107.08430, 2021.
41. K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 596–608.