Academic Education Publishing House
Industry Science and Engineering Vol. 3 No. 8, 2026

Improved Yolov10 for Real-Time Multi-Scale Detection of Floor Contaminants in Precision Machining Workshops

Juhua Chen*, Fan Chen
Chongqing Industry and Trade Polytechnic, Chongqing, People's Republic of China
**Corresponding Author*

Abstract: **Floors in precision machining workshops accumulate contaminants that differ enormously in scale, from sub-centimetre metal swarf to metre-wide patches of spilled coolant. Fine debris scratches finished workpieces carried across it, while extensive oily films create slipping hazards and contaminate guide rails, yet routine inspection still relies on the human eye, which misses small debris and reacts late. YOLOv10 offers end-to-end inference without non-maximum suppression, but its detection levels and loss function are poorly matched to millimetre-sized targets lying against a cluttered floor. This paper makes three changes to YOLOv10s: DCNv4 deformable convolution is embedded at the neck fusion nodes so that the sampling grid follows the irregular outline of stains; a shallow P2 branch preserves the high-resolution detail of minute debris; and the CIoU loss is replaced by the WIoUv3 dynamic non-monotonic focusing loss to rebalance large and small instances. On a self-collected dataset of 3746 workshop floor images the improved model reaches 92.6% mAP@0.5, 4.2 percentage points above the baseline, and runs at 66 FPS with only 0.8 M extra parameters, which meets the real-time budget of embedded monitoring devices on the shop floor.**

Keywords: YOLOv10; Ground Pollutants; Multi-Scale Object Detection; Deformable Convolution; Real-Time Detection

1. Introduction

Cleanliness is a hard requirement on the shop floors where precision components are machined. Metal swarf, leakage from cutting-fluid systems and agglomerated dust settle on the floor throughout a shift: fine chips scratch workpieces carried across them, oily patches make walkways slippery and accelerate wear on machine-guide rails, and all of these effects lower yield and shorten equipment life. Inspection is still done largely by eye. Human observers miss a large share of sub-centimetre debris, and their reports arrive long after the contamination appears, so the feedback loop of workshop environment control cannot close in real time.

Single-stage detectors of the YOLO family provide a practical route to automatic floor surveillance. YOLOv10 removes the non-maximum-suppression stage at inference through a consistent dual-assignment head, which cuts end-to-end latency and suits edge controllers [1]. Applied directly to a workshop floor, however, the stock model runs into two difficulties. The contaminants span an extreme range of scales: a millimetre-sized chip occupies only a handful of pixels, and its features are attenuated by successive downsampling. The background is equally hostile, since tile joints, shadows cast by machine tools and specular reflections all resemble true targets and trigger false alarms [2,3].

Progress in generic object detection has been rapid. YOLOv9 alleviates the information bottleneck of deep networks with programmable gradient information and an auxiliary reversible branch, allowing lighter models to retain accuracy [4]. DCNv4 revisits the deformable convolution operator itself: dynamic sparse attention is recast as dense memory access with re-designed weight normalisation, so a dynamic receptive field comes at a much lower computational cost [5]. The vision transformer has shown that stacks of pure attention can carry the backbone, pushing detector design towards convolution–attention hybrids [6], and a systematic survey of the YOLO line confirms that multi-scale feature fusion and decoupled heads have become standard practice in real-time detectors [7]. All of these designs were tuned on generic benchmarks; none accounts for the mixture of extremely small targets and large, shapeless stains that a workshop floor presents.

Within industrial visual inspection, most improved YOLO variants address surface defects or safety monitoring. DN-YOLO fits strip-steel surface defects by means of deformable convolution [8], and RD-YOLO adds shallow detection branches for small objects in UAV imagery [9]. Attention modules and feature-pyramid refinements have likewise been applied to debris and stain recognition, but such models are usually heavy and weak at retaining the features of minute targets. Detecting millimetre swarf and metre-scale coolant films in the same frame remains largely unaddressed, and no purpose-built detector exists for the multi-scale floor contamination of precision workshops.

Against this background, the present work builds on YOLOv10 and adjusts it in three places: a DCNv4 deformable-convolution block is embedded in the neck so that the sampling grid can follow the irregular outline of stains; a shallow P2 branch is added to keep the high-resolution detail of small chips; and the CIoU loss is replaced by the WIoUv3 dynamic non-monotonic focusing loss to rebalance large and small instances. A floor-contaminant dataset covering swarf, coolant stains and dust agglomerates is constructed and augmented, the value of each module is verified through ablation, and the finished model is deployed on an edge device for field testing.

2. Improved YOLOv10 Network

2.1 Baseline Architecture

YOLOv10 consists of a backbone, a neck and a detection head. The backbone extracts multi-scale features with C2f, SCDown and PSA blocks, a PAN–FPN neck fuses shallow and deep features bidirectionally, and the head is trained with one-to-many and one-to-one label assignments so that NMS disappears at inference [1]. Gradient flow is split inside C2f to balance fine shallow detail against deep semantics, SCDown trims redundant computation at the downsampling stages, and the PSA block placed after SPPF introduces position-sensitive attention for global context modelling [7]; the gradient-splitting blocks inherit from the programmable gradient information of YOLOv9 [4]. With 7.2 M parameters and 21.6 GFLOPs, YOLOv10s reaches 46.7% mAP@0.5:0.95 on COCO, and the architecture has already generalised well to domains such as medical image analysis [10].

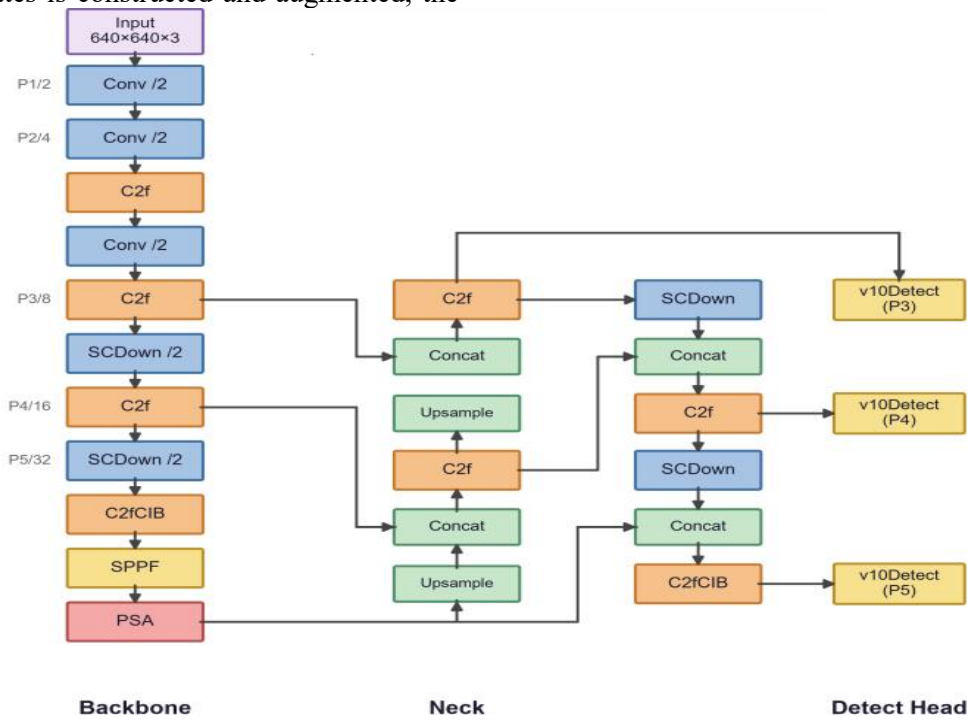


Figure 1. Architecture of the Baseline YOLOv10 Detector

The stock network emits three feature maps, P3 to P5, matched to medium and large objects, as sketched in Figure 1. On a workshop floor most swarf is far smaller: by the P3 stage its detail has already been diluted. Coolant stains have no fixed outline, which a fixed sampling grid copes with poorly. Moreover, the original CIoU loss is scale-sensitive, and because large stains

outnumber the other categories, training drifts towards big boxes and the small chips end up under-fitted.

2.2 Network Modification

Three changes follow from the analysis above, acting on feature extraction, feature preservation and loss design: DCNv4 replaces the standard convolution at the neck fusion nodes; a P2 branch carries the stride-4 feature map to a fourth detection head; and WIoUv3 supervises box regression. The fusion paths are rearranged to match the extra scale.

2.2.1 DCNv4 deformable convolution in the neck

To follow the irregular contours of contaminants, DCNv4 blocks are inserted at the up-sampling and down-sampling fusion nodes of the neck. The operator predicts sampling offsets and aggregation weights on the fly, abandoning the fixed grid of ordinary convolution: the receptive field stretches along the edge of an oily patch or tightens around a chip, while tile joints and shadows are treated as background. The block keeps the cross-stage-partial split-and-concat pattern; its main branch stacks DCNv4NeXt transformations and the shortcut branch passes the raw features through, so the cost of the deformation machinery stays small. The sampling procedure is drawn in Figure 2.

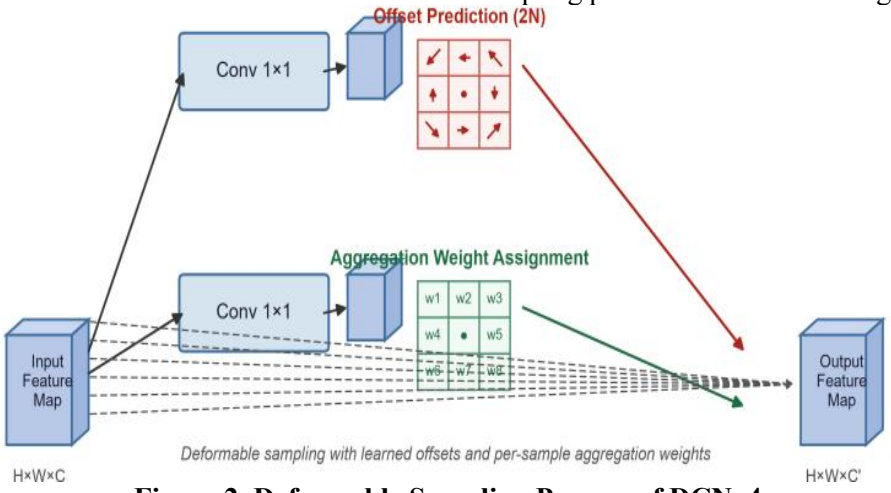


Figure 2. Deformable Sampling Process of DCNv4

Formally, the aggregation of sampled values reads

$$y(p_0) = k = 1\sum_{K} x(p_0 + p_k + \Δ p_k)\, A_{w_k} \quad (1)$$

where x denotes the input feature, p0 the output location, p_k the k-th point of the regular grid, Δpk the learned offset, A_{w_k} the spatial aggregation weight, wk the convolution weight, τ a scaling factor and K the number of sampling points per group, set to 9 here. The weight decomposes into a per-sample term predicted together with the offsets by the parallel 1×1 convolution of

$$A_{w_k} = \frac{A_{p_k}\, w_k}{\tau} \quad (2)$$

in which A_{pk} is predicted from the input feature and σ(·) is a point-wise mapping.

Three fusion nodes, two on the up-sampling path and one on the down-sampling path, receive the block, with 3×3 kernels, four offset groups (G = 4) and output channels of 128/256 matched to the original concatenations; τ is set to the reciprocal of the group size to steady the gradients. Unlike DCNv3, which normalises its weights with softmax, DCNv4 drops that constraint and expands the sparse sampling into a dense memory-access pattern, which raises parallelism and inference speed, a property that matters for embedded deployment on the shop floor.

2.2.2 Shallow detection branch for small objects

The finest feature map of the baseline has a stride of 8, whereas a millimetre chip occupies fewer than 8×8 pixels in a 640×640 frame, so its features barely survive downsampling. A P2 branch taken from the second stage of the backbone, with a stride of 4 and hence a 160×160 map for 640×640 input, therefore feeds a fourth head alongside P3 to P5. The branch joins the PAN path through bidirectional fusion with the P3 level. To restrain the cost, its head uses half the channels of the P3 head, and DCNv4 fusion nodes are reused on its up-sampling path. The addition costs about 2.1 GFLOPs and 0.6 M parameters but preserves the features of sub-8-pixel targets intact.

2.2.3 WIoUv3 dynamic focusing loss

CIoU penalises the aspect ratio geometrically and treats every sample alike; since large stains

dominate the dataset, training gravitates towards them. The regression loss is therefore switched to WIoUv3, which attaches a distance-wise attention to the IoU term so that small, poorly localised boxes receive proportionally more gradient [11–13]. Its building blocks are

$$R_{WIoU} = exp(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(g_2^W+g_2^H)^*}) \quad (3)$$

$$L_{WIoUv1} = R_{WIoU} \cdot L_{IoU} \quad (4)$$

where (x, y) and (x_{gt}, y_{gt}) are the centres of the predicted and ground-truth boxes, Wg and Hg the width and height of their minimal enclosing box, and the superscript * marks quantities detached from the backward pass so that no harmful gradient blocks convergence [11–13]. A dynamic non-monotonic focusing coefficient then rescales the loss:

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \quad (5)$$

$$\beta = \frac{IoU_*^L}{\overline{L_{IoU}}} \quad (6)$$

in which β is the outlier degree of an anchor and the barred term a momentum average of the IoU loss over the mini-batch; α and δ are hyper-parameters, fixed at 1.9 and 3, the setting under which an anchor with β = δ receives unit gain. Whereas CIoU imposes a static geometric penalty through the aspect ratio, WIoUv3 assigns gradient gain by outlier degree, muting the harmful gradients of low-quality labels and concentrating the effort on anchors of ordinary quality; it also cooperates with the distribution-based regression supervision of the head to sharpen localisation.

The overall objective combines the three supervision terms as

$$L = \lambda_1 L_{WIoUv3} + \lambda_2 L_{cls} + \lambda_3 L_{DFL} \quad (7)$$

with L_{cls} the classification loss, L_{WIoUv3} the focusing regression loss above, L_{DFL} the distribution focal loss and λ_i dynamic weights. The weighting raises the share of the tiny chips in the total loss, offsetting the imbalance in the instance distribution.

Putting the pieces together: a 640×640 floor image enters the backbone; the neck reinforces the multi-scale features through the DCNv4 nodes; the P2 branch keeps the fine detail of the swarf; and the four levels P2 to P5 reach the one-to-one head, which emits classes and box coordinates without any NMS stage. The complete architecture appears in Figure 3.

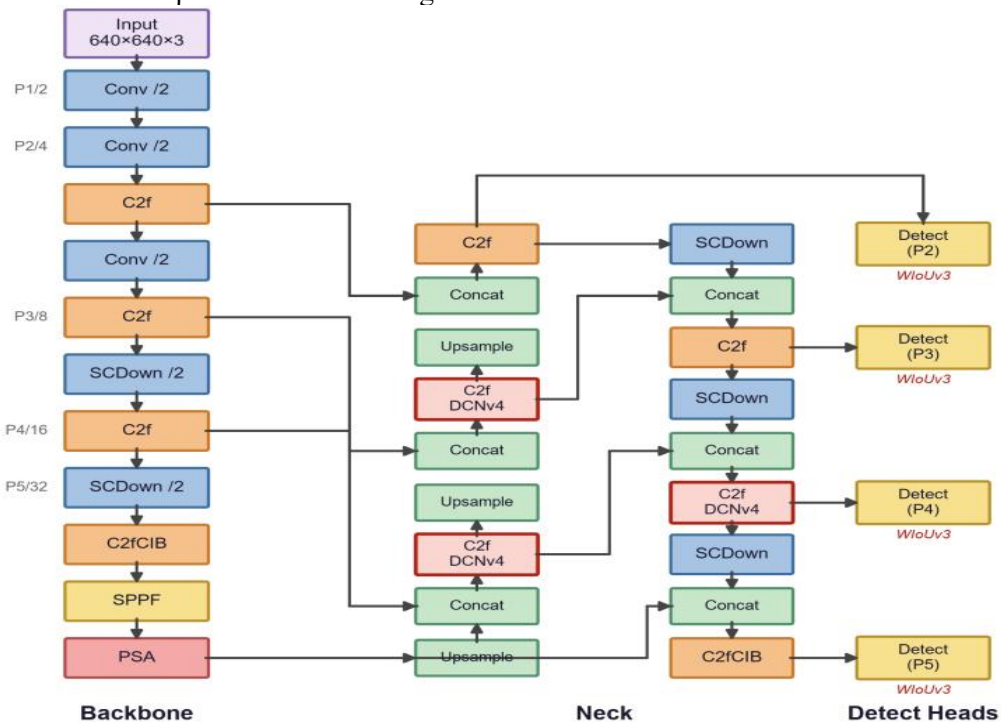


Figure 3. Architecture of the Improved YOLOv10

3. Dataset Construction and Experimental Setup

3.1 Dataset

Images were captured in a precision component plant. Eight cameras were mounted above walkways and machining stations, 2.2–2.6 m above the floor at a depression angle of about 60°, covering day shifts, night shifts and the light transitions between them. Blurred, over-exposed and target-free frames were discarded,

leaving 3746 valid images with three labelled categories: metal swarf (chip), coolant stain (stain) and dust agglomerate (dust). Labelling was done in LabelImg with PASCAL VOC format: chips were boxed with their minimal enclosing rectangle, contiguous coolant films were boxed as a whole, and dust piles were boxed along the fringe of the covered area. Two annotators cross-checked every image, keeping box-position errors within four pixels. The data were split 7:2:1 into training, validation and test sets of 2622, 749 and 375 images, with the category and scale distribution kept consistent across the subsets, following the construction protocol of the FLAME aerial wildfire dataset.

Mosaic composition, random flipping, brightness jitter, Gaussian blur and random scaling simulate camera shake and the lighting swings of a real shift. The collection holds 1847 small instances, 2306 medium ones and 1542 large ones, so all three scales are well represented; the split is visualised in Figure 4.

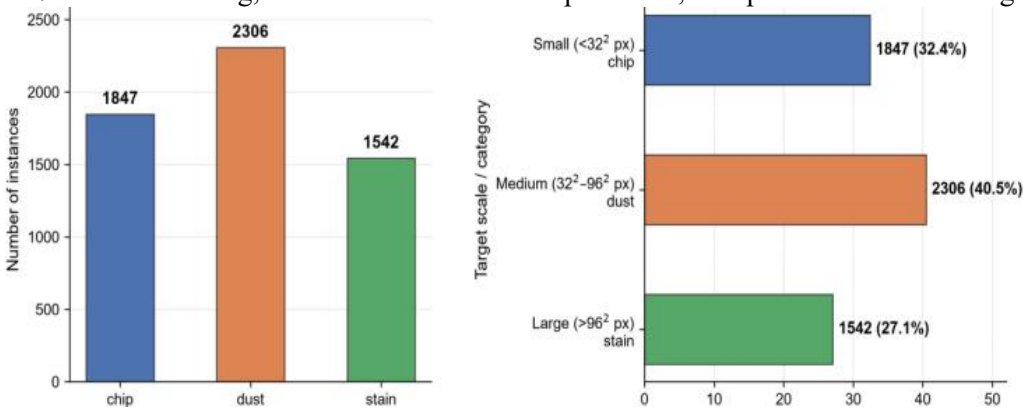


Figure 4. Dataset Statistics: (a) Instances per Category; (b) Scale-wise Instance Proportion

The three classes differ sharply in appearance, as Table 1 records. Swarf appears as silver-grey slivers or grains with bright specular edges and occupies very few pixels. Coolant stains spread as milky or pale-yellow translucent films with soft, reflection-contaminated boundaries. Dust piles form loose grey-black flocs without a stable outline and fall between the other two in scale.

Table 1. Characteristics of the Three Contaminant Categories

Category	Typical appearance	Pixel size	Instances
chip	silver-grey slivers or grains, specular edges	6×6 to 60×60	1847
stain	milky or pale-yellow films, blurred boundary	100×80 to 640×480	1542
dust	grey-black loose flocs	40×40 to 300×300	2306

3.2 Training Configuration and Evaluation Metrics

All experiments ran on an Intel Core i7-12700 with an NVIDIA GeForce RTX 3070 (8 GB) and 32 GB of memory, under Ubuntu 20.04, Python 3.8, PyTorch 1.13.1 and CUDA 11.7. Training used SGD with an initial learning rate of 0.01, momentum of 0.937 and weight decay of 0.0005, a batch size of 16, 300 epochs and 640×640 inputs; the first three epochs warmed up, and mosaic augmentation was switched off over the final ten epochs to protect convergence.

Performance is reported with the usual detection measures. Precision counts the fraction of correct positive predictions, recall the fraction of ground-truth objects actually found, AP is the area under the precision–recall curve, and mAP@0.5 averages AP over the classes at an IoU threshold of 0.5; the parameter count (M) and the frame rate (FPS) describe model complexity and real-time behaviour:

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

$$AP = 0\int_{1} P(R)\, dR \tag{10}$$

$$mAP = \frac{1}{N} i = 1\sum_{N} AP_i \tag{11}$$

where TP, FP and FN denote true positives, false positives and false negatives, and N is the number of classes.

3.3 Comparison with Mainstream Detectors

The improved model was benchmarked against YOLOv8s, YOLOv9s and the unmodified YOLOv10s on the same test split. YOLOv8s is a widely used deployment baseline, YOLOv9s leans on programmable gradient information for multi-scale feature learning, and YOLOv10s is the starting point of this work. All four were trained to convergence under identical settings,

so the comparison is fair; Table 2 lists the outcome.

Table 2. Detection Performance of Different Models

Model	Precision/%	Recall/%	mAP@0.5/%	FPS	Params/M
YOLOv8s	87.1	84.9	88.3	53	11.2
YOLOv9s	89.4	87.2	89.8	47	20.1
YOLOv10s	90.0	88.5	88.4	62	8.4
improved YOLOv10s	93.3	91.7	92.6	66	9.2

Three observations stand out. First, mAP@0.5 rises by 4.2 percentage points over the baseline, with recall up by 3.2 points, so missed swarf, the main failure mode, drops markedly, and precision climbs by 3.3 points as tile joints and shadows stop triggering false alarms. Second, the frame rate reaches 66 FPS, showing that the NMS-free design survives the extra modules almost untouched. Third, the parameter budget grows by only 0.8 M. Against YOLOv8s the improved model is 4.3 points more accurate and 13 FPS faster while being 2.0 M smaller; YOLOv9s, at 20.1 M parameters, is more than twice the size and still 2.8 points behind. The improved variant tops all four performance columns of Table 2, which indicates a reasonable balance between accuracy and speed.

3.4 Ablation Study

Each module was validated by progressive stacking on YOLOv10s: DCNv4 first, then the P2 branch, then WIoUv3, as reported in Table 3. Configurations other than the added modules were identical to those of Section 3.2.

Table 3. Ablation Results

Model	DCNv4	P2 branch	WIoUv3	mAP@0.5/%	FPS
baseline (YOLOv10s)	×	×	×	88.4	62
A	√	×	×	89.7	64
B	√	√	×	91.3	65
C	√	√	√	92.6	66

DCNv4 alone contributes 1.3 points, confirming that an adaptive receptive field fits irregular contours better; the P2 branch adds another 1.6 points, evidence that the stride-4 features retain the detail of millimetre chips; and WIoUv3 contributes a further 1.3 points by rebalancing large and small instances. The gains accumulate to 4.2 points while the frame rate stays between 62 and 66 FPS and the model grows only from 8.4 M to 9.2 M parameters. No module duplicates the effect of another, and the three together account for the overall improvement.

3.5 Detection Results

Per-class AP breaks down as 91.1% for swarf, 94.4% for coolant stains and 92.3% for dust. The baseline manages only 85.8% on swarf, the hardest class, so the shallow branch and the focusing loss act exactly where they were intended to; stains, which are already easy, gain slightly, and dust moves accordingly. Figure 5 compares the precision–recall curves: the baseline curves collapse inwards and decay quickly at high recall (Figure 5a), whereas the improved curves stay flatter and enclose visibly more area (Figure 5b), consistent with the 4.2-point mAP margin of Table 2.

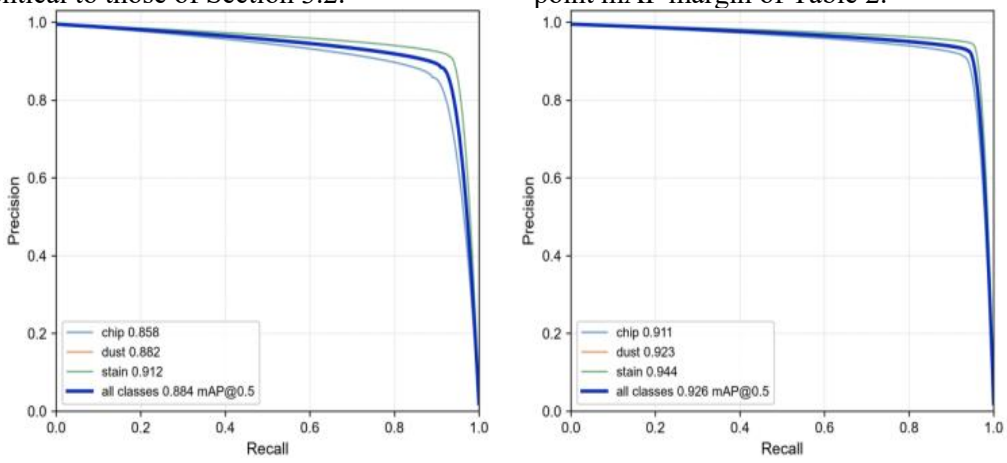


Figure 5. Precision–Recall Curves of (a) the Baseline and (b) the Improved Model

4. Conclusion

This paper has addressed the mixed-scale floor contamination of precision machining shops with a modified YOLOv10s. DCNv4 deformable convolution in the neck learns the spatial form of irregular stains, a shallow P2 branch preserves the detail of millimetre swarf, and the WIoUv3 focusing loss counters the imbalance between large and small instances. On a purpose-built dataset the model reaches 92.6% mAP@0.5 at 66 FPS with 9.2 M parameters, 4.2 points above the baseline, and runs comfortably on edge hardware, giving the plant a practical means of

locating and classifying swarf, stains and dust in real time. Future work will extend the category set, add low-light night-shift samples, and apply pruning and quantisation for an even leaner edge deployment.

Acknowledgments
Chongqing Industry and Trade Polytechnic School-level Scientific Research Project (Project Number: ZR202626).

References

[1] Wang, L., Fang, H.: 'Research on UAV detection algorithm for low-altitude small targets based on improved YOLOv10', Science and Technology Innovation, 2026, (17), pp. 120–123.
[2] Shamsoshoara, A., Afghah, F., Razi, A., et al.: 'Aerial imagery pile burn detection using deep learning: the FLAME dataset', Computer Networks, 2021, 193, p. 108001.
[3] Padilla, R., Passos, W.L., Dias, T.L.B., et al.: 'A comparative analysis of object detection metrics with a companion open-source toolkit', Journal of Imaging, 2021, 7, (8), p. 119.
[4] Wang, C.-Y., Yeh, I.-H., Liao, H.-Y.M.: 'YOLOv9: learning what you want to learn using programmable gradient information', Proc. European Conf. Computer Vision (ECCV), 2024, pp. 1–18.
[5] Xiong, Y., Liao, H., Li, X., et al.: 'Efficient deformable ConvNets: rethinking dynamic and sparse operator for vision applications', Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2024.
[6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: 'An image is worth 16×16 words: transformers for image recognition at scale', Proc. Int. Conf. Learning Representations (ICLR), 2021.
[7] Terven, J., Cordova-Esparza, D.M., Romero-Gonzalez, J.A.: 'A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS', Machine Learning and Knowledge Extraction, 2023, 5, (4), pp. 1680–1716.
[8] Zhang, S., Hu, Y., Zhou, J.: 'Strip steel surface defect detection algorithm based on DN-YOLO', Shanxi Metallurgy, 2026, 49, (7), pp. 127–130.
[9] Wu, S., Yu, L.: 'RD-YOLO: an improved YOLOv11 algorithm for small object detection in UAV imagery', Journal of Harbin University of Commerce (Natural Science Edition), 2026, 42, (4), pp. 419–429.
[10] Nandal, P., Gaur, V., Bhutani, M., et al.: 'Automated urine sediment microscopic image analysis using YOLOv10 with chimpanzee leader election optimization', Journal of Visualized Experiments, 2026, (233).
[11] Wu, Z., Li, H., Xue, R.: 'Lightweight PCB defect detection model based on improved YOLO', Journal of Southwest Minzu University (Natural Science Edition), pp. 1–13, 2026.
[12] Zheng, Z., Wang, P., Liu, W., et al.: 'Distance-IoU loss: faster and better learning for bounding box regression', Proc. AAAI Conf. Artificial Intelligence, 2020, 34, (7), pp. 12993–13000.
[13] Li, X., Wang, W., Wu, L., et al.: 'Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection', Advances in Neural Information Processing Systems, 2020, 33, pp. 21002–21012