applied sciences

MDPI

Article

Lightweight Road Segmentation and Multiscale YOLO11 for Street-Sweeper Litter Detection

Qingbing Zeng [1,*], Zihua Chen [1], Zhongwen Chen [1], Yixiao Zhang [1], Minghui Wu [1] and Heng Yang [2]

[1] School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; m315124339@sues.edu.cn (Z.C.); m315124308@sues.edu.cn (Z.C.); m315124237@sues.edu.cn (Y.Z.); wuminghui@sues.edu.cn (M.W.)
[2] Hunan Niu En Chi New Energy Vehicle Co., Ltd., Majiahe, Zhuzhou 412007, China; chenhui8044@163.com
* Correspondence: qingbingzeng@sues.edu.cn

Featured Application

Camera-based street-sweeper litter perception with semantic cleaning-region selection. Runtime is measured on a desktop CPU and Jetson Orin Nano Super under explicitly defined workloads.

Abstract

Street-sweeper litter perception requires object localization and an explicit road-region selection rule. We evaluate MobileNetV2-based DeepLabv3+ with a four-scale YOLO11n detector incorporating P2 prediction, scale-sequence fusion and channel-and-position attention. A predicted class-index mask selects full-image detections by road coverage. A new date-grouped protocol uses 100 training, 38 validation and 142 test images. Six controlled configurations are each trained with three seeds, separating Zoom_cat, ScalSeq and P2-subpath effects. The joint configuration obtains a mean AP50 of 16.38%, lower than 35.18% for standard YOLO11n under this fixed budget. On 233 public road images, the MobileNetV2 checkpoint achieves road IoU of 70.18%, road Dice of 82.48%, binary mIoU of 78.09% and pixel accuracy of 89.47%, using 89.37% fewer parameters than the evaluated Xception checkpoint. Retrospective region selection removes both false and true detections. On Jetson Orin Nano Super with TensorRT 10.3, the FP16 serial pipeline takes 80.45 ms per frame (12.43 FPS), or 10.00 FPS including file decoding, with mean module-input power of 9.78 W under the recorded workload. The results identify trade-offs in detector design, road segmentation, and region selection under the evaluated protocols.

Keywords: street sweeper; road litter detection; semantic segmentation; DeepLabv3+; YOLO11; multiscale feature fusion

Check for updates

Academic Editor: Yutaka Ishibashi

Received: 24 October 2026
Revised: 16 November 2026
Accepted: 17 November 2026
Published: 20 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Street sweepers operate in scenes where the presence of visible litter does not necessarily imply that an item belongs to the area being cleaned. Leaves, paper, bottles, and other small objects may appear on the carriageway, on pavements, or within roadside planting. A forward-facing camera observes these regions together. For a perception system intended to support sweeping operations, recognizing an object and identifying its relevant spatial context are therefore related requirements. The visual output should distinguish roadway litter from similar objects outside the operational region.

Small-object recognition presents an additional difficulty. Distant litter occupies few pixels, and thin or irregular objects provide weak boundaries against textured road surfaces.

Repeated feature downsampling can remove details needed to distinguish these objects from cracks, shadows, and markings. Increasing the amount of high-resolution processing can help preserve detail, but it also enlarges intermediate feature maps. A practical vehicle system must consequently consider the detector's representation and the segmentation network's storage demands together.

Previous research has connected visual garbage recognition to road-sweeper operation. Donati et al. investigated a deep-vision approach to garbage detection for an energy-saving road sweeper [1]. Gomez et al. combined litter identification and adaptive cleaning in a pavement-sweeping robot [2]. These studies demonstrate the relevance of linking visual perception with cleaning equipment. Within this application, scene interpretation and object localization provide complementary information: the former describes the road environment, while the latter identifies the litter to be considered by the vehicle system.

This paper addresses these requirements through a two-network perception architecture. A lightweight DeepLabv3+ model estimates the road region, and the proposed multiscale YOLO11n model localizes litter. The detector processes the full image independently of segmentation. A subsequent region-selection stage is evaluated both with the original color-derived mask and with a revised direct class-index mask. The detector adds a higher-resolution prediction scale, aligned fusion of different feature levels, and attention refinement to the YOLO11n framework. Comparative experiments and cumulative ablation evaluate how these adaptations affect litter-detection accuracy.

The contribution is a task-specific multiscale architecture and an explicit segmentation-to-detection selection rule. One ScalSeq branch aggregates backbone features at P3 resolution; the other combines a P2 backbone feature with refined neck features at P2 resolution. Channel-and-position attention connects each fused representation to the corresponding prediction path, and the detector predicts at P2–P5. Segmentation then selects full-image detections by their coverage within the designated cleaning-road region. The operators are adopted from established methods, while their placement and integration are tailored to the litter-perception task. The evaluation separately examines detector configuration, road-region selection, and computational cost. The original cumulative comparison characterizes the construction sequence. A new date-grouped, three-seed factorial experiment separately estimates conditional Zoom_cat, ScalSeq and P2-subpath effects.

2. Related Work

2.1. Road-Litter Perception and Operational Context

Litter recognition spans object detection, scene understanding, and application-specific decision making. A road sweeper needs the surrounding road structure as well as the position of individual items. Deep-vision sweeper research [1] and adaptive pavement-cleaning research [2] illustrate how a visual recognition component can be incorporated into a larger operational system. For the present task, the road estimate supplies scene context, while the detector supplies object locations. Their combination is motivated by the difference between objects visible to the camera and objects relevant to the road-cleaning area.

Public litter datasets also demonstrate the appearance diversity of the problem. TACO provides litter annotations in contextual scenes [3], while pLitterStreet addresses street-level plastic litter detection and mapping [4]. Their camera viewpoints, object composition, and annotation objectives provide useful context for application-specific dataset design. Our dataset concentrates on a sweeper's forward view and typical litter encountered during road operations, with images collected under four weather and illumination conditions.

RoLID-11K provides a recent dashcam dataset for small roadside litter, emphasizing sparse targets, cluttered road verges, and size and frequency imbalance [5]. Its vehicle viewpoint is relevant to sweeper perception, although roadside monitoring and cleaning

applications can differ in viewing geometry and target relevance. This work complements TACO and pLitterStreet in defining the context of vehicle-based litter recognition.

2.2. Lightweight Semantic Segmentation

Semantic segmentation assigns spatial labels, whereas detection localizes discrete objects. Minaee et al. review fully convolutional prediction, encoder–decoder reconstruction, multiscale context, and attention in segmentation [6]. These mechanisms provide a foundation for road-region estimation, whose extended spatial support complements the localized representations needed for individual litter items.

DeepLabv3+ combines atrous spatial pyramid pooling with an encoder–decoder structure to recover spatial detail after contextual feature extraction [7]. The architecture is suitable for separating an extended road surface from its surroundings because both local boundaries and larger scene context are relevant. However, a segmentation backbone that is convenient on a workstation can impose substantial storage demands in a vehicle deployment.

MobileNetV2 uses inverted residual blocks, linear bottlenecks, and depthwise separable convolutions to build compact feature extractors [8]. We use it as the DeepLabv3+ backbone in place of Xception to reduce the segmentation model's storage footprint while preserving the encoder–decoder organization. The backbone comparison reports storage, counted operations, and pixel-level quality on a common public road-image reference set.

PIDNet explicitly separates detail, context, and boundary processing for efficient segmentation [9]. This differs from replacing the backbone of an existing DeepLabv3+ decoder. The present storage comparison is therefore insufficient to establish an accuracy–latency advantage over dedicated efficient segmentation networks; that comparison requires common labels, evaluation images, and hardware measurements.

2.3. Multiscale Detection and Attention

The YOLO family provides the detector background. YOLOv3 predicts at multiple feature resolutions [10], and YOLOv4 examines architectural and training choices for balancing accuracy and speed [11]. YOLOv5 provides an established training and deployment implementation [12]. Together, these developments motivate examining prediction resolution and fusion within an executable detection framework.

Later families refine different parts of this framework. YOLOv8 provides a widely used software implementation [13]. YOLOv9 introduces programmable gradient information and GELAN for efficient learning and representation [14], while YOLOv10 investigates end-to-end detection without conventional non-maximum suppression at inference [15]. These references identify the detector families; they do not establish that the original comparison used identical training settings, which remain to be linked to its run records.

YOLO11 provides an established detection framework and deployment ecosystem [16]. Feature pyramids and path aggregation offer general foundations for exchanging shallow spatial detail and deeper semantic information [17,18]. For litter, the relevant tension is that shallow maps preserve small structures but provide limited semantic discrimination, whereas deeper maps can distinguish object appearance while losing spatial detail. Connecting these levels is therefore central to the detector design.

Dense prediction also encounters foreground–background imbalance. Focal loss reduces the influence of easy background examples during training [19], illustrating an optimization issue distinct from feature resolution. Our adaptations concentrate on resolution, fusion, and attention to improve the representation of sparse litter against pavement texture.

Zoom_cat and ScalSeq are adopted from the scale-fusion ideas of ASF-YOLO [20] and adapted within the present YOLO11n architecture. Our related work on manhole-cover

recognition for refuse collection vehicles [21] also considers the onboard visual setting, with an emphasis on cover-condition recognition. Here, the focus is on road litter and the integration of road context with object localization. Channel and spatial weighting follow established attention formulations [22]. The methodological contribution lies in adapting and combining these building blocks for the sweeper's perception task.

ASF-YOLO contributes scale–sequence fusion in its original cell-segmentation setting [20]; those operators are adopted here, rather than introduced as new primitives. RT-DETR instead uses an efficient hybrid encoder and query-based end-to-end detection [23]. Our configuration retains dense YOLO11n predictions and introduces a stride-four branch. Its proposed advantage is the preservation of spatial detail for road litter, with an associated increase in activation size. No superiority over RT-DETR, dedicated segmentation networks, or alternative fusion designs is established without matched experiments.

Table 1 compares the design choices relevant to the present application. It separates a rationale for selecting operators from evidence of their effectiveness under a common protocol.

Table 1. Qualitative design comparison. This table is not a matched accuracy or runtime benchmark.

Approach	Relevant Design Distinction	Implication and Evidence Boundary
DeepLabv3+/MobileNetV2 [7,8]	Encoder–decoder segmentation with a compact feature extractor.	Motivates the storage choice; file size does not establish boundary accuracy or device speed.
PIDNet [9]	Explicit detail, context, and boundary branches.	A different efficient-segmentation design; this study does not include a matched comparison of road masks.
ASF-YOLO [20]	Scale–sequence fusion and attention in cell instance segmentation.	Source of adopted operators; transfer into YOLO11n is a task adaptation, not a new primitive.
RT-DETR [23]	Hybrid encoding and query-based end-to-end detection.	Differs from the present dense P2–P5 detector; no accuracy or speed ranking is inferred.
Vehicle-view litter data [3–5]	Different viewpoints, object distributions, and annotation objectives.	Relevant to external validity; the present aggregate scores do not demonstrate transfer to these datasets.

2.4. Task-Specific Waste Detection

Attention and temporal association have also been investigated for waste recognition. Ma et al. developed DSYOLO-trash, combining attention-assisted solid-waste detection with object tracking [24]. Zailan et al. investigated an optimized YOLO model for floating-garbage detection [25], where water surfaces and target appearances differ from road pavement and roadside backgrounds. These studies illustrate the value of compact detection and feature refinement across waste-management applications. For a street sweeper, the particular challenge is to represent small litter together with the surrounding road structure. We therefore combine road segmentation with multiscale detection and compare detector configurations on a common task-specific dataset.

Waste segmentation provides another relevant perspective. Wang et al. combine scene-, object-, and pixel-level processing and introduce the MJU-Waste RGBD dataset [26]. Haldorai et al. develop an SSD-based visual component for a water-surface cleaning robot [27]. Their segmentation objectives, optional depth information, and aquatic backgrounds differ from road litter, reinforcing the need to relate a recognition architecture to its operating scene.

3. Materials and Methods

3.1. Overall Perception Framework

The segmentation and detection networks operate separately on the same image. The detector processes the full image without masking its input or modifying its features using segmentation. Region selection is applied after detection. The revised implementation uses the predicted class-index tensor directly; the original visualization-based implementation instead extracts a selection mask from a blended image. Figure 1 distinguishes the revised data path and the legacy comparator. Neither implementation jointly trains the networks or issues a brush-control command.

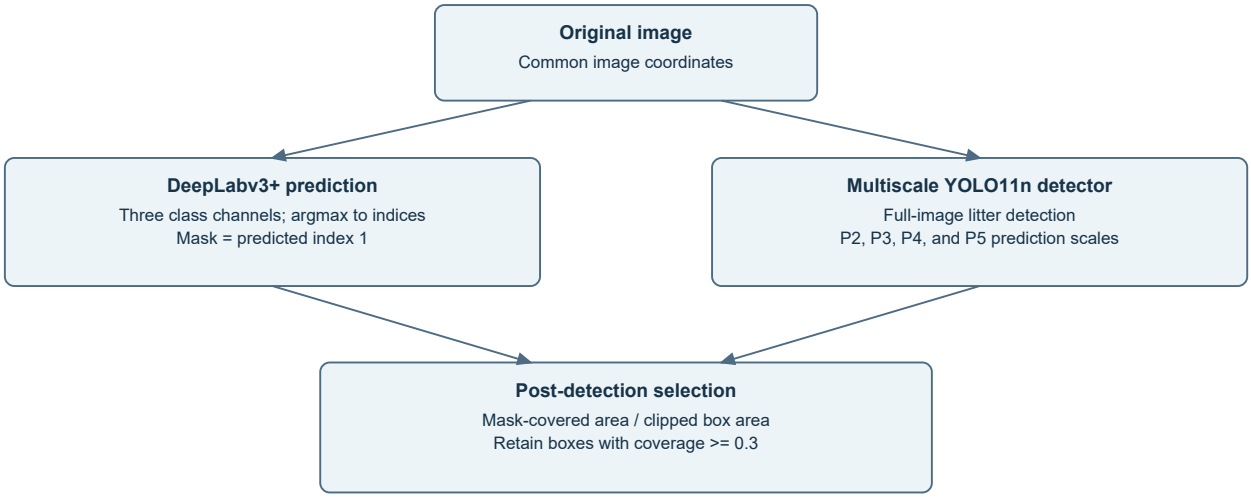

Figure 1. Revised inference path. A direct predicted class-index mask selects full-image detector boxes after inference. The original HSV mask is retained only as a separately evaluated comparator.

For class probabilities $p_k(u, v)$, the revised selection mask for the prototype's region index 1 is

$$M_1(u, v) = \mathbf{1}\left[\arg\max_{k\in\{0,1,2\}} p_k(u, v) = 1\right]. \quad (1)$$

A predicted box is clipped to the image and retained if its mask-covered fraction satisfies

$$\rho(B; M) = \frac{\sum_{(u,v)\in B} M(u, v)}{|B|} \geq 0.3. \quad (2)$$

Degenerate clipped boxes are rejected. Clipped coordinates are truncated to integer pixel bounds, and the numerator and denominator use the same bounds. The denominator is the clipped box area; this is a coverage rule, not box–mask IoU or a box-center test. The 0.3 setting is retained from the prototype rather than tuned to the diagnostic results. Class index 1, shown in red, denotes the designated road area to be cleaned. Class index 2 is excluded from the selection mask. This image-space task definition does not establish calibrated three-dimensional brush reach.

For the legacy comparator, M is obtained by HSV thresholding of the 0.7-alpha blended segmentation visualization. Red hue intervals are 0–10 and 160–180, with saturation and value at least 100. A computed green mask is not used by that test. The direct-index revision removes dependence on recovering labels from visualization colors, including the original image colors. The paired evaluation in Section 4.4 quantifies both implementations.

The supplementary evaluation uses the 33-layer multiscale detector described in Section 3.3, with the confidence and NMS settings specified in each protocol. Appendix A lists its complete architecture. Vehicle-installation photographs illustrate the application setting; quantitative comparisons use the specified evaluation models.

3.2. MobileNetV2-Based DeepLabv3+ Segmentation

The segmentation network follows the DeepLabv3+ encoder–decoder organization [7]. Its encoder extracts contextual features, and atrous spatial pyramid pooling combines information obtained with different receptive fields. The decoder combines contextual features with higher-resolution information to recover the road boundary. For a 512 × 512 input, the three-class segmentation network produces a 1 × 3 × 512 × 512 score tensor with class indices 0, 1 and 2. The segmentation palette maps index 1 to red, RGB (128,0,0), and index 2 to green, RGB (0,128,0). The red class is defined as the road region designated for cleaning, so the revised rule uses index 1 directly.

The detector uses one litter class, independently of the three semantic classes predicted by the segmentation network. The number of trained detection classes is determined by the dataset configuration rather than the architecture template.

MobileNetV2 replaces the heavier Xception feature extractor. Depthwise convolution performs spatial filtering separately for each channel, and pointwise convolution combines channels. Inverted residual blocks expand and transform the representation while retaining a compact bottleneck [8]. This structure motivates the backbone choice for an embedded application, although the actual hardware cost also depends on implementation and acceleration support. Figure 2 summarizes the segmentation network and its road mask output.

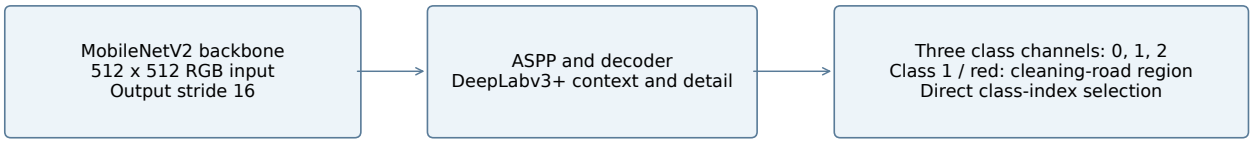

Figure 2. MobileNetV2-based DeepLabv3+ road segmentation. The encoder, atrous spatial pyramid pooling, and decoder produce a segmentation prediction. The network predicts three semantic classes; the red class (index 1) denotes the designated cleaning-road region.

The segmentation images were resized to 512 × 512, and the output stride was 16. The evaluated three-class MobileNetV2 and Xception checkpoints occupy 22.441 MiB and 209.712 MiB, respectively, where one MiB is 2^{20} bytes. The smaller serialized model reduces the storage required by the segmentation component in the two-network vehicle system.

The MobileNetV2–Xception comparison uses matched input and output dimensions for storage and computational measurements. Pixel-level quality is additionally evaluated on the same 233 publicly annotated CamVid images under the binary road protocol in Section 3.11. This reference evaluation separates visible-road segmentation quality from the local task definition of a cleaning region.

3.3. Four-Scale YOLO11n Detector

The proposed detector retains the YOLO11 backbone through layers 0–10 and uses a multiscale neck. Layer indices below are zero-based and correspond to Figure 3. Two Zoom_cat operations at layers 13 and 17 feed C3k2 blocks at layers 14 and 18. A bottom-up path produces layers 21 and 24. The first ScalSeq branch takes backbone layers [4,6,8] and generates layer 25 at P3 resolution; attention fusion of layers [18,25] generates layer 26. Layers 27–29 upsample this tensor and concatenate backbone layer 2 to form a P2 path. A second ScalSeq takes [2,26,21], generates layer 30 at P2 resolution, and is fused with layer 29 to produce layer 31. Detect at layer 32 consumes [31,26,21,24].

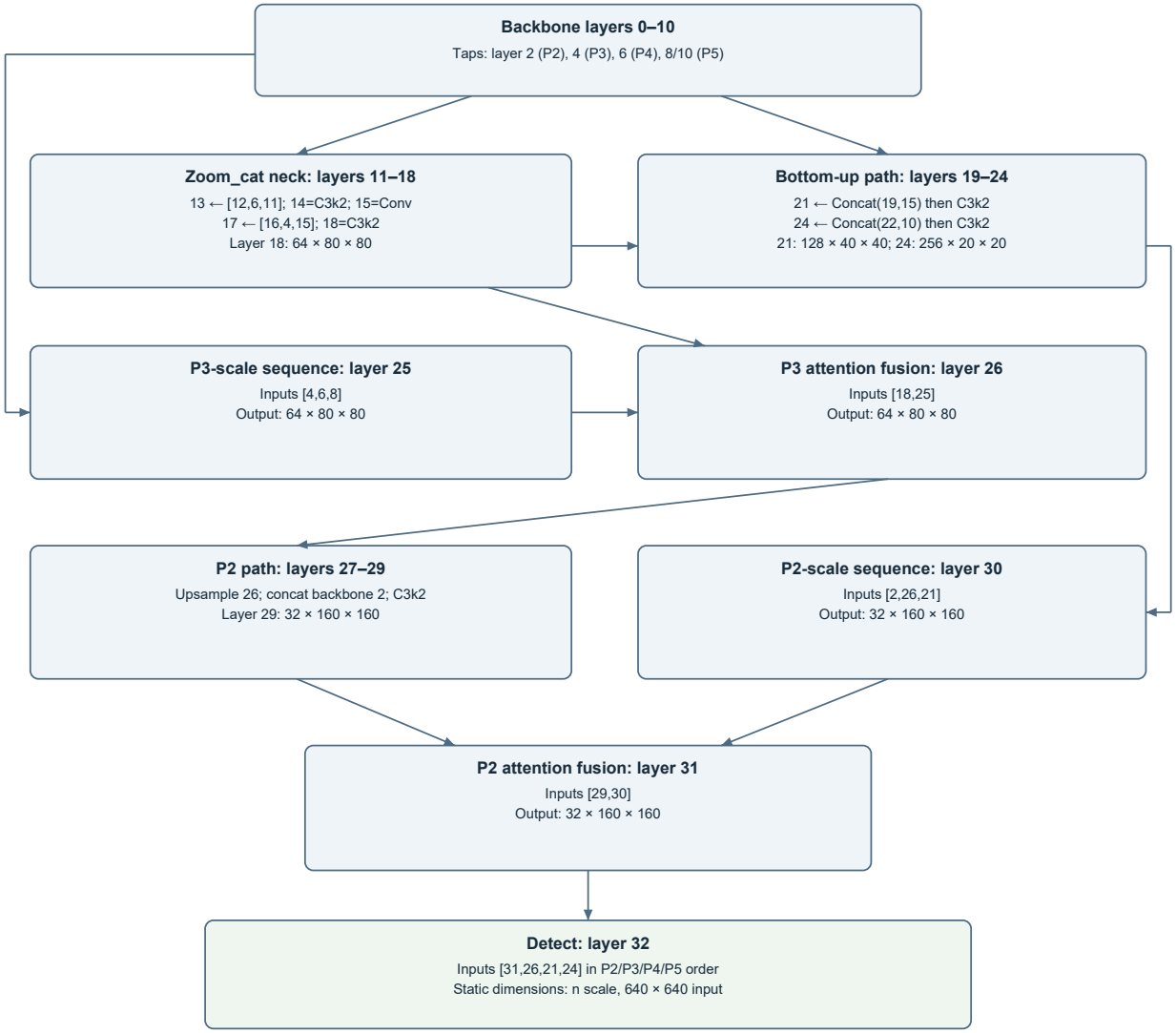

Figure 3. Architecture of the proposed multiscale YOLO11n detector. Two ScalSeq branches and two attention-fusion blocks refine the P3 and P2 features. The four-scale detection head receives P2, P3, P4, and P5 features. Layer indices identify feature connections; tensor dimensions correspond to a 640 × 640 input.

For the n scale (depth 0.50, width 0.25, maximum nominal channels 1024) and a 640 × 640 input, the source-derived dimensions are listed in Table 2. A forward pass of the proposed detector confirms the listed intermediate tensors at layers 13, 17, 25, 26, 29, 30, and 31. The loaded detector contains 33 network layers and 2,737,622 parameters, with a decoded output shape of 1 × 5 × 34,000 for one trash class. This verifies the executable configuration used for the supplementary diagnostic. The original experimental observations and supplementary checkpoint-linked measurements are reported separately.

For an input of height H_0 and width W_0 divisible by 32, the four prediction grids have strides $s \in \{4, 8, 16, 32\}$ and dimensions $(H_0/s) \times (W_0/s)$. At 640 × 640, P2 contains 160 × 160 positions compared with 80 × 80 at P3. This fourfold increase in positions is a geometric property, not a measured small-object accuracy gain or a prediction of device latency. Appendix A lists every network layer and its source-derived tensor dimensions.

Table 2. Intermediate feature dimensions of the proposed YOLO11n detector for a 640 × 640 input, verified by a forward pass. Batch dimension is omitted.

Layer	Operation and Input Layers	Channels	Spatial Grid
13	Zoom_cat [12,6,11]	384	40 × 40
17	Zoom_cat [16,4,15]	256	80 × 80
25	ScalSeq [4,6,8]	64	80 × 80
26	ASF attention [18,25]	64	80 × 80
29	P2 C3k2 after upsampling and concatenation	32	160 × 160
30	ScalSeq [2,26,21]	32	160 × 160
31	ASF attention [29,30]	32	160 × 160
32	Detect [31,26,21,24]	32/64/128/256	P2/P3/P4/P5

3.4. Zoom _CAT Alignment and ScalSeq Interaction

Zoom_cat takes three tensors denoted L, M, S, with M defining the target grid. The larger spatial input is aligned by adding adaptive maximum- and average-pooling outputs; the smaller input uses nearest-neighbor interpolation. Concatenation is along the channel axis:

$$Z = \mathrm{Concat}_C[\mathrm{MaxPool}_{H_M,W_M}(L) + \mathrm{AvgPool}_{H_M,W_M}(L),\ M,\ \mathrm{Resize}_{\mathrm{nearest}}(S)]. \tag{3}$$

The module itself contains no convolutional projection. Its following C3k2 block processes the concatenated tensor. In the proposed architecture, layer 13 aligns [12,6,11] to layer 6, and layer 17 aligns [16,4,15] to layer 4.

ScalSeq first applies 1 × 1 Conv blocks to align channels; projection of the first input is conditional on a channel mismatch. The second and third inputs are resized to the first input grid by nearest-neighbor interpolation. The aligned tensors of shape $B \times C \times H \times W$ are unsqueezed and stacked into $B \times C \times 3 \times H \times W$. A Conv3D with kernel $(1, 1, 1)$, BatchNorm3d, and LeakyReLU with slope 0.1 transform this tensor. MaxPool3D with kernel $(3, 1, 1)$ collapses the scale axis, which is squeezed to recover a 2D feature map. The pointwise Conv3d does not itself mix adjacent scale positions; cross-scale selection occurs in the pooling stage. The axis represents scales within one image, not consecutive video frames. This scale-sequence fusion operation follows the ASF-YOLO formulation [20].

For three source features F_1, F_2, F_3, let Π_i denote the implemented 1 × 1 Conv block, with Π_1 replaced by the identity when the first input already has C channels. If H, W are the first input's spatial dimensions, the aligned tensors are

$$\hat{F}_1 = \Pi_1(F_1), \qquad \hat{F}_i = \mathrm{Resize}^{\mathrm{nearest}}_{H,W}(\Pi_i(F_i)), \quad i = 2, 3. \tag{4}$$

Using Stack_s to introduce an explicit scale axis, the implemented sequence is

$$\begin{aligned} T &= \mathrm{Stack}_s(\hat{F}_1, \hat{F}_2, \hat{F}_3) \in \mathbb{R}^{B\times C\times 3\times H\times W}, \\ S &= \mathrm{Squeeze}_s\big[\mathrm{MaxPool3D}_{(3,1,1)}\big(\phi\big(\mathrm{BN3D}\big(\mathrm{Conv3D}_{(1,1,1)}(T)\big)\big)\big)\big]. \end{aligned} \tag{5}$$

Here, ϕ is LeakyReLU with negative slope 0.1, and $S \in \mathbb{R}^{B\times C\times H\times W}$. The pointwise Conv3D acts independently at each scale position; the size-three maximum-pooling operation combines the scale positions. The fixed input ordering identifies source feature levels and does not introduce temporal modeling. Unlike Equation (3), which describes channel concatenation in Zoom_cat, Equations (4) and (5) describe the explicit scale axis and its reduction. Figure 4 shows both instances in the implemented network.

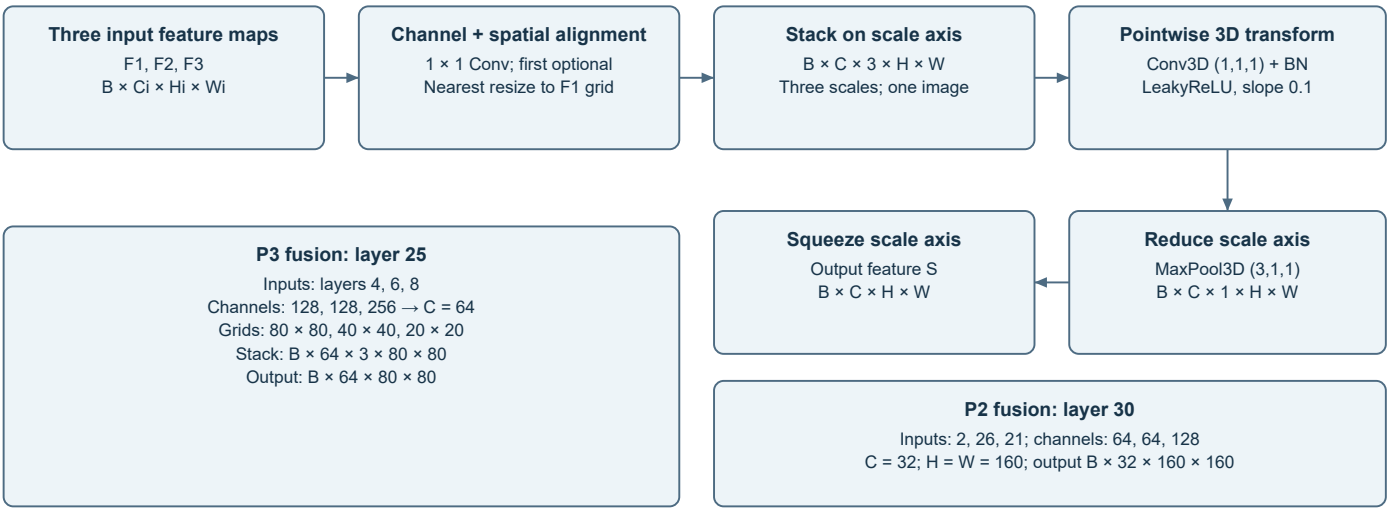

Figure 4. ScalSeq operations and tensor shapes. The scale axis has length three before pooling and length one after pooling. Layer indices correspond to Figure 3; tensor shapes are shown for the proposed YOLO11n detector with a 640 × 640 input.

3.5. Channel-and-Position Attention Fusion

The channel-and-position attention block takes two equal-shaped features X and Z. Channel attention is applied to the first feature, and the second feature is then added:

$$U = X \odot \sigma(\mathrm{Conv1D}_k(\mathrm{GAP}(X))) + Z. \tag{6}$$

Global average pooling yields a channel descriptor; the one-dimensional convolution runs over channels. Its odd kernel size is obtained from $\lfloor |(\log_2 C + 1)/2| \rfloor$, incremented by one when even. There is no maximum-pooling branch or MLP in this channel gate.

The local attention stage averages U separately along width and height. After axis reordering, these directional descriptors are concatenated, passed through a 1 × 1 bottleneck with reduction 16, BatchNorm2d, and ReLU, and split back into height and width descriptors. Separate 1 × 1 projections followed by sigmoid gates generate $A_h(U) \in \mathbb{R}^{B\times C\times H\times 1}$ and $A_w(U) \in \mathbb{R}^{B\times C\times 1\times W}$. The output is

$$Y = U \odot A_h(U) \odot A_w(U). \tag{7}$$

At P3, X is layer 18 and Z is layer 25; at P2, they are layers 29 and 30. These directional position gates differ from the average/maximum-pooling and spatial-convolution formulation of CBAM [22]. The implemented fusion follows the ASF attention design [20].

3.6. Dataset and Annotation

The original collection is described as 1086 images acquired using a front-mounted camera (SENSING SL258BALF3, SENSING Tech Co., Ltd., Shenzhen, China) on the street sweeper during road operations. This acquisition description follows the onboard application context of previous vehicle-vision work [21]; the annotations and partition records remain specific to the present litter task. Approximately one frame was retained for every six video frames to reduce adjacent-frame redundancy. The acquisition covers fine weather, rain, overcast conditions, and bright light. Table 3 reports the image totals for these environmental groups. Litter includes leaves, crumpled paper, plastic bottles, cigarette butts, cans, stones, and other small debris.

Table 3. Original image collection by environmental condition.

Scenario	Images
Fine weather	340
Rainy	280
Overcast	240
Bright light	226
Total	1086

Translation, rotation, and horizontal and vertical flipping expanded the image collection to 2172 images. Translations were within ±15 pixels in each image direction, and rotations were within ±15°. These geometric transformations belong to the transformation-based augmentation strategies reviewed by Xu et al. [28]. They vary the presentation of acquired examples without adding independently collected road scenes. Segmentation annotations were prepared with Labelme using the two region labels. Object boxes were annotated in YOLO format with T-Rex Label. All identifiable litter boxes use the single category "trash", which combines the different litter materials into one detection task.

The original experimental description assigns segmentation and detection annotations to the same image collection and reports training:test:validation = 7:2:1 for segmentation and training:validation:test = 8:1:1 for detection. These are original reported ratios, rather than partitions verified from an image-level manifest. Table 4 distinguishes this description from the unresolved run-specific settings.

Table 4. Original workstation and evaluation context. The new controlled training protocol is specified in Section 3.12.

Component	Configuration
Training CPU	Intel Core i7-14700, 2.10 GHz (Intel Corp., Santa Clara, CA, USA)
Training GPU	NVIDIA GeForce RTX 4060 Ti, 8 GB VRAM (NVIDIA Corp., Santa Clara, CA, USA)
Workstation memory	64 GB
Python	3.10.0
Deep-learning framework	PyTorch 2.5.1
Image-processing library	OpenCV 4.10.0
Segmentation input	512 × 512; output stride 16
Segmentation partition	Originally reported training:test:validation = 7:2:1; manifest not specified
Detection input	Original input not specified in retained evaluation records; supplementary input 640 × 640
Detection partition	Original training:validation:test = 8:1:1; new date-grouped split in Table 5
Original run-level settings	Optimizer and initialization records are not fully associated with the original tables; the new controlled protocol is specified in Section 3.12
Measured revision host	Intel Core i7-13650HX CPU; six threads; FP32
Desktop revision backend	PyTorch 2.7.1 CPU; Jetson protocol reported separately

Table 5. Prespecified date-grouped supplementary split. Dates refer to EXIF capture metadata. Only training images receive stochastic augmentation.

Subset	Capture Dates in 2025	Images	Litter Boxes
Training	7, 8, 9 and 28 August	100	1485
Validation	12 November	38	316
Test	13 November	142	1517

The original study reports partition ratios but does not document augmentation ancestry or recording-level membership for each image. The order of augmentation and splitting, and separation of related frames across subsets, therefore cannot be reconstructed from those ratios alone. The retrospective protocol below excludes exact duplicates and explicitly retains the remaining acquisition-group dependence as an evaluation limitation.

The original study reports the detection training/validation/test ratio given in Table 4. The archived detection configuration specifies training and validation paths but no active test path; it therefore does not identify an independent test manifest for the original evaluation. An audit of the archived detection split found 1178 training images and 297 validation images, with 75 identical decoded-image pairs crossing these subsets. The supplementary diagnostic excludes these validation duplicates. This exclusion addresses exact-copy overlap in the archived split but does not establish recording-level independence or reconstruct the original partition protocol.

Labels in the new date-grouped training, validation, and test subsets are checked for valid class indices and positive box dimensions before use. No coordinate signs are changed automatically. The original results are retained separately from this newly defined training experiment.

Usable EXIF capture dates are available for 280 of the 1475 detection images. They define the new temporal protocol in Section 3.12; descriptive date groups for the earlier retrospective cohort are retained in Appendix B. Per-image weather labels and recording identifiers are not reconstructed from file names. Road-pixel quality is evaluated separately using the annotated public reference set in Section 3.11.

3.7. Implementation and Evaluation Metrics

Table 4 distinguishes the originally reported workstation from the host used for the supplementary measurements. The retained records do not associate a complete set of optimizer, learning-rate, batch-size, initialization, and input-size settings with every original result. Segmentation uses 512 × 512 input and output stride 16; the supplementary detector evaluation uses 640 × 640 input with the protocol specified below. YOLOv9n denotes the lightweight configuration label used in the original comparison. Desktop execution is reported in Section 4.8; the separate Jetson protocol and results are reported in Sections 3.14 and 4.13. Vehicle-installation photographs provide application context independently of these measured workloads.

The detector comparison uses precision, recall, AP50, and AP50–95. True positives (TP), false positives (FP), and false negatives (FN) describe matched predictions, unmatched predictions, and unmatched ground-truth objects, respectively. Precision, recall, and box overlap are defined as

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad \mathrm{IoU}(B, G) = \frac{|B \cap G|}{|B \cup G|}, \tag{8}$$

where B and G are predicted and ground-truth boxes. AP50 summarizes the precision–recall curve at a box-IoU threshold of 0.50. AP50–95 averages AP over thresholds from 0.50 to 0.95 in steps of 0.05. These paired AP measures follow the object-localization evaluation terminology associated with COCO [29]. With a single detection class, AP and mean AP coincide for the litter category.

The precision–recall evaluator selects the index maximizing the mean F1-confidence curve after smoothing with fraction 0.1 and returns precision and recall at that index. Its confidence grid has 1000 points over [0,1], and the P/R curve uses the first matching-IoU column (0.50 in the detector validator). These source semantics establish how this evaluator works but do not prove that the original table values were produced by this version. The effective confidence values, checkpoint association, evaluation split, and NMS settings remain to be linked to each reported row. AP integrates a confidence-ranked curve and must be distinguished from precision and recall at an operating point.

The selection rule implemented by the evaluation routine can be written explicitly. On its confidence grid $c_j = j/999$, $j = 0, \ldots, 999$, define $P_k(c_j)$ and $R_k(c_j)$ for each evaluated class k at the first matching-IoU column. The routine forms

$$j^\star = \arg\max_j \; \mathcal{S}_{0.1}\left[\frac{1}{K}\sum_{k=1}^{K}\frac{2P_k(c_j)R_k(c_j)}{P_k(c_j)+R_k(c_j)+\epsilon}\right], \tag{9}$$

where $\mathcal{S}_{0.1}$ denotes the source routine's smoothing operation. It returns $P_k(c_{j^\star})$ and $R_k(c_{j^\star})$, not smoothed precision and recall. For the single trash class, $K = 1$. This evaluator-derived summary selects an operating point from the evaluated predictions; it is not evidence of a fixed deployment threshold selected on a separate validation set. The retained description of the original comparison does not specify its numerical per-row confidence operating points; the supplementary thresholds are not retrospectively assigned to those rows. A confidence threshold, a box-matching IoU threshold, the NMS IoU setting, and the color-derived mask box-coverage threshold are different quantities. No one of them can be substituted for another.

3.8. Paired Retrospective Evaluation Protocol

The supplementary retrospective evaluation is distinct from the original comparison in Tables 6 and 7. It uses the proposed four-scale YOLO11n detector and the three-class MobileNetV2-based DeepLabv3+ segmenter. The complete detector architecture is provided in Appendix A. Both models run in FP32 evaluation mode on a CPU using PyTorch 2.7.1 and Ultralytics 8.3.9. The custom scale-fusion and attention operators follow their original implementation. The detector uses the conventional prediction branch with NMS; it does not use the alternative end-to-end prediction branch. These measurements evaluate inference, rather than repeated training or Jetson execution.

Table 6. Originally reported single-class detector comparison. Values are percentages and exclude a quantified road-filtering effect. Numerical per-row confidence operating points were not specified in the retained evaluation description.

Model	AP50	AP50–95	Precision	Recall
YOLOv8n	82.4	56.4	79.9	77.4
YOLOv9n	82.2	56.7	79.1	77.3
YOLOv10n	83.7	57.2	80.1	76.3
YOLO11n	84.2	58.4	80.2	78.1
Proposed YOLO11n	87.3	61.5	84.0	83.2

Table 7. Originally reported cumulative detector ablation. Component flags describe the reported stages; no independent Zoom_cat or ScalSeq effect is isolated. Metric values are percentages.

Stage	P2	Zoom	ScalSeq	ASF	AP50	AP50–95	P	R
YOLO11n	No	No	No	No	84.2	58.4	80.2	78.1
A	Yes	No	No	No	85.8	59.6	81.6	79.6
B	Yes	Yes	Yes	No	86.5	60.7	82.9	81.3
Complete	Yes	Yes	Yes	Yes	87.3	61.5	84.0	83.2

The archived validation subset contains 297 images and 3758 trash boxes. An exact decoded-pixel comparison with the archived training subset removes 75 overlapping validation images, leaving 222 images and 2967 boxes for the primary diagnostic. Excluding exact copies does not establish acquisition-group independence or that the original checkpoint never encountered related images. Both cohorts are retained in the reproducibility records.

Detector input uses aspect-preserving 640 × 640 letterboxing; segmentation uses the specified 512 × 512 padding and resize procedure. Detector candidates use confidence

at least 0.001, NMS IoU 0.7, and at most 300 detections per image. Three paired variants share the same detector predictions: no region selection, the legacy HSV rule, and the revised class-index rule. Both selection variants use coverage 0.3. All annotated trash boxes remain evaluation targets, including any outside the predicted region; the annotations do not identify physical sweeping-area membership.

AP is computed using the evaluator functions at IoUs 0.50:0.05:0.95. Table 8 additionally fixes confidence at 0.25 for precision, recall, and error counts; greedy IoU matching at 0.50 is recomputed after that threshold. This fixed-threshold calculation is distinct from the evaluator-selected F1 operating point used to summarize an AP-confidence curve and is not retroactively assigned to the original tables. Checkpoint hashes, predictions, settings, and per-image statistics are recorded. No threshold is selected to optimize these diagnostic outcomes.

Table 8. Paired diagnostic on 222 available images after excluding exact training duplicates. AP, precision, and recall are percentages. P/R and FP/FN use confidence 0.25 and matching IoU 0.50. Remaining acquisition-group dependence is unresolved.

Selection	AP50	AP50–95	P	R	FP	FN
Unfiltered	64.67	40.10	50.39	72.50	2118	816
Legacy HSV	64.58	40.27	51.11	71.96	2042	832
Class index	64.57	40.26	51.09	71.96	2044	832

Available capture dates are used only for exploratory subset descriptions in Appendix B; unknown dates remain unknown. They cannot provide the missing field-weather classification or independent-route identity.

3.9. Object-Size, Error, and Computational Analysis

The baseline is the standard 24-layer YOLO11n detector; the proposed model is the proposed 33-layer multiscale detector. Both are evaluated on the same 222-image cohort using the inference and matching settings in Section 3.8. Their training records identify separate runs and data paths but do not establish identical training samples or initialization. The comparison therefore evaluates the available trained models under a common inference protocol; it does not isolate architectural causation or reproduce Table 6. Appendix A details the proposed detector architecture.

For size analysis, original-coordinate box area is multiplied by r^2, where $r = \min(640/H, 640/W)$ and H, W are original image dimensions. Small boxes have area below 32^2, medium boxes have area in $[32^2, 96^2)$, and large boxes have area at least 96^2 pixels at this input scale. These cutoffs define the present recall diagnostic; the result is not COCO size-specific AP. All ground-truth and prediction boxes are matched globally at confidence 0.25 and IoU 0.50 before assigning matched or missed ground truth to size bins. This avoids counting a valid detection from another size group as a false positive. Unmatched predictions are counted separately, and any size assigned to them describes predicted-box size, not a true-object category.

New error overlays use the same unique greedy matches. Green solid boxes denote matched predictions (TP), red solid boxes unmatched predictions (FP), and cyan dashed boxes unmatched annotations (FN). Classification is relative to the reference box annotations, whose completeness has not been independently audited. Examples are selected by explicit extrema in the 222-image cohort: greatest TP increase, greatest FP increase, greatest filter-induced FP reduction without TP loss, and greatest filter-induced TP loss. These are illustrative extrema, not representative error frequencies.

Computational profiling counts multiply-accumulates in executed Conv1D, Conv2D, Conv3D, and Linear modules for a batch-one 640 × 640 input. One multiply-accumulate is

one MAC. Batch normalization, pooling, interpolation, nonlinearities, elementwise operations, decoding, and NMS are excluded; the resulting GMAC count is not a complete FLOP total. Desktop forward timing uses the shared prepared-real-image protocol in Section 3.10, with the same five images, two warm-up calls per model, and five timed calls per model. Timing excludes preprocessing and downstream post-processing, except operations embedded inside a model head. These forward times are distinct from the complete-pipeline measurement and do not establish Jetson performance.

3.10. Desktop Integrated-Pipeline and Brightness Protocols

The full-pipeline benchmark uses the same identified detector and MobileNetV2 segmenter, with the red class defining the selection region. We select 12 paths evenly across the sorted 222-image manifest, decode the RGB images before timing, warm up on three images, and measure three sequential cycles (36 calls). A timed call includes segmentation resizing/letterboxing, tensor conversion, forward inference, probability resizing and class selection, detector preprocessing and forward inference, NMS, coordinate restoration, and box-coverage selection. It excludes disk decoding, camera acquisition, visualization, and actuator control. Input images retain their original sizes for resizing the output masks so that cost can vary across frames. Sequential throughput is total calls divided by total measured time, not the reciprocal of an averaged stage median.

The host is an Intel Core i7-13650HX with approximately 16.94 GB of system RAM, using Windows 11 (23H2), Python 3.10.11, and PyTorch 2.7.1 CPU, six threads, FP32 with CPU subnormal flushing enabled, and batch one. Detector input is 640 × 640; segmentation input is 512 × 512. Confidence, NMS IoU, and region coverage remain 0.25, 0.70, and 0.30. A 10 ms sampler records process resident memory during timed calls; this includes the interpreter, both models, decoded inputs, and temporary buffers and is not isolated model memory or a guaranteed instantaneous peak. The desktop software environment also used Ultralytics 8.3.9, NumPy 2.0.2, Pillow 11.3.0, and opencv-python 4.11.0.86. No validated power measurement is available for this desktop run.

The evaluated three-class Xception checkpoint is loaded with strict state-dictionary checking, using output stride 16 and 512 × 512 input. Both backbones' dimensions and convolution/linear cost are measured without inferring matched accuracy or training equivalence. For forward timing, all seven detectors and the MobileNetV2 segmenter are loaded in one process and evaluated on five evenly spaced real images from the sorted 222-image manifest. Prepared tensors are shared across methods at the corresponding 640/512 resolution, with CPU subnormal flushing enabled. Each model receives two warm-up calls and five timed calls, one per image. Model order is shuffled deterministically within each round using ordering seed 0; this is unrelated to a training seed. Prediction agreement against the prior records is checked on these five timing images, not claimed for an independently acquired set. The records retain every call. This shared protocol replaces exploratory zero-input timing traces in comparative tables. Its simultaneous eight-model residency differs from the two-model integrated-pipeline process, so their timings must not be added or interpreted as the same deployment state. Xception is excluded from the comparable timing table because its exploratory traces and shared warm-up were unstable; its storage, parameter and operation counts remain reported. Five calls per model are a limited desktop diagnostic; cache residency, scheduling, thermal state, and background load can affect timing.

To assess variation with image brightness, we partition the 222 images by quartiles of original-image mean encoded luminance, $\overline{Y'} = 0.299\overline{R} + 0.587\overline{G} + 0.114\overline{B}$. Channel means use unpadded original RGB images on the 0–255 scale. Cutoffs depend only on image brightness, not detection scores. Groups are not weather labels, measured illuminance,

or independent routes; content and camera exposure confound this image-level measure. Appendix D reports all groups, and Appendix E extends the checkpoint inventory and common-cohort diagnostics.

3.11. Public Road-Pixel Reference Protocol

CamVid provides manually annotated driving-scene pixels [30]. We use the published 233-image test split distributed with the SegNet authors' tutorial at the distributed 480 × 360 resolution [31]. Both segmentation checkpoints are evaluated without fine-tuning, subset selection, or threshold adjustment. RGB inputs are divided by 255 and resized with aspect-preserving bicubic interpolation into a 512 × 512 canvas filled with 128. Softmax probabilities are cropped to remove padding and bilinearly restored to the original image size before argmax.

The reference label index 3 is road, and index 11 is unlabeled; unlabeled pixels are ignored. The predicted index 1 is mapped to road, and predicted indices 0 and 2 are mapped to non-road. The remaining labeled reference classes form non-road. This binary mapping permits comparison of road extraction without equating the models' three semantic outputs with CamVid's eleven labeled classes. Both checkpoints are evaluated on all 38,719,017 valid pixels, using PyTorch FP32 on the same GPU.

For binary confusion counts C_{ij}, where i is ground truth and j is prediction, the per-class quantities are

$$\mathrm{IoU}_c = \frac{C_{cc}}{\sum_j C_{cj} + \sum_i C_{ic} - C_{cc}}, \qquad \mathrm{Dice}_c = \frac{2C_{cc}}{\sum_j C_{cj} + \sum_i C_{ic}}. \tag{10}$$

Binary mIoU is the unweighted mean over road and non-road; pixel accuracy is $\sum_c C_{cc} / \sum_{ij} C_{ij}$. Counts are accumulated over valid pixels before computing the ratios. The historical segmentation-training membership is not documented, so these fixed-checkpoint measurements are not presented as a certified unseen-domain test. They measure visible-road agreement with public annotations, rather than local cleaning-area membership or calibrated brush reach.

3.12. Date-Grouped Controlled Detector Experiment

A new supplementary experiment defines image subsets before applying augmentation. All 280 images with usable EXIF capture dates are assigned by complete calendar date: four August dates for training, 12 November for validation, and 13 November for testing (Table 5). Unknown-date images are not added to this experiment. The image-level manifest is fixed before training. The last validation timestamp is 12 November at 12:39:51, and the first test timestamp is 13 November at 16:14:28, separated by 27 h 34 min 37 s. Decoded-pixel hashes identify no exact cross-subset duplicates. An additional 64-bit DCT perceptual-hash screen evaluates all 23,396 cross-subset pairs under four flip orientations; no pair has Hamming distance at most 8. This screening checks for near-copy similarity but does not certify recording identity. Grouping uses recorded dates rather than inferred vehicle-trip identifiers.

The training, validation and test subsets contain 1485, 316 and 1517 litter annotations, respectively. Under the input-scaled area bins, small objects account for 97.17%, 81.01% and 79.17%. Mean original-image grayscale is 128.11, 150.50 and 142.31 on the 0–255 scale. These measurements describe differences in target size and image appearance across the fixed subsets; they are not weather labels. All 280 dated images carry the same EXIF camera-model code.

The controlled configurations are standard YOLO11n, a common four-scale reference with both fusion operators disabled, Zoom_cat only, ScalSeq only, their joint configuration,

and the joint configuration without the P2 subpath. The four factorial configurations retain P2, attention, and all remaining connections. Disabling Zoom_cat replaces its dual-pooling alignment with ordinary nearest-neighbor resize and concatenation. Disabling ScalSeq retains only its first-input channel projection, removing cross-scale sequence mixing. These replacements preserve output dimensions and define the common reference; that reference is not standard YOLO11n.

The P2-off control removes the P2 fusion/refinement subpath and its prediction scale. P3–P5 prediction widths, initialization and incoming features are preserved; only the P2 subpath is removed. This estimates the subpath's conditional effect, rather than the effect of a prediction operator in isolation. All models use one output category. The fixed DFL integral projection retains bins 0–15 and is not randomly reset or optimized.

Training uses 640-input images, 100 epochs, batch size 2, SGD with initial learning rate 0.01, final learning-rate fraction 0.01, momentum 0.937, weight decay 0.0005, and three warm-up epochs. The nominal batch size is 64, with the trainer applying gradient accumulation to the batch-two iterations. Mosaic is enabled in training and disabled for the final ten epochs; MixUp and copy–paste are disabled. HSV augmentation parameters are 0.015, 0.7 and 0.4; translation and scale ranges use 0.1 and 0.5, respectively. Horizontal flipping has probability 0.5; vertical flips, rotation, shear and perspective augmentation are disabled. Validation and testing have no stochastic augmentation. The three seeds are 0, 1 and 2, with deterministic execution requested and automatic mixed precision enabled. Early stopping is disabled within the fixed epoch budget. The new training host uses Python 3.10.11, Ultralytics 8.3.9, PyTorch 2.7.1 with CUDA 12.8, and an NVIDIA GeForce RTX 4070 Laptop GPU. Validation uses 640-pixel maximum-side scaling with rectangular batch padding; the final test uses fixed 640 × 640 letterboxing.

Each model transfers only backbone layers 0–10 from the same general 80-class COCO initialization, while the neck and prediction heads are initialized afresh. The serialized model graph is used only to construct module topology. All learned neck/head tensors and their batch-normalization buffers are reset before the common COCO backbone is transferred. No litter-trained parameters or batch-normalization statistics are retained for initialization. Shared modules in the four factorial configurations have identical initial tensors for each seed. The checkpoint maximizing validation fitness 0.1 AP50 + 0.9 AP50–95 is selected, then evaluated on the fixed 142-image test subset using confidence floor 0.001, NMS IoU 0.70, maximum 300 predictions per image, and matching IoUs 0.50:0.05:0.95. Fixed-threshold P/R and error counts use confidence 0.25 with matching recomputed at IoU 0.50. Detector ablations do not apply road-region filtering.

All three seeds are reported, with means and sample standard deviations. For a score Y, the factorial interaction is $Y_{11} - Y_{10} - Y_{01} + Y_{00}$, where the subscripts indicate enabled Zoom_cat and ScalSeq. The P2 effect compares the joint configuration with and without its P2 subpath. These are conditional comparisons under the specified data and training budget; no positive interaction or superiority is assumed in advance. Object-size recall uses the same globally matched 640-input-area bins as Section 3.9.

3.13. Sampling and Matching of Temporal-Test Error Cases

New error illustrations use the prespecified seed 0 of standard YOLO11n and the joint four-scale model. The 142 test images are ranked by mean original-image grayscale intensity, with image ID breaking ties, then divided into four groups of 36, 36, 35 and 35 images. In each group, the image nearest the median ground-truth object count is selected, with ties resolved by image ID. Thus, this selection uses image content and annotation density rather than detector accuracy. The quartiles are relative brightness strata within this cohort, not weather or nighttime labels.

Separate failure panels display the full model's largest FP count and largest FN count, selecting unique images and resolving ties by image ID. These deliberately selected extrema are labeled as such. All panels use confidence 0.25, NMS IoU 0.70 and one-to-one matching at IoU 0.50. Green solid boxes denote TP predictions, red solid boxes denote unmatched predictions, and cyan dashed boxes denote unmatched annotations. Counts refer to the full image and the reference annotations. This protocol separates stratified illustrations from selected failures and aggregate error frequencies.

3.14. Jetson TensorRT Measurement Protocol

We additionally measured the exported original four-scale detector and MobileNetV2 segmenter on an NVIDIA Jetson Orin Nano Super developer kit (NVIDIA Corp., Santa Clara, CA, USA) with approximately 8 GB shared memory. The recorded software is Linux for Tegra R36.5.2, CUDA driver API 12.6, TensorRT 10.3.0, Python 3.10.12, NumPy 1.26.4, OpenCV 5.0.0, Pillow 9.0.1, and PyCUDA 2025.1.2. The deployment checkpoints are distinct from the newly trained temporal-ablation checkpoints; deployment timing is not paired with their accuracy scores. Detector and segmenter input shapes are fixed at $1 \times 3 \times 640 \times 640$ and $1 \times 3 \times 512 \times 512$, respectively. Both engines were built on the device with TF32 disabled and a 512 MiB workspace limit. The FP16 configuration permits mixed FP16/FP32 tactics with FP32 inputs and outputs; it is not a guarantee that every layer executes in FP16.

The existing MAXN_SUPER mode (ID 2) and board fan settings were retained without locking clocks. A LITEON PA-1450-26 adapter (LITEON Technology Corp., Taipei, Taiwan), rated at 19 V, 2.37 A, and 45 W, supplied the board; this rating is not measured consumption. The graphical desktop, development and remote-access services remained active. Ambient temperature was not recorded. These conditions are reported rather than describing the measurements as an idle, clock-locked benchmark.

For each precision, three separate runs each used 100 warm-up frames and 1000 timed frames. A fixed list of 12 supplied road images was cycled, so 3000 timed frames per precision do not constitute 3000 independent scenes. The serial, batch-one interval includes both preprocessing stages, host/device transfers, detector inference, segmentation inference, NMS, full-resolution label recovery and class-index region selection. Confidence, NMS IoU and box-road coverage thresholds are 0.25, 0.70 and 0.30, respectively. File reading and decoding precede that interval; separate wall throughput includes them. Neither timing boundary includes camera acquisition, vehicle communication or actuation.

Each run records per-frame stage times and timestamped `tegrastats` samples at a requested 200 ms interval. Module-input power is the arithmetic mean of the instantaneous VDD_IN samples whose timestamps fall within the actual timed loop, including file decoding. The VDD_IN rail denotes total module power in the device documentation [32]. It is neither wall-plug power nor an isolated model power increment. Shared-system RAM and process peak resident memory are reported separately; neither is labeled exclusive GPU or model-activation memory. We report the mean and sample standard deviation of three run-level means or throughputs; latency medians and 95th percentiles pool the 3000 timed intervals per precision. Single-network `trtexec` measurements use synthetic inputs and are reported separately from image-pipeline timing.

Three supplied desktop FP32 output samples were used for numerical checking. FP32 engine outputs met the predefined tolerances (absolute 0.005, relative 0.001). FP16 errors and label agreement were retained instead of requiring FP32 tolerance. The current detector preprocessing differs from the saved reference inputs by at most one 8-bit intensity step; verification compares engines on the saved input tensors. This output check does not establish dataset-wide accuracy equivalence between precisions.

4. Results

4.1. Detector Comparison

Table 6 presents the detector comparison from the original study. Its proposed-model row reports 87.3% AP50, 61.5% AP50–95, 84.0% precision, and 83.2% recall, with higher point estimates than its reported baselines. Supplementary checkpoint-linked diagnostics appear separately in Sections 4.4–4.9 and Appendix E, with their own evaluation cohort and settings. The two evaluation contexts are not pooled, and the supplementary analysis is not presented as a repeat of the original comparison.

In the reported point estimates, the proposed model gains 3.1 percentage points in each AP measure relative to YOLO11n. Precision rises by 3.8 percentage points and recall by 5.1 percentage points. The AP50–95 difference includes stricter box-overlap thresholds, but the aggregate scores do not isolate localization-error types. These descriptive differences concern the reported evaluation only. Their stability across training runs, data partitions, and independent roads has not been established.

4.2. Cumulative Ablation

Table 7 presents the progressive construction of the proposed detector. Variant A adds P2 prediction to YOLO11n. Variant B further incorporates Zoom_cat and ScalSeq together. The complete configuration adds ASF channel-and-position attention. The component columns identify these original cumulative stages. The new independent factorial configurations are specified separately in Section 3.12.

Adding P2 increases AP50 from 84.2% to 85.8% and recall from 78.1% to 79.6%, consistent with the motivation to preserve high-resolution object information. The fusion group in Variant B then raises AP50 to 86.5% and AP50–95 to 60.7%; the latter is a 1.1-point increment over Variant A. The precision and recall point estimates also rise for the combined fusion configuration; this comparison does not isolate the mechanism responsible. With channel-and-position attention, the complete configuration reaches 87.3% AP50 and 83.2% recall, adding 0.8 AP50 points and 1.9 recall points relative to Variant B. Thus, the four point estimates increase at each stage of the original reported construction sequence.

The progression records point-estimate changes conditional on this construction order. The largest reported AP50 increment occurs when P2 is added. The grouped fusion stage increases AP50–95, followed by increases in precision and recall after attention is included. These observations do not establish an order-independent causal contribution or statistical significance. The results document conditional changes along this construction order. They do not determine separate Zoom_cat and ScalSeq effects, an interaction effect, or a size-specific P2 gain.

4.3. Vehicle Installation and Combined Perception Display

Ground-truth-matched detector illustrations are reported in Sections 4.6 and 4.12. The latter uses an explicit image/annotation sampling rule and separately labeled failure extrema. The installation photographs below document the vehicle prototype and combined image-based output.

The implementation was installed on the sweeper and exercised in campus and urban street settings. Figure 5 shows the camera installation, onboard equipment, and a combined field output. The road-segmentation map and litter boxes provide complementary descriptions of the scene in the same image coordinate frame. The photographs document the installation and an illustrative combined output. They do not by themselves verify a TensorRT execution path, sustained throughput, end-to-end latency, or closed-loop sweeping control. In the right-hand panel, blue boxes mark detections of the single litter class.

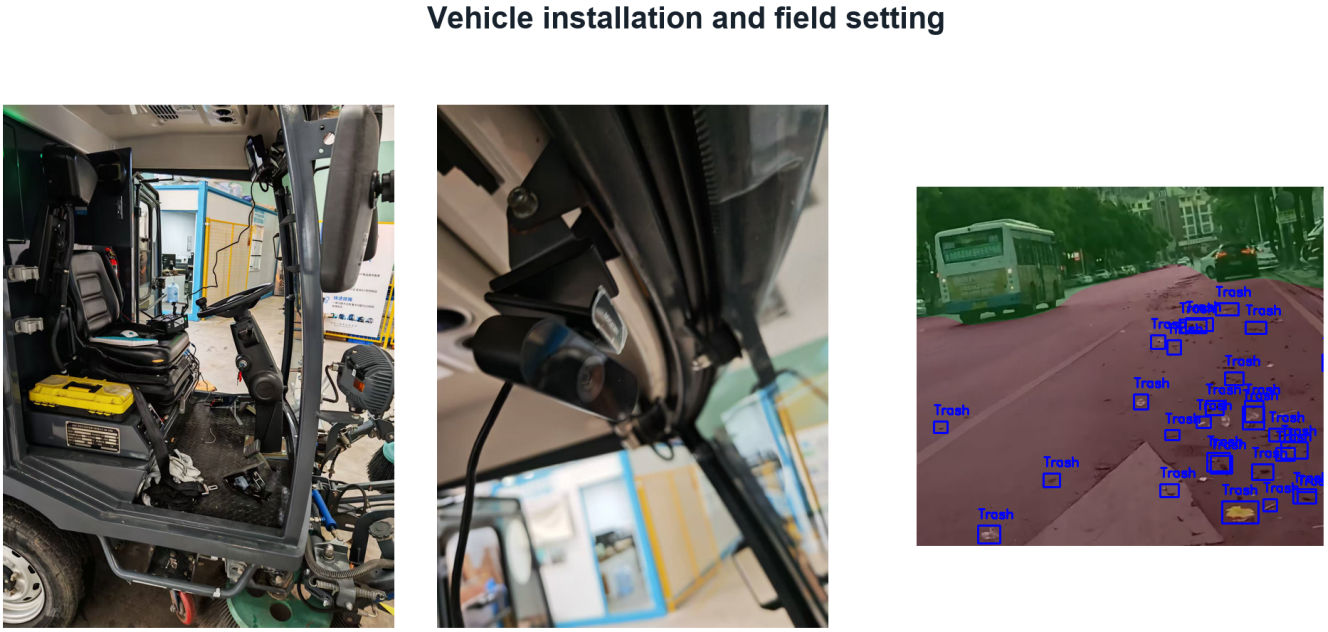

Figure 5. Street-sweeper camera installation, onboard equipment, and field image illustrating the combined road-segmentation and litter-detection application.

4.4. Paired Diagnostic of Region Selection

Table 8 reports the new 222-image comparison. These values are checkpoint-linked retrospective diagnostics; they neither reproduce nor replace the original results. All three variants use identical images, annotations, detector predictions, and inference settings before region selection.

At the fixed operating point, the legacy filter reduces false positives from 2118 to 2042 but also reduces true positives from 2151 to 2135. Precision changes from 50.39% to 51.11%, while recall changes from 72.50% to 71.96%. AP50 changes from 64.67% to 64.58% and AP50–95 from 40.10% to 40.27%. The filter therefore changes the precision–recall trade-off rather than providing an across-metric improvement. Because all annotated trash remains a target, these error counts cannot be interpreted as physical cleaning-area correctness.

The direct-index filter yields 2044 false positives and the same 2135 true positives. Its AP50/AP50–95 are 64.57%/40.26%. Across all 297 evaluated images, the mean pixel disagreement between the extracted HSV mask and the predicted class-1 mask is 0.345%; this is agreement between two predicted representations, not segmentation accuracy. Direct indexing removes the color-recovery dependency, but these results do not establish a detection-accuracy gain over the legacy rule.

The full 297-image diagnostic, its exact-duplicate exclusion manifest, confidence-selected summaries, and exploratory date subsets are retained separately. These diagnostics concern the available material and fixed checkpoints; they are not independent-scene tests, repeated-training statistics, or evidence of weather robustness.

4.5. Checkpoint Comparison and Confidence Operating Points

Table 9 reports a same-image diagnostic of the two evaluated detector checkpoints. Relative to the available baseline, the proposed checkpoint has AP50 higher by 1.66 percentage points and AP50–95 higher by 0.93 points. At confidence 0.25, it yields 247 more matched detections and 845 more false positives. Precision is lower, while recall is higher. This comparison does not demonstrate an improvement in every metric, controlled training equivalence, or an independent contribution from P2.

Table 9. Checkpoint-linked unfiltered detector comparison on the same 222 images. All percentages refer to the new diagnostic, not original Table 6. P/R and counts use confidence 0.25 and matching IoU 0.50.

Checkpoint	AP50	AP50–95	P	R	TP	FP	FN
Available baseline	63.01	39.17	59.93	64.17	1904	1273	1063
Improved, unfiltered	64.67	40.10	50.39	72.50	2151	2118	816

Table 10 resolves the numerical P/R operating points for the new diagnostic. It applies the evaluator's shared grid and smoothed-F1 selection rule in Equation (9); the displayed values are the unsmoothed P/R at that selected index. This confidence depends on the model and selection variant. These AP-curve summaries differ from re-matching after a fixed 0.25 cutoff and must not be combined with the fixed-threshold TP/FP/FN counts. These supplementary confidence values are not retrospectively assigned to the original P/R observations.

Table 10. Evaluator-selected confidence and P/R on the 222-image diagnostic. Matching for the curve uses IoU 0.50. P/R are percentages. The confidence grid is $j/999$.

Variant	Selected Confidence	P	R
Available baseline	0.482482	74.65	56.39
Improved, unfiltered	0.576577	73.49	59.12
Improved, legacy HSV	0.574575	73.54	59.02
Improved, class index	0.574575	73.54	59.02

4.6. Size-Stratified Recall and Matched Errors

Table 11 separates recall by target area at the detector input scale. Small annotations comprise 2878 of 2967 boxes (97.0%). Their recall is 63.86% for the available baseline and 72.48% for the unfiltered proposed checkpoint, a difference of 8.62 percentage points under the fixed reporting threshold. Medium recall is 75.00% and 73.86%, respectively. The single large annotation is missed by all variants, so that bin cannot support a general conclusion. The size distribution and unmatched-prediction breakdown are retained in the evaluation records. These model-level observations support examining the small-target trade-off but do not attribute it exclusively to P2 or demonstrate a size-specific AP improvement.

Table 11. Matched recall by ground-truth size at confidence 0.25 and IoU 0.50. Size cutoffs refer to area after scaling for the 640-input letterbox. Counts are TP/GT, followed by recall in percent. Large-bin GT count is one.

Variant	Small	Medium	Large
Available baseline	1838/2878 (63.86)	66/88 (75.00)	0/1 (0.00)
Improved, unfiltered	2086/2878 (72.48)	65/88 (73.86)	0/1 (0.00)
Improved, legacy HSV	2070/2878 (71.92)	65/88 (73.86)	0/1 (0.00)
Improved, class index	2070/2878 (71.92)	65/88 (73.86)	0/1 (0.00)

Figure 6 presents dense litter on textured pavement. The first row is the image with the greatest increase in matched detections: TP changes from 48 to 62, while FP changes from 7 to 10. The second is the image with the greatest FP increase: TP changes from 113 to 119, while FP changes from 37 to 86. Thus, greater detection count can reflect both fewer misses and more unmatched predictions. These labels follow annotation matching rather than visual judgments of correctness.

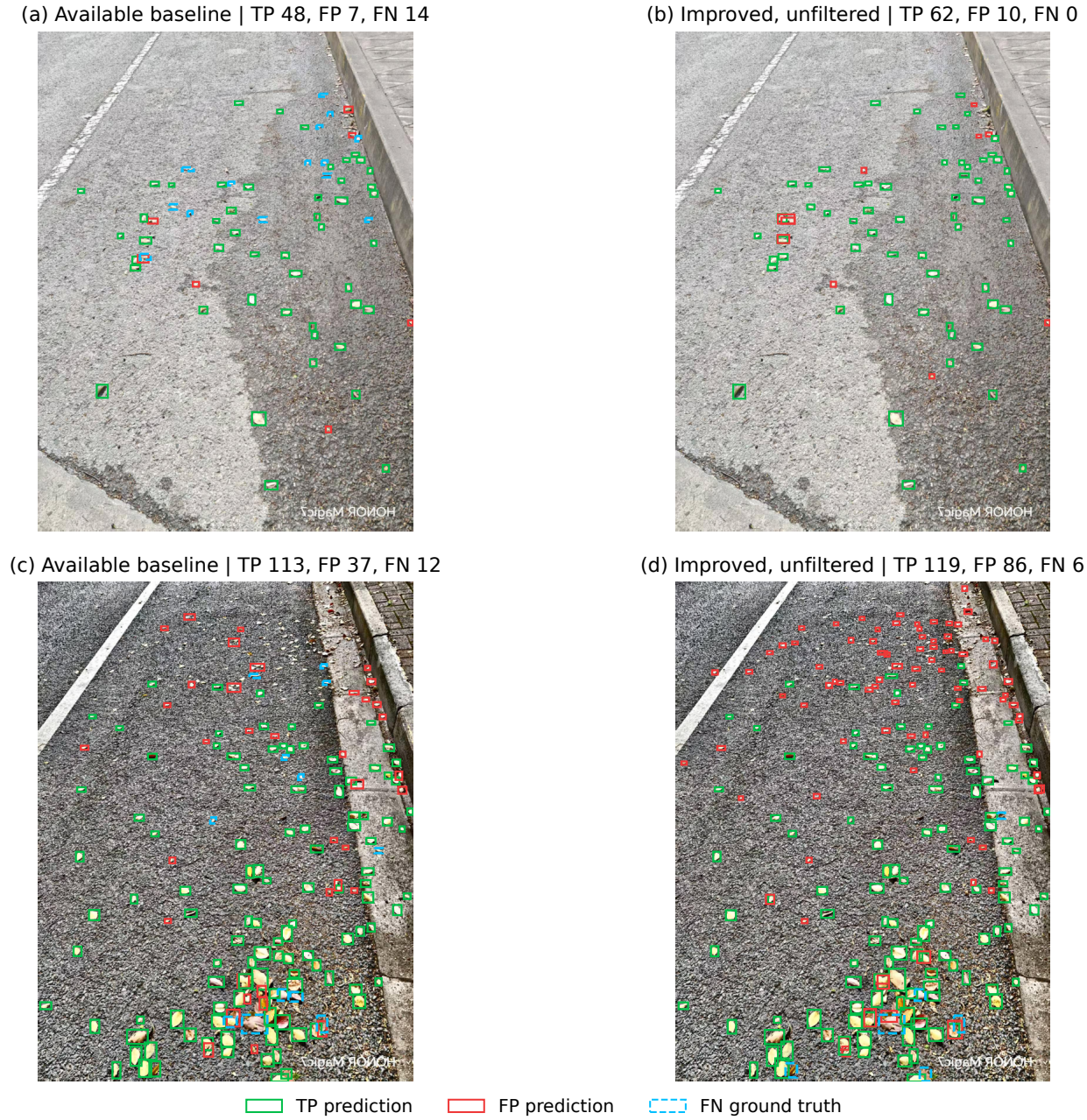

Figure 6. Matched detector errors at confidence 0.25 and IoU 0.50. (**Left**): available baseline; (**right**): unfiltered proposed checkpoint. Green solid: TP prediction; red solid: FP prediction; cyan dashed: FN annotation. Rows show the maximum TP increase and maximum FP increase in the diagnostic cohort, respectively. Selection is deliberately illustrative. The mirrored HONOR Magic7 text is a camera watermark in the source images.

Figure 7 shows both a favorable and an unfavorable region-selection outcome. In the first row, five unmatched predictions on the building facade are removed without losing the matched road object. In the second row, which includes shadows and visible onboard equipment, selection reduces TP from four to one and increases FN from 19 to 22. The latter example illustrates a risk of removing annotated detections; neither the image nor the binary rule establishes physically sweepable road membership. No occlusion-specific error rate is inferred.

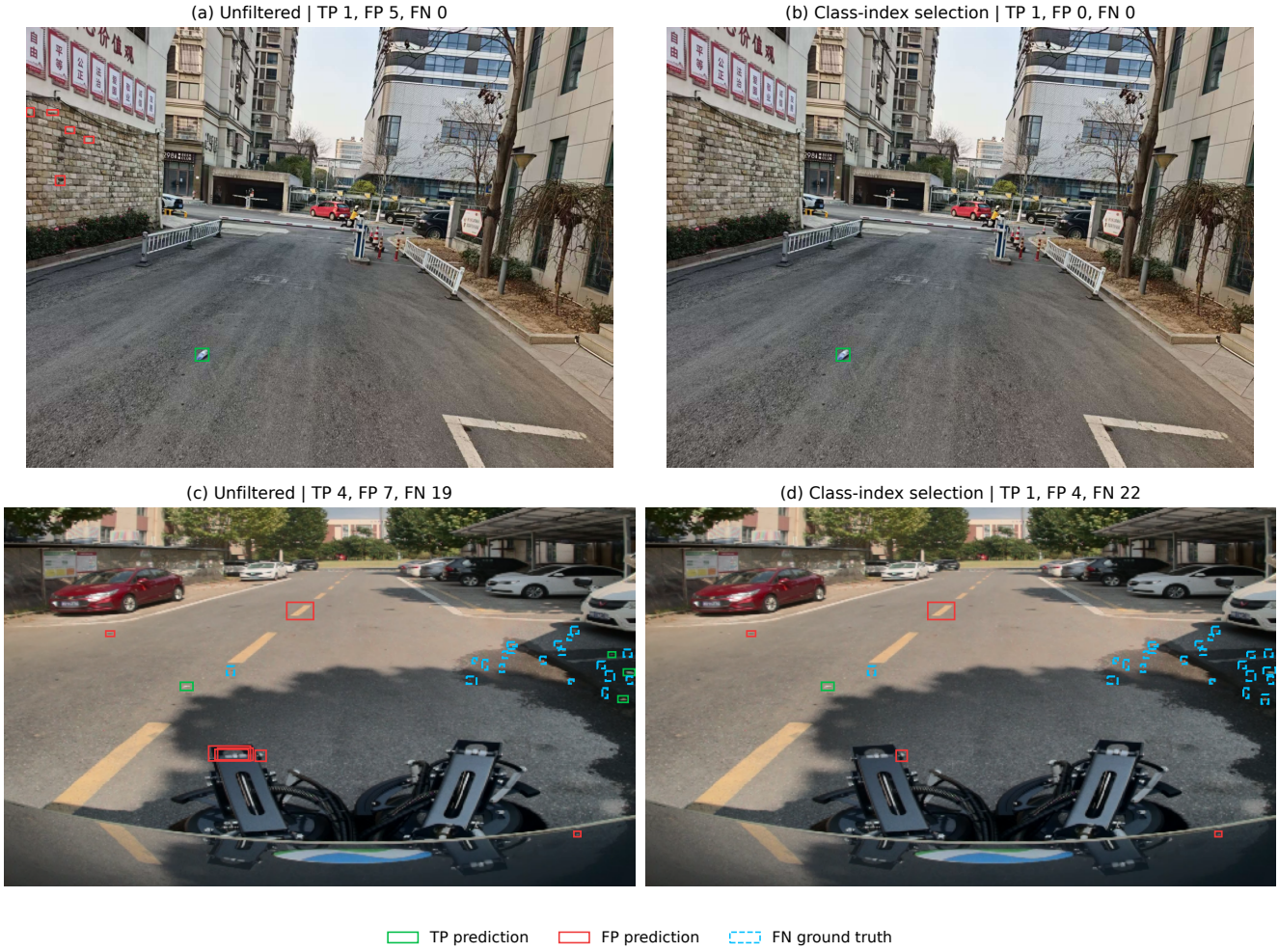

Figure 7. Matched region-selection outcomes with the same legend and thresholds as Figure 6. (**Left**): unfiltered proposed detector; (**right**): direct class-index selection. Rows are selected by the greatest FP decrease without TP loss and greatest TP loss, respectively. Classification is relative to available annotations; road-membership ground truth is not asserted.

4.7. Model Complexity and Desktop Inference Time

Table 12 records costs of the two trained models evaluated above. Parameters increase by approximately 5.70%, while counted convolution/linear MACs increase by approximately 71.82%. The high-resolution branches therefore have a modest parameter increment but a much larger arithmetic increment. The shared prepared-image CPU timings provide a separate execution diagnostic under the stated convention. A smaller parameter change must not be interpreted as proportionally smaller execution cost. These measurements cover only the two identified checkpoints and are not attached to the other original detector-family rows.

Table 12. Model complexity and CPU forward-inference time, FP32, batch one, 640 × 640, six threads. MB is 10^6 bytes. Time is median [10th, 90th percentile] of five shared prepared-image calls after two warm-ups; it excludes the rest of the pipeline. Counted GMACs exclude non-convolution/linear operations.

Checkpoint	Parameters	File (MB)	Counted GMACs	CPU Time (ms)
Available baseline	2,590,035	5.489	3.155	521.61 [493.19,667.20]
Improved	2,737,622	5.980	5.421	810.55 [753.59,875.71]

Single-process CPU timing is a desktop diagnostic, subject to scheduling, thermal state, and order effects. It is not an embedded benchmark, a complete end-to-end latency, or a repeatability confidence interval. No target Jetson memory or power result follows from it.

4.8. Desktop Integrated-Pipeline Performance

Table 13 reports measured stage times and complete sequential perception time under the Section 3.10 boundary. The recorded complete-pipeline median is 2128.74 ms, and the sequential throughput is 0.63 frames/s. Stage medians do not add exactly to the median of total time. These values concern in-memory images on the specified CPU; they are not camera-to-actuator latency or sustained vehicle-stream FPS. The result does not support a real-time claim.

Table 13. Desktop CPU pipeline timing in milliseconds: 12 images, three cycles, 36 timed calls after three warm-up images. All values use the stated input and post-processing settings. Percentiles describe calls, not confidence intervals.

Stage	Median	10th Percentile	90th Percentile
Segmentation preprocessing	51.54	9.03	76.30
Segmentation forward	1047.42	131.34	1223.78
Segmentation postprocessing	91.43	16.83	158.25
Detector preprocessing	10.74	1.84	13.47
Detector forward	820.07	92.12	989.41
Detector postprocessing	6.29	0.85	8.67
Region selection	5.23	1.26	10.27
Complete timed pipeline	2128.74	260.54	2306.42

Sampled process resident memory reaches 679.76 MB during the timed calls. It includes the interpreter, loaded networks, decoded image buffers, and intermediate processing; it must not be relabeled as model-only memory. The sampling interval can miss a shorter transient peak. This CPU experiment contains no whole-system power measurement. Separate target-device latency, shared-memory and module-input power measurements are reported in Section 4.13.

4.9. Segmentation Model Complexity

The evaluated three-class MobileNetV2 and Xception checkpoints both load strictly and produce $1 \times 3 \times 512 \times 512$ outputs. Table 14 provides model-specific storage, parameter, and counted-operation measurements. Exploratory Xception timing traces had substantial variability and are retained separately, rather than presented as a matched latency comparison. The evaluated MobileNetV2 pipeline's stage timing is reported in Table 13. The common-image road-pixel reference evaluation in Section 4.10 complements these costs; Section 4.13 reports the MobileNetV2-based integrated pipeline on the target device; a matched MobileNetV2–Xception target-device comparison is not inferred from that measurement.

Table 14. Computational and storage costs of the three-class segmentation models at 512×512, output stride 16, FP32, batch one. File size is MiB (2^{20} bytes). Counted GMACs exclude non-convolution/linear operations. Unstable Xception timings are not used to infer a matched latency ranking.

Backbone	Parameters	File (MiB)	Counted GMACs
MobileNetV2	5,813,523	22.441	26.235
Xception	54,708,931	209.712	82.745

4.10. Pixel-Level Road Segmentation Quality

Table 15 reports common-image road-pixel measurements. MobileNetV2 obtains road IoU of 70.18% and road Dice of 82.48%, compared with 63.47% and 77.65% for Xception. Its binary mIoU and pixel accuracy are 78.09% and 89.47%. The corresponding differences in road IoU and Dice are 6.71 and 4.82 percentage points. These are comparisons of the evaluated fixed checkpoints under one evaluation protocol; they do not isolate the causal effect of changing the backbone under matched retraining.

Table 15. Common-image binary road-pixel reference evaluation on all 233 published CamVid test images. Scores are percentages, with unlabeled pixels excluded. mIoU averages road and non-road, not the three model-output classes.

Backbone	Road IoU	Non-Road IoU	Road Dice	Binary mIoU	Pixel Accuracy
Xception	63.47	84.73	77.65	74.10	87.93
MobileNetV2	70.18	86.00	82.48	78.09	89.47

MobileNetV2 produces 9,595,999 road TP, 3,283,170 road FP and 794,662 road FN pixels; road precision is 74.51%, and recall is 92.35%. Xception produces 8,117,984 road TP, 2,399,978 road FP and 2,272,677 road FN pixels; road precision is 77.18%, and recall is 78.13%. Thus, the MobileNetV2 checkpoint recovers more road pixels while also predicting more non-road pixels as road. This distinction matters when the mask is used to select litter near a road boundary.

The MobileNetV2 checkpoint uses approximately 89.37% fewer parameters than Xception. In this reference evaluation, that reduction is accompanied by higher road-pixel agreement, rather than merely a smaller model file (Figure 8). CamVid does not label the vehicle-specific cleaning boundary, so this observation cannot determine which locally visible litter is operationally reachable.

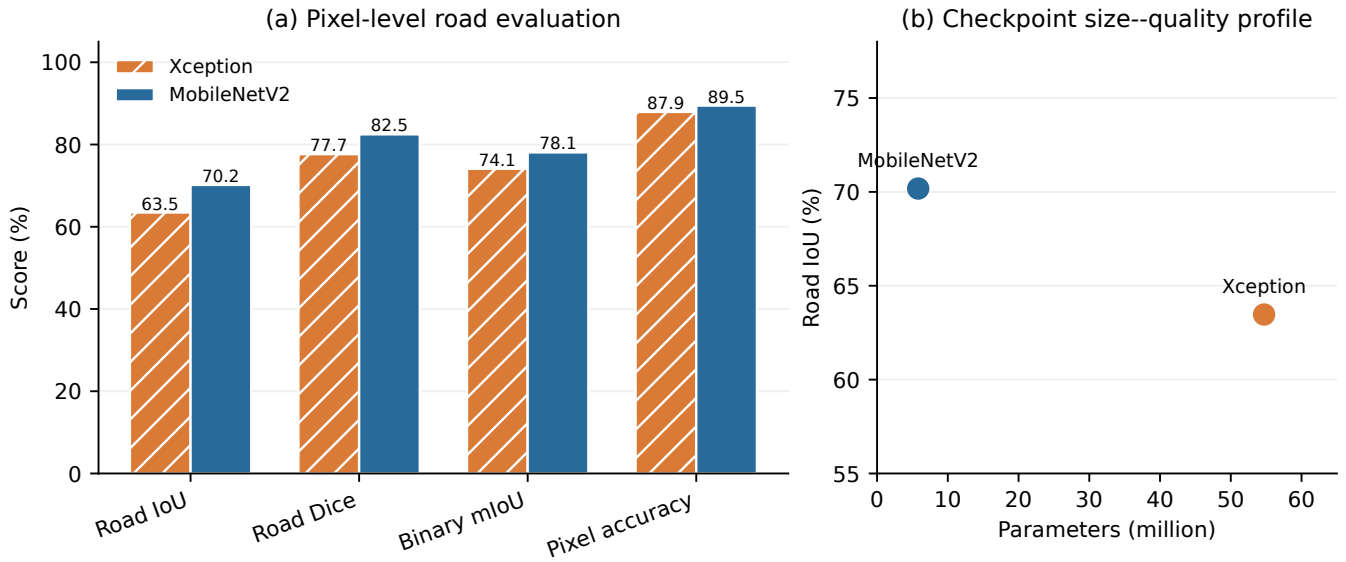

Figure 8. Fixed-checkpoint road segmentation on a common public reference set: (**a**) pixel-quality scores; (**b**) parameter count versus road IoU. Both checkpoints use the same images and binary label mapping. The comparison does not control their historical training procedures.

4.11. Controlled Component Effects on the Date-Grouped Test

Table 16 reports all six configurations under the fixed date-grouped protocol, averaged over seeds 0, 1 and 2. No seed or configuration is omitted. These results use 100 training images and a 142-image test subset, rather than the earlier comparison protocol or the

222-image retrospective diagnostic. The standard baseline and four-scale reference are distinct controls.

Table 16. Date-grouped controlled test. Scores are mean ± sample standard deviation in percent over three training seeds. Fixed P/R use confidence 0.25 and matching IoU 0.50. Z and S indicate Zoom_cat and ScalSeq; attention is fixed in the four-scale reference and its variants.

Configuration	Z	S	P2	Params (M)	AP50	AP50–95	P	R
Standard YOLO11n	–	–	No	2.590	35.18 ± 9.72	16.97 ± 5.15	55.69 ± 8.41	61.79 ± 37.16
Reference	No	No	Yes	2.701	15.72 ± 4.63	7.00 ± 2.05	39.12 ± 3.16	50.96 ± 14.92
Zoom_cat only	Yes	No	Yes	2.701	17.09 ± 6.29	7.10 ± 3.21	40.10 ± 12.40	58.21 ± 11.86
ScalSeq only	No	Yes	Yes	2.738	20.41 ± 5.02	9.68 ± 2.58	42.11 ± 12.51	59.46 ± 6.90
Joint	Yes	Yes	Yes	2.738	16.38 ± 6.64	7.18 ± 3.24	42.16 ± 7.15	57.42 ± 18.27
Joint without P2	Yes	Yes	No	2.657	25.66 ± 7.78	12.51 ± 4.24	50.58 ± 6.90	65.57 ± 13.16

The joint model obtains mean AP50 of 16.38%, compared with 35.18% for standard YOLO11n. The full configuration therefore does not outperform the standard baseline under this date-grouped, backbone-transfer and fixed-budget protocol. These observations qualify a general accuracy-improvement claim; they are not pooled with the original point estimates.

Table 17 separates operator effects from the full-model comparison. Individual contrasts keep the other operator disabled; the interaction is $Y_{11} - Y_{10} - Y_{01} + Y_{00}$. Positive values favor the first named configuration. Standard deviations describe variation across three paired training seeds, not uncertainty over independent roads.

Table 17. Paired component contrasts, in percentage points, mean ± sample standard deviation over three seeds. Small-object recall uses the fixed operating threshold.

Contrast	AP50 Difference	AP50–95 Difference	Small-Recall Difference
Zoom_cat with ScalSeq off	+1.37 ± 1.86	+0.10 ± 1.22	+8.38 ± 3.67
ScalSeq with Zoom_cat off	+4.69 ± 9.28	+2.68 ± 4.45	+9.69 ± 21.21
Joint versus common reference	+0.66 ± 2.52	+0.18 ± 2.09	+5.02 ± 3.06
Factorial interaction	−5.39 ± 10.58	−2.60 ± 5.32	−13.04 ± 21.81
P2 subpath in the joint configuration	−9.28 ± 12.04	−5.32 ± 5.25	−3.64 ± 24.05
Joint versus standard YOLO11n	−18.80 ± 13.16	−9.79 ± 5.80	−0.25 ± 42.94

Zoom_cat alone changes AP50 by +1.37 ± 1.86 percentage points relative to the common reference; ScalSeq alone changes it by +4.69 ± 9.28. The joint contrast is +0.66 ± 2.52, and the factorial interaction is −5.39 ± 10.58 percentage points. The interaction changes sign across seeds, so the experiment does not establish a stable positive synergy. Appendix C reports the individual seed scores.

The test annotations comprise 1201 small and 316 medium objects under the input-scaled area bins; there are no large-object annotations. Table 18 reports recall after global matching. The P2-off comparison preserves the P3–P5 predictor widths and initialization, enabling a conditional estimate for the complete P2 subpath.

Retaining the P2 subpath changes AP50 by −9.28 ± 12.04 and small-object recall by −3.64 ± 24.05 percentage points. Prediction grids alone do not determine the direction of these measured effects. The comparison concerns a complete refinement/fusion subpath and its scale, not a single head operator.

Table 18. Date-grouped size-stratified recall, mean ± sample standard deviation in percent. Confidence is 0.25, and matching IoU is 0.50. Large-object recall is not estimable because its denominator is zero.

Configuration	Small Recall (GT 1201)	Medium Recall (GT 316)
Standard YOLO11n	61.09 ± 34.44	64.45 ± 47.92
Reference	55.81 ± 17.04	32.49 ± 8.26
Zoom_cat only	64.20 ± 13.44	35.44 ± 9.06
ScalSeq only	65.50 ± 5.14	36.50 ± 14.54
Joint	60.84 ± 14.29	44.41 ± 33.40
Joint without P2	64.47 ± 10.99	69.73 ± 21.59

Figure 9 summarizes the date-grouped results across the six configurations.

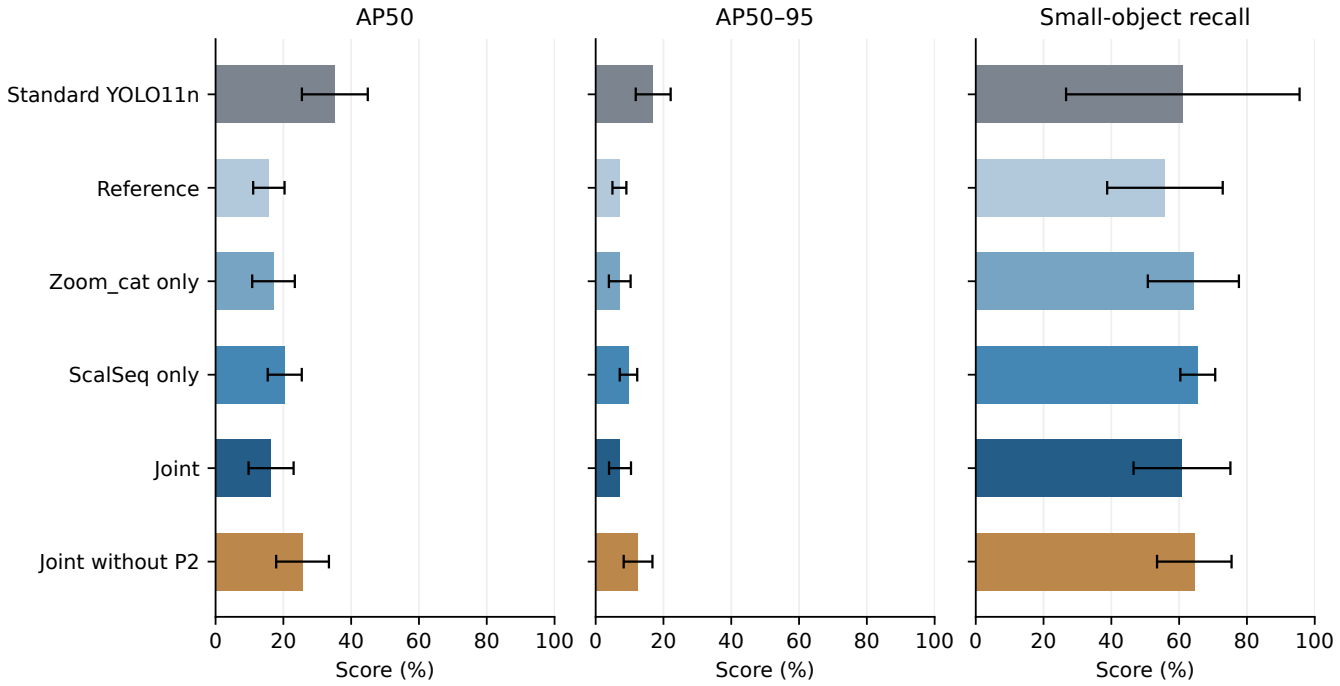

Figure 9. Date-grouped controlled results. Bars are three-seed means, and error bars are sample standard deviations. These are training-run variations on one fixed temporal test subset, not confidence intervals over routes.

4.12. Date-Grouped Test Error Cases

Figures 10 and 11 show the image/annotation-stratified cases. The examples include scattered low-contrast litter near curbs, road texture and markings, and objects at different apparent sizes. Both models produce unmatched predictions around pavement detail and overlapping boxes; the joint model also misses annotated items in several panels. These matched overlays distinguish additional boxes from verified additional detections.

For the illustrated seed over all 142 test images, standard YOLO11n records 1301 TP, 1528 FP and 216 FN; the joint configuration records 1009 TP, 1155 FP and 508 FN. The joint model reduces unmatched predictions while missing more annotated objects. The pictures therefore support an explicit error analysis rather than an unconditional improvement claim. These seed-0 counts are not substituted for the three-seed training summary.

Figure 12 separately illustrates selected failure extrema. The highest-FP case exposes erroneous or duplicate predictions against textured road surfaces; the highest-FN case exposes missed annotated litter. Their frequency is described by the cohort counts, not by the number of panels. The dataset contains no independently verified occlusion or weather strata for these examples, so the visual observations remain tied to the pictured scenes.

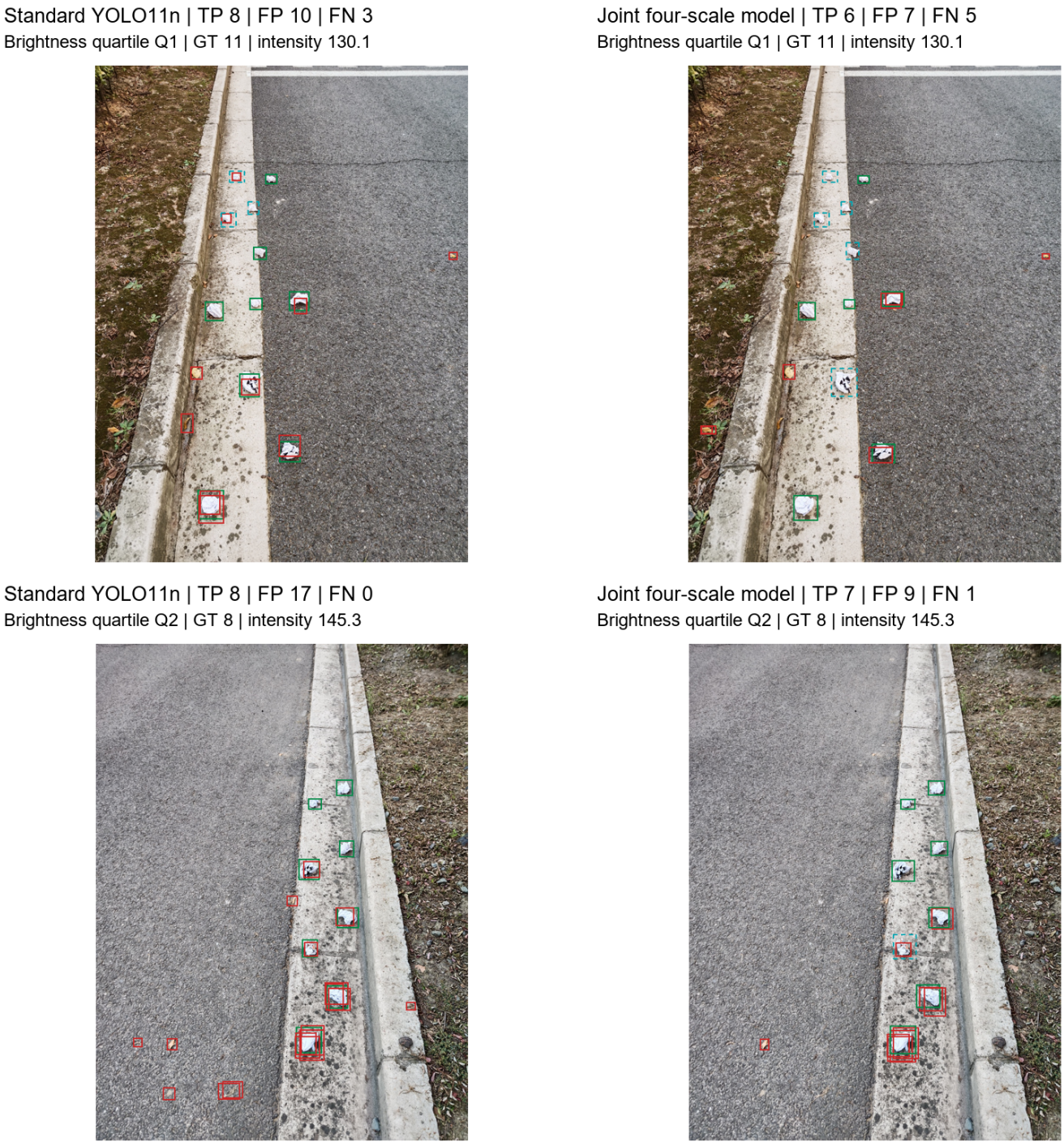

Figure 10. Date-grouped test cases from the first two brightness quartiles. (**Left**): standard YOLO11n; (**right**): joint four-scale model, both seed 0. Each image is chosen by brightness rank and median annotation count without using detection accuracy. Green solid: TP prediction; red solid: unmatched prediction; cyan dashed: unmatched annotation. Confidence is 0.25, and matching IoU is 0.50.

4.13. Target-Device TensorRT Execution

All six target-device runs completed, with 6000 timed frames in total. Table 19 reports the integrated pipeline. FP16 reduces mean serial latency from 121.39 to 80.45 ms (33.72%) and increases throughput from 8.24 to 12.43 FPS under this workload. Including image reading and decoding gives 7.02 and 10.00 FPS. These are different timing boundaries, and the latter is not camera-to-actuator throughput. Figure 13 separates the four measured pipeline stages and the module-input power comparison.

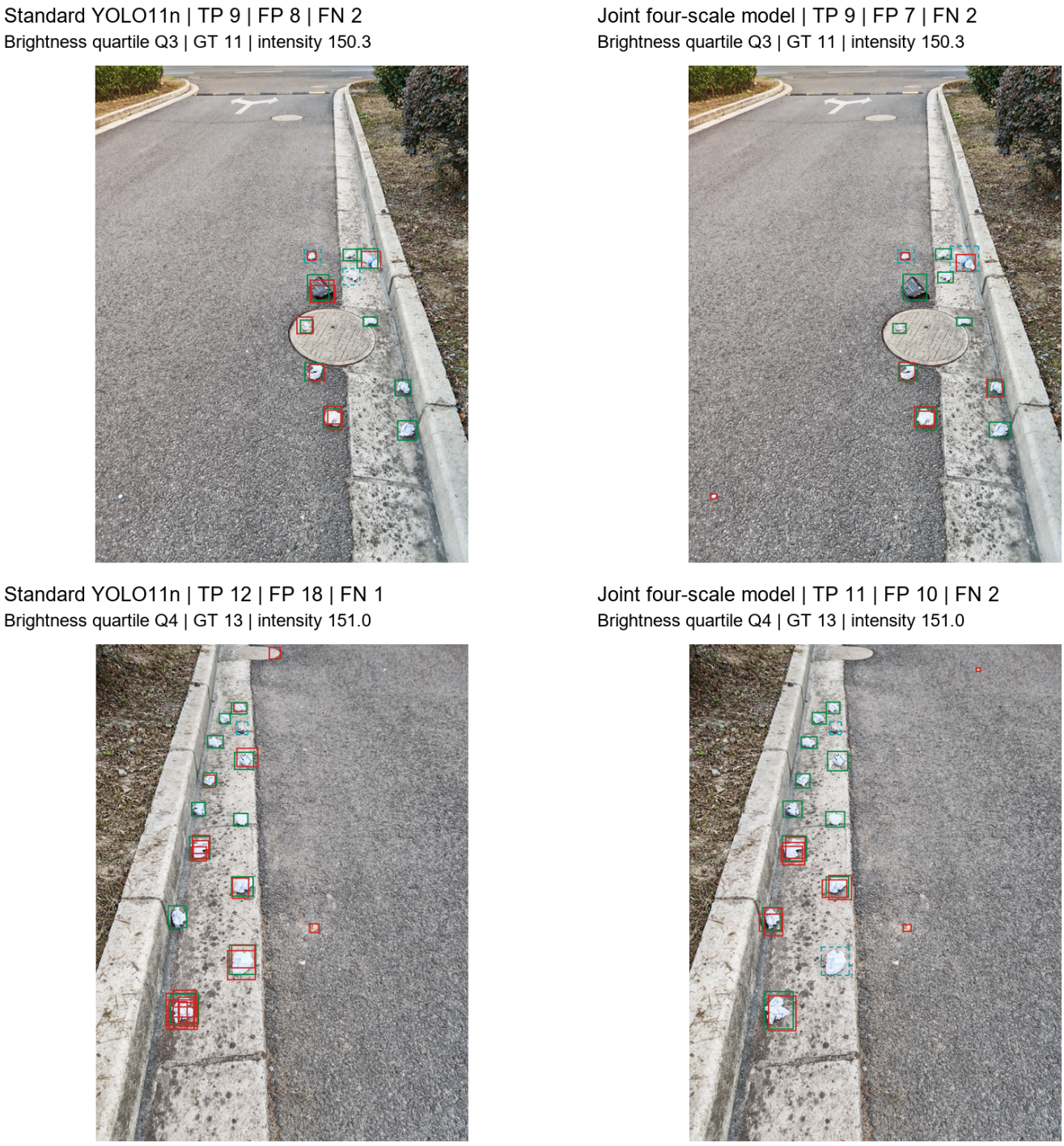

Figure 11. Date-grouped test cases from the third and fourth brightness quartiles. (**Left**): standard YOLO11n; (**right**): joint four-scale model, both seed 0. Each image is chosen by brightness rank and median annotation count without using detection accuracy. Green solid: TP prediction; red solid: unmatched prediction; cyan dashed: unmatched annotation. Confidence is 0.25, and matching IoU is 0.50.

Table 20 gives the complementary single-network measurements. The `trtexec` logs flag GPU compute variability (coefficients of variation 9.78–19.29% across networks and precisions), consistent with an unlocked, shared device. Therefore, synthetic-input throughput is not substituted for application throughput, and small variation between repeated pipeline means is not interpreted as clock stability or independent-scene generalization.

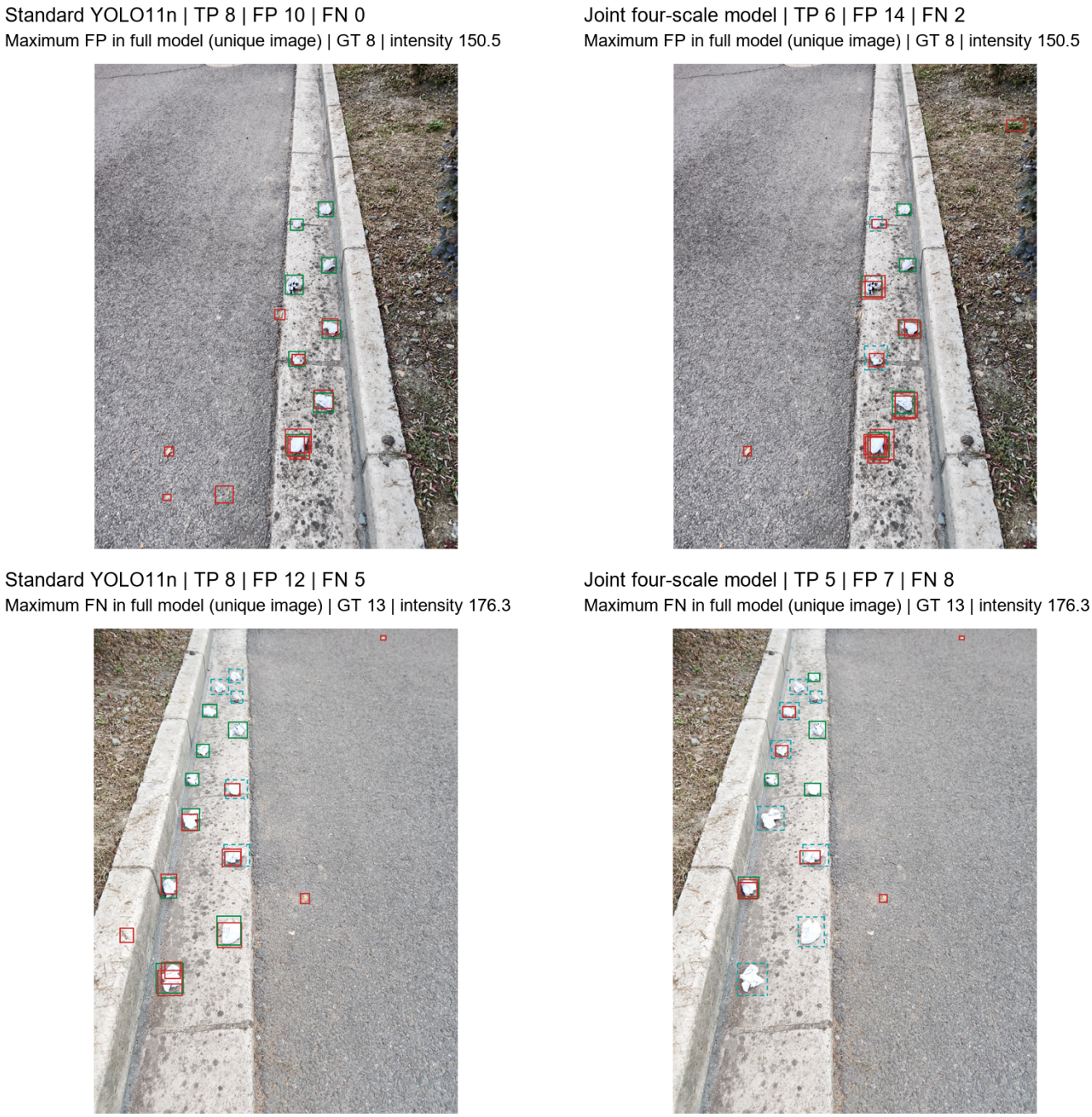

Figure 12. Deliberately selected failure extrema on the date-grouped test set: the joint model's highest FP count and highest FN count, using unique images. (**Left**): standard YOLO11n; (**right**): joint model. These panels are error illustrations, not representative sampling. Symbols and thresholds follow Figures 10 and 11.

The FP16 configuration has mean VDD_IN power 9.78 W, compared with 12.43 W for FP32. These are module-input readings with background services active, not wall-plug measurements, idle-subtracted model power or evidence of lower cleaning energy. The reported RAM maxima include system services; the process RSS also includes the interpreter, runtime and buffers. No dedicated GPU-memory peak is inferred.

On the three numerical reference images, FP16 full-resolution label agreement is 99.73–99.94%, and road-class IoU against the desktop output is 99.68–99.93%. Selected-box counts agree (12, 13 and 13), while detector confidence differences reach 0.00923. These are output-agreement measures against a computational reference, not ground-truth IoU or mAP. No accuracy-preservation claim is made from those three checks.

Table 19. Jetson Orin Nano Super integrated-pipeline results. Values with ± are mean ± sample standard deviation across three runs; each run has 1000 timed frames after 100 warm-up frames. Median and P95 pool 3000 intervals. Power is the VDD_IN sample mean over the loop including file decoding.

Measure	FP32	FP16
Serial pipeline latency (ms)	121.39 ± 0.23	80.45 ± 0.05
Pooled median/P95 (ms)	120.17/138.14	79.84/92.81
Serial throughput (FPS)	8.24 ± 0.02	12.43 ± 0.01
Wall throughput including decoding (FPS)	7.019 ± 0.014	10.005 ± 0.005
Module-input power (W)	12.43 ± 0.02	9.78 ± 0.07
Peak shared-system RAM across runs (MB)	5776	5585
Maximum process peak RSS (MiB)	761.38	728.07

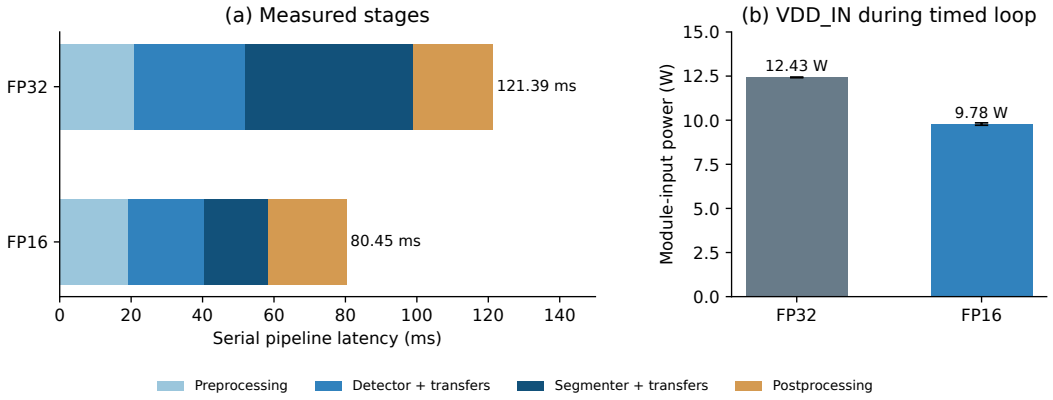

Figure 13. Measured Jetson serial-pipeline stages and module-input power. Stage bars average 3000 frame intervals per precision; power bars show the mean and sample standard deviation of three run means. Pipeline stages exclude file reading and decoding, whereas the power sampling interval includes them. Power is not vehicle energy use.

Table 20. Single-network TensorRT measurements on Jetson. GPU time excludes transfers; host latency includes H2D/D2H transfers, but both exclude image preprocessing and task postprocessing. Each network has one 30 s synthetic-input benchmark. Engine sizes are MiB (2^{20} bytes).

Network	Precision	Engine	GPU (ms)	Host (ms)	Throughput (s^{-1})
Detector	FP32	15.68	24.48	24.84	40.81
Detector	FP16	9.65	10.12	10.48	98.71
Segmenter	FP32	26.88	41.42	41.79	24.11
Segmenter	FP16	11.90	9.19	9.63	108.69

5. Discussion

5.1. Interpretation of the Detector Adaptations

The architecture targets the loss of spatial detail caused by downsampling: P2 increases prediction density, while scale alignment and attention connect representations at different resolutions. This is a design rationale rather than a guarantee of improved accuracy. The original cumulative point estimates and the new controlled experiment use different documented protocols and are interpreted separately.

Under the new date-grouped protocol, the joint model's mean AP50 is 16.38%, lower than the standard baseline's 35.18%. The paired full-model difference is -18.80 ± 13.16 percentage points. These results show why higher feature resolution and additional fusion cannot by themselves establish a performance advantage. The restricted training set, backbone-only transfer and fixed optimization budget define the conditions of this comparison; their separate influence is not isolated by the experiment.

Within the four-scale factorial reference, the AP50 changes are $+1.37 \pm 1.86$ percentage points for Zoom_cat alone and $+4.69 \pm 9.28$ for ScalSeq alone; the interaction is -5.39 ± 10.58. Retaining the P2 subpath changes small-object recall by -3.64 ± 24.05 percentage points. These contrasts estimate conditional component effects, with the remaining connections controlled. Three-seed variation supports inspection of run stability, without proving superiority across routes or optimization regimes.

5.2. Road Context and Vehicle Integration

The paired diagnostic quantifies a region-selection trade-off on the available annotations: some false positives are removed together with some correct detections. The revised direct class-index rule removes dependence on recovering class labels from visualization colors, while giving nearly the same diagnostic scores as the legacy rule. The task definition now explicitly identifies the red class as the designated cleaning-road region. The remaining evidence gap concerns matched per-image region ground truth: the present all-trash annotations cannot determine region-specific correctness. Calibration and vehicle geometry would be additionally needed for a physical three-dimensional brush-reach claim.

Road pixels occupy 26.84% of the public reference set's valid pixels. Pixel accuracy is therefore interpreted alongside both class IoUs and road Dice. The MobileNetV2 checkpoint recovers more road pixels but also produces more false road pixels than Xception. A high road recall can consequently coexist with incorrect region selection near boundaries; the public pixel comparison does not quantify local cleaning-area correctness.

The MobileNetV2 checkpoint combines lower storage and parameter counts with higher pixel agreement on the public road reference set. This comparison concerns the evaluated checkpoints, whose historical training procedures are not matched. Practical cleaning efficiency, energy savings, and successful closed-loop sweeping have not been measured in the reported comparison. The prototype selects displayed boxes but does not establish a brush-control policy, successful pickup rate, or reduced cleaning energy. These operational outcomes remain untested application objectives.

5.3. Validity and Application Limits

The new experiment fixes a complete-date partition before training and keeps the validation and test roles separate. Its test subset is one held-out capture date with 142 images, and training uses only 100 dated images. This design provides a reproducible temporal comparison, while recording identity, route membership and historical offline-augmentation ancestry are not inferred from timestamps. Exact and perceptual-copy screening cannot establish independence of road scenes. The earlier 222-image evaluation remains a retrospective checkpoint diagnostic and is not pooled with the new retraining results.

The new test subset contains 1201 small and 316 medium objects, with no large objects under the specified bins. The three training seeds quantify variation conditional on this one test cohort; they are not three independent collections. Brightness quartiles characterize image intensity and annotation density, not verified weather, occlusion or illuminance groups. Stratified examples and explicitly labeled error extrema therefore serve different descriptive purposes.

The public road-pixel comparison evaluates the two fixed checkpoints on the same visible-road annotations. It does not control their historical training procedures or establish that every reference image was unseen during training. Its binary mapping also does not validate a vehicle-specific cleaning boundary or brush footprint. Local pixel annotations and geometric calibration would be needed to assess that operational interpretation.

Cross-domain studies reinforce the importance of documented configurations and data coverage. The operational-AHU study examines sensitivity to learning configurations [33]; the multi-building study examines fault-class coverage and domain transfer [34]. These findings motivate controlled comparisons and explicit domain definitions, although their tabular-data results do not provide uncertainty estimates for road-image detection.

5.4. Computational and Deployment Evidence

The term lightweight primarily concerns the reported segmentation file-size comparison. The new detector measurements in Table 12 show that the proposed checkpoint has higher parameter count and counted MACs than the available baseline; forward timing is measured separately under a shared prepared-image protocol. This quantifies an accuracy–cost trade-off under the stated inference configuration rather than an efficiency optimum. Serialized artifact size, counted operations, and CPU forward time measure different properties; none substitutes for peak activation memory, engine size, or target-device latency.

The revised desktop benchmark now measures the combined segmentation, detection, and selection pipeline, including its preprocessing and post-processing, with the boundary specified in Section 4.8. It also records sampled process memory. Those desktop measurements are complemented by the actual TensorRT experiment in Section 4.13. Its FP16 pipeline achieves 12.43 FPS, excluding file decoding, and 10.00 FPS, including decoding. The lower module-input power relative to FP32 concerns these software configurations under the recorded workload, not vehicle cleaning energy savings. Neither experiment establishes camera-to-actuator latency or model-only peak memory. Unlocked clocks, background desktop services and repeated images define the scope of the device result.

6. Conclusions

We evaluated a street-sweeper perception pipeline combining MobileNetV2-based DeepLabv3+ and a four-scale YOLO11n detector. The detector uses scale alignment, scale–sequence fusion, channel-and-position attention and P2–P5 prediction. A separate class-index mask selects full-image detections using a fixed road-coverage rule.

The new date-grouped, three-seed experiment reports all six controlled configurations. The joint model reaches mean AP50 of 16.38%, lower than 35.18% for standard YOLO11n under the prescribed training budget. The factorial interaction is -5.39 ± 10.58 percentage points, and the P2-subpath effect on small-object recall is -3.64 ± 24.05 percentage points. These measurements support conditional component comparisons rather than an unconditional benefit from additional modules.

On the 233-image public road reference set, the MobileNetV2 checkpoint obtains road IoU 70.18%, road Dice 82.48%, binary mIoU 78.09% and pixel accuracy 89.47%, with substantially fewer parameters than Xception. The fixed-checkpoint comparison complements the computational profile without establishing a matched-training backbone advantage or local cleaning-boundary accuracy. In the retrospective 222-image evaluation, direct region selection removes 74 false positives together with 16 true detections, demonstrating a measured precision–recall trade-off.

On Jetson Orin Nano Super, the FP16 batch-one pipeline takes 80.45 ± 0.05 ms and reaches 12.43 ± 0.01 FPS, excluding decoding; measured wall throughput, including file decoding, is 10.00 ± 0.005 FPS. Mean module-input power is 9.78 ± 0.07 W across three

repeated runs. These results establish target-device execution for the fixed workload, while physical cleaning success, calibrated brush reach and vehicle energy savings remain outside the demonstrated perception results.

Author Contributions: Conceptualization, Z.C. (Zhongwen Chen), Q.Z. and M.W.; methodology, Z.C. (Zhongwen Chen); software, Z.C. (Zhongwen Chen); validation, Z.C. (Zhongwen Chen); formal analysis, Z.C. (Zhongwen Chen); investigation, Q.Z. and M.W.; resources, Z.C. (Zhongwen Chen), Z.C. (Zihua Chen) and H.Y.; data curation, Z.C. (Zhongwen Chen) and H.Y.; writing—original draft preparation, Z.C. (Zhongwen Chen); writing—review and editing, Z.C. (Zhongwen Chen), Q.Z. and M.W.; visualization, Z.C. (Zhongwen Chen); supervision, Q.Z., M.W., Z.C. (Zihua Chen), Y.Z. and H.Y.; project administration, Q.Z., M.W., Z.C. (Zihua Chen), Y.Z. and H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no direct financial funding. Hunan Niu En Chi New Energy Vehicle Co., Ltd. provided non-financial, in-kind support consisting of an autonomous sweeping vehicle, camera equipment, testing facilities, road-scene data, and field engineering assistance.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The locally collected street-sweeper images and annotations are not publicly available due to confidentiality restrictions. Requests for access may be directed to the corresponding author at qingbingzeng@sues.edu.cn. The public CamVid reference images and annotations are available from the source repository cited in the Methods. Requests for access to the image assignments and supporting experimental records may also be directed to the corresponding author.

Acknowledgments: The authors thank Qihao Wang for his assistance with dataset construction, image acquisition and annotation, camera installation on the street sweeper, and review and revision of the experimental section of the manuscript.

Conflicts of Interest: Author Heng Yang is employed as Technical Director of Hunan Niu En Chi New Energy Vehicle Co., Ltd. The company provided non-financial, in-kind support consisting of an autonomous sweeping vehicle, camera equipment, testing facilities, road-scene data, and field engineering assistance. The company provided no direct financial funding for this study. Beyond the provision of these resources and field engineering assistance, the company had no role in the study design, annotation protocol, algorithm development, model training, experimental evaluation, statistical analysis, interpretation of the results, manuscript preparation, or the decision to submit the manuscript for publication. The remaining authors declare no competing interests.

Appendix A. Layer-Wise Network Architecture

Table A1 records every layer of the proposed detector at the n scale and a 640 × 640 input. Indices and inputs are zero-based, the batch dimension is omitted, and an effective stride refers to the original input image. These dimensions are statically derived from the configuration and module definitions; they are not a device profile or proof of correspondence with the original reported scores. The final row lists the four input feature tensors to Detect rather than an inferred detection-output shape.

The listed intermediate tensors at layers 13, 17, 25, 26, 29, 30, and 31 have additionally been checked by real forward hooks on the diagnostic checkpoint. Its decoded detector output and segmentation output are retained in the evaluation records.

Table A1. Layer-wise architecture and feature dimensions of the proposed detector.

Layer	Inputs	Operation	Channels	Stride	Grid
0	image	Conv	16	2	320 × 320
1	0	Conv	32	4	160 × 160
2	1	C3k2	64	4	160 × 160
3	2	Conv	64	8	80 × 80
4	3	C3k2	128	8	80 × 80
5	4	Conv	128	16	40 × 40
6	5	C3k2	128	16	40 × 40
7	6	Conv	256	32	20 × 20
8	7	C3k2	256	32	20 × 20
9	8	SPPF	256	32	20 × 20
10	9	C2PSA	256	32	20 × 20
11	10	Conv	128	32	20 × 20
12	4	Conv	128	8	80 × 80
13	12,6,11	Zoom_cat	384	16	40 × 40
14	13	C3k2	128	16	40 × 40
15	14	Conv	64	16	40 × 40
16	2	Conv	64	4	160 × 160
17	16,4,15	Zoom_cat	256	8	80 × 80
18	17	C3k2	64	8	80 × 80
19	18	Conv	64	16	40 × 40
20	19,15	Concat	128	16	40 × 40
21	20	C3k2	128	16	40 × 40
22	21	Conv	128	32	20 × 20
23	22,10	Concat	384	32	20 × 20
24	23	C3k2	256	32	20 × 20
25	4,6,8	ScalSeq	64	8	80 × 80
26	18,25	Attention fusion	64	8	80 × 80
27	26	Upsampling	64	4	160 × 160
28	27,2	Concat	128	4	160 × 160
29	28	C3k2	32	4	160 × 160
30	2,26,21	ScalSeq	32	4	160 × 160
31	29,30	Attention fusion	32	4	160 × 160
32	31,26,21,24	Detect	32/64/128/256	4/8/16/32	160/80/40/20

Appendix B. Exploratory Capture-Date Diagnostics

The subsets in Table A2 are formed only from available EXIF dates within the 222-image diagnostic. They are not weather groups, verified trips, independent routes, or independent training runs. Model settings are identical across subsets. Small group sizes and missing metadata prevent a generalization claim.

Table A2. Exploratory capture-date subset diagnostics. AP values are percentages; HSV denotes the legacy filter.

Date	Images	GT	Raw AP50	Raw AP50–95	HSV AP50	HSV AP50–95
8 August 2025	13	104	34.52	18.57	32.26	18.61
9 August 2025	7	106	68.31	49.69	68.44	49.98
28 August 2025	9	32	23.74	7.93	23.77	7.94
12 November 2025	10	79	87.26	42.86	87.80	43.38
13 November 2025	33	256	98.59	54.82	97.96	54.54
Unknown	150	2390	62.10	39.91	62.12	40.07

Appendix C. Per-Seed Controlled Detector Results

Table A3 reports every prespecified seed on the same fixed temporal test subset. It is not a selection of the best-performing seed. AP50, AP50–95, P and R are percentages; fixed P/R use confidence 0.25 and matching IoU 0.50.

Table A3. All 18 controlled test runs.

Configuration	Seed	AP50	AP50–95	P	R
Standard YOLO11n	0	43.37	21.27	45.99	85.76
Standard YOLO11n	1	37.73	18.39	60.19	80.62
Standard YOLO11n	2	24.45	11.26	60.89	18.98
Reference	0	16.83	7.45	38.51	58.80
Reference	1	10.64	4.76	36.31	33.75
Reference	2	19.70	8.80	42.54	60.32
Zoom_cat only	0	17.46	7.20	40.02	67.17
Zoom_cat only	1	10.63	3.85	27.74	44.76
Zoom_cat only	2	23.18	10.26	52.54	62.69
ScalSeq only	0	16.39	7.65	33.81	51.55
ScalSeq only	1	26.04	12.58	56.50	64.21
ScalSeq only	2	18.79	8.82	36.03	62.62
Joint	0	19.93	10.04	46.63	66.51
Joint	1	8.72	3.66	33.91	36.39
Joint	2	20.49	7.86	45.94	69.35
Joint without P2	0	31.36	15.81	46.14	69.02
Joint without P2	1	28.82	14.00	58.53	76.66
Joint without P2	2	16.79	7.72	47.08	51.02

Appendix D. Image-Brightness-Stratified Diagnostics

Mean encoded-luminance quartile cutoffs are 118.65, 138.14, 153.57 on the 0–255 scale. Q1 is the lowest-brightness group and Q4 the highest. Each row in Table A4 uses the same images and annotations across methods. These groups measure image intensity, not meteorological conditions, and do not establish scene independence.

Table A4. Brightness-quartile diagnostic on the 222-image cohort. AP50 is in percent. Baseline and improved denote identified checkpoints; selected means direct class-index region selection.

Group	Images	GT	Baseline AP50	Improved AP50	Selected AP50
Q1	56	533	23.35	25.54	25.56
Q2	55	580	69.76	72.03	71.71
Q3	55	1025	85.13	85.07	84.96
Q4	56	829	54.02	57.21	57.22

Appendix E. Supplementary Comparisons of Trained Models

The supplementary comparison includes available trained YOLOv8n, YOLOv9n, and YOLOv10n models, together with P2 and P2/Zoom configurations. Table A5 records the evaluated checkpoints on the same 222-image diagnostic. YOLOv9n remains the study label for its lightweight configuration. File hashes, embedded training settings, graph flags, candidate predictions, and per-operator cost counts are recorded. Distinct embedded training paths and missing original partition manifests prevent interpreting this table as controlled architectural ablation. The v10Detect checkpoint uses its NMS-free output path; other rows use NMS IoU 0.70. Confidence floor is 0.001, candidate cap 300, and detector input 640 × 640.

The P2 and P2/Zoom configurations respectively include the additional high-resolution prediction branch alone and together with aligned feature fusion. Runtime strides and graph flags confirm their scale/fusion arrangement. Neither is relabeled as a separately verified training run for the original ablation. A model with a different attention variant, scale, input size, or training endpoint is not substituted for a missing independent ScalSeq comparison.

Table A5. Supplementary comparison of trained models on the common evaluation cohort: detection accuracy and computational cost. Parameters are millions; MB is 10^6 bytes; time is median CPU forward milliseconds under the shared prepared-image protocol in Section 3.10. Counts are convolution/linear GMACs, not full FLOPs. These comparisons evaluate existing trained models; they do not reproduce the original experimental protocol or isolate individual component effects.

Model Configuration	AP50	AP50–95	Params (M)	File (MB)	GMACs	CPU (ms)
YOLOv8n	61.92	38.80	2.690	5.641	3.410	411.15
YOLOv9n	63.94	39.02	1.727	3.944	3.307	767.24
YOLOv10n	57.79	36.24	2.707	5.768	4.113	622.20
P2 configuration	58.71	35.81	2.677	5.829	4.923	702.90
P2/Zoom configuration	59.69	36.72	2.702	5.878	5.002	704.69

References

1. Donati, L.; Fontanini, T.; Tagliaferri, F.; Prati, A. An Energy Saving Road Sweeper Using Deep Vision for Garbage Detection. *Appl. Sci.* **2020**, *10*, 8146. [CrossRef]
2. Gomez, B.F.; Yi, L.; Ramalingam, B.; Rayguru, M.M.; Hayat, A.A.; Thejus, P.; Leong, K.; Elara, M.R. Deep Learning based Litter Identification and Adaptive Cleaning using Self-reconfigurable Pavement Sweeping Robot. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 20–24 August 2022; pp. 2301–2306. [CrossRef]
3. Proença, P.F.; Simões, P. TACO: Trash Annotations in Context for Litter Detection. *arXiv* **2020**, arXiv:2003.06975. [CrossRef]
4. Mandhati, S.R.; Deshapriya, N.L.; Mendis, C.L.; Gunasekara, K.; Yrle, F.; Chaksan, A.; Sanjeev, S. pLitterStreet: Street Level Plastic Litter Detection and Mapping. *arXiv* **2024**, arXiv:2401.14719. [CrossRef]
5. Wu, T.; Xu, Q.; He, X.; Weekes, O.; Brown, J.; Duan, W. RoLID-11K: A Dashcam Dataset for Small-Object Roadside Litter Detection. *arXiv* **2026**, arXiv:2601.00398. [CrossRef]
6. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [CrossRef] [PubMed]
7. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
8. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
9. Xu, J.; Xiong, Z.; Bhattacharyya, S.P. PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 19529–19539.
10. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
11. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
12. Jocher, G. *Ultralytics YOLOv5*; Ultralytics GitHub Repository (Computer Software); Zenodo: Geneva, Switzerland, 2020 . [CrossRef]
13. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. Ultralytics GitHub Repository (Computer Software). 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 23 October 2026).
14. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *Computer Vision—ECCV 2024*; Springer: Cham, Switzerland, 2025; Volume 15089, pp. 1–21. [CrossRef]
15. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): San Diego, CA, USA, 2024; Volume 37, pp. 107984–108011. [CrossRef]
16. Jocher, G.; Qiu, J. Ultralytics YOLO11. Ultralytics Software and Official Documentation. 2024. Available online: https://docs.ultralytics.com/models/yolo11/ (accessed on 23 October 2026).
17. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
18. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.

19. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
20. Kang, M.; Ting, C.M.; Ting, F.F.; Phan, R.C.W. ASF-YOLO: A novel YOLO model with attentional scale sequence fusion for cell instance segmentation. *Image Vis. Comput.* **2024**, *147*, 105057. [CrossRef]
21. Chen, Z.; Zeng, Q.; Chen, Z.; Zhang, Y. Recognition of Manhole Cover Status for Refuse Collection Vehicles Utilizing YOLO26-ABF. *Appl. Sci.* **2026**, *16*, 8482. [CrossRef]
22. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
23. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.
24. Ma, W.; Chen, H.; Zhang, W.; Huang, H.; Wu, J.; Peng, X.; Sun, Q. DSYOLO-trash: An attention mechanism-integrated and object tracking algorithm for solid waste detection. *Waste Manag.* **2024**, *178*, 46–56. [CrossRef] [PubMed]
25. Zailan, N.A.; Mohd Khairuddin, A.S.; Hasikin, K.; Junos, M.H.; Khairuddin, U. An automatic garbage detection using optimized YOLO model. *Signal Image Video Process.* **2024**, *18*, 315–323. [CrossRef]
26. Wang, T.; Cai, Y.; Liang, L.; Ye, D. A Multi-Level Approach to Waste Object Segmentation. *Sensors* **2020**, *20*, 3816. [CrossRef] [PubMed]
27. Haldorai, A.; R, B.L.; M, S.; Balakrishnan, M. An improved single short detection method for smart vision-based water garbage cleaning robot. *Cogn. Robot.* **2024**, *4*, 19–29. [CrossRef]
28. Xu, M.; Yoon, S.; Fuentes, A.; Park, D.S. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognit.* **2023**, *137*, 109347. [CrossRef]
29. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Springer: Cham, Switzerland, 2014; Volume 8693, pp. 740–755. [CrossRef]
30. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.* **2009**, *30*, 88–97. [CrossRef]
31. Kendall, A. SegNet Tutorial: CamVid Images, Annotations, and Evaluation Scripts. Available online: https://github.com/alexgkendall/SegNet-Tutorial (accessed on 15 November 2026).
32. NVIDIA. Jetson Linux Developer Guide R36.5: Platform Power and Performance for Jetson Orin Series. 2026. Available online: https://docs.nvidia.com/jetson/archives/r36.5/DeveloperGuide/SD/PlatformPowerAndPerformance/JetsonOrinNanoSeriesJetsonOrinNxSeriesAndJetsonAgxOrinSeries.html (accessed on 16 November 2026).
33. Wang, S. Automated Fault Detection and Diagnosis of AHUs via Tabular-Based Methods Using Operational Data from a Large Office Building. *J. Comput. Civ. Eng.* **2026**, *40*, 04026018. [CrossRef]
34. Wang, S. Fault-class coverage-aligned combined training for AFDD of AHUs across multiple buildings. *Sci. Rep.* **2025**, *15*, 41192. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.