Scientific Research Publishing

Open Journal of Applied Sciences, 2026, 16(11), 4283-4303
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

A Lightweight Weighted Multi-Scale Fusion Network for Small-Person Detection in Post-Disaster UAV Imagery

Danah Alotaibi

School of Computer Science, China University of Geosciences (Wuhan), Wuhan, China
Email: daanaalotibi@gmail.com

How to cite this paper: Alotaibi, D. (2026) A Lightweight Weighted Multi-Scale Fusion Network for Small-Person Detection in Post-Disaster UAV Imagery. *Open Journal of Applied Sciences*, **16**, 4283-4303.
https://doi.org/10.4236/ojapps.2026.1611233

Received: September 12, 2026
Accepted: November 2, 2026
Published: November 5, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

Person detection in post-disaster unmanned aerial vehicle imagery requires sensitivity to small targets and a compact deployment budget. This study presents LMF-YOLO, a YOLO11n variant that retains the pretrained backbone and neck and adds three residual positive-normalized fusion blocks at P3, P4, and P5. A single P3 projection increases the model from 2,582,347 to 2,590,609 parameters and from 6.4 to 6.5 GFLOPs. The PDD audit produced a group-isolated split of 619 training images with 1783 person instances, 112 validation images with 313 instances, and 146 frozen-test images with 481 instances. Across seeds 17, 43, and 97, A0 and A2 achieved validation mAP50-95 of 0.53095 ± 0.01065 and 0.53053 ± 0.03128, a paired mean change of −0.04 percentage points. Frozen-test mAP50-95 changed from 0.45675 ± 0.00389 to 0.44530 ± 0.03469, a paired mean change of −1.15 points. Small-object test AR increased descriptively by 4.44 ± 4.60 points. A2 seed 97 completed 97 epochs and reached zero mosaic-free epochs, whereas the planned schedule assigned the final 15 epochs to mosaic-free training. The evidence therefore characterizes LMF-YOLO as a low-cost mechanism for shifting scale-sensitive coverage, with fixed-schedule repetitions providing the next confirmatory test.

Keywords

UAV Imagery, Post-Disaster Search and Rescue, Person Detection, Small-Object Detection, Weighted Feature Fusion, YOLO11n, Lightweight Neural Networks

1. Introduction

Unmanned aerial vehicles can survey damaged areas while keeping responders out-

side unstable structures, blocked roads, and other immediate hazards. Their overhead viewpoint also supports rapid coverage of large areas. The visual conditions remain demanding. People often occupy only a small portion of the frame, debris creates repeated textures, and altitude, occlusion, smoke, illumination, and camera motion alter target appearance. A practical detector therefore benefits from small-target sensitivity, repeatable evaluation, and a compact computational profile.

Modern one-stage detectors are attractive for this setting because they combine localization and classification in a single forward pass. The original YOLO formulation established this real-time design direction [1], and subsequent detectors have improved multi-scale prediction, training objectives, and deployment efficiency. YOLO11n offers a compact baseline with feature-pyramid outputs at three spatial resolutions. Its standard neck follows a fixed fusion pattern, while the proposed extension learns the contribution of a scale-matched backbone correction after top-down and bottom-up aggregation.

Feature pyramids address scale variation by joining high-resolution spatial information with lower-resolution semantic information. FPN introduced a top-down pathway with lateral connections [2], PANet strengthened bottom-up information flow [3], and BiFPN used normalized trainable weights to fuse features efficiently [4]. These designs established that the routing and weighting of cross-scale information can affect detection quality. They also illustrate a recurring tradeoff: adding feature paths may help small targets, but additional heads, large attention blocks, or repeated bidirectional stages can increase memory use and latency. For a nano-scale detector, the value of an architectural change should be judged against both accuracy and deployment cost.

This work studies a deliberately small modification. LMF-YOLO retains the stock YOLO11n backbone, neck, and three detection scales. It adds one residual weighted fusion to each P3, P4, and P5 output. The main input to every fusion is the pretrained stock neck feature, and the secondary input is the scale-matched backbone feature. The resulting architecture keeps the stock information path dominant at initialization while allowing training to learn a scale-specific correction.

The study also addresses reproducibility when a custom YAML architecture relocates the detection layer. A direct layer-index load transfers the backbone and stock neck, while an explicit remapping step aligns the pretrained detection tensors with the relocated head. Architecture validation confirmed 499 transferred tensors, 468,034 transferred values, 87.24% target-tensor coverage, and a successful transfer through the rebuilt training model.

The main contributions are as follows:

A residual P3-P5 weighted-fusion design that preserves the stock YOLO11n neck and adds only three pairs of scalar fusion parameters plus one P3 channel projection.

A pretrained-transfer procedure that explicitly remaps a moved detection head and verifies the transfer after the training model is rebuilt.

A three-seed evaluation covering standard validation and frozen-test metrics, COCO-style object-size strata, parameter and FLOP counts, checkpoint size, and matched FP32 and FP16 GPU latency.

The results reveal a change in operating characteristics. Validation recall and test precision increased, small-object recall rose descriptively, and FP16 batch-one throughput improved on the measured GPU. Frozen-test overall AP favored A0, and seed 97 exposed schedule sensitivity. The study therefore reports the method as a compact, auditable design whose accuracy claim centers on measured trade-offs and protocol-controlled follow-up.

2. Related Work

2.1. Object Detection for Aerial and Disaster Imagery

Aerial object detection differs from conventional ground-level detection because targets are viewed from a variable altitude, occupy few pixels, and appear in scenes with strong background repetition. The VisDrone benchmark helped formalize these challenges through drone-captured images containing dense objects, occlusions, and large scale variation [5]. The TinyPerson study likewise showed that very small human instances require evaluation and training strategies that account for the mismatch between ordinary detection datasets and tiny-person imagery [6]. Although the present post-disaster dataset is smaller than these benchmarks, it shares the operational difficulty that person instances may be visually weak relative to the surrounding scene.

Search-and-rescue systems add a decision consequence to this visual problem. A camera-equipped UAV can assist field teams by highlighting candidate locations for human review, and trained personnel provide the final confirmation. Goodrich *et al.* described the broader role of small UAVs in wilderness search and rescue [7], while recent disaster-oriented human-detection work has expanded benchmark coverage [8].

2.2. Lightweight One Stage Detection

Two-stage detectors such as Faster R-CNN use proposal generation followed by final prediction and can provide strong accuracy [9]. One-stage detectors integrate these operations into a direct prediction pipeline. YOLO formulated detection as grid-based regression and achieved real-time inference [1]. The family has since adopted multi-scale heads, improved label assignment, and more efficient feature-processing blocks. For UAV use, parameter count, memory behavior, execution graph, precision mode, and target hardware jointly determine deployment suitability.

Lightweight convolutional networks commonly use depthwise separable operations or inverted residual blocks to reduce computation. Xception demonstrated the effectiveness of depthwise separable convolution [10], while MobileNet and MobileNetV2 developed this idea for mobile models [11] [12]. A detector can nevertheless become slower after a nominal FLOP reduction if an added operation is

poorly optimized for a device. For this reason, the present study reports both analytical complexity and measured latency. It also treats the FP32 and FP16 results separately instead of assuming that lower precision always produces a proportional speedup.

2.3. Multi Scale Feature Fusion

FPN combines semantically strong low-resolution features with higher-resolution features through lateral connections [2]. PANet adds a bottom-up path that shortens the route from low-level features to later prediction layers [3]. EfficientDet's BiFPN emphasizes multi-input nodes, repeats bidirectional fusion, and learns normalized weights for different inputs [4]. Weighted fusion has a small parameter cost, and its effect depends on insertion point and initialization. A pretrained-path-dominant initialization supports a smooth start to fine-tuning when the lateral feature enters a new fusion point.

LMF-YOLO applies the weighting principle in a restricted residual setting. It uses a single fusion stage at each stock neck output, with the pretrained path dominant at initialization. This choice isolates the effect of scale-matched residual fusion and keeps the additional computation explicit.

2.4. Evaluation of Small Objects

Average precision summarizes the precision-recall curve over a selected intersection-over-union threshold, while COCO-style AP averages over thresholds from 0.50 to 0.95 [13]. Aggregate AP can hide a detector's behavior on small instances because medium and large objects may dominate the annotation set. Size-stratified AP and average recall therefore provide useful complementary evidence. At the same time, a stratum with only one or two instances is too small for stable inference. This study reports the tiny category for completeness but bases interpretation primarily on the small and medium-plus categories, which contain more observations.

Repeated training is valuable when model differences approach optimization variance. Detector training depends on initialization, mini-batch order, and augmentation. The current experiment therefore uses three deterministic seeds and reports the mean, sample standard deviation, paired differences, and per-seed results. These summaries support descriptive interpretation, while additional fixed-schedule repetitions can tighten uncertainty intervals.

3. Proposed Method

3.1. Design Objective

The design objective is to increase access to scale-matched backbone detail while preserving the nano-scale deployment budget. Four constraints guide the design: three detection strides of 8, 16, and 32; retention of the pretrained stock neck; a small parameter increment; and explicit checkpoint transfer into the relocated detection head.

3.2. Baseline Architecture

A0 is the standard YOLO11n detector configured for one class. The backbone uses convolutional downsampling, C3k2 feature blocks, spatial pyramid pooling, and a C2PSA block. The neck constructs P3, P4, and P5 outputs through a top-down path followed by bottom-up aggregation. The detection module receives feature maps at 80 × 80, 40 × 40, and 20 × 20 for a 640 × 640 input. Both A0 and A2 share the baseline label assignment, classification objective, distribution-based box regression, and post-processing settings.

3.3. Residual Lightweight Weighted Fusion

A2 first executes the complete stock YOLO11n neck and then fuses each neck output with the backbone feature at the same spatial scale. Figure 1 traces this data flow: the stock P3-P5 neck outputs remain the main branch, scale-matched backbone features provide residual inputs, and each fused feature enters the decoupled Detect head. At nano scale, the P3 projection is a 1 × 1 convolution from 128 to 64 channels with stride 1, zero-padding, and bias set to False. The Ultralytics Conv block applies BatchNorm2d and SiLU after the convolution. PyTorch initializes the convolution with its default Kaiming-uniform rule; BatchNorm starts with unit scale and zero offset, and Ultralytics sets epsilon to 10^{-3} and momentum to 0.03. The P4 and P5 pairs already match in spatial size and channel count.

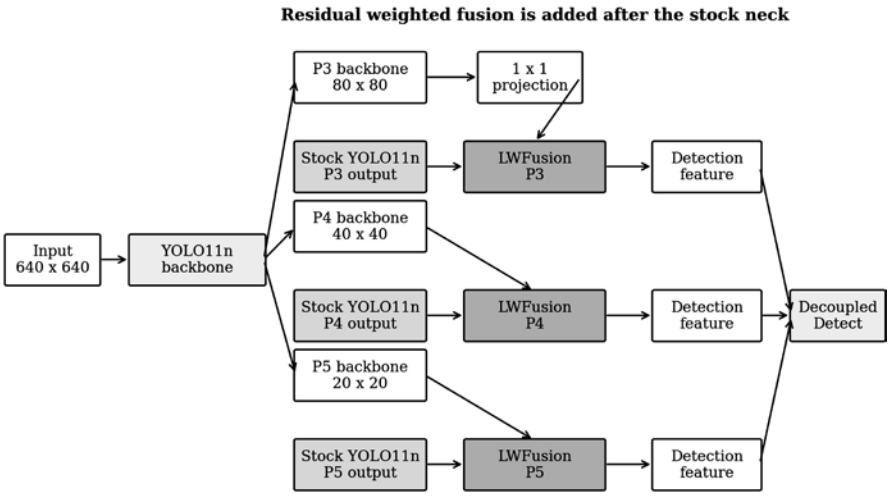


Figure 1. Revised LMF-YOLO architecture. The stock YOLO11n neck is retained, and residual weighted fusion is applied only to the P3, P4, and P5 outputs.

Let X1 and X2 denote aligned feature tensors with identical batch, channel, height, and width dimensions. Each fusion block owns two trainable scalar parameters w_1 and w_2. A softplus transform maps them to positive values. The normalized coefficient for input i is

$$\alpha_i = \text{softplus}(w_i) \Big/ \left(\varepsilon + \sum_j \text{softplus}(w_j)\right) \quad (1)$$

where epsilon = 10^{-4} ensures stable normalization. The fused feature is

$$Y = \sum_i \alpha_i X_i \tag{2}$$

Figure 2 makes each LWFusion step explicit: the two raw scalar weights pass through softplus, the positive terms are normalized into coefficients, and the weighted sum produces the fused feature Y. With the initialization $w = [1, -3]$, the stock-neck path begins at an effective coefficient of approximately 0.964 and the backbone path at approximately 0.036, preserving the pretrained feature as the dominant starting signal. Each block adds two scalar parameters, while projection and spatial alignment remain explicit in the model definition, which keeps the computational-cost audit transparent.

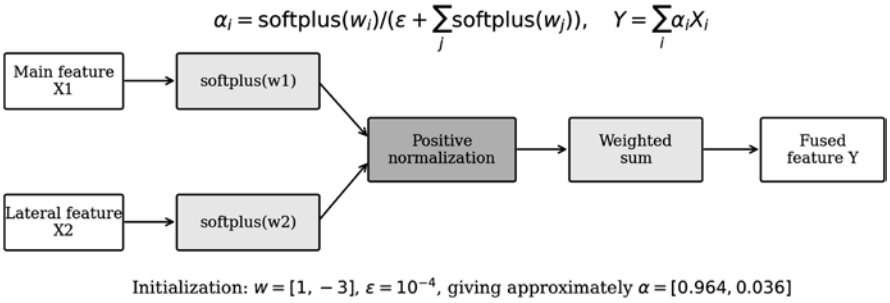


Figure 2. Computation and initialization of one LWFusion block.

3.4. Pretraining Preserving Initialization

The main feature X1 is the pretrained stock-neck output, and X2 is the corresponding backbone feature. Raw weights start at [1, −3]. After softplus transformation and normalization, the effective coefficients are approximately alpha_neck = 0.9643 and alpha_backbone = 0.0357. The model therefore begins close to the pretrained stock detector and learns scale-specific corrections during fine-tuning. Final effective coefficients are obtained directly from each evaluated A2 best.pt checkpoint.

Positive coefficients preserve the direction of every input feature, and normalization keeps the combined scale bounded relative to the inputs. These properties follow the fast normalized-fusion principle [4], applied once per output scale in the present architecture.

3.5. Transfer of the Displaced Detection Head

Adding three fusion layers changes the sequential index of the final Detect module. Ultralytics checkpoints use layer-indexed state-dictionary names, so the revised loader maps the source detection layer 23 to target layer 30. Direct transfer covered 378 tensors, and remapping added 121 tensors, yielding 499 transferred tensors and 468,034 parameter values across 572 target tensors. The resulting pretrained-tensor coverage was 0.8724.

A second model is built through the training path and loaded from the first model. A probe tensor in the box-regression branch remains identical after recon-

struction. The validation script also confirms strides [8, 16, 32]. Together, these tests verify detection-head transfer and model-rebuild consistency.

3.6. Computational Cost

Table 1 isolates the controlled architectural difference between the two models. A0 and A2 use the same stock YOLO11n backbone, P3-P5 neck, detection strides, and detection loss; A2 adds the residual backbone-to-output fusion path, three pairs of normalized scalar weights, and one P3 projection. At 640 × 640 resolution, this change increases the parameter count from 2,582,347 to 2,590,609 and computation from 6.4 to 6.5 GFLOPs, corresponding to increments of 0.32% and 1.56%.

Table 1. Structural comparison of the evaluated models.

Item	A0 baseline	A2 LMF-YOLO
Backbone	Stock YOLO11n	Stock YOLO11n
Neck	Stock P3-P5 neck	Stock P3-P5 neck
Additional path	Reference path	Residual backbone-to-output fusion
Fusion weights	Fixed reference path	3 pairs of positive normalized scalars
Detection strides	8, 16, 32	8, 16, 32
Loss	Stock detection loss	Stock detection loss
Parameters	2,582,347	2,590,609
GFLOPs	6.4	6.5

4. Experiments

4.1. Dataset and Split Protocol

Experiments used a cleaned single-class subset of the Post-Disaster Dataset (PDD) [14], audited from repository commit 70884690b2aaf3c44f9f507c8dbffe4116db4532. The public primary subset began with 879 images and 2578 valid person boxes. Cleaning excluded one image whose empty label file conflicted with visible people and one member of an exact duplicate pair. The locked pool contained 877 images and 2577 boxes. Label checks required five finite YOLO fields, class 0, centers in [0, 1], positive width and height within (0, 1], and corner coordinates within a tolerance of 10^{-6}.

Exact duplicate groups were identified by SHA-256. Near-duplicate candidates used perceptual-hash distance 4 or lower, and sequence groups used filename families with a frame gap of 25 or lower. All members of each connected group entered one partition. A seeded search with random seed 20260909 evaluated 30,000 group assignments while balancing image count, box count, tiny-at-640 count, and small-at-640 count. Table 2 links the final counts to their analytical roles: 619 training images with 1783 boxes support parameter optimization, 112

validation images with 313 boxes govern checkpoint selection and repeated evaluation, and 146 frozen-test images with 481 boxes provide the final generalization assessment. Model development therefore uses training and validation before one common evaluation procedure is applied to the frozen test set. **Appendix A** provides the complete audit record.

Table 2. Dataset split and annotation summary.

Split	Images	Person instances	Use
Training	619	1783	Parameter optimization
Validation	112	313	Checkpoint selection and repeated evaluation
Frozen test	146	481	Final generalization assessment

4.2. Metrics

The standardized evaluator reports precision, recall, AP50, AP75, and mAP50-95 for the person class. IoU thresholds span 0.50 to 0.95 in increments of 0.05. The evaluator uses confidence threshold 0.001, NMS IoU threshold 0.70, and a maximum of 300 detections per image. COCO precision uses 101 recall samples, maxDets is [1, 10, 300], and the reported AP and AR use maxDet 300.

The scale evaluator maps each original-image ground-truth box to the 640 × 640 letterboxed input before computing area. Tiny uses $[0, 16^2)$ pixels, small uses $[16^2, 32^2)$ pixels, and medium-plus uses $[32^2, \text{infinity})$ pixels. Validation contains 2 tiny, 32 small, and 279 medium-plus instances; the frozen test contains 1, 39, and 441. These counts support descriptive scale analysis, with the small and tiny strata interpreted in relation to their sample sizes.

For each model, the tables report the arithmetic mean and sample standard deviation over seeds 17, 43, and 97. Differences use paired A2-minus-A0 values in percentage points. Means, standard deviations, and differences are calculated from the full-precision JSON and CSV records. Per-seed values appear in **Appendix C**.

4.3. Training Configuration

Both models began from the same YOLO11n checkpoint and used Ultralytics 8.4.144, Python 3.11.16, PyTorch 2.11.0 with CUDA 12.8, and one NVIDIA GeForce RTX 4060 Ti with 8 GB memory. Training used 640 × 640 images, batch size 8, and a maximum of 200 epochs. AdamW [15] used an initial learning rate of 0.001, weight decay 0.0005, and a cosine schedule.

Augmentation settings were constant across configurations: HSV hue 0.015, saturation 0.4, value 0.3, rotation 5 degrees, translation 0.1, scale 0.35, shear 2 degrees, horizontal flip probability 0.5, mosaic probability 0.5, and MixUp probability 0.0. The planned schedule started mosaic-free training at epoch 186 and allocated the final 15 epochs to that phase when a run reached 200 epochs. Early-

stopping patience was 30, seeds were 17, 43, and 97, automatic mixed precision was enabled, and workers were set to 0 for Windows reproducibility. As documented in Table 3, the input, optimizer, augmentation, schedule, repetition, and runtime controls were shared across A0 and A2, keeping the architectural modification as the planned comparison.

Table 3. Common training configuration.

Category	Setting	Value
Input	Image size	640 × 640
Input	Batch size	8
Schedule	Maximum epochs	200
Schedule	Early-stopping patience	30
Schedule	Mosaic-free phase	Epochs 186 - 200 when reached
Selection	Validation fitness	mAP50-95
Optimization	Optimizer	AdamW
Optimization	Initial learning rate	0.001
Optimization	Weight decay	0.0005
Optimization	Cosine schedule	Enabled
Augmentation	Mosaic/MixUp	0.5/0.0
Repetition	Random seeds	17, 43, 97
Runtime	Workers/AMP	0/enabled

4.4. Training Behavior

Figure 3 shows how validation mAP50-95 evolves across all six runs. The three A0 curves extend to 197, 200, and 198 epochs and include 12, 15, and 13 mosaic-free epochs. A2 seeds 17 and 43 extend through 200 epochs, while A2 seed 97 ends at epoch 97 after reaching its best value at epoch 67. This run therefore contributes zero mosaic-free epochs, compared with 15 for each of the other two A2 runs. The unequal schedule exposure makes seed 97 the principal source of protocol sensitivity in the three-seed result. **Appendix C** records every completion and best-epoch value.

Both models reach broad validation plateaus. The checkpoint rule selects the epoch with the highest Ultralytics detection fitness. In version 8.4.144, the fitness weights are [0, 0, 0, 1] for precision, recall, AP50, and mAP50-95, so fitness equals validation mAP50-95. Checkpoint selection used validation fitness exclusively, and frozen-test plus scale-stratified metrics were generated after selection.

4.5. Standardized Validation Results

Table 4 identifies a recall-oriented validation profile across the three seeds. A2

raises mean recall from 0.7410 to 0.7614, a paired gain of 2.04 percentage points, while mean precision changes from 0.8681 to 0.8358. AP50 and AP75 increase by 0.39 and 0.31 points, and mAP50-95 changes from 0.53095 to 0.53053, a paired difference of −0.04 points. The aggregate mAP values remain closely matched while the precision-recall balance shifts toward recall.

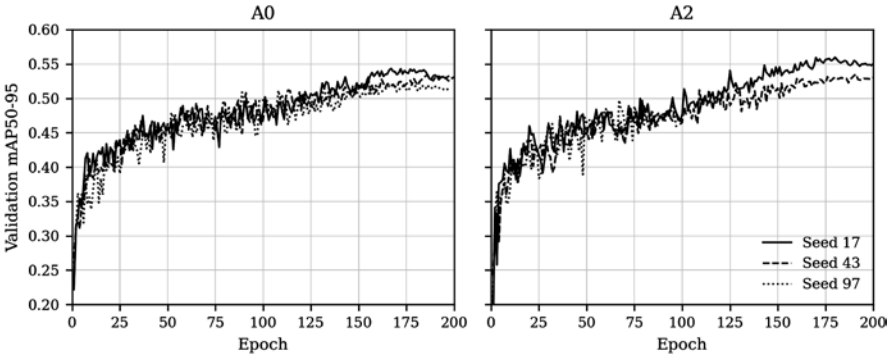


Figure 3. Validation mAP50-95 curves for the three A0 and A2 training runs. Curves terminate at the completed epoch for each seed.

Table 4. Standardized validation results across three seeds. Means and paired differences use unrounded JSON values.

Model	Precision	Recall	AP50	AP75	mAP50-95
A0	0.8681 ± 0.0287	0.7410 ± 0.0149	0.8360 ± 0.0094	0.5720 ± 0.0292	0.5309 ± 0.0106
A2	0.8358 ± 0.0317	0.7614 ± 0.0213	0.8399 ± 0.0050	0.5751 ± 0.0505	0.5305 ± 0.0313
Paired delta pp	−3.23	+2.04	+0.39	+0.31	−0.04

The paired changes are modest for AP50, AP75, and mAP50-95, while the precision and recall shifts are larger. This pattern characterizes A2 as a model with a different validation operating balance and an average validation accuracy level close to A0.

4.6. Frozen Test Results

Table 5 records a different frozen-test profile for the same checkpoints. A2 raises mean precision from 0.8190 to 0.8405, while recall changes from 0.6958 to 0.6656. AP50, AP75, and mAP50-95 change by −0.30, −2.23, and −1.15 percentage points. The mAP50-95 standard deviation increases from 0.0039 for A0 to 0.0347 for A2, making seed-level variation central to the frozen-test interpretation.

Table 5. Frozen-test results across three seeds. Means and paired differences use unrounded JSON values.

Model	Precision	Recall	AP50	AP75	mAP50-95
A0	0.8190 ± 0.0085	0.6958 ± 0.0142	0.7609 ± 0.0107	0.4818 ± 0.0184	0.4568 ± 0.0039
A2	0.8405 ± 0.0202	0.6656 ± 0.0150	0.7579 ± 0.0245	0.4595 ± 0.0502	0.4453 ± 0.0347
Paired delta pp	+2.15	−3.02	−0.30	−2.23	−1.15

The frozen-test evidence supports a cautious, split-specific interpretation: A2 emphasizes precision, while overall AP favors A0. The wider dispersion across A2 checkpoints also motivates direct inspection of the three paired seeds.

The paired bars in Figure 4 expose the source of this seed dependence. A2 exceeds A0 for seeds 17 and 43 on both validation and frozen test, while seed 97 contributes the lower A2 value and the larger standard deviation. A sensitivity summary restricted to seeds 17 and 43 gives validation mAP50-95 of 0.5362 ± 0.0076 for A0 and 0.5472 ± 0.0169 for A2, with a paired change of +1.10 ± 0.92 points. The corresponding frozen-test values are 0.4589 ± 0.0019 and 0.4651 ± 0.0069, with a paired change of +0.63 ± 0.51 points. This n = 2 summary remains descriptive and motivates fixed-schedule confirmation.

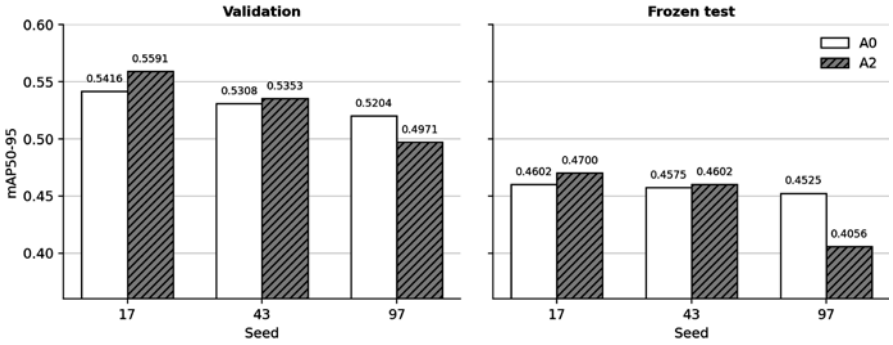


Figure 4. Standardized mAP50-95 by model, seed, and split. Labels retain four decimal places from the raw JSON records.

4.7. Object Size Analysis

Table 6 reveals the scale-specific pattern behind the aggregate metrics. For small objects, A2 changes validation AP by +2.30 ± 3.82 points and frozen-test AR by +4.44 ± 4.60 points. Across the small and medium-plus strata, recall deltas are positive on both partitions, while AP deltas vary by split. This combination describes broader candidate coverage together with seed-dependent confidence ranking and localization.

Table 6. COCO-style scale-stratified AP and AR. Values are mean ± sample SD; paired deltas are A2 minus A0 in percentage points.

Split	Area	A0 AP	A2 AP	Paired delta AP pp	A0 AR	A2 AR	Paired delta AR pp
Validation	Tiny	0.252 ± 0.050	0.158 ± 0.122	−9.44 ± 9.13	0.250 ± 0.050	0.267 ± 0.076	+1.67 ± 12.58
Validation	Small	0.279 ± 0.032	0.302 ± 0.017	+2.30 ± 3.82	0.451 ± 0.005	0.459 ± 0.017	+0.83 ± 1.54
Validation	Medium-plus	0.536 ± 0.018	0.545 ± 0.017	+0.93 ± 0.18	0.640 ± 0.020	0.654 ± 0.016	+1.42 ± 3.06
Validation	All	0.508 ± 0.015	0.518 ± 0.016	+0.97 ± 0.40	0.618 ± 0.018	0.632 ± 0.014	+1.36 ± 2.77
Frozen test	Tiny	0.000 ± 0.000	0.001 ± 0.002	+0.09 ± 0.15	0.000 ± 0.000	0.100 ± 0.173	+10.00 ± 17.32
Frozen test	Small	0.082 ± 0.022	0.087 ± 0.012	+0.53 ± 3.38	0.214 ± 0.046	0.258 ± 0.020	+4.44 ± 4.60

Continued

Frozen test	Medium-plus	0.470 ± 0.013	0.464 ± 0.030	−0.69 ± 1.79	0.585 ± 0.009	0.597 ± 0.003	+1.13 ± 1.15
Frozen test	All	0.439 ± 0.013	0.433 ± 0.029	−0.56 ± 1.66	0.554 ± 0.010	0.568 ± 0.004	+1.42 ± 1.35

Figure 5 translates the small-object rows of Table 6 into paired bars. On validation, A2 raises mean AP50-95 from 0.279 to 0.302 and AR50-95 from 0.451 to 0.459. On the frozen test, the corresponding means rise from 0.082 to 0.087 for AP and from 0.214 to 0.258 for AR. The error bars show material cross-seed variation, especially around the frozen-test recall increase, so the result is reported as a descriptive pattern for fixed-schedule confirmation.

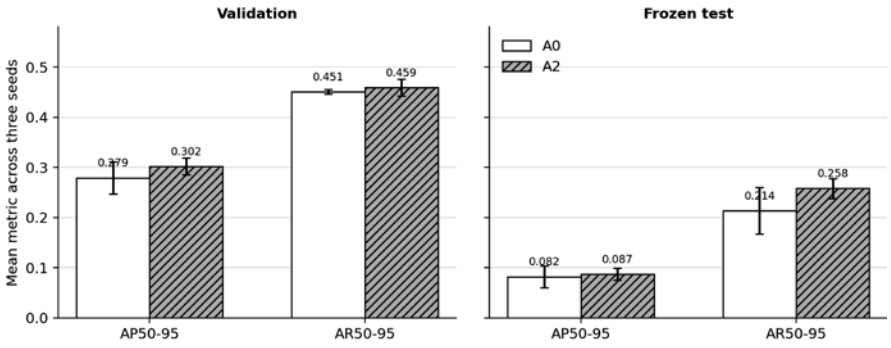


Figure 5. Small-object AP50-95 and AR50-95 across three seeds. Bars show means and error bars show sample standard deviations.

The medium-plus category dominates both partitions. A2 changed validation medium-plus AP and AR by +0.93 ± 0.18 and +1.42 ± 3.06 points. On the frozen test, the paired changes were −0.69 ± 1.79 points for AP and +1.13 ± 1.15 points for AR. Together with the small stratum, this pattern characterizes a coverage-versus-ranking trade-off whose magnitude varies by seed.

4.8. Complexity and Deployment Benchmark

Table 7 quantifies the deployment cost of residual fusion. A2 adds 8,262 parameters, increasing the model by 0.32%, and adds 0.1 GFLOPs, an increase of 1.56%. Its checkpoint grows from 5.236 to 5.257 MB, a difference of 0.021 MB or 0.39%. These increments keep the modification within the nano-scale computational range established by A0.

Table 7. Model complexity and checkpoint size.

Model	Parameters	GFLOPs	Checkpoint MB	Change from A0
A0	2,582,347	6.4	5.236	Reference
A2	2,590,609	6.5	5.257	+0.32% params; +1.56% FLOPs; +0.39% file

Table 8 shows precision-dependent runtime behavior on the RTX 4060 Ti. Un-

der FP32, A2 changes median latency from 11.09 to 11.56 ms and throughput from 90.20 to 86.49 FPS. Under FP16, median latency changes from 13.70 to 12.53 ms and throughput from 72.98 to 79.83 FPS. Peak memory differs by 0.57 MB in FP32 and 0.06 MB in FP16, indicating closely matched memory demand.

Table 8. Batch-one GPU inference benchmark.

Precision	Model	Median ms	P95 ms	FPS	Peak memory MB
FP32	A0	11.09	19.62	90.20	59.65
FP32	A2	11.56	19.63	86.49	60.22
FP16	A0	13.70	21.25	72.98	45.78
FP16	A2	12.53	20.67	79.83	45.84

Figure 6 makes the runtime contrast visible: the A2 bar has higher latency and lower FPS under FP32, whereas its FP16 latency is lower and its FPS is higher. The benchmark used one fixed validation image, batch size one, 50 warm-up iterations, and 500 timed iterations, so the plotted differences describe the measured PyTorch implementation on the RTX 4060 Ti.

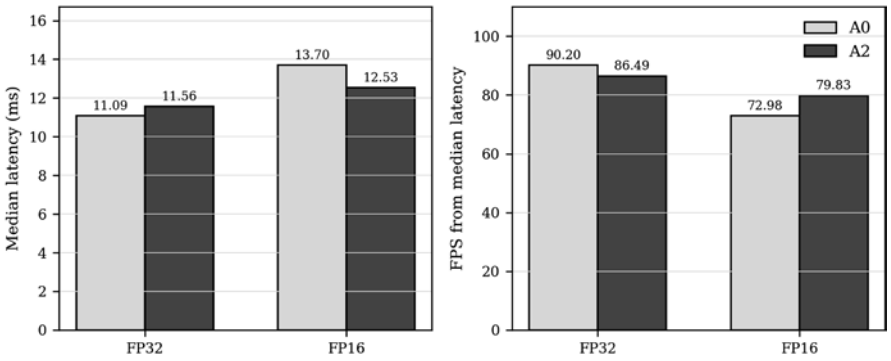


Figure 6. Median latency and derived throughput under the matched batch-one GPU benchmark.

The FP16 result is an implementation-and-hardware observation. Both models achieved higher throughput in FP32 in this batch-one PyTorch test, indicating a strong contribution from conversion, kernel-launch, and synchronization overhead. The matched procedure supports comparison on this platform, while deployment-specific engines remain the appropriate basis for embedded claims.

5. Discussion

5.1. Main Findings

The central finding is that A2 changes the detector's operating profile at minimal model cost. Validation recall, frozen-test precision, and descriptive small-object recall increased, while frozen-test overall AP favored A0. The full-precision vali-

dation means are 0.53095 for A0 and 0.53053 for A2; the frozen-test means are 0.45675 and 0.44530. These values replace rounded cross-source summaries throughout the manuscript.

The residual design contributes a compact and auditable mechanism. It adds 0.32% parameters, preserves the stock neck, begins with the pretrained path at effective weight 0.9643, and exposes scale-specific coefficients for checkpoint-level inspection. The transfer audit also confirms that the relocated detection head receives the intended pretrained tensors.

5.2. Interpretation for Search and Rescue

In an assisted search workflow, the preferred operating point depends on the review process. A candidate-generation workflow can favor recall on small people through a lower confidence threshold and a larger human-review queue. A bandwidth-constrained workflow can favor precision through a stricter threshold. The reported precision-recall trade-offs therefore guide threshold calibration more directly than a single aggregate rank.

The frozen-test small-object AR increase suggests that A2 can retrieve more candidate regions when the evaluator permits multiple detections. The smaller AP change indicates that confidence ranking and localization remain key optimization targets. Operational evaluation can connect these findings to false positives per image, recall at a fixed review budget, and time to verify an alert.

5.3. Sources of Variability

Three factors shape uncertainty. First, 112 validation images and 146 frozen-test images allow individual predictions to influence aggregate metrics. Second, the tiny and small strata contain 1 and 39 frozen-test instances. Third, A2 seed 97 completed 97 epochs and remained in the mosaic phase throughout training. The per-seed tables and paired standard deviations make each source of variability visible.

Scene content, object scale, occlusion, and annotation difficulty can also differ between validation and frozen test. The current archive supports aggregate, per-seed, and scale-stratified analysis. A subsequent error study can extend this evidence with box-area distributions, crowding, blur, illumination, altitude proxies, and scene-level false-positive categories.

5.4. Study Scope and Evidence Boundaries

The present confirmatory scope is one cleaned post-disaster dataset. External evaluation on VisDrone, TinyPerson, or another rescue-oriented dataset can measure transfer beyond this split. The current three-seed design quantifies first-order optimization variability, and additional fixed-schedule repetitions can narrow the interval around the paired effects.

The current architecture comparison contrasts A0 with the complete A2 residual-fusion design. A controlled component study can separate symmetric initial-

ization, P3-only fusion, P4-P5 fusion, and a fixed P3 projection. Direct extraction from the three evaluated A2 best.pt files will provide the final trained alpha_neck and alpha_backbone values at P3, P4, and P5. The initialization values remain separately documented for provenance.

Deployment evidence currently covers a desktop RTX 4060 Ti and the native PyTorch runtime. TensorRT FP16 and INT8 exports on the target platform can extend the benchmark to power, video decoding, telemetry, and end-to-end response time. The 5.257 MB checkpoint and batch-one timing provide the present computational reference.

Human-centered metrics complete the operational picture. Missed-location rate, alert clustering, review time, and confidence calibration can quantify decision support for rescue personnel. The intended workflow combines model proposals, geospatial tracking, and trained human verification.

5.5. Confirmatory Training and Evaluation Plan

The confirmatory protocol keeps the frozen A0 and A2 definitions, uses seeds 17, 43, and 97, trains every run for 200 epochs, sets patience to 0 or above 200, and applies close_mosaic = 15 so epochs 186 - 200 share the same mosaic-free phase. For each run, best.pt is selected by validation mAP50-95. The frozen test is evaluated after all six checkpoints and analysis scripts are locked. This protocol directly tests the seed-97 sensitivity under equal schedule exposure.

The next analysis will regenerate **Figure 3**, **Figure 4**, **Tables 4-6**, and the appendix from the six fixed-schedule checkpoints. It will also extract P3, P4, and P5 effective fusion coefficients from the evaluated best.pt files. A subsequent external-dataset study can add confidence-calibration curves, false positives per image, recall at a fixed review budget, and TensorRT FP16 and INT8 benchmarks.

6. Conclusions

This paper presented LMF-YOLO, a lightweight YOLO11n variant that adds residual positive-normalized feature fusion at P3, P4, and P5 while preserving the stock neck and pretrained detection head. The implementation adds 0.32% parameters and 1.56% FLOPs. Across three archived runs, validation mAP50-95 changed from 0.53095 ± 0.01065 to 0.53053 ± 0.03128, and frozen-test mAP50-95 changed from 0.45675 ± 0.00389 to 0.44530 ± 0.03469. Small-object frozen-test AR showed a descriptive paired increase of 4.44 ± 4.60 points.

The evidence supports LMF-YOLO as a low-cost mechanism for adjusting scale-sensitive coverage and deployment behavior. The checkpoint-transfer audit, group-isolated data split, full-precision metric pipeline, and per-seed appendix strengthen reproducibility. The fixed 200-epoch confirmation plan and direct best.pt coefficient extraction provide clear next steps for a protocol-matched accuracy claim.

Data Availability

The source images are available from the public Post-Disaster Dataset repository

associated with Reference 14 (https://github.com/HaoqianSong/Post-Disaster-Dataset). The audit records repository commit 70884690b2aaf3c44f9f507c8dbffe4116db4532. The project materials retain the cleaned split manifests, group assignments, exact- and near-duplicate reports, evaluation outputs, and scale manifests used in this study.

Code Availability

The project package contains the revised model YAML, LWFusion module, pretrained detection-head remapping logic, architecture verification scripts, training configuration, standardized evaluator, scale evaluator, result collector, and deployment benchmark. The package records the exact environment versions, seeds, configuration fields, and audit hashes needed to reproduce the reported pipeline.

Ethical and Operational Considerations

UAV imagery can contain identifiable people and sensitive location information. Collection and use should follow applicable aviation, privacy, data-protection, and emergency-response procedures. Trained personnel should review model predictions, and every rescue decision should combine independent confirmation with established operational protocols.

Declaration of AI-Assisted Technology Use

During the preparation of this manuscript, the authors used OpenAI ChatGPT and Codex to support English-language refinement, improve textual clarity and organization, assist with document formatting, and review code presentation.
The authors independently designed and conducted the experiments and verified the training data, model outputs, statistical calculations, references, tables, figures, interpretations, and scientific conclusions. All AI-assisted suggestions were critically reviewed and revised by the authors, who assume full responsibility for the accuracy, originality, and final content of the manuscript.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/cvpr.2016.91

[2] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017) Feature Pyramid Networks for Object Detection. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 936-944. https://doi.org/10.1109/cvpr.2017.106

[3] Liu, S., Qi, L., Qin, H., Shi, J. and Jia, J. (2018) Path Aggregation Network for Instance Segmentation. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recog-*

nition, Salt Lake City, 18-23 June 2018, 8759-8768. https://doi.org/10.1109/cvpr.2018.00913

[4] Tan, M., Pang, R. and Le, Q.V. (2020) EfficientDet: Scalable and Efficient Object Detection. 2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 13-19 June 2020, 10778-10787. https://doi.org/10.1109/cvpr42600.2020.01079

[5] Zhu, P., Wen, L., Bian, X., Ling, H. and Hu, Q. (2018) Vision Meets Drones: A Challenge. arXiv: 1804.07437.

[6] Yu, X., Gong, Y., Jiang, N., Ye, Q. and Han, Z. (2020) Scale Match for Tiny Person Detection. 2020 *IEEE Winter Conference on Applications of Computer Vision* (*WACV*), Snowmass, 1-5 March 2020, 1246-1254. https://doi.org/10.1109/wacv45572.2020.9093394

[7] Goodrich, M.A., Morse, B.S., Gerhardt, D., Cooper, J.L., Quigley, M., Adams, J.A., *et al.* (2007) Supporting Wilderness Search and Rescue Using a Camera-Equipped Mini UAV. *Journal of Field Robotics*, **25**, 89-110. https://doi.org/10.1002/rob.20226

[8] Nihal, R.A., Yen, B., Itoyama, K. and Nakadai, K. (2024) UAV-Enhanced Combination to Application: Comprehensive Analysis and Benchmarking of a Human Detection Dataset for Disaster Scenarios. In: Antonacopoulos, A., Chaudhuri, S., Chellappa, R., Liu, CL., Bhattacharya, S. and Pal, U., Eds., *Lecture Notes in Computer Science*, Springer, 145-162. https://doi.org/10.1007/978-3-031-78341-8_10

[9] Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Advances in Neural Information Processing Systems*, **28**, 91-99. https://proceedings.neurips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[10] Chollet, F. (2017) Xception: Deep Learning with Depthwise Separable Convolutions. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 1800-1807. https://doi.org/10.1109/cvpr.2017.195

[11] Howard, A.G., Zhu, M., Chen, B., *et al.* (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861.

[12] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/cvpr.2018.00474

[13] Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., *et al.* (2014) Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B. and Tuytelaars, T., Eds., *Lecture Notes in Computer Science*, Springer International Publishing, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[14] Song, H., Song, W., Cheng, L., Wei, Y. and Cui, J. (2024) PDD: Post-Disaster Dataset for Human Detection and Performance Evaluation. *IEEE Transactions on Instrumentation and Measurement*, **73**, 1-14. https://doi.org/10.1109/tim.2023.3346508

[15] Loshchilov, I. and Hutter, F. (2019) Decoupled Weight Decay Regularization. https://openreview.net/forum?id=Bkg6RiCqY7

Appendix

Appendix A. Dataset Cleaning and Split Audit

The audit used the public PDD repository at commit 70884690b2aaf3c44f9f507c8dbffe4116db4532. The public primary subset contained 879 decodable images and 2578 valid person boxes. The supervised study retained the primary subset. The auxiliary 112TinyPerson subset was reserved for source-data inspection because its public release contained 47 images whose label files were pending. The restricted 155real survivor material was placed outside the experimental pool because the public repository supplied access screenshots rather than source image-label pairs.

Cleaning removed one image with an empty label file despite visible people and one member of an exact duplicate pair. Every retained label contained five finite YOLO fields, class 0, normalized centers in [0, 1], positive normalized width and height within (0, 1], and box corners within a tolerance of 10^{-6} around the image domain. The locked pool therefore comprised 877 images and 2577 person instances (Table A1).

Table A1. Cleaning and split audit.

Audit item	Images	Person boxes	Recorded action
Public primary subset	879	2578	Starting audit pool
Empty-label image with visible people	1	0 recorded	Excluded
Exact duplicate member	1	1	Excluded
Locked primary pool	877	2577	Grouped split source
Training	619	1783	Model optimization
Validation	112	313	Checkpoint selection
Frozen test	146	481	Final assessment

Leakage control combined SHA-256 exact-duplicate grouping, perceptual-hash candidate detection at Hamming distance 4 or lower, and filename-family sequence grouping with a frame gap of 25 or lower. The audit identified one exact-duplicate group containing two images and 30 near-duplicate candidate pairs. All members of each connected group were assigned to one partition. The split search used seed 20260909 and 30,000 random group assignments while balancing image count, box count, tiny-at-640 count, and small-at-640 count. The resulting image proportions were 70.58%, 12.77%, and 16.65% for training, validation, and frozen test. Source metadata supports group-level leakage control, while event-level independence remains an explicit target for future metadata collection.

Appendix B. Evaluation Configuration

The standardized and scale-stratified evaluations used a 640 × 640 letterboxed in-

put. Ground-truth areas were computed after mapping each original-image box into the letterboxed input coordinates. Tiny covers area [0, 256), small covers [256, 1024), and medium-plus covers [1024, infinity), so each boundary belongs to exactly one stratum. The evaluation and checkpoint-selection settings are summarized in Table B1.

Table B1. Evaluation and checkpoint-selection settings.

Component	Setting	Value
Input	Letterboxed resolution	640 × 640
Prediction	Confidence threshold	0.001
Prediction	NMS IoU threshold	0.70
Prediction	Maximum detections per image	300
COCO evaluation	IoU thresholds	0.50:0.05:0.95
COCO evaluation	Recall sampling	101 points
COCO evaluation	maxDets	[1, 10, 300]
Reported AP and AR	Maximum detections	300
Model selection	Validation fitness	mAP50-95
Checkpoint	Selection rule	Highest validation fitness

Ultralytics 8.4.144 defines detection fitness with weights [0, 0, 0, 1] over precision, recall, AP50, and mAP50-95. Accordingly, fitness equals validation mAP50-95. Checkpoint selection used this validation quantity throughout training. Frozen-test metrics and area-stratified metrics were generated after checkpoint selection and served as confirmatory analyses.

Appendix C. Per-Seed Results and Training Protocol Audit

Tables C1-C4 summarize the per-seed detection metrics, scale-stratified results, and training protocol audit.

Table C1. Standard metrics for each evaluated checkpoint.

Split	Model	Seed	Precision	Recall	AP50	AP75	mAP50-95
Validation	A0	17	0.8871	0.7279	0.8304	0.6000	0.5416
Validation	A0	43	0.8352	0.7572	0.8308	0.5742	0.5308
Validation	A0	97	0.8821	0.7380	0.8469	0.5417	0.5204
Validation	A2	17	0.8616	0.7508	0.8405	0.6215	0.5591
Validation	A2	43	0.8455	0.7859	0.8445	0.5824	0.5353
Validation	A2	97	0.8004	0.7476	0.8347	0.5213	0.4971

Continued

Frozen test	A0	17	0.8121	0.7006	0.7674	0.4996	0.4602
Frozen test	A0	43	0.8165	0.7069	0.7668	0.4629	0.4575
Frozen test	A0	97	0.8285	0.6798	0.7486	0.4831	0.4525
Frozen test	A2	17	0.8365	0.6807	0.7682	0.5039	0.4700
Frozen test	A2	43	0.8625	0.6653	0.7756	0.4696	0.4602
Frozen test	A2	97	0.8226	0.6507	0.7299	0.4050	0.4056

Table C2. Scale-stratified AP50-95 for each checkpoint.

Split	Model	Seed	Tiny	Small	Medium-plus	All
Validation	A0	17	0.2020	0.2581	0.5555	0.5246
Validation	A0	43	0.2525	0.2631	0.5298	0.5010
Validation	A0	97	0.3030	0.3158	0.5216	0.4981
Validation	A2	17	0.0196	0.3194	0.5635	0.5354
Validation	A2	43	0.2525	0.2859	0.5411	0.5141
Validation	A2	97	0.2020	0.3006	0.5301	0.5033
Frozen test	A0	17	0.0000	0.1058	0.4809	0.4495
Frozen test	A0	43	0.0000	0.0620	0.4740	0.4423
Frozen test	A0	97	0.0000	0.0779	0.4565	0.4246
Frozen test	A2	17	0.0000	0.0732	0.4816	0.4505
Frozen test	A2	43	0.0000	0.0943	0.4801	0.4489
Frozen test	A2	97	0.0026	0.0941	0.4292	0.4002

Table C3. Scale-stratified AR50-95 for each checkpoint.

Split	Model	Seed	Tiny	Small	Medium-plus	All
Validation	A0	17	0.2000	0.4500	0.6616	0.6371
Validation	A0	43	0.2500	0.4562	0.6366	0.6157
Validation	A0	97	0.3000	0.4469	0.6219	0.6019
Validation	A2	17	0.3500	0.4406	0.6523	0.6288
Validation	A2	43	0.2500	0.4719	0.6398	0.6201
Validation	A2	97	0.2000	0.4656	0.6706	0.6466
Frozen test	A0	17	0.0000	0.2154	0.5950	0.5630
Frozen test	A0	43	0.0000	0.2590	0.5825	0.5551
Frozen test	A0	97	0.0000	0.1667	0.5787	0.5441

Continued

Frozen test	A2	17	0.0000	0.2359	0.5934	0.5632
Frozen test	A2	43	0.0000	0.2744	0.5977	0.5703
Frozen test	A2	97	0.3000	0.2641	0.5991	0.5713

Table C4. Training completion and mosaic-free exposure.

Model	Seed	Completed epochs	Best epoch	Best log mAP50-95	Mosaic-free epochs
A0	17	197	167	0.5435	12
A0	43	200	186	0.5339	15
A0	97	198	168	0.5208	13
A2	17	200	180	0.5599	15
A2	43	200	191	0.5352	15
A2	97	97	67	0.4977	0

The model definition initializes every LWFusion pair at effective coefficients alpha_neck = 0.9643 and alpha_backbone = 0.0357. Final trained coefficients are checkpoint attributes. Their definitive report therefore uses direct extraction from the three evaluated A2 best.pt files at P3, P4, and P5. The current portable evidence archive contains training logs and evaluation outputs, while the checkpoint files remain with the training workspace. The submission package will pair the final coefficient table with those exact evaluated checkpoints and will keep initialization values in a separate provenance field.