STRANDED CAPACITY: DEEP PYRAMID LEVELS EARN NO RETURN ON SMALL OBJECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In UAV imagery, remote sensing, and traffic surveillance, small instances account for more than 70% of all objects, against about one third on COCO. Existing work appends modules to a general-purpose detector but inherits the allocation of capacity across pyramid depth, channel width, and input resolution, with the computational budget fixed for a different scale distribution and left largely unexamined. We vary each axis in turn and measure its marginal return per unit of compute. The optimum departs from convention on all three axes, and on two of them the sign reverses: removing deep heads or compressing deep channels raises AP_{50}. We call this condition stranded capacity and trace it to a single geometric quantity, the effective feature area a target occupies on the grid from which the target is predicted. This area scales inversely with the squared pyramid stride and quadratically with input scale. These two dependencies predict the sign of the return on every axis. Reading the configuration from the measured returns yields a three-step transformation that adds no modules. Extensive experiments show that the configuration holds across both detection paradigms. On VisDrone, R-YOLOv11-L matches an identically resolved baseline (41.5 versus 40.8 AP_{50}) with **96% fewer parameters and 76% less compute, and runs 2.3× faster**. R-D-FINE-L gains 5.1 AP_{50} over the corresponding baseline with 87% fewer parameters. The same configuration transfers to five further datasets without retuning, gaining 20.2 and 18.0 AP_{50} on TinyPerson and SODA-D. The code is included in the supplementary material.

1 INTRODUCTION

Modern detector architectures took shape on benchmarks such as COCO (Lin et al., 2014), where object sizes span a wide range, so that every level of the feature pyramid has targets to serve. The YOLO and DETR families differ in head design and label assignment but share the same depth-stratified prediction structure. Most deployment settings, including UAV imagery, remote sensing, and traffic surveillance (Wu et al., 2022; Cheng & Han, 2016; Tang et al., 2019), lack this scale profile. Among the ten benchmarks in Figure 1, all nine deployment datasets have at least 70% of their instances in the smallest size band, compared with about one third on COCO. The supports of the COCO and deployment size distributions barely overlap, and this difference exceeds what is typically considered a domain shift.

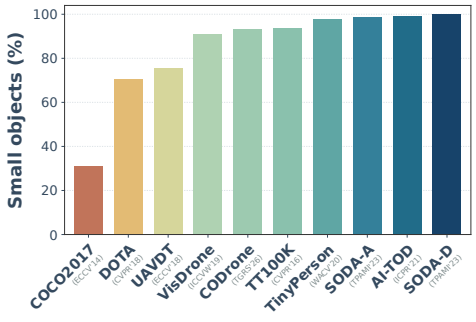


Figure 1: Share of small instances ($\sqrt{wh} < 32$ after resizing to 640) across ten detection benchmarks. Protocols in Appendix A.1.

Research addressing this difference follows three directions: preserving a high-resolution pathway (Xiao et al., 2025; Peng et al., 2026; Zhang et al., 2024), incorporating attention and context (Han et al., 2025; Hu et al., 2025b; Huang et al., 2024), and modifying the supervision signal (Xu et al., 2022b;a; Yuan et al., 2026a). Although these approaches intervene at different points, all retain the capacity allocation of a general-purpose detector across pyramid depth, channel width, and input resolution. The added modules therefore build on a computational budget set for a different scale

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

distribution. Nothing guarantees that this allocation remains appropriate once small objects dominate. In a depth-stratified architecture, each pyramid level serves a particular range of object sizes. When instances are concentrated in the smallest ranges, the levels provisioned for large objects continue to consume compute while the ranges they serve remain nearly empty. Adding modules to this base does not correct the underlying mismatch. The budget itself remains unexamined: how to allocate a fixed amount of compute across the three axes, and whether this allocation can be determined without a dataset-specific search.

Answering this requires examining how existing compute is allocated and whether that allocation matches the task, rather than how much compute to add. On VisDrone, we add a fourth detection head to the official YOLOv11-L at its highest-resolution pyramid level, so that every level produces predictions. We vary one axis at a time and estimate its marginal return as the ratio of the accuracy change to the compute change. The optimal direction differs from conventional detection practice on all three axes, with two favoring capacity reduction over expansion. Removing a deep detection head *raises* accuracy; reducing deep channel widths to one eighth of their original values incurs no accuracy loss. In the initial measured interval of each reinvestment trajectory, increasing input resolution yields a higher accuracy gain per GFLOP than widening channels. The first two findings indicate that capacity allocated to deep pyramid levels is *stranded*: it consumes compute without improving accuracy.

Both stranding and the high return on resolution follow from a geometric quantity we call the effective feature area: the area a target occupies on the feature map from which it is predicted. This area scales inversely with the squared pyramid stride and quadratically with input scale. Deep capacity is stranded because the size bands it serves contain no samples; resolution yields the highest return because it alone directly enlarges the effective feature area. This account predicts the sign of the return on each axis and can therefore be falsified on data with the opposite scale distribution. We conduct this control on COCO, where the ordering of head importance reverses completely under an identical protocol (Table 1). On VisDrone, closing the two deepest heads leaves accuracy unchanged; on COCO, closing them causes the largest accuracy losses. The shallowest head shows the opposite pattern. The detector is unchanged; the difference lies in the scale distributions shown in Figure 1.

The measured returns guide a three-step allocation procedure: remove heads with non-positive returns, compress deep channel groups according to their measured accuracy–compute trade-offs, and reinvest the released compute in input resolution. The operating points are calibrated once through axis-wise sweeps on VisDrone. The steps share one budget and must be applied together. The retention ratios form an inverted pyramid (Figure 2), contrary to the deeper-is-wider convention used since FPN and PANet (Lin et al., 2017; Liu et al., 2018). No modules are added; transformed models carry the prefix R-. On VisDrone, R-YOLOv11-L matches baseline accuracy at the same input resolution using $\mathbf{4\%}$ **of the parameters and** $\mathbf{24\%}$ **of the compute, and runs** $\mathbf{2.3\times}$ **faster**. On the DETR side, R-D-FINE-L raises its baseline by $\mathbf{5.1\ AP_{50}}$ **with** $\mathbf{87\%}$ **fewer parameters**.

Our contributions are as follows.

- On small-object data, capacity allocated to pyramid depth earns a non-positive marginal return, a condition we name *stranded capacity*; a negative control on COCO reverses the ordering of head importance, showing that stranding follows the scale distribution rather than the detector.
- We explain the reversal on all three axes by the effective feature area and define the cross-axis-comparable *capacity return* (CR), whose three sign predictions are tested separately; the detectability threshold it leaves free is a dataset-level constant shown not to transfer.
- The configuration read off the measured returns is calibrated once on VisDrone and transfers to five further datasets and two detector families without further search, giving a reusable allocation rule rather than a tuned architecture, whose cost and validity limits we report.

2 Related Work

Compensating for small objects. The dominant approach to small-object detection adds mechanisms to a general-purpose detector. One family adds high-resolution capacity: FBRT-

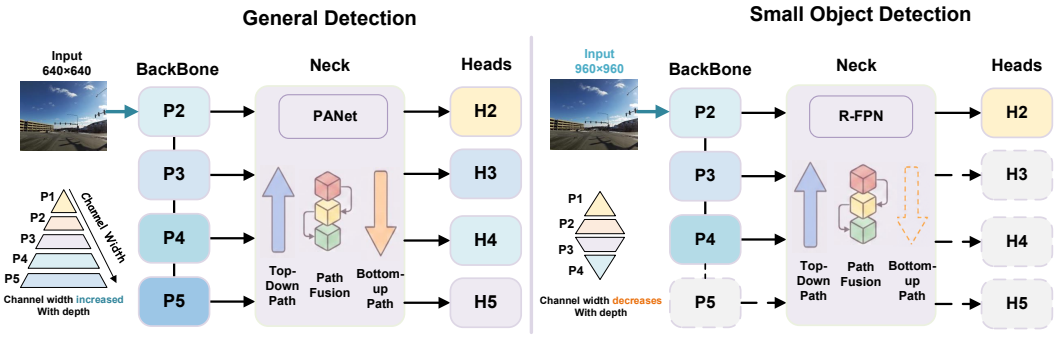


Figure 2: The inherited configuration (left) against the one read off the measured returns (right). General detection widens channels with depth and predicts at all four levels. The transformed network inverts the retention ratio, keeps only H_2, and discards P_5 together with the bottom-up pathway, while the input grows from 640 to 960. No module is added.

YOLO (Xiao et al., 2025) maintains a high-resolution backbone branch with cross-level complementation, while HR-SemNet (Peng et al., 2026) extracts classification semantics directly from high-resolution layers. Cross-layer fusion combines pyramid features for aerial targets (Zhang et al., 2024; Du et al., 2025; Liu et al., 2024; Yang et al., 2025; Wang et al., 2025). Other methods preserve the original signal before it collapses (Zheng et al., 2025; Guo et al., 2024). Tiled inference (Akyon et al., 2022; Unel et al., 2019) avoids spatial collapse by processing crops at near-native resolution, at a cost quadratic in image size. A second family adds discriminative capacity through attention and context, using contrastive multi-domain fusion (Han et al., 2025), query guidance based on local object density (Hu et al., 2025b; Huang et al., 2024; Zhang et al., 2025; Lei et al., 2026; Ma et al., 2025), or spatial-frequency component selection (Jia et al., 2026; Hu et al., 2025a; Sun et al., 2025). A third family refines the learning signal without altering feature structure, using Gaussian receptive-field assignment (Xu et al., 2022b), scale-robust box similarity (Xu et al., 2022a), uncertainty modelling (Yuan et al., 2026a), or detection-oriented rectification (Yuan et al., 2026b). All three families treat the inherited resource configuration as fixed and add small-object capability on top of it.

Questioning the configuration itself. A smaller literature questions individual configuration choices. YOLOF (Chen et al., 2021) shows that a single feature level matches a full pyramid under a general scale distribution and attributes FPN success to divide-and-conquer optimisation rather than multi-scale fusion. EFF (Gong et al., 2021) finds that the top-down pathway introduces large-object spatial information into shallow levels as noise; a dataset-specific fusion factor suppresses this interference. HRDNet (Liu et al., 2021) pairs shallow networks with high-resolution inputs. A sweep over sixty configurations reports the same inversion on another task: the smallest model at high resolution dominates the largest, and resolution is the strongest lever (Mbobda-Kuate & Kasmi, 2026). Each challenges one established practice but changes only one axis and is refitted per dataset. These axes are otherwise set through architecture search (Zoph et al., 2018; Real et al., 2019; Tan et al., 2019; Wu et al., 2019; Cai et al., 2020; Ghiasi et al., 2019; Du et al., 2020; Chen et al., 2019; Robinson et al., 2025) or pruning based on importance proxies such as BN scaling factors (Liu et al., 2017), reconstruction error (Luo et al., 2017), or task-level redundancy (Wei et al., 2026). Compound scaling (Tan & Le, 2019; Tan et al., 2020) is a third. All three differ in method but share one assumption: accuracy increases monotonically with capacity, so reducing capacity can at best preserve it. This premise fails on small-object-dominated data. Removing a deep head or compressing a deep channel group *raises* accuracy, an improvement that importance criteria based on monotonicity cannot capture. The released compute yields greater returns when allocated to input scale than retained as channel width.

3 Effective Feature Area

Small-object detection is limited not by the information a target carries but by its area on the feature map used for prediction. For a target of native side length d predicted at P_k after input rescaling, its

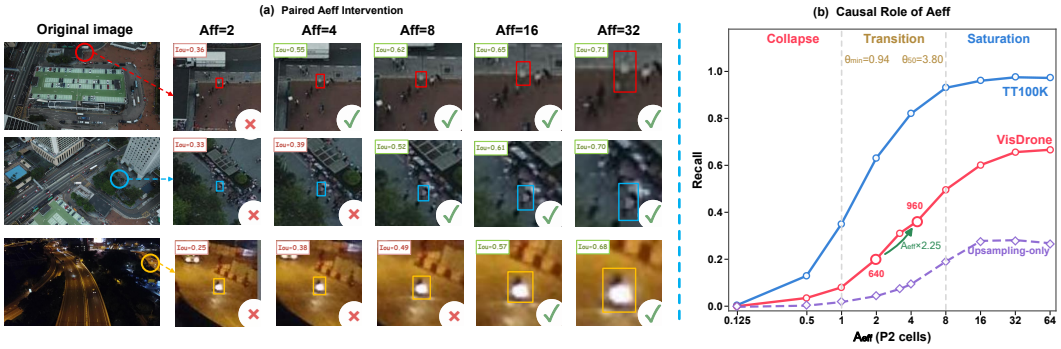


Figure 3: (a) Paired intervention: each target is cropped and rescaled to a prescribed A_{eff} on P_2 with input fixed at 960; all examples are strictly upsampled. (b) Recall versus A_{eff} in P_2 cells. Purple denotes 29 upsampling-only targets (95% CI) used as the causal control. The band begins at the VisDrone activation threshold $\theta_{\min} = 0.94$. The arrow shows the $2.25\times$ increase in median A_{eff} from 640 to 960.

area in feature-cell units is

$$A_{\text{eff}}(d, k) = \left(\frac{d\, s_{\text{in}}}{s_{\text{orig}}\, s_k} \right)^2, \tag{1}$$

where $s_k = 2^k$ is the stride of P_k, and s_{in} and s_{orig} denote the input and native image sizes. Each pyramid level has one detection head H_k, which maps the corresponding features to class scores and box regressions. Since H_k predicts only on P_k, its ability to detect targets of a given size depends on A_{eff} at that level. YOLOv11-L uses H_3 through H_5; the reference model in Section 4 adds H_2.

A_{eff} scales inversely with squared pyramid stride and quadratically with the input scale ratio. Moving one level deeper doubles the stride, so doubling input resolution exactly offsets the resulting area reduction; the computational costs differ substantially. Under the VisDrone setting ($s_{\text{in}} = 640$, $s_{\text{orig}} = 1400$), a target with side length 32 pixels covers 13.4 cells on P_2 but only 0.209 on P_5. A target with side length 8 covers less than one cell at each of the four levels.

Combined with the data scale distribution, A_{eff} yields three signed predictions about compute allocation in a general-purpose architecture. First, deep heads yield no return when small targets dominate because their size bands contain no samples. Second, deep channels have non-positive returns: they process regions unsupported by targets, and the top-down pathway propagates these responses into the shallow detection path, a structural effect verified separately in Appendix B.3. The cost therefore extends beyond wasted compute. Third, resolution yields the highest return across the three axes but has a point of exhaustion. Only resolution enlarges A_{eff}; the other two axes change capacity without changing coverage. Once targets generally exceed the detectability threshold, the return on further enlargement declines. All three predictions specify a direction and are tested separately in Section 4. The first also specifies when the ordering of head importance should reverse: on data with the opposite scale distribution.

The argument above attributes detectability to A_{eff}, but target size, sampled pixel count, and A_{eff} are strongly collinear; correlation alone cannot establish causality. We therefore intervene. At a fixed input of 960, we crop around each target and rescale it to a prescribed A_{eff} on P_2. Undetected targets still yield boxes with IoU below 0.5, indicating localisation failure rather than target disappearance. Upsampling introduces no new sampled detail, yet the 29 targets upsampled throughout the sweep show significantly higher recall as A_{eff} increases (95% confidence interval, Figure 3). This rules out an information deficit as the competing explanation. Cropping also changes context, so this experiment supports a causal role for A_{eff} without establishing it as the sole determinant.

A_{eff} transfers across datasets as a control coordinate; the activation threshold $\theta_{\min}$, below which targets are undetectable, does not. Median A_{eff} at peak accuracy is 31.35 on TT100K, 18.58 on VisDrone, and 8.46 on TinyPerson, a several-fold variation. TinyPerson provides the clearest counterexample: accuracy peaks at input size 1600 and then declines. At 2208, median area reaches

16.11 while accuracy continues to fall; $A_{\text{eff}} \approx 16$ is therefore not a universal optimum. $\theta_{\min}$ varies with target recognisability, background ambiguity, and the number of sub-threshold targets, requiring dataset-specific calibration. What fails is treating any particular area as a universal constant, not the mechanism.

4 Capacity Return Along Three Axes

4.1 Testing the three predictions

Testing the three predictions requires comparing axes with different units: discrete heads, channel-width proportions, and input-resolution scales. Two additional components are relevant. Neck-P_k denotes the neck pathway at level k, with neck and backbone channels counted separately. The top-down pathway comprises edges that propagate deep responses to shallow levels. To *remove* a head, we delete H_k from the computation graph and retrain. Leave-one-out removes one head and retains the rest to measure its marginal contribution. Single-head retention keeps one head and removes the rest to measure its standalone capability. To *close* a head, we retain the weights and discard its candidates before applying the same non-maximum suppression to the remaining candidates. This measures what the head currently contributes. These three interventions answer different questions.

Definition 1 (Capacity return)**.** *For two configurations x_i and x_j differing only along axis d, the capacity return of d over this interval is*

$$\mathrm{CR}_d(x_i \to x_j) = \frac{\mathrm{AP}_{50}(x_j) - \mathrm{AP}_{50}(x_i)}{\mathrm{FLOPs}(x_j) - \mathrm{FLOPs}(x_i)}, \tag{2}$$

in percentage points per GFLOP (pp/G).

Each measurement changes one axis while holding the other two and the training protocol fixed. For expansion, CR measures accuracy gained per added GFLOP; for withdrawal, it measures accuracy lost per GFLOP saved. Negative CR therefore means that removing compute improves accuracy. Within the measured interval, non-positive CR permits withdrawing compute without accuracy loss and reallocating it to the highest-return axis.

Deep heads earn no return. The four-head reference adds H_2 to the official YOLOv11-L, enabling predictions from P_2 through P_5 and achieving 37.7 AP_{50} at 109.4 GFLOPs. Under leave-one-out, removing any deep head *raises* AP_{50}: $-H_3$, $-H_4$, and $-H_5$ yield 37.9, 38.1, and 38.0, respectively. Removing the shallowest head reduces accuracy to 30.3. Under single-head retention, keeping only H_5, H_4, H_3, or H_2 yields 16.1, 28.5, 35.2, and 37.8, respectively. Accuracy increases monotonically as the retained head becomes shallower; the shallowest head alone reaches four-head accuracy at 80.8 GFLOPs. Prediction one holds.

Table 1: Peak band denotes the instance-size band with the largest recall decrease after closing a head; "none" means no band changes. Closing preserves weights and structure, so these values measure current head contributions, not capacity requirements, and support no speed claim. Per-band changes: Appendix B.2.

	Peak band (px)		ΔAP	
Closed	VisDrone	COCO	VisDrone	COCO
H_2	16–32	8–16	−13.27	−1.58
H_3	64–96	32–64	−3.54	−10.12
H_4	none	96–128	0.00	−12.45
H_5	none	≥ 128	0.00	−13.45

The two deepest heads are not suppressed; they find almost no matches. We run non-maximum suppression separately for each head and match predictions to ground truth across 75,102 boxes at five input resolutions. Across 375,510 observations, P_4 matches 3 (0.0008%), while P_5 matches none. Nor is their contribution merely absorbed during retraining after removal: closing the two deepest heads with unchanged weights leaves recall unchanged in every size band (Table 1).

On COCO, the same analysis yields the opposite ordering: closing H_4 and H_5 is most costly (-12.45 and -13.45 AP), while closing the shallowest costs 1.58. Size-stratified recall explains this pattern. Each head primarily affects one instance-size band, which shifts from smaller to larger targets with increasing depth. The three bands at and above 64 px are those served by the two deepest heads and contain 260 VisDrone instances versus 17,601 COCO instances. Because the datasets differ in more than scale, this is a consistency check rather than a causal test.

Deep channels earn a negative return. Prediction two requires only non-positive returns; the measurements are stronger. Using the model after head removal as the reference, we fix the heads and compress one channel group at a time, keeping all others at full width. The resulting order is $P_2 >$ Neck-$P_2 > P_3 >$ Neck-$P_3 > P_4$, with negative return at P_4. This is not an isolated point: compressing P_4 from 75% to 12.5% retention keeps $\Delta\mathrm{AP}_{50}$ non-negative throughout. This defines a free-compression region; an accuracy cost appears only below 12.5% (Appendix C.2). The shallow groups behave oppositely: any compression of P_2 or Neck-P_2 immediately reduces accuracy.

A negative marginal utility is what an importance criterion cannot express once it assumes that returns are monotone, and that assumption is shared by pruning and architecture search. Prediction two holds.

Resolution offers strong returns that diminish with scale. Two reinvestment trajectories start from the post-compression model: one increases input resolution, while the other uniformly widens the remaining channels. In the measured sweep, increasing resolution to 800 reaches 41.3 AP_{50} at 57.3 GFLOPs, compared with 39.7 AP_{50} at 66.3 GFLOPs for $1.5\times$ channel widening. The adjacent-point return of resolution decreases from 0.213 pp/G over 640–800 to 0.099 over 800–960, followed by 0.030 and 0.017 over the next two intervals. These measurements show diminishing returns at higher resolutions. We use 960 as an empirical operating point for the subsequent experiments.

4.2 The configuration that follows

We use the measured returns to select a three-step configuration. The steps follow a fixed order; the first two release compute for the final reinvestment. The first removes heads with non-positive returns, retains only the shallowest, and discards the corresponding deep stages and bottom-up pathway. The second selects an empirical accuracy–compute trade-off for the deep channel groups, retaining 50% of P_3, 12.5% of P_4, and 50% of Neck-P_3. Compressing these groups together is safe because the losses are approximately additive: the joint loss differs from the sum of the individual losses by less than 0.3 points, so the groups can be withdrawn in the order the single-factor measurements give, without searching over combinations. The third jointly increases resolution and compresses the shallow path. Compressing P_2 and Neck-P_2 alone is costly because their returns are highest. At a fixed pyramid stride, increasing the input scale by a factor r increases each target's effective feature area by r^2. We therefore jointly evaluate higher resolution and shallow-path compression to measure their net accuracy–compute trade-off. Input resolution increases from 640 to 960, while P_1, P_2, and Neck-P_2 retain 50%, 75%, and 50%, respectively. Section 5 reports the step-wise trajectory.

The resulting structure (Figure 2) forms an inverted pyramid in retention ratios: P_2, P_3, and P_4 retain 0.75, 0.50, and 0.125, respectively. Shallow levels retain a larger *fraction* of capacity, but not more absolute channels (P_2 has 192 versus 256 for P_3). Appendix A.3 provides the full specification. Transfer to another dataset repeats none of these measurements and follows a fixed rule: $s_{\mathrm{in}} = 960$. This fixed-input rule tests whether one configuration transfers without further search, rather than targeting a dataset-specific optimum. Transformed models are denoted R-YOLOv11-L and R-D-FINE-L.

5 Experiments

5.1 Setup

We evaluate on three public benchmarks with a high proportion of small objects, VisDrone (Du et al., 2019), AI-TOD (Wang et al., 2021), and TT100K (Zhu et al., 2016; Wang et al., 2024), covering UAV aerial and ground traffic domains, using their official splits and COCO-style metrics (Lin et al., 2014): AP averaged over IoU thresholds from 0.50 to 0.95 in steps of 0.05, AP_{50} at the single threshold 0.50, and AP_s over instances below 32^2 px, all computed with `pycocotools` on original-image areas. The size bands of Figure 1, Table 1 and Appendix B.2 follow a separate and finer convention, $\sqrt{wh}$ after letterboxing to the input; the two are not interchangeable, and no quantity is compared across them. TinyPerson (Yu et al., 2020), SODA-D (Cheng et al., 2023), and CODrone (Ye et al., 2026) serve as transfer tests. The R- models are trained and evaluated at 960×960 on all datasets. Baselines use 640×640 unless the tables state otherwise. The trans-

Table 2: VisDrone test-set results. Gray rows are models produced by the transformation; YOLOv11-L at 960 × 960 controls for resolution. Red/blue mark best/second-best values (lower for parameters/FLOPs). Small teal annotations give AP_{50} gains (points) and parameter/FLOP reductions (%) against each model's 640 × 640 baseline. FPS uses batch size 1.

Model	Input	Params (M)	FLOPs (G)	FPS	AP	AP_{50}	AP_S	AP_M	AP_L
YOLO-based detectors									
YOLOv8-L [Ultralytics'23]	640×640	43.61	164.9	32.3	20.3	34.9	9.8	31.2	42.8
YOLOv11-M [Ultralytics'24]	640×640	20.04	67.7	82.0	19.9	34.5	9.9	30.4	39.9
YOLOv11-L [Ultralytics'24]	640×640	25.29	86.6	59.9	20.4	35.1	10.0	31.7	40.0
YOLOv11-L [Ultralytics'24]	960×960	25.29	196.5	38.8	24.0	40.8	14.6	34.9	36.0
YOLO26-L [Ultralytics'26]	640×640	24.75	86.1	64.1	19.9	35.3	10.7	29.6	40.6
FBRT-YOLO-M [AAAI'25]	640×640	7.36	58.7	68.2	19.6	34.4	9.4	30.9	42.1
YOLO-Master-M [CVPR'26]	640×640	52.12	116.4	63.9	19.8	34.4	10.0	30.6	40.8
HR-SemNet [TIP'26]	640×640	5.91	181.1	48.5	24.9	42.9	14.9	36.6	45.7
R-YOLOv11-M	960×960	**0.74** -96.3%	34.3 -49.3%	**111.1**	22.6	39.2 +4.7	14.0	33.3	31.3
R-YOLOv11-L	960×960	0.99 -96.1%	46.6 -46.2%	**88.5**	24.0	41.5 +6.4	15.2	35.3	40.3
R-YOLO26-L	960×960	**0.98** -96.0%	44.3 -48.5%	86.2	24.0	41.3 +6.0	14.7	34.7	37.0
DETR-based detectors									
D-FINE-S [ICLR'25]	640×640	10.2	27.6	62.0	21.9	38.3	12.5	32.1	44.1
D-FINE-M [ICLR'25]	640×640	19.2	60.0	47.7	23.7	40.9	13.9	33.9	46.6
D-FINE-L [ICLR'25]	640×640	30.7	96.1	33.1	24.3	42.2	15.1	36.1	48.5
Dome-DETR-S [ACMMM'25]	800×800	13.2	176.5	17.2	26.9	45.5	17.1	37.2	**50.0**
DEIM-D-FINE-M [CVPR'25]	640×640	19.2	56.4	50.8	24.2	41.7	13.9	34.4	48.5
DEIMv2-S [arXiv'25]	640×640	9.7	25.4	60.3	20.4	36.3	10.9	29.9	45.1
RT-DETRv4-S [ECCV'26]	640×640	10.38	**24.9**	63.5	21.9	38.4	11.8	32.0	42.5
CSFPR-RTDETR [TGRS'25]	640×640	14.09	63.9	44.6	22.8	39.2	13.6	32.6	39.9
SFS-DETR-S [CVPR'26]	640×640	21.1	68.5	45.2	23.6	41.7	14.5	35.5	48.2
DINO [ICLR'23]	1333×750	47.6	274.0	13.2	25.3	44.5	15.0	37.1	**50.3**
R-D-FINE-S	960×960	1.69 -83.4%	**24.8** -10.3%	64.2	24.7	42.1 +3.8	16.5	33.6	41.0
R-D-FINE-M	960×960	2.88 -85.0%	55.9 -6.9%	49.4	**27.1**	**46.0** +5.1	**17.6**	**37.4**	42.4
R-D-FINE-L	960×960	3.95 -87.1%	84.1 -12.5%	44.0	**28.0**	**47.3** +5.1	**18.6**	**38.3**	43.0
Other detectors									
GFL [NeurIPS'20]	1333×750	32.3	206.0	16.5	19.3	32.1	9.4	30.1	40.9
RTMDet-L [arXiv'22]	640×640	52.3	80.2	23.2	19.9	32.9	9.8	29.9	43.7

formation is applied to two families: the YOLO line (Jocher et al., 2023; Jocher & Qiu, 2024; 2026) and the real-time DETR line that D-FINE (Peng et al., 2025) continues (Zhao et al., 2024; Huang et al., 2025b). We compare against recent small-object designs on the same benchmarks (Lin et al., 2026; Huang et al., 2025a; Chen et al., 2025) and against a high-resolution DETR-based reference (Zhang et al., 2023). Apart from the input resolution and the structural changes induced by the transformation, training hyperparameters follow the official implementation of each baseline; Appendix A.2 gives the complete configuration and Appendix A.1 the splits. Training runs on a single NVIDIA L20. The FPS reported in Tables 2 and 3 is measured in native PyTorch on an RTX 5070 Ti at batch size 1, without export or graph optimisation, so it is a conservative figure; the edge profiling of Table 4 uses TensorRT FP16 and is not comparable with it.

5.2 Main Results

Table 2 compares two detection paradigms at three model scales on VisDrone. Every R- variant simultaneously reduces compute and raises accuracy relative to its own baseline, contrary to the assumption that accuracy and capacity are monotonically related. The decisive row is the same-resolution control: the unmodified YOLOv11-L at a 960 input reaches 40.8 AP_{50} and requires 196.5 GFLOPs at 38.8 FPS, while R-YOLOv11-L at the same input reaches 41.5 using 46.6 GFLOPs at 88.5 FPS. The two are comparable in accuracy, while the latter uses one quarter of the compute and one twenty-sixth of the parameters, and runs 2.3× faster. Reallocation does not buy new accuracy; it makes that accuracy affordable. The same holds on the DETR side, where R-D-FINE-L attains the highest AP_{50} (47.3) and AP_s (18.6) in the table, above Dome-DETR-S (45.5 at 176.5 GFLOPs) and HR-SemNet (42.9 at 181.1 GFLOPs) at under half their compute.

Table 3: Transfer across five datasets, using the configuration derived on VisDrone without per-dataset tuning, tiling, or test-time augmentation. Gray rows are models produced by the transformation; red/blue mark best/second-best within each dataset block. Small teal annotations are gains over the 640 baseline of the same detector and therefore include a resolution component. All transformed models use 960×960 inputs. AP_m and AP_l are unavailable for AI-TOD and TT100K.

Model	Input	Params (M)	FLOPs (G)	AP	AP_{50}	AP_S
AI-TOD						
YOLOv11-L [Ultralytics'24]	640×640	25.29	86.6	26.3	54.1	24.7
D-FINE-L [ICLR'25]	640×640	30.7	96.1	**27.8**	55.6	**27.3**
CAEM-DETR [TGRS'25]	640×640	19.9	93.0	25.6	**56.7**	24.9
R-YOLOv11-L	960×960	**0.99** -96.1%	**46.6** -46.2%	26.5 +0.2	56.1 +2.0	25.9
R-D-FINE-L	960×960	**3.95** -87.1%	**84.1** -12.5%	**29.1** +1.3	**58.3** +2.7	**28.4**
TT100K						
YOLOv11-L [Ultralytics'24]	640×640	25.29	86.6	69.0	87.2	50.2
D-FINE-L [ICLR'25]	640×640	30.7	96.1	67.5	88.6	50.5
YOLO-TS [T-ITS'25]	640×640	11.1	99.1	72.3	91.2	58.9
R-YOLOv11-L	960×960	**0.99** -96.1%	**46.6** -46.2%	**74.4** +5.4	**93.4** +6.2	**61.6**
R-D-FINE-L	960×960	**3.95** -87.1%	**84.1** -12.5%	**74.1** +6.6	**92.7** +4.1	**60.5**
TinyPerson						
YOLOv11-L [Ultralytics'24]	640×640	25.29	86.6	7.1	22.0	5.0
R-YOLOv11-L	960×960	0.99 -96.1%	46.6 -46.2%	**15.1** +8.0	**42.2** +20.2	**13.5**
SODA-D						
YOLOv11-L [Ultralytics'24]	640×640	25.29	86.6	5.1	12.9	3.4
R-YOLOv11-L	960×960	0.99 -96.1%	46.6 -46.2%	**13.3** +8.2	**30.9** +18.0	**10.9**
CODrone						
YOLOv11-L [Ultralytics'24]	640×640	25.29	86.6	16.4	31.9	1.7
R-YOLOv11-L	960×960	0.99 -96.1%	46.6 -46.2%	**19.4** +3.0	**36.5** +4.6	**5.0**

Table 4: Jetson Orin Nano Super profiling with TensorRT FP16 in MAXN SUPER mode. Latency is the median of 1,000 runs; power is sampled by the INA3221 sensor at 100 Hz. S2/S3 denote transformation steps 2/3. FPS multipliers compare models at the same input; AP_{50} gains use the 640 baseline. Input denotes the square image side length; engine and GPU memory are in MiB. Best values are bold, including ties.

Model	Input	Latency (ms)	FPS	Power (W)	Energy (mJ)	Engine (M)	GPU (M)	AP	AP_{50}	AP_S
YOLOv11-L	640	25.91	38.6	**12.28**	318.2	51.8	84	20.4	35.1	10.0
YOLOv11-L	960	52.99	18.9	13.15	696.7	51.8	128	**24.0**	40.8	14.6
R-YOLOv11-L (S2)	640	**17.25**	**58.0** (1.50×)	12.58	**217.0**	**4.7**	**45**	20.9	36.8 **+1.7**	11.6
R-YOLOv11-L (S3)	960	31.53	31.7 (1.68×)	12.47	393.3	**4.7**	106	**24.0**	**41.5** **+6.4**	**15.2**

5.3 Transfer without retuning

The configuration derived on VisDrone is applied unchanged to the remaining five datasets, with no repeated measurement, no per-dataset tuning, no tiled inference, and no test-time augmentation (Table 3). On AI-TOD, R-YOLOv11-L gains 2.0 points while reducing parameters by 96.1%, and R-D-FINE-L exceeds the 96.1-GFLOP D-FINE-L using 84.1 GFLOPs and 12.9% of its parameters. On TT100K, R-YOLOv11-L gains 6.2 points in AP_{50} and 11.4 points in AP_s. On TinyPerson, SODA-D, and CODrone the gains are 20.2, 18.0, and 4.6 points; the first two are fed as whole images rather than tiles, so the downsampling ratio is at its most extreme and coverage area is compressed hardest, and the gain is correspondingly largest. CODrone sits at the other end: its oriented annotations are converted to horizontal bounding boxes for the HBB protocol, which raises the effective size of every instance and leaves less stranded capacity to reclaim. This ordering agrees with the mechanism: the more severe the spatial collapse, the more stranded capacity there is to reclaim. Appendix D.4 places the same models against every published method we could compare on for AI-TOD and TT100K, and Appendix E shows the comparison on individual scenes.

Table 5: Step-wise ablation of YOLOv11-L on VisDrone. Steps 1–3 are applied cumulatively; parentheses show the change in AP_{50} from the preceding step. The last row is the unmodified baseline at 960×960. Best values are bold, including ties.

Step 1	Step 2	Step 3	Input (px)	Params (M)	FLOPs (G)	AP	AP_{50}
–	–	–	640×640	25.29	86.6	20.4	35.1
✓	–	–	640×640	9.16	74.4	21.2	37.4 **(+2.3)**
✓	✓	–	640×640	1.36	**36.2**	20.9	36.8 **(-0.6)**
✓	✓	✓	960×960	**0.99**	46.6	**24.0**	**41.5** **(+4.7)**
–	–	–	960×960	25.29	196.5	**24.0**	40.8

5.4 Edge deployment

FLOPs are a proxy, and the operating points have to survive the hardware they are meant for. Table 4 profiles them on a Jetson Orin Nano Super. At a fixed 640 input the model after the second step dominates the baseline on every axis at once — $1.50\times$ the throughput, 31.8% less energy per image, an engine one eleventh the size and 46% less GPU memory — while raising AP_{50} by 1.7 points. The claim that removing capacity can raise accuracy is therefore not an artefact of counting FLOPs. At 960 the comparison is against the identically-resolved baseline, which runs at 18.9 FPS on 697 mJ per image, below real time on this device; the transformed model is $1.68\times$ faster on 43.5% less energy and at higher accuracy. Reallocation turns an operating point that cannot be deployed into one that can.

5.5 Ablation

Step-wise ablation. Table 5 accumulates the three steps while listing compute and parameters. At a fixed 640 input the first step reduces parameters to 36% while raising AP_{50} by 2.3 points; that row is independent of resolution and on its own contradicts the monotonic-capacity assumption. The second step trades 0.6 points for half the compute and frees the budget the third step needs. The third raises the resolution while jointly compressing the shallow path, adding 4.7 points; its gain falls with target scale as the coverage account requires, by 31.0%, 12.4%, and 7.8% in relative terms on small, medium, and large objects.

Where the operating points come from. The operating points are selected from the axis-wise sweeps on VisDrone. Retaining 12.5% of P_4 preserves accuracy relative to the post-Step-1 reference, whereas retaining 50% of P_3 and Neck-P_3 incurs individual losses of 0.7 and 0.1 AP_{50}, respectively. Applying these three reductions together lowers compute from 74.4 to 36.2 GFLOPs with a 0.6-point loss. These settings therefore represent an empirical accuracy–compute trade-off. The resolution sweep shows diminishing incremental returns beyond 960, which is the input size used for the subsequent shallow-path compression.

6 Conclusion

Detectors carry capacity that earns no return once small objects dominate the scale distribution. We separately measure the marginal returns of pyramid depth, channel width, and input resolution, find sign reversals on two axes, and trace all three to the feature-map area occupied by a target. The account is falsifiable: on COCO, where the size bands served by deep heads are populated, the same measurement yields the opposite ordering. A three-step transformation calibrated by axis-wise sweeps on VisDrone adds no modules, matches an identically resolved baseline at one quarter of the compute, and transfers under one fixed rule to five further datasets and two detector families. The detectability threshold is not constant across datasets and must be calibrated rather than assumed. What generalises is the direction of reallocation, not a particular area. A natural next step is to read the stranded levels directly from the training-set distribution of effective feature area before any training, so that only the detectability threshold, and with it the input resolution, remains to be calibrated. Whether the same account governs other dense prediction tasks with skewed scale distributions is left to future work.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to assist with writing and editing the manuscript for readability. We also used them to identify relevant literature and to suggest a structure for the paper. We did not use generative AI tools to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate or prove mathematical claims, to propose or refine hypotheses, to design research methodology or experiments, to implement methods, to clean or reformat datasets, or to interpret results. All experimental design, implementation, measurement, and interpretation reported here were carried out by the authors. We have reviewed all AI-assisted text against the underlying measurements and take responsibility for the final content of this work, including its text, claims, and artifacts.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

This work uses publicly available detection benchmarks under their original licences and introduces no new data collection. Several of them, including VisDrone, TinyPerson, and CODrone, contain images of people captured from aerial platforms; we use them only for measuring detector behaviour and perform no identification, tracking, or re-identification. Methods that make small-object detection cheaper to deploy also lower the cost of aerial surveillance, and we note that the reallocation described here is not specific to any application. The reduction in compute and parameters lowers the energy cost of both training and deployment.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

The appendix reports the complete hyperparameter configuration for each detector family, dataset splits and preprocessing, and the full capacity-return sweep from which every operating point in Section 4.2 is read. The supplementary material includes code with the exact scripts and command lines used to reproduce every table and figure, pinned dependencies and an environment file, random seeds and deterministic flags, and the transformed model configurations.

REFERENCES

Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. In *IEEE International Conference on Image Processing*, pp. 966–970, 2022. doi: 10.1109/ICIP46576.2022.9897990.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-All: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018. doi: 10.1109/CVPR.2018.00644.

Junzhou Chen, Heqiang Huang, Ronghui Zhang, Nengchao Lyu, Yanyong Guo, Hong-Ning Dai, and Hong Yan. YOLO-TS: Real-time traffic sign detection with enhanced accuracy using optimized receptive fields and anchor-free fusion. *IEEE Transactions on Intelligent Transportation Systems*, 26(11):19995–20011, 2025. doi: 10.1109/TITS.2025.3597710.

Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13039–13048, 2021. doi: 10.1109/CVPR46437.2021.01284.

Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. DetNAS: Backbone search for object detection. In *Advances in Neural Information Processing Systems*, 2019.

Gong Cheng and Junwei Han. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:11–28, 2016. doi: 10.1016/j.isprsjprs.2016.03.014.

Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023. doi: 10.1109/TPAMI.2023.3290594.

Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7369–7378, 2021. doi: 10.1109/CVPR46437.2021.00729.

Bowei Du, Yecheng Huang, Jiaxin Chen, and Di Huang. Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13435–13444, 2023. doi: 10.1109/CVPR52729.2023.01291.

Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Lin, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In *IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V. Le, and Xiaodan Song. SpineNet: Learning scale-permuted backbone for recognition and localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11592–11601, 2020. doi: 10.1109/CVPR42600.2020.01161.

Zewen Du, Zhenjiang Hu, Guiyu Zhao, Ying Jin, and Hongbin Ma. Cross-layer feature pyramid transformer for small object detection in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–14, 2025. doi: 10.1109/TGRS.2025.3572706.

Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R. Scott, and Weilin Huang. TOOD: Task-aligned one-stage object detection. In *IEEE/CVF International Conference on Computer Vision*, pp. 3490–3499, 2021. doi: 10.1109/ICCV48922.2021.00349.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, 2019. doi: 10.1109/CVPR.2019.00720.

Yuqi Gong, Xuehui Yu, Yao Ding, Xiaoke Peng, Jian Zhao, and Zhenjun Han. Effective fusion factor in FPN for tiny object detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1160–1168, 2021. doi: 10.1109/WACV48630.2021.00120.

Guangqian Guo, Pengfei Chen, Xuehui Yu, Zhenjun Han, Qixiang Ye, and Shan Gao. Save the tiny, save the all: Hierarchical activation network for tiny object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(1):221–234, 2024. doi: 10.1109/TCSVT.2023.3284161.

Zhangheng Han, Yang Xu, Jun Li, Javier Plaza, Antonio Plaza, Zhihui Wei, and Zebin Wu. CAEM-DETR: Small aerial target detection via contrastive attention-enhanced multidomain fidelity fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–14, 2025. doi: 10.1109/TGRS.2025.3614370.

Lei Hu, Jiwen Yuan, Bailiang Cheng, and Qizhi Xu. CSFPR-RTDETR: Real-time small object detection network for UAV images based on cross spatial frequency domain and position relation. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–19, 2025a. doi: 10.1109/TGRS.2025.3601828.

Zhangchi Hu, Peixi Wu, Jie Chen, Huyue Zhu, Yijun Wang, Yansong Peng, Hebei Li, and Xiaoyan Sun. Dome-DETR: DETR with density-oriented feature-query manipulation for efficient tiny object detection. In *ACM International Conference on Multimedia*, pp. 101–110, 2025b. doi: 10.1145/3746027.3754861.

Shihua Huang, Yongjie Hou, Longfei Liu, Xuanlong Yu, and Xi Shen. Real-time object detection meets DINOv3, 2025a.

Shihua Huang, Zhichao Lu, Xiaodong Cun, Yongjun Yu, Xiao Zhou, and Xi Shen. DEIM: DETR with improved matching for fast convergence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15162–15171, 2025b. doi: 10.1109/CVPR52734.2025.01412.

Yi-Xin Huang, Hou-I Liu, Hong-Han Shuai, and Wen-Huang Cheng. DQ-DETR: DETR with dynamic query for tiny object detection. In *European Conference on Computer Vision*, pp. 290–305, 2024. doi: 10.1007/978-3-031-73116-7_17.

D. Jia, J. Wang, L. Zhang, Z. Liu, and X. Wang. SFS-DETR: Spatial-frequency selection for UAV object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Glenn Jocher and Jing Qiu. Ultralytics YOLO11. `https://github.com/ultralytics/ultralytics`, 2024.

Glenn Jocher and Jing Qiu. Ultralytics YOLO26. `https://github.com/ultralytics/ultralytics`, 2026.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. `https://github.com/ultralytics/ultralytics`, 2023.

Huan Lei, Lei Shang, Hong Zhao, and Wenyuan Yang. Polarity aware detection transformer with hierarchical cross attention for unmanned aerial vehicle small object detection. *Pattern Recognition*, 179:113812, 2026. doi: 10.1016/j.patcog.2026.113812.

Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *Advances in Neural Information Processing Systems*, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

Xu Lin, Jinlong Peng, Zhenye Gan, Jiawen Zhu, and Jun Liu. YOLO-Master: MoE-accelerated with specialized transformers for enhanced real-time detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Hou-I Liu, Yu-Wen Tseng, Kai-Cheng Chang, Pin-Jyun Wang, Hong-Han Shuai, and Wen-Huang Cheng. A denoising FPN with Transformer R-CNN for tiny object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. doi: 10.1109/TGRS.2024.3396489.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018. doi: 10.1109/CVPR.2018.00913.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *IEEE/CVF International Conference on Computer Vision*, pp. 2736–2744, 2017. doi: 10.1109/ICCV.2017.298.

Ziming Liu, Guangyu Gao, Lin Sun, and Zhiyuan Fang. HRDNet: High-resolution detection network for small objects. In *IEEE International Conference on Multimedia and Expo*, pp. 1–6, 2021. doi: 10.1109/ICME51207.2021.9428241.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. ThiNet: A filter level pruning method for deep neural network compression. In *IEEE/CVF International Conference on Computer Vision*, pp. 5058–5066, 2017. doi: 10.1109/ICCV.2017.541.

Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. RTMDet: An empirical study of designing real-time object detectors, 2022.

Shuai Ma, Yihong Zhang, Ling Peng, Chen Sun, Baolong Ding, and Yongdong Zhu. OWRT-DETR: A novel real-time transformer network for small object detection in open-water search and rescue from UAV aerial imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–13, 2025. doi: 10.1109/TGRS.2025.3560928.

Kwame Mbobda-Kuate and Gabriel Kasmi. When bigger is worse: A practitioner's guide to model selection under data scarcity, 2026.

Can Peng, Manxin Chao, Ruoyu Li, Zaiqing Chen, Lijun Yun, and Yuelong Xia. HR-SemNet: A high-resolution network for enhanced small object detection with local contextual semantics. *IEEE Transactions on Image Processing*, 35:1096–1110, 2026. doi: 10.1109/TIP.2026.3654770.

Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-FINE: Redefine regression task in DETRs as fine-grained distribution refinement. In *International Conference on Learning Representations*, 2025.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 4780–4789, 2019. doi: 10.1609/aaai.v33i01.33014780.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. RF-DETR: Neural architecture search for real-time detection transformers, 2025.

Huixin Sun, Runqi Wang, Yanjing Li, Linlin Yang, Shaohui Lin, Xianbin Cao, and Baochang Zhang. SET: Spectral enhancement for tiny object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4713–4723, 2025. doi: 10.1109/CVPR52734.2025.00444.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114, 2019.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. MnasNet: Platform-aware neural architecture search for mobile. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2815–2823, 2019. doi: 10.1109/CVPR.2019.00293.

Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and efficient object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10778–10787, 2020. doi: 10.1109/CVPR42600.2020.01079.

Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8789–8798, 2019. doi: 10.1109/CVPR.2019.00900.

F. Ozge Unel, Burak O. Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 582–591, 2019. doi: 10.1109/CVPRW.2019.00084.

Hezheng Wang, Jiahui Liu, Jian Zhao, Jianzhong Zhang, and Dong Zhao. Precision and speed: LSOD-YOLO for lightweight small object detection. *Expert Systems with Applications*, 269: 126440, 2025. doi: 10.1016/j.eswa.2025.126440.

Jinwang Wang, Wen Yang, Haowen Guo, Ruixiang Zhang, and Gui-Song Xia. Tiny object detection in aerial images. In *International Conference on Pattern Recognition*, pp. 3791–3798, 2021. doi: 10.1109/ICPR48806.2021.9413340.

Junfan Wang, Yi Chen, Xiaoyue Ji, Zhekang Dong, Mingyu Gao, and Chun Sing Lai. Vehicle-mounted adaptive traffic sign detector for small-sized signs in multiple working conditions. *IEEE Transactions on Intelligent Transportation Systems*, 25(1):710–724, 2024. doi: 10.1109/TITS.2023.3309644.

Ziteng Wei, Qiang He, Bing Li, Feifei Chen, Hai Jin, and Yun Yang. NuWa: Deriving lightweight class-specific vision transformers for edge devices. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 156–166, 2026.

Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. FBNet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10726–10734, 2019. doi: 10.1109/CVPR.2019.01099.

Xin Wu, Wei Li, Danfeng Hong, Ran Tao, and Qian Du. Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 10(1):91–124, 2022. doi: 10.1109/MGRS.2021.3115137.

Yao Xiao, Tingfa Xu, Yu Xin, and Jianan Li. FBRT-YOLO: Faster and better for real-time aerial image detection. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 8673–8681, 2025. doi: 10.1609/aaai.v39i8.32937.

Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:79–93, 2022a. doi: 10.1016/j.isprsjprs.2022.06.002.

Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. RFLA: Gaussian receptive field based label assignment for tiny object detection. In *European Conference on Computer Vision*, pp. 526–543, 2022b. doi: 10.1007/978-3-031-20077-9_31.

Yuchuan Yang, Jiahui Dai, Yong Wang, and Yafei Chen. FM-RTDETR: Small object detection algorithm based on enhanced feature fusion with mamba. *IEEE Signal Processing Letters*, 32: 1570–1574, 2025. doi: 10.1109/LSP.2025.3553426.

Kai Ye, Haidi Tang, Bowen Liu, Pingyang Dai, and Liujuan Cao. CODrone: A comprehensive oriented object detection benchmark for UAV. *IEEE Transactions on Geoscience and Remote Sensing*, 64:1–15, 2026. doi: 10.1109/TGRS.2026.3677376.

Xuehui Yu, Yuqi Gong, Nan Jiang, Qixiang Ye, and Zhenjun Han. Scale match for tiny person detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1257–1265, 2020. doi: 10.1109/WACV45572.2020.9093394.

Xiang Yuan, Gong Cheng, Jiacheng Cheng, Ruixiang Yao, and Junwei Han. Unc-SOD: An uncertainty learning framework for small object detection. *IEEE Transactions on Image Processing*, 35:1127–1142, 2026a. doi: 10.1109/TIP.2026.3654892.

Xiang Yuan, Junwei Han, and Gong Cheng. Breathing new life into small object detection with detection-oriented rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2026b. doi: 10.1109/TPAMI.2026.3704810.

H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2023.

Huaxiang Zhang, Hao Zhang, Kai Liu, Zhongxue Gan, and Guo-Niu Zhu. UAV-DETR: Efficient end-to-end object detection for unmanned aerial vehicle imagery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 15143–15149, 2025. doi: 10.1109/IROS60139.2025.11246176.

Yin Zhang, Mu Ye, Guiyi Zhu, Yong Liu, Pengyu Guo, and Junhua Yan. FFCA-YOLO for small object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. doi: 10.1109/TGRS.2024.3363057.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2024. doi: 10.1109/CVPR52733.2024.01605.

PeiHuang Zheng, Yunlong Zhao, Zheng Cui, and Yang Li. PRNet: Original information is all you have, 2025.

Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2110–2118, 2016. doi: 10.1109/CVPR.2016.232.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710, 2018. doi: 10.1109/CVPR.2018.00907.

A IMPLEMENTATION DETAILS

This section records everything needed to reproduce the runs: the data, the training recipe, and the two transformed architectures in full.

A.1 DATASETS AND SPLITS

VisDrone provides 10,209 drone-captured images annotated with over 542,000 instances across ten traffic categories; we follow the official split of 6,471 training, 548 validation, and 1,610 test images. AI-TOD comprises 28,036 aerial images with roughly 700,000 instances, split 11,214 / 2,804 / 14,018; its mean object extent is 12.8 pixels. TT100K is built from 100,000 street-view panoramas with about 30,000 traffic-sign annotations; following common practice we retain the 45 categories with more than 100 instances each, giving 6,105 training and 3,069 test images. TinyPerson, SODA-D, and CODrone are evaluated under a full-image protocol with no tiling; CODrone's oriented annotations are converted to horizontal enclosing boxes for HBB evaluation.

The shares in Figure 1 are measured in each benchmark's own annotation format, so the area definition is not uniform across the ten bars. For CODrone we compute it ourselves from the annotated quadrilaterals under a full-image stretch to 640, over the 443,583 instances that remain after removing the ignored and zero-area ones, which gives 93.4%; the horizontal enclosing boxes used for the evaluation above give 90.1% over the same instances, and letterboxing instead of stretching would raise the share further. The bar is therefore an estimate of the same quantity under a neighbouring convention, not a value carried over from the CODrone paper.

A.2 TRAINING CONFIGURATIONS

For each detector family the transformed variant follows the training recipe of its baseline unless stated otherwise. The only deliberate changes are the input resolution and the architecture induced by the transformation.

Table 6: Training configuration. Everything not listed follows the official implementation of the corresponding baseline. Note the asymmetry in mixed precision: it is disabled on the YOLO side and enabled on the DETR side, in both cases following the baseline.

	R-YOLO series	R-D-FINE series
Framework	Ultralytics	D-FINE
Optimiser	SGD	AdamW
Initial LR	0.01	4×10^{-4} (VisDrone), 10^{-4} (others)
LR schedule	linear decay, final factor 0.01	—
Head / backbone LR	—	1×10^{-4} / 2×10^{-4}
Momentum	0.937	—
Weight decay	5×10^{-4}	1×10^{-4}
Warmup	3 epochs	500 iterations
Gradient clip	—	0.1
EMA	—	decay 0.9999
Loss weights	box / cls / DFL $= 7.5/0.5/1.5$	VFL, box, GIoU, FGL, DDF as in D-FINE
Mixed precision	disabled	enabled
Batch size	8	4
Epochs	300 / 600 / 600	500 / 220 / 220
Initialisation	pretrained	random (VisDrone), pretrained (others)
Input	960 / 960 / 960	960 / 960 / 960

Epoch, initialisation, and input rows are given for VisDrone / AI-TOD / TT100K in that order. TinyPerson, SODA-D, and CODrone reuse the VisDrone configuration unchanged. Data augmentation on the YOLO side is Mosaic with probability 1.0, horizontal flip with probability 0.5, translation 0.1, scale 0.5, and HSV jitter $(h, s, v) = (0.015, 0.7, 0.4)$, with Mosaic switched off for the last ten epochs. For the edge profiling of Table 4, models are exported to ONNX and compiled into TensorRT engines at FP16.

A.3 Full architecture specification

Table 7 gives the level-by-level specification, with a retention row underneath: the ratio of each transformed width to the corresponding YOLOv11-L width. That row is the compact statement of what the transformation does. Retention falls with depth, 0.75 at P_2 against 0.125 at P_4, and the network keeps 3.9% of the parameters while running at $1.5\times$ the input scale.

Table 7: Architecture specification against YOLOv11-L. The transformed network forms an inverted capacity pyramid in retention ratio, where shallow levels keep a larger fraction of their original capacity than deep ones.

	Backbone channels					Neck		Topology		Cost	
Model	P_1	P_2	P_3	P_4	P_5	N-P_3	N-P_2	Heads	Input	Params	FLOPs
YOLOv11-L	64	256	512	512	512	256	128	P_3–P_5	640	25.29 M	86.6 G
Ours	32	192	256	64	–	128	64	P_2 only	960	0.99 M	46.6 G
Retention	0.50	**0.75**	**0.50**	**0.125**	–	0.50	0.50	–	1.50	0.039	0.538

A.4 Structural configuration on the DETR side

The transformation applies to D-FINE through the same three steps. The backbone loses its deepest stage and the pyramid shifts one octave shallower; the decoder follows.

One setting departs from the official release, and it applies to both sides of the comparison. D-FINE decodes 300 object queries by default, a budget chosen on COCO-scale data that averages fewer than ten instances per image. VisDrone averages above fifty by the counts in Appendix A.1, and its crowded scenes run far higher, so that budget risks capping how many boxes the decoder can emit before detection quality is tested at all. We therefore raise the query budget to 600 for the baseline and for the transformed model alike. Every D-FINE number reported in this paper is measured under that setting and is consequently higher in FLOPs than the published 300-query figure; the two are not interchangeable.

Table 8: Structural configuration of R-D-FINE against D-FINE-L.

	Component	D-FINE-L	R-D-FINE
Backbone	HGNetv2 variant	B4	Pruned B2
	Stage-1 channels	128	128
	Stage-2 channels	512	256
	Stage-3 channels	1024	256
	Stage-4 channels	2048	removed
	Output stride	$[8, 16, 32]$	$[4, 8, 16]$
Encoder–decoder	Encoder hidden	256	128
	Decoder channels	$[256, 256, 256]$	$[128, 128]$
	Decoder stride	$[8, 16, 32]$	$[4, 8]$
	Feature levels	3	2
	Decoder layers	6	4
	Object queries	600	600
	Deformable points	$[3, 6, 3]$	$[2, 3]$

B Head-wise measurements

The three parts below support prediction one and the COCO control of Section 4: what each head contributes, which size bands it serves, and what the top-down pathway carries back into the shallow levels.

B.1 HEAD-WISE ABLATION ON VISDRONE

Section 4 reports the outcome of the two structural protocols in prose. The full table is given here, with the compute released alongside the accuracy change. The reference is the four-head model. Single-head retention also removes the part of the bottom-up pathway associated with the discarded heads. The last row is Step 1 of the transformation.

Table 9: Head-wise ablation on VisDrone with YOLOv11-L, measured against the four-head reference. Δ values are relative to that reference.

Configuration	AP_{50}	FLOPs (G)	ΔAP_{50}	ΔFLOPs (G)
Leave-one-out: delete one head, retrain, keep the rest				
All heads (P_2–P_5)	37.7	109.4	–	–
$-H_2$	30.3	100.7	-7.4	-8.7
$-H_3$	37.9	106.3	$+0.2$	-3.1
$-H_4$	38.1	108.0	$+0.4$	-1.4
$-H_5$	38.0	105.2	$+0.3$	-4.2
Single-head retention: keep one head, delete the rest				
H_5 only	16.1	98.5	-21.6	-10.9
H_4 only	28.5	97.7	-9.2	-11.7
H_3 only	35.2	88.3	-2.5	-21.1
H_2 only	37.8	80.8	$+0.1$	-28.6
H_2 only, $-$Stage 5 (Step 1)	37.4	74.4	-0.3	-35.0

B.2 PER-BAND RECALL UNDER HEAD CLOSING

Table 1 measures what a head is worth in AP; Table 10 asks which objects that value comes from. The protocol is the close protocol of Section 4: weights and structure are preserved and the head's candidates are dropped before NMS, so a row records what a head currently contributes and not what the dataset requires. Size here is $\sqrt{wh}$ after letterboxing, a finer convention than the `pycocotools` original-image areas behind AP_s, and the two are not compared. The two datasets separate cleanly: on VisDrone the H_4 and H_5 rows are zero in every band, while on COCO those same two heads own the two largest bands.

Table 10: Change in recall at IoU 0.5 for every instance-size band when one head is closed, against the unclosed model, in points. Instance size is $\sqrt{wh}$ after letterboxing. VisDrone is the validation split (38,759 instances, baseline 30.39 AP); COCO is `val2017` (36,781 instances, baseline 49.93 AP). Largest drop of each row in bold. Closing H_4 or H_5 on VisDrone leaves every band identical to the unclosed model.

		< 8	8–16	16–32	32–64	64–96	96–128	≥ 128
Instances per band								
	VisDrone	12,003	15,106	8,718	2,672	225	21	14
	COCO	1,008	3,282	6,744	8,146	4,721	3,050	9,830
H_2	VisDrone	-18.8	-5.6	$\mathbf{-63.9}$	-25.7	$+3.1$	0.0	0.0
	COCO	-30.1	$\mathbf{-37.9}$	-20.4	-1.1	$+0.7$	$+0.6$	$+0.3$
H_3	VisDrone	-0.7	$+0.1$	$+0.2$	-41.1	$\mathbf{-94.2}$	-85.7	-92.9
	COCO	-0.1	$+0.9$	-0.1	$\mathbf{-34.9}$	-29.1	-1.2	$+0.1$
H_4	VisDrone	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	COCO	$+0.1$	$+0.3$	$+0.3$	$+0.1$	-13.3	$\mathbf{-53.8}$	-25.0
H_5	VisDrone	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	COCO	$+0.1$	$+0.2$	$+0.1$	$+0.1$	$+0.1$	-0.1	$\mathbf{-47.4}$

B.3 FEATURE-LEVEL EVIDENCE FOR THE TOP-DOWN BACKFLOW

Section 3 predicts more than idle compute at the deep levels: what those levels respond to is carried back down the top-down pathway into the one level that still detects. Figure 4 follows one dense

VisDrone scene through that path in four panels: the input with ground truth, the backbone's own P_2, what P_4 and P_5 send down to P_2, and the fused P_2 the head actually reads. The evidence is qualitative and is offered for the mechanism rather than as a measurement of it; its quantitative counterpart is the leave-one-out block of Table 9, where deleting H_2 costs 7.4 points while deleting any deep head costs nothing at all.

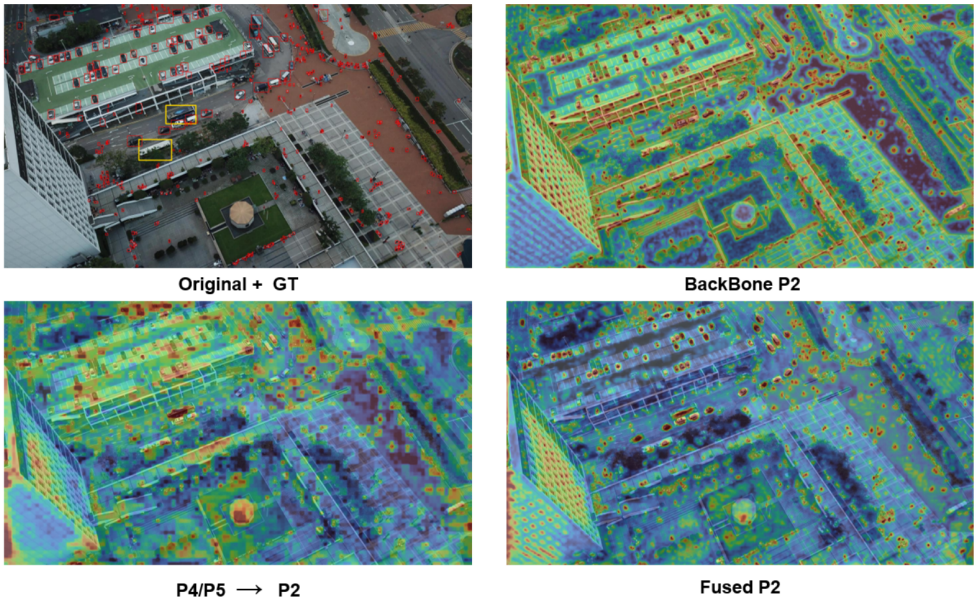


Figure 4: Backbone P_2 keeps fine-grained responses on the dense small targets. What P_4 and P_5 propagate down to P_2 activates broadly over background instead, and the fused P_2 carries that background energy into the shallow detection path. This is the observation behind the second prediction of Section 3: deep channels compute over space no target supports, and the cost is not merely the wasted compute.

C Capacity return

The complete measurements behind predictions two and three, first as a table of operating points and then in graphical form.

C.1 Full sweep of operating points

The deep-channel retention ratios and input resolution used in Section 4.2 are selected from the axis-wise sweeps reported here. Step 2 compresses one channel group at a time from the post-Step-1 reference; Step 3 reinvests from the post-Step-2 reference. Budget is the compute released or consumed relative to the corresponding reference, and CR is Definition 1 applied to that interval. The selected points are 50% for P_3, 12.5% for P_4, 50% for Neck-P_3, and an input of 960.

C.2 Channel sensitivity and the two reinvestment trajectories

Figure 5 redraws the sweep, which makes two things visible that the table cannot show. Panels (a)–(c) show that the free-compression region on P_4 is an interval and not an isolated point, so the 12.5% retention selected in Section 4.2 is not a knife edge. Panel (e) plots the resolution and channel trajectories against compute and labels their adjacent-point returns. The resolution sweep shows diminishing returns as input size increases; we select 960 as an empirical operating point.

D Additional results

Results that are referenced from the main text but not required by the argument.

Table 11: Capacity return on VisDrone. $\Delta\mathrm{AP}_{50}$ and ΔFLOPs denote modified minus block-reference values. $\mathrm{CR} = \Delta\mathrm{AP}_{50}/\Delta\mathrm{FLOPs}$ (pp/G) is calculated from the displayed values. Negative CR under compression indicates improved accuracy. These returns are relative to each block's reference; Figure 5(e) shows adjacent-point returns.

Target	Setting	FLOPs (G)	AP	AP_{50}	AP_s	$\Delta\mathrm{AP}_{50}$	CR
Step 2: channel compression, reference 74.4 *G at* 37.4 AP_{50}							
P_4	75%	63.8	21.6	37.8	11.9	+0.4	−0.038
	50%	59.4	21.5	38.0	11.9	+0.6	−0.040
	25%	54.5	21.6	37.9	12.0	+0.5	−0.025
	12.5%	52.7	21.6	37.8	12.0	+0.4	−0.018
	1.5%	51.5	20.8	36.6	11.6	−0.8	+0.035
P_3	75%	66.7	21.0	37.2	11.6	−0.2	+0.026
	50%	60.0	20.8	36.7	11.3	−0.7	+0.049
	25%	54.3	20.5	36.0	11.2	−1.4	+0.070
	12.5%	51.9	20.3	36.0	11.2	−1.4	+0.062
	1.5%	50.0	19.5	34.7	10.9	−2.7	+0.111
Neck-P_3	50%	67.8	21.1	37.3	11.6	−0.1	+0.015
	25%	65.5	20.7	36.7	11.4	−0.7	+0.079
P_2	75%	68.1	20.3	35.9	11.0	−1.5	+0.238
	50%	61.9	20.1	35.7	10.9	−1.7	+0.136
	25%	57.1	19.6	35.2	10.7	−2.2	+0.127
Neck-P_2	75%	69.4	20.9	36.7	11.2	−0.7	+0.140
	50%	65.4	20.7	36.5	11.1	−0.9	+0.100
	25%	62.1	19.9	35.4	10.8	−2.0	+0.163
P_3+P_4	50/12.5%	41.6	21.2	37.4	11.7	0.0	0.000
P_3+N-P_3	50/50%	53.8	20.6	36.5	11.5	−0.9	+0.044
P_4+N-P_3	12.5/50%	46.8	21.5	37.6	11.9	+0.2	−0.007
P_3+P_4+N-P_3	50/12.5/50%	36.2	20.9	36.8	11.6	−0.6	+0.016
Step 3: reinvestment, reference 36.2 *G at* 36.8 AP_{50}							
Resolution	800	57.3	23.8	41.3	14.4	+4.5	+0.213
	960	82.5	25.5	43.8	16.6	+7.0	+0.151
	1120	112.2	25.9	44.7	17.7	+7.9	+0.104
	1280	146.6	26.3	45.3	18.1	+8.5	+0.077
Channel	1.5×	66.3	22.8	39.7	13.0	+2.9	+0.096
	2.0×	106.7	23.3	40.7	13.6	+3.9	+0.055
	2.5×	140.9	23.8	41.3	14.1	+4.5	+0.043

D.1 Scale-stratified gain from the resolution step

Section 5 states that the gain of the third step falls with target scale. The stratification is given here. Sizes follow the `pycocotools` original-area convention, as elsewhere for AP.

Table 12: Scale-stratified gain from 640 to 960 on VisDrone. The relative gain decreases monotonically with object scale, which is what the coverage account requires.

Scale	AP@640	AP@960	ΔAP (pp)	Relative gain
Small ($< 32^2$)	11.6	15.2	+3.6	+31.0%
Medium	31.4	35.3	+3.9	+12.4%
Large	37.4	40.3	+2.9	+7.8%

D.2 Cross-platform inference speed

The throughput gain is not tied to one GPU. All measurements use batch size 1 in native PyTorch, the same protocol as Table 2.

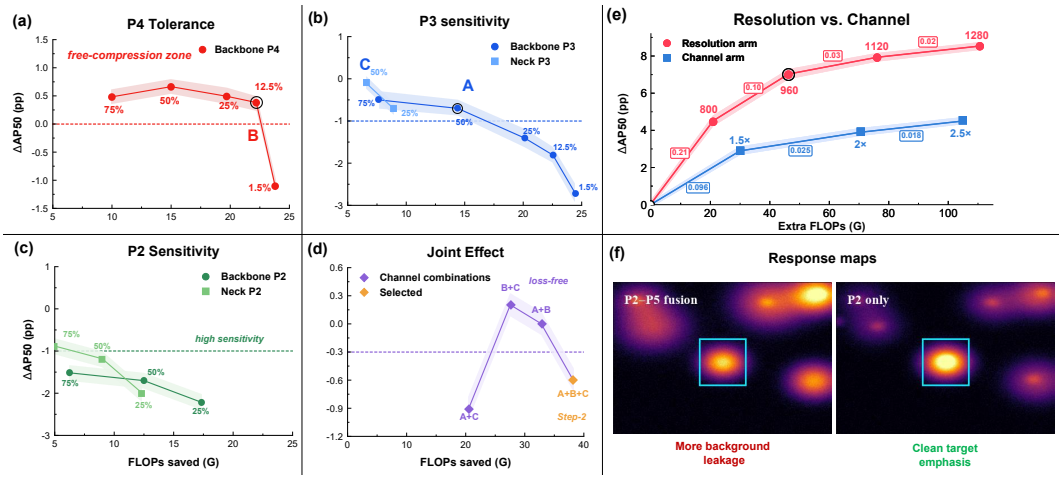


Figure 5: The graphical form of Table 11. (a)–(c) ΔAP_{50} against compute saved, level by level: P_4 stays non-negative from 75% down to 12.5% retention and only turns costly at 1.5%, P_3 and Neck-P_3 are moderately sensitive, and P_2 and Neck-P_2 lose accuracy at any compression. Markers A, B and C are the three selected settings, 50% of P_3, 12.5% of P_4 and 50% of Neck-P_3. (d) Their combinations, which are approximately additive. (e) Resolution against channel widening from the Step-2 reference, with the adjacent-point return labelled on each interval. (f) Response maps of the same target with and without the deep levels in the fusion.

Table 13: Latency and throughput on three desktop GPUs, reported as ms / FPS. Parenthesised values are the speedup over the baseline on that device.

Model	Input	AP_{50}	Params (M)	RTX 5070 Ti	RTX 4090	RTX 3090
YOLOv11-L	640	35.1	25.29	16.7 / 59.9	15.8 / 63.3	16.8 / 59.5
R-YOLOv11-L	960	41.5	0.99	11.3 / 88.5	8.9 / 112.4	11.3 / 88.5
throughput gain				**1.48**×	**1.78**×	**1.49**×

D.3 General-purpose detectors on VisDrone

These six rows — Cascade R-CNN (Cai & Vasconcelos, 2018), GFL (Li et al., 2020), ATSS-DyHead (Dai et al., 2021), TOOD (Feng et al., 2021), RTMDet (Lyu et al., 2022) and CEASC-GFL (Du et al., 2023) — are referenced by Table 2 and held out of it because their input protocol differs. Five of them run at 1333×750 rather than a square input.

Table 14: General-purpose detectors on the VisDrone test set. Entries marked "–" are not reported in the original papers.

Model	Input	AP	AP_{50}	AP_s	Params (M)	FLOPs (G)
Cascade R-CNN [CVPR'18]	1333×750	19.7	32.6	9.9	69.3	236.0
GFL [NeurIPS'20]	1333×750	19.3	32.1	9.4	32.3	206.0
ATSS-DyHead [CVPR'21]	1333×750	20.4	33.8	10.0	38.9	110.0
TOOD [ICCV'21]	1333×750	20.4	33.9	10.2	32.0	199.0
RTMDet-L [arXiv'22]	640	19.9	32.9	9.8	52.3	80.2
CEASC-GFL [CVPR'23]	1333×750	20.0	36.2	9.7	–	150.2

D.4 Full comparison on AI-TOD and TT100K

Table 3 keeps one baseline per family so that the paired comparison stays legible. The two tables here place the same R- models against every published method we could compare on, including tiny-object specialists that report no compute figures. Nothing is re-tuned: these are the transferred configurations of Section 4.2, evaluated at 960 as everywhere else.

Table 15: Full comparison on AI-TOD. Annotations on R- rows are measured against the corresponding 640 baseline. Entries marked "–" are not reported in the original papers.

Model	AP	AP_{50}	AP_s	Params (M)	FLOPs (G)
YOLO-based detectors					
YOLOv8-L	26.6	53.7	25.4	43.6	164.9
YOLOv11-L	26.3	54.1	24.7	25.29	86.6
YOLO26-L	25.6	53.6	24.6	24.75	86.1
R-YOLOv11-L	26.5 ↑ 0.2	56.1 ↑ 2.0	25.9	**0.99**	**46.6**
DETR-based detectors					
RT-DETRv2-S	23.8	51.3	23.5	20.0	60.0
RT-DETRv2-M	26.1	55.0	25.2	42.0	136.0
CAEM-DETR	25.6	56.7	24.9	19.9	93.0
FM-RTDETR-S	26.9	56.3	27.1	22.0	144.0
DINO	23.2	56.6	28.3	47.6	274.0
D-FINE-L	27.8	55.6	27.3	30.7	96.1
R-D-FINE-L	**29.1** ↑ 1.3	**58.3** ↑ 2.7	28.4	3.95	84.1
Tiny-object specialists					
RFLA	21.7	50.5	24.5	–	–
HANet	22.1	53.7	27.3	–	–
RFLA w/ SET	24.5	54.5	**31.0**	–	–
DNTR	26.2	56.7	**31.0**	46.5	167.0

Table 16: Full comparison on TT100K. Annotations on R- rows are measured against the corresponding 640 baseline.

Model	AP	AP_{50}	AP_s	Params (M)	FLOPs (G)
YOLO-based detectors					
YOLOv8-L	67.5	85.9	51.2	43.6	165.4
YOLOv11-L	69.0	87.2	50.2	25.29	86.6
YOLO26-L	68.3	87.0	56.3	24.75	86.1
YOLO-TS	72.3	91.2	58.9	11.1	99.1
R-YOLOv11-L	74.4 ↑ 5.4	93.4 ↑ 6.2	**61.6**	0.99	46.6
R-YOLO26-L	**74.6** ↑ 6.3	**93.6** ↑ 6.6	**61.6**	**0.98**	44.3
DETR-based detectors					
RT-DETRv2-S	64.3	86.2	44.2	20.0	60.0
D-FINE-L	67.5	88.6	50.5	30.7	96.1
DEIM-D-FINE-L	67.0	89.3	49.7	30.7	90.7
R-D-FINE-L	74.1 ↑ 6.6	92.7 ↑ 4.1	60.5	3.95	84.1
Traffic-sign and small-object specialists					
VATSD	63.1	82.0	46.4	7.9	**16.6**
Unc-SOD	64.2	84.1	41.9	41.5	298.5

E QUALITATIVE COMPARISON

Figure 6 shows the transferred configuration on the three datasets it was never calibrated on. The panels are illustrative rather than representative, chosen to span the three domains, and no claim in this paper rests on them. What they show is consistent with the coverage account of Section 3: the detections the transformed model recovers are concentrated in the crowded regions and at the small end of each scene.

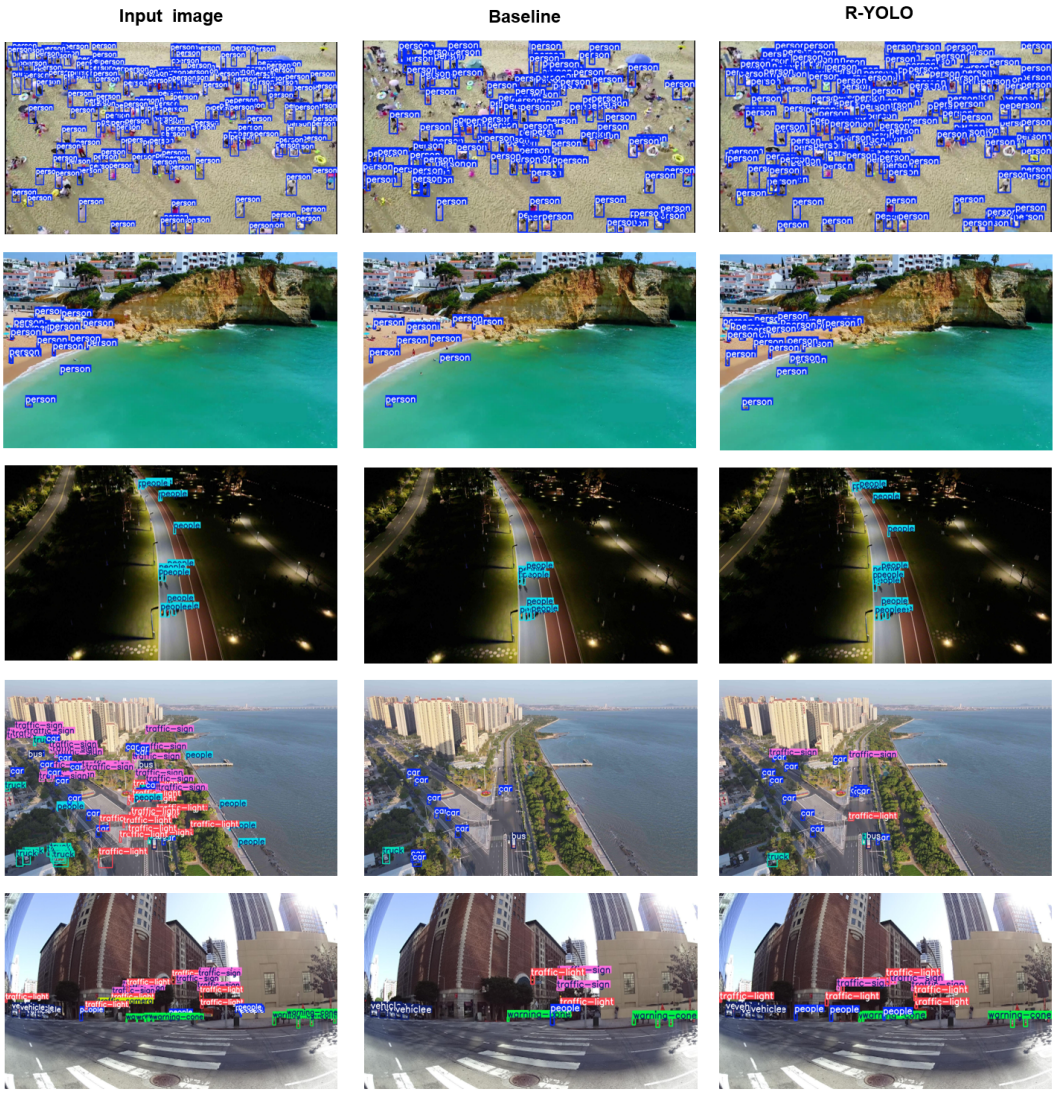


Figure 6: Cross-dataset detection under the fixed transfer rule, with no dataset-specific retuning. Rows 1–2 are TinyPerson, rows 3–4 are CODrone, and row 5 is SODA-D. Columns, left to right: the input with its annotations, the YOLOv11-L baseline, and the transformed model.