

A Lagrangian View of Trading Pixels for Parameters

Kabir Murjani
Nirma University
Ahmedabad, India
23bee064@nirmauni.ac.in

Aum Joshi
Nirma University
Ahmedabad, India
23bee015@nirmauni.ac.in

Kuntal Bhattacharjee
Nirma University
Ahmedabad, India
kuntal.bhattacharjee@nirmauni.ac.in

Abstract

Computational constraints couple the selection of image preprocessing and object detectors. This study formulates their joint selection as a resource allocation problem in which detection accuracy is maximized subject to an inference budget. A Lagrangian relaxation characterizes the marginal returns associated with preprocessing and detector computation, including the conditions governing interior and boundary allocations. The formulation also defines a substitution hypothesis under which the accuracy increment from a fixed transformation decreases with detector capacity. An underwater detection study examines this increment using classical image compensation and detectors from the You Only Look Once (YOLO) family, trained separately on raw and compensated images. Compensation improves validation mean average precision across the evaluated configurations, with decreasing gains under several checkpoint and training trajectory summaries. The accompanying precision and recall changes vary across configurations. The analysis identifies the measurements required to assess preprocessing within a constrained deployment decision.

CCS Concepts

• **Computing methodologies** → Neural networks; Control methods; Object detection; • **Theory of computation** → Mathematical optimization.

Keywords

constrained optimization, Lagrangian analysis, resource allocation, image preprocessing, object detection

ACM Reference Format:

Kabir Murjani, Aum Joshi, and Kuntal Bhattacharjee. 2026. A Lagrangian View of Trading Pixels for Parameters. In *Proceedings of 17th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP2026)*. ACM, New York, NY, USA, 17 pages.

1 Introduction

Allocating computation between image preprocessing and object detection is a joint design problem. Each component incurs an inference cost, and the accuracy obtained from either component depends on the other. Increasing detector size or input resolution can improve recognition, but consumes resources that might otherwise support an input transformation. Model scaling studies demonstrate the importance of coordinating computational

choices across network depth, width and resolution [1]. When preprocessing is included, the corresponding design objective is detection accuracy under a budget on the complete inference pipeline.

Underwater perception provides a setting in which this allocation is particularly relevant. Wavelength-dependent attenuation and backscatter alter color, contrast and the visibility of object boundaries [2, 3]. Color correction, local contrast adjustment and filtering can modify these features before detection. Their computational requirements are additional to those of the detector, while their effects depend on the information preserved in the recorded image. A useful preprocessing decision must therefore account for both its detection benefit and the resources required to obtain that benefit.

Prior studies establish that improved visual appearance is insufficient to predict improved recognition. Sun et al. [4] show that conventional restoration can provide limited or adverse benefits for detection and investigate restoration training guided by the downstream objective. In underwater detection, Wang et al. [5] evaluate several enhancement methods across multiple detectors and report performance losses associated with changes in image content and background interference. These findings motivate evaluation of the transformation together with the detector trained on its outputs. They also indicate that preprocessing value is conditional on the task, data and detection procedure.

From an optimization perspective, this dependence makes the accuracy increment from preprocessing a property of the joint allocation. A positive increment must be considered alongside its computational cost and the accuracy available from competing detector configurations. Classical resource allocation theory expresses this comparison through a budget constraint and its Lagrange multiplier [6, 7]. For a differentiable relaxation, an interior optimum equalizes marginal returns per unit expenditure. Boundary allocations and discrete architecture choices require corresponding inequality conditions or finite comparisons, which are developed in the supplementary analysis.

The paired representation in Figure 1 places a static feedforward compensator before a detector trained for the corresponding input transformation [8]. Detectors are trained separately on raw and transformed inputs, so the measured increment includes adaptation to each input representation. A substitution hypothesis specifies that this increment decreases as detector capacity increases within a controlled family. The hypothesis requires an assumption about the interaction between preprocessing and capacity, formalized through a decreasing-differences condition. This separates the mathematical conditions for substitution from an ordering observed across particular trained configurations.

The empirical study examines a fixed classical compensator with nano variants of the You Only Look Once (YOLO) family [9].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICVGIP2026, Kolkata, India

© 2026 Copyright held by the owner/author(s).

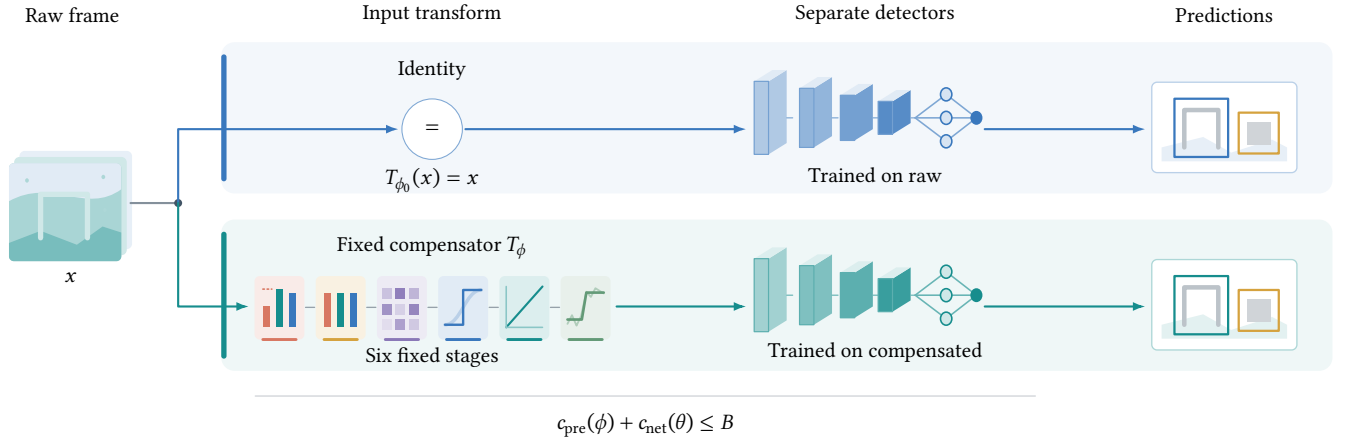


Figure 1: The allocation framework compares separately trained baseline and compensated detectors while accounting for both preprocessing and detector cost. The six colored tiles represent the stages of Algorithm 1. Image and network symbols are schematic.

Each baseline and compensated pair shares images, annotations, split assignments and training settings. Validation gains are positive across the evaluated configurations and decrease in release order under several checkpoint and trajectory summaries. Architecture and training differences restrict this ordering to a configuration comparison, while the single-seed design and validation reuse limit its statistical interpretation. The study contributes a joint allocation formulation, an explicit account of the assumptions governing substitution, and a paired analysis of compensation gains. The available results characterize validation accuracy, with independent test performance and complete inference cost remaining outside the measured scope.

2 Related Work

The formulation draws on resource allocation theory and empirical studies of image enhancement for recognition. The relevant literature concerns marginal returns, model scaling and the dependence of detection accuracy on preprocessing.

2.1 Resource allocation

Classical allocation theory relates constrained optima to marginal returns per unit cost [6, 7]. For discrete decisions, diminishing returns can be expressed through submodular set functions [10]. The present formulation uses continuous and discrete allocation models to specify the accuracy and cost comparisons required for preprocessing selection. Establishing submodularity would require additional assumptions on the joint objective.

Scaling studies examine performance as model and data sizes change [11, 12]. Architecture studies characterize accuracy–computation relationships [1, 13, 14]. Release order alone supplies no quantitative capacity measure. A controlled capacity comparison therefore requires a common training procedure and explicit measurements of parameter counts, floating-point operations and deployment latency.

2.2 Enhancement for recognition

Pei et al. [15] investigate haze removal before classification. Sun et al. [4] examine the limitations of restoration objectives for detection and introduce restoration training directed toward the downstream task. These studies support evaluating preprocessing through detection accuracy, since perceptual quality and task performance can respond differently to the same transformation.

Wang et al. [5] compare eighteen underwater enhancement methods with seven detectors, covering both classical and learned enhancement. Their experiments establish prior evaluation of enhancement–detector interactions. The present study considers one fixed transformation across detector configurations and interprets the resulting differences through a resource allocation model.

2.3 Underwater image processing

Underwater images exhibit wavelength-dependent attenuation and backscatter [2, 3]. Classical correction methods include gray-world and shades-of-gray approaches [16, 17], color balance and fusion [18, 19], and restoration using transmission estimates or channel priors [20–22]. Adaptive histogram equalization, contrast limitation and bilateral filtering provide established operators for local contrast adjustment and smoothing [23–25].

Learned enhancement methods include convolutional and adversarial models [26–29]. A classical pipeline permits fixed preprocessing settings across the detector configurations considered here. Comparisons with learned alternatives require evaluation of both detection accuracy and complete inference cost.

3 Problem Formulation

The allocation model treats preprocessing and detection as coupled uses of a computational budget. The image formation model motivates the transformation, while the objective measures its contribution through the performance of a detector trained on the transformed inputs.

3.1 Image formation

Let x_0 denote the radiance that would reach the sensor in clear water and x the recorded frame. For red, green and blue (RGB) channels indexed by $k \in \{R, G, B\}$, a simplified formation model is

$$x_k(u) = x_{0,k}(u)e^{-\beta_k z(u)} + B_k^\infty(1 - e^{-\beta_k z(u)}), \quad (1)$$

where u indexes pixels, $z(u)$ is distance along the viewing ray, β_k is an attenuation coefficient and B_k^∞ is asymptotic backscatter [19–21]. Attenuation and the distance-dependent offset motivate color and contrast correction. Recovery of the original radiance remains limited by noise, quantization and the identifiability of the formation parameters.

3.2 Constrained allocation

Let $T_\phi : \mathcal{X} \rightarrow \mathcal{X}$ be a static compensator on the image space $\mathcal{X}$ and f_θ a trained detector. For an evaluation set $\mathcal{S}$ of images x and annotations y , define

$$\begin{aligned} U_\mathcal{S}(\theta, \phi) &= \text{mAP}_{50-95}(\mathcal{P}_\mathcal{S}(\theta, \phi)), \\ \mathcal{P}_\mathcal{S}(\theta, \phi) &= \{(f_\theta(T_\phi(x)), y) : (x, y) \in \mathcal{S}\}. \end{aligned} \quad (2)$$

Mean average precision (mAP) averages classwise average precision (AP). Here mAP_{50-95} also averages over intersection-over-union (IoU) thresholds from 0.50 to 0.95 in increments of 0.05. IoU is the intersection area divided by the union area of predicted and reference boxes. AP is computed from ranked detections across the evaluation set.

Let $U(\theta, \phi) = \mathbb{E}_\mathcal{S}[U_\mathcal{S}(\theta, \phi)]$ for evaluation sets drawn under a fixed sampling protocol from the target distribution. Let $c_{\text{net}}(\theta)$ and $c_{\text{pre}}(\phi)$ denote detector and preprocessing costs in common units. For serial execution under budget B , the allocation problem is

$$\begin{aligned} \max_{\phi \in \Phi, \theta \in \Theta(\phi)} \quad & U(\theta, \phi) \\ \text{subject to} \quad & c_{\text{net}}(\theta) + c_{\text{pre}}(\phi) \leq B. \end{aligned} \quad (3)$$

The set Φ contains candidate transformations, and $\Theta(\phi)$ contains detectors trained for transformation ϕ . The allocation is made over trained systems. If cost represents latency, the measurement boundary must include input preparation, device transfers and postprocessing. End-to-end timing is required to assess the additive approximation.

For a continuous relaxation, let p and n denote preprocessing and detector expenditure, and let $F(p, n)$ be a differentiable utility model. The Lagrangian associated with this budget constraint is

$$\mathcal{L}(p, n, \lambda) = F(p, n) - \lambda(p + n - B), \quad \lambda \geq 0. \quad (4)$$

At an interior allocation with an active budget, stationarity yields [6, 7]

$$\frac{\partial F}{\partial p} = \frac{\partial F}{\partial n} = \lambda. \quad (5)$$

Appendix A derives the corresponding boundary conditions and sufficiency under concavity. Appendix A.2 treats discrete alternatives. Empirical average precision requires separate treatment. The experiments below measure accuracy differences without estimating F , its derivatives or an optimal budget allocation.

3.3 Marginal value

Let T_{ϕ_0} be the identity transformation with zero additional preprocessing cost. For a controlled detector family indexed by capacity κ , define

$$\begin{aligned} V(\kappa) &= U(\hat{\theta}(\kappa, \phi), \phi) \\ &\quad - U(\hat{\theta}(\kappa, \phi_0), \phi_0), \end{aligned} \quad (6)$$

where $\hat{\theta}$ is the output of a specified training and checkpoint-selection procedure. Both detectors are trained on their respective input representations. Across capacities, the training protocol is held fixed, with repeated runs required to quantify training variability.

The substitution hypothesis imposes decreasing compensation gains,

$$\kappa' > \kappa \implies V(\kappa') \leq V(\kappa). \quad (7)$$

Appendix B gives a sufficient condition for this ordering in a smooth model. An empirical test requires a controlled capacity axis. Compensation can alter useful information and interact with architecture or training, so the inequality is an additional hypothesis. A deployment preference further depends on the accuracy and costs of the competing systems.

In the case study, $g \in \{1, 2, 3\}$ indexes the YOLOv5n, YOLOv8n and YOLO11n configurations, including their training procedures. The quantity $\Delta_{\text{val}}(g)$ denotes the observed compensated-minus-baseline validation difference. Release order determines g , while capacity remains unmeasured.

3.4 Fixed compensation

The settings ϕ are selected before detector training and held constant across images and configurations. The compensator receives no detector feedback, consistent with a static feedforward configuration [8]. This restriction defines the empirical comparison and permits the same transformation to be evaluated with successive detector configurations. Jointly trained or adaptive transformations constitute additional alternatives in Equation (3) and require their own accuracy and cost measurements.

4 Compensation Method

The transformation consists of six classical operations applied offline to every image. The first five operate on individual color channels, and the last filters the merged image. Algorithm 1 specifies the reported procedure. The parameter selection is described in Section 4.2.

4.1 Operators

Red channel compensation adds the difference between the global green and red means,

$$x'_R = \text{clip}(x_R + (\overline{x_G} - \overline{x_R}), 0, 255). \quad (8)$$

The overbar denotes the spatial mean. This global correction differs from the local green-dependent compensation used in fusion-based enhancement [19].

Algorithm 1 The six-stage compensator T_ϕ

Require: frame of height H and width W , $x \in \{0, \dots, 255\}^{H \times W \times 3}$
Require: $\phi = (c_{\text{clip}}, G, \alpha, D, \sigma_c, \sigma_s)$
 $= (2.0, 10 \times 10, 0.3, 5, 75, 75)$
Ensure: compensated frame $\tilde{x}$

- 1: $(x_R, x_G, x_B) \leftarrow \text{Split}(x)$
- 2: $x_R \leftarrow \text{Clip}(x_R + (\overline{x_G} - \overline{x_R}), 0, 255)$
- 3: $\bar{\mu} \leftarrow \frac{1}{3}(\overline{x_R} + \overline{x_G} + \overline{x_B})$
- 4: **for all** $k \in \{R, G, B\}$ **do**
- 5: $x_k \leftarrow \text{Clip}(x_k \cdot \bar{\mu} / \overline{x_k}, 0, 255)$
- 6: **end for**
- 7: **for all** $k \in \{R, G, B\}$ **do**
- 8: $x_k \leftarrow \text{CLAHE}(x_k; c_{\text{clip}}, G)$
- 9: $x_k \leftarrow \text{Clip}(x_k - \alpha \nabla^2 x_k, 0, 255)$
- 10: **if** $\max(x_k) > \min(x_k)$ **then**
- 11: $x_k \leftarrow 255 \cdot \frac{x_k - \min(x_k)}{\max(x_k) - \min(x_k)}$
- 12: **end if**
- 13: **end for**
- 14: $\tilde{x} \leftarrow \text{Bilateral}(\text{Merge}(x_R, x_G, x_B); D, \sigma_c, \sigma_s)$
- 15: **return** $\tilde{x}$

Gray-world white balance rescales each channel toward the mean of the channel means [16],

$$x'_k = \text{clip}\left(x_k \frac{\bar{\mu}}{\overline{x_k}}, 0, 255\right), \quad \bar{\mu} = \frac{1}{3} \sum_k \overline{x_k}. \quad (9)$$

Clipping can prevent exact equality of the resulting means. The formula assumes a positive channel mean. The available implementation record does not specify the treatment of zero means or the numerical precision.

Contrast-limited adaptive histogram equalization (CLAHE) [23, 24] is applied independently to each channel, using clip limit $c_{\text{clip}} = 2.0$ and tile grid $G = 10 \times 10$. Laplacian sharpening then gives

$$x'_k = \text{clip}(x_k - \alpha \nabla^2 x_k, 0, 255), \quad \alpha = 0.3, \quad (10)$$

with the Laplacian evaluated in floating point before clipping. Contrast stretching rescales each nonconstant channel according to

$$x'_k = 255 \frac{x_k - \min(x_k)}{\max(x_k) - \min(x_k)}. \quad (11)$$

Constant channels are passed through unchanged. Bilateral filtering [25] is applied to the merged image with kernel diameter $D = 5$ and color and spatial standard deviations $\sigma_c = \sigma_s = 75$.

The complete operator is

$$T_\phi = S_{D, \sigma_c, \sigma_s} \circ \mathcal{C}[N \circ L_\alpha \circ E_{c_{\text{clip}}, G}] \circ W \circ C, \quad (12)$$

where C denotes red compensation, W gray-world balance, E local equalization, L sharpening, N stretching and S bilateral filtering. For a scalar-channel operator Q , $\mathcal{C}[Q]$ applies Q independently to the three channels.

Interactions between the stages affect the complete transformation. Gray-world balance rescales the red channel after mean compensation, while local equalization and stretching can alter chromatic relationships. Sharpening and filtering also affect common image boundaries. The experiment evaluates the complete composition. Attributing contributions to individual stages requires controlled ablations. Figure 2 illustrates the resulting appearance and

intensity distributions. Appendix H provides additional examples for tub and gate scenes.

4.2 Parameter selection

The parameter vector $\phi = (c_{\text{clip}}, G, \alpha, D, \sigma_c, \sigma_s)$ is fixed at $(2.0, 10 \times 10, 0.3, 5, 75, 75)$. Selection used visual and histogram inspection. The CLAHE grid specifies ten tiles along each axis. Increasing the number of tiles decreases their spatial extent. The sharpening weight was compared with 0.1 and 0.5, and the filter diameter with 9 and 15. Appendix E records the parameter comparisons used during selection. Detection-based parameter optimization and stage ablations were outside the experimental design.

5 Experimental Setup

The study compares raw and compensated inputs within each detector configuration. Shared image identities, annotations and split assignments define the paired design. Each comparison estimates the effect of the complete transformation under one training procedure, with validation accuracy as the observed outcome.

5.1 Dataset

The underwater collection was created and annotated using Roboflow. It contains eight classes relevant to underwater robotic tasks, comprising flare, gate, red tub, blue tub, red pole, yellow pole, blue pole and sea-floor objects. Several classes are distinguished by color, which makes the collection suitable for examining color and contrast transformations. Its application-specific composition limits generalization to other environments and object categories.

The collection contains approximately 800 images with reported training, validation and test proportions of 70%, 20% and 10%, corresponding to approximately 560, 160 and 80 images. Dataset versioning preserves the reported split assignments. Exact counts, split manifests and separation of related frames are not documented in the available dataset record. Figure 3 summarizes the paired experimental design.

All reported detection results use the validation split, which also determines checkpoint selection. The nominal test split has not been evaluated for these comparisons. Consequently, selection bias can affect both the reported accuracies and their differences.

5.2 Detector training

The evaluated models are YOLOv5n [30], YOLOv8n [31] and YOLO11n [32], where the suffix n denotes the nano variant. Initialization uses weights pretrained on Common Objects in Context (COCO) [33]. The models share a size designation, but their parameter counts, operation counts and latency have not been matched.

Each run uses 640×640 inputs, batch size 16, random seed 0 and at most 100 epochs. The selected checkpoint maximizes validation mAP_{50–95} within the run. Early-stopping patience is 20 epochs for YOLOv5n and YOLOv8n. The available YOLO11n descriptions report inconsistent patience values of 20 and 30, leaving the stopping setting uncertain. The paired trajectory horizons are 100, 83 and 100 epochs, respectively.

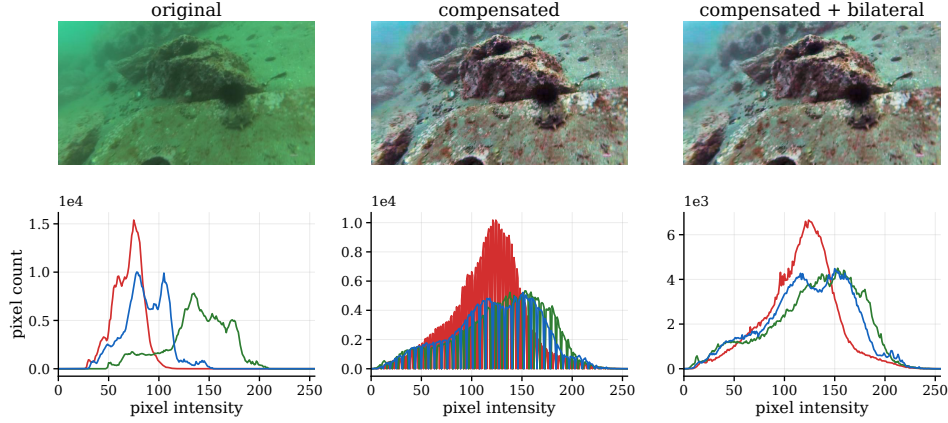


Figure 2: A sea-floor object frame before processing, after the five per-channel stages and after bilateral filtering, shown from left to right with RGB histograms below.

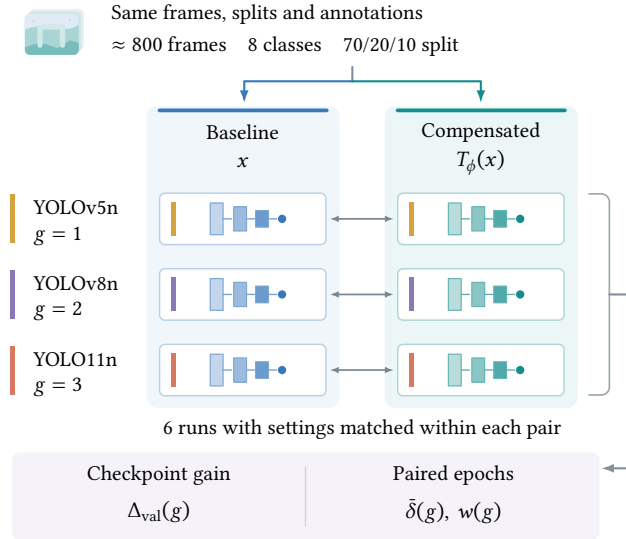


Figure 3: The measurement design compares raw and compensated versions of the same dataset across three detector and training configurations, with training settings matched within each pair.

Each baseline and compensated pair uses matched training settings. All detector configurations include mosaic augmentation during training. It is disabled during the final ten epochs for YOLOv8n and YOLOv11n. YOLOv5n uses its reported default settings and an initial learning rate of 0.01. YOLOv8n uses adaptive moment estimation with decoupled weight decay (AdamW) [34], an initial learning rate of 0.001 and cosine decay. YOLOv11n uses automatic optimizer selection, an initial learning rate of 0.01 and cosine decay. Appendix D gives the augmentation settings. Architecture, optimization and augmentation vary across configurations, preventing attribution of the observed ordering to capacity alone.

5.3 Computing environment

Training used NVIDIA T4 graphics processing units (GPUs) in Google Colab. The reported inference system contained eight gigabytes of fourth-generation double data rate (DDR4) random-access memory (RAM) and an NVIDIA GeForce GTX 1650 GPU with four gigabytes of sixth-generation graphics double data rate (GDDR6) memory. Unavailable software versions and incomplete implementation records restrict reproduction of the complete pipeline. The code will be made publicly available at <https://github.com/Kcbir/pixels> before submission.

5.4 Evaluation metrics

The primary comparison uses validation mAP_{50-95} at each run’s selected checkpoint. Additional metrics are mAP_{50} , defined at IoU threshold 0.50, and framework-reported precision and recall. The scalar operating confidence thresholds are undocumented. Appendix G includes precision–recall and confidence curves. The F1 score is the harmonic mean of precision and recall. Framework figures use *enhanced* for compensated inputs.

Three additional summaries characterize the validation histories. These are the difference between means over each run’s final ten epochs, the difference between means over each run’s five highest values, and the mean difference at matched epochs. These quantities describe distinct features of the recorded histories and remain subject to validation reuse and training variability [35, 36].

For compensated and baseline validation values m_t^{enh} and m_t^{base} , respectively, define

$$\begin{aligned}\bar{\delta} &= \frac{1}{T} \sum_{t=1}^T (m_t^{enh} - m_t^{base}), \\ w &= \frac{1}{T} \sum_{t=1}^T \mathbf{1}[m_t^{enh} > m_t^{base}].\end{aligned}\tag{13}$$

Here T is the smaller completed epoch count and $\mathbf{1}$ is the indicator function. The mean $\bar{\delta}$ depends on the training horizon, while w is the fraction of shared epochs with positive differences. The shared training history and validation images induce dependence across

Table 1: Baseline and compensated validation accuracy at each run’s best validation mAP_{50-95} checkpoint, with checkpoint selection and reporting using the same split.

Model	Data	mAP_{50}	mAP_{50-95}	P	R
YOLOv5n	baseline	0.541	0.361	0.878	0.455
YOLOv5n	compensated	0.810	0.582	0.710	0.802
$\Delta_{val}(1)$		+0.269	+0.221	-0.168	+0.347
YOLOv8n	baseline	0.691	0.471	0.647	0.707
YOLOv8n	compensated	0.869	0.616	0.839	0.861
$\Delta_{val}(2)$		+0.178	+0.145	+0.192	+0.154
YOLO11n	baseline	0.750	0.617	0.683	0.759
YOLO11n	compensated	0.839	0.646	0.813	0.837
$\Delta_{val}(3)$		+0.089	+0.029	+0.130	+0.078

Table 2: Compensated-minus-baseline validation mAP_{50-95} differences under best-checkpoint, final-10, top-5 and matched-epoch summaries, with the fraction of shared epochs ahead shown as w .

Model	ΔmAP_{50-95} by rule			δ	(w)
	best	final-10	top-5		
YOLOv5n	+0.221	+0.231	+0.223	+0.216	(99/100)
YOLOv8n	+0.145	+0.165	+0.151	+0.170	(82/83)
YOLO11n	+0.029	+0.083	+0.041	+0.119	(100/100)

epochs. These summaries therefore provide no independent replication or estimate of uncertainty across seeds. Appendix C makes this distinction explicit.

5.5 Independent evaluation

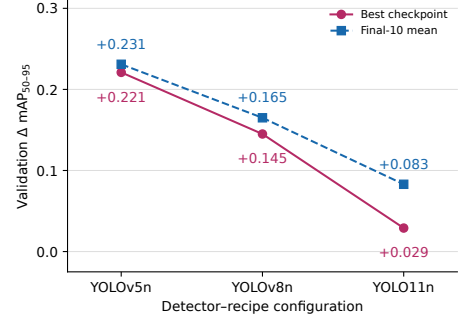
A confirmatory study requires checkpoint selection by validation performance followed by evaluation on an independent test set. Preprocessing, checkpoints and evaluation settings must be fixed before test inspection. Both members of a pair should use identical test images, with documented separation of related frames. Repetition across independent seeds would permit paired uncertainty estimates that account for the experimental sampling units. This protocol specifies the evaluation needed beyond the present retrospective analysis.

6 Results

Table 1 reports the selected-checkpoint validation results. Compensation increases mAP_{50-95} by 0.221, 0.145 and 0.029 for YOLOv5n, YOLOv8n and YOLO11n, respectively. The corresponding mAP_{50} increases are 0.269, 0.178 and 0.089. Both sequences decrease in release order. Each row represents one seed-0 run, and Δ denotes the within-configuration difference. The columns P and R denote precision and recall.

6.1 Summary sensitivity

Table 2 compares the four validation summaries. The YOLO11n baseline reaches 0.617 at epoch 88 and ends at 0.537. Its reported final-25-epoch mean is 0.556, compared with 0.626 for the compensated run. The selected-checkpoint difference is 0.029, the final-ten-epoch difference is 0.083 and the matched-epoch mean difference is 0.119. These values refer to distinct estimands, so their range cannot be interpreted as an uncertainty interval.

**Figure 4: Validation gains decrease across detector and training configurations in release order under both best-checkpoint and final-ten-epoch summaries.**

Positive differences occur at 99 of 100, 82 of 83 and 100 of 100 shared epochs (Figure 5). Compensation gains therefore extend beyond the selected checkpoints in the recorded histories. Their persistence remains conditional on the seed, validation sample and training procedure. Appendix F provides the complete recorded validation trajectories.

The decreasing order is preserved across all four summaries. For the matched-epoch means, the ratio of the YOLO11n gain to the YOLOv5n gain is $0.119/0.216 \approx 0.55$. Its interpretation is restricted to the observed validation histories because detector capacity and inference costs were not measured.

6.2 Precision and recall

For YOLOv5n, compensation increases reported recall from 0.455 to 0.802 and decreases precision from 0.878 to 0.710. The result indicates a change in the reported operating behavior toward higher recall and lower precision. Undocumented confidence thresholds prevent attributing this change to a specific alteration in the detection mechanism.

Both metrics increase for YOLOv8n and YOLO11n. Their precision gains are 0.192 and 0.130, and their recall gains are 0.154 and 0.078, respectively. Figure 6 shows their evolution during training. The supplementary confusion matrices and confidence curves provide further descriptive information. Evaluation at common thresholds and verified predictions on paired frames would be required to distinguish changes in localization, classification and confidence calibration.

6.3 Computational cost

The complete preprocessing pipeline was applied offline. Its inference latency and the end-to-end cost of the compensated system were not measured. The available detector timings exclude compensation and lack a reproducible protocol. The reported measurements consequently leave the feasibility constraint in Equation (3) unevaluated.

A deployment comparison requires the original implementation on specified target hardware, with software versions, numerical precision, thread count, resolution and batch size documented. The timing boundary should state the inclusion of decoding,

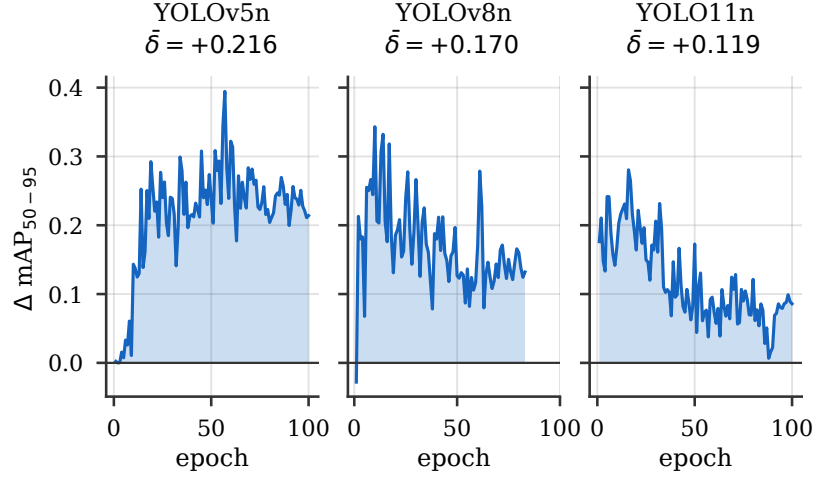


Figure 5: Paired per-epoch differences in validation mAP_{50-95} for each detector and training configuration, with positive shading indicating that the compensated run leads.

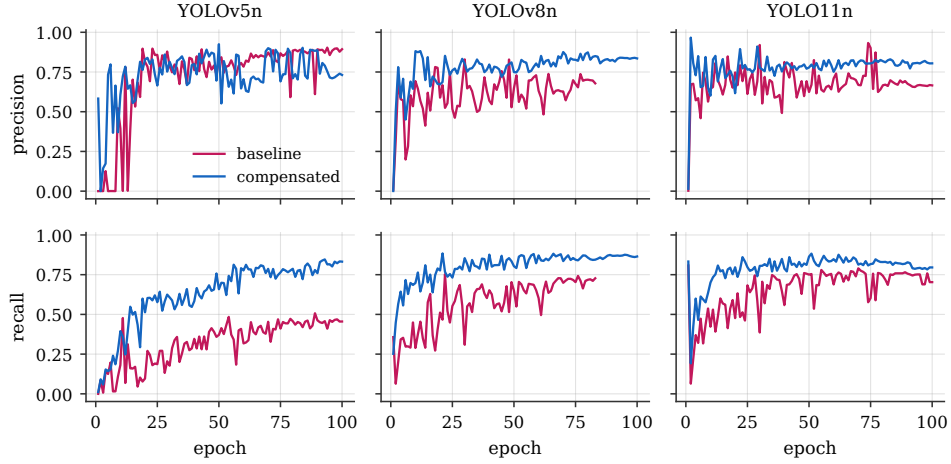


Figure 6: Validation precision (top) and recall (bottom) across training for baseline and compensated runs of each detector and training configuration.

resizing, device transfers and nonmaximum suppression. Warm-up, repeated synchronized measurements, and central and tail latency summaries are required where applicable. Identical timing boundaries for baseline and compensated paths are necessary to estimate the additional cost of compensation.

6.4 Result Discussion

The validation results indicate that the contribution of a fixed transformation depends on the detector configuration. The common ordering across checkpoint and trajectory summaries supports further study of this dependence under controlled experimental conditions. The variation in precision and recall also indicates that compensation can alter several aspects of reported detection performance.

Within the allocation model, a deployment decision depends on both the accuracy increment and its associated computational cost. For feasible alternatives, the Lagrangian comparison subtracts the

additional cost, weighted by the budget multiplier, from the accuracy gain. An allocation preference therefore requires a measured accuracy–cost comparison. Preprocessing expenditure also reduces the budget available for detector computation.

The current experiment evaluates the accuracy component of this comparison on a validation sample. Its architecture and training differences prevent identification of a capacity effect, while missing latency measurements prevent estimation of the cost penalty. A controlled family of detectors, independent test evaluation and complete pipeline timing would connect the observed differences to the theoretical allocation conditions. Such measurements would permit an empirical assessment of substitution across computational budgets.

7 Limitations and Future Work

The study is limited to a small underwater dataset and one training seed per configuration. Checkpoint selection and reporting share

the validation split, introducing selection bias. Architecture and training settings vary jointly, preventing isolation of detector capacity. Correlated epochs and unequal training horizons restrict interpretation of trajectory summaries. The complete compensator is evaluated without stage ablations or comparisons with learned enhancement. Preprocessing latency and total inference cost remain unmeasured. Reproducibility also requires verified split manifests, software versions, operating thresholds and resolution of the inconsistent stopping parameter. Future work should prioritize independent test evaluation, repeated training and synchronized pipeline timing. A controlled size sweep within one detector family would test the substitution hypothesis. Stage ablations and comparisons with adaptive transformations would clarify the source and cost of the observed gains. Additional datasets and imaging conditions are required to assess transferability and support allocation decisions under deployment constraints with specified computational budgets.

8 Conclusion

This study formulates the joint selection of image preprocessing and object detectors as a constrained resource allocation problem. A Lagrangian relaxation relates interior allocations to equal marginal returns, while supplementary derivations characterize boundary solutions, discrete alternatives and conditions for decreasing compensation gains. The underwater case study evaluates a fixed classical transformation with three YOLO detector configurations. Compensation improves validation mean average precision in every configuration, with gains decreasing in release order across checkpoint and trajectory summaries. Precision and recall responses vary, indicating that preprocessing affects multiple aspects of detection performance. These findings remain conditional on the dataset, training procedures and validation protocol. Architecture differences and validation reuse prevent establishing a general capacity relationship, while unmeasured inference costs preclude deployment recommendations. Independent test evaluation, repeated training, controlled capacity comparisons and complete pipeline timing are required to connect the observed accuracy gains to allocation decisions under specified computational budgets for underwater perception.

References

- [1] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. 6105–6114.
- [2] Muwei Jian, Xiangyu Liu, Hanjiang Luo, Xiangwei Lu, Hui Yu, and Junyu Dong. 2021. Underwater image processing and analysis: A review. *Signal Processing: Image Communication* 91 (2021), 116088.
- [3] Salma P. González-Sabbagh and Antonio Robles-Kelly. 2023. A survey on underwater computer vision. *Comput. Surveys* 55, 13s (2023), 1–39.
- [4] Shangquan Sun, Wenqi Ren, Tao Wang, and Xiaochun Cao. 2022. Rethinking image restoration for object detection. In *Advances in Neural Information Processing Systems*, Vol. 35. 4461–4474.
- [5] Yudong Wang, Jichang Guo, Wanru He, Huan Gao, Huihui Yue, Zenan Zhang, and Chongyi Li. 2024. Is underwater image enhancement all object detectors need? *IEEE Journal of Oceanic Engineering* 49, 2 (2024), 606–621.
- [6] Hugh Everett. 1963. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research* 11, 3 (1963), 399–417.
- [7] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, Cambridge, UK.
- [8] Karl Johan Åström and Richard M. Murray. 2008. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, USA.
- [9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 779–788.
- [10] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming* 14, 1 (1978), 265–294.
- [11] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. arXiv:1712.00409 [cs.LG]
- [12] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv:2001.08361 [cs.LG]
- [13] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napolitano. 2018. Benchmark analysis of representative deep neural network architectures. *IEEE Access* 6 (2018), 64270–64277.
- [14] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 [cs.CV]
- [15] Yanting Pei, Yaping Huang, Qi Zou, Yuhang Lu, and Song Wang. 2018. Does haze removal help CNN-based image classification?. In *European Conference on Computer Vision*. 682–697.
- [16] Gershon Buchsbaum. 1980. A spatial processor model for object colour perception. *Journal of the Franklin Institute* 310, 1 (1980), 1–26.
- [17] Graham D. Finlayson and Elisabetta Trezzi. 2004. Shades of gray and colour constancy. In *Color and Imaging Conference*. 37–41.
- [18] Cosmin Ancuti, Codruta Orniara Ancuti, Tom Haber, and Philippe Bekaert. 2012. Enhancing underwater images and videos by fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*. 81–88.
- [19] Codruta O. Ancuti, Cosmin Ancuti, Christophe De Vleeschouwer, and Philippe Bekaert. 2018. Color balance and fusion for underwater image enhancement. *IEEE Transactions on Image Processing* 27, 1 (2018), 379–393.
- [20] Kaiming He, Jian Sun, and Xiaoou Tang. 2011. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (2011), 2341–2353.
- [21] Paulo L. J. Drews, Erickson R. Nascimento, Silvia S. C. Botelho, and Mario Fernando Montenegro Campos. 2016. Underwater depth estimation and image restoration based on single images. *IEEE Computer Graphics and Applications* 36, 2 (2016), 24–35.
- [22] Adrian Galdran, David Pardo, Artzai Picón, and Aitor Alvarez-Gila. 2015. Automatic red-channel underwater image restoration. *Journal of Visual Communication and Image Representation* 26 (2015), 132–145.
- [23] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and Karel Zuiderveld. 1987. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing* 39, 3 (1987), 355–368.
- [24] Karel Zuiderveld. 1994. Contrast limited adaptive histogram equalization. In *Graphics Gems IV*. Academic Press, 474–485.
- [25] Carlo Tomasi and Roberto Manduchi. 1998. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*. 839–846.
- [26] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. 2020. An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing* 29 (2020), 4376–4389.
- [27] Chongyi Li, Saeed Anwar, and Fatih Porikli. 2020. Underwater scene prior inspired deep underwater image and video enhancement. *Pattern Recognition* 98 (2020), 107038.
- [28] Cameron Fabbri, Md Jahidul Islam, and Junaed Sattar. 2018. Enhancing underwater imagery using generative adversarial networks. In *IEEE International Conference on Robotics and Automation*. 7159–7165.
- [29] Md Jahidul Islam, Youya Xia, and Junaed Sattar. 2020. Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3227–3234.
- [30] Glenn Jocher. 2020. YOLOv5 by Ultralytics. <https://github.com/ultralytics/yolov5>.
- [31] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. Ultralytics YOLOv8. <https://github.com/ultralytics/ultralytics>.
- [32] Rahima Khanam and Muhammad Hussain. 2024. YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725 [cs.CV]
- [33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*. 740–755.
- [34] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- [35] Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. 2021. Accounting for variance

- in machine learning benchmarks. *Proceedings of Machine Learning and Systems* 3 (2021), 747–769.
- [36] David Picard. 2021. `Torch.manual_seed(3407)` is all you need: On the influence of random seeds in deep learning architectures for computer vision. arXiv:2109.08203 [cs.CV]
- [37] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.
- [38] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. RandAugment: Practical automated data augmentation with a reduced search space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.
- [39] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *AAAI Conference on Artificial Intelligence*, Vol. 34. 13001–13008.

Supplementary Material

A Allocation Analysis

This section derives the continuous optimality conditions and the finite-candidate comparison associated with Equation (3). The assumptions are stated for each model. The experimental results provide no estimate of the utility surface or deployment costs used in these derivations.

A.1 Continuous optimality conditions

Let F be continuously differentiable on a neighborhood of the feasible set

$$\mathcal{D}_B = \{(p, n) : p \geq 0, n \geq 0, p + n \leq B\}, \quad B > 0. \quad (14)$$

A maximum exists by compactness and continuity. The nonnegativity constraints must be included when considering boundary allocations. With multipliers $\lambda, \mu_p, \mu_n \geq 0$, the full Lagrangian for maximization is

$$\mathcal{L} = F(p, n) - \lambda(p + n - B) + \mu_p p + \mu_n n. \quad (15)$$

The affine constraints admit a strictly feasible point. At an optimum the Karush–Kuhn–Tucker (KKT) conditions give [7]

$$\begin{aligned} F_p - \lambda + \mu_p &= 0, & F_n - \lambda + \mu_n &= 0, \\ \lambda(p + n - B) &= 0, & \mu_p p &= \mu_n n = 0, \end{aligned} \quad (16)$$

together with primal and dual feasibility. Subscripts on F denote partial derivatives. If $p, n > 0$, complementary slackness yields $\mu_p = \mu_n = 0$ and hence $F_p = F_n = \lambda$. If $p = 0$ and $n > 0$, the conditions instead yield $F_p \leq \lambda = F_n$. An infinitesimal transfer from detection to preprocessing then has nonpositive first-order return. The symmetric inequality applies when $n = 0$. An inactive budget constraint implies a zero multiplier, $\lambda = 0$.

For concave F , these conditions are sufficient. Let $z^* = (p^*, n^*)$ satisfy them and let $z = (p, n)$ be feasible. The supporting hyperplane inequality gives

$$F(z) - F(z^*) \leq \nabla F(z^*)^\top (z - z^*). \quad (17)$$

Substitution of Equation (16) expresses the right-hand side as

$$\lambda[(p + n) - (p^* + n^*)] - \mu_p p - \mu_n n \leq 0, \quad (18)$$

where feasibility and complementary slackness give the inequality. Thus z^* is a global maximizer. Strict concavity of the utility function guarantees uniqueness of this maximizer. Without concavity, the conditions provide only a local stationarity test.

For an active budget, define $G_B(p) = F(p, B - p)$. Differentiation along the budget boundary gives

$$G'_B(p) = F_p - F_n, \quad G''_B(p) = F_{pp} - 2F_{pn} + F_{nn}. \quad (19)$$

The first derivative is the gain from transferring expenditure from detection to preprocessing. A negative second derivative at an interior stationary point establishes a strict local maximum along that boundary. This identifies the opportunity cost omitted by a comparison of preprocessing accuracy alone.

A.2 Discrete allocation

Let $\mathcal{J}$ be a finite set of trained pipelines, each with expected accuracy u_j and measured total cost c_j . For a budget admitting at least one candidate, the exact selection is

$$j_B \in \arg \max_{j \in \mathcal{J} : c_j \leq B} u_j. \quad (20)$$

A candidate can be discarded if another candidate has no greater cost and strictly greater accuracy. The retained accuracy–cost pairs define the relevant finite comparison set.

For a specified multiplier $\lambda \geq 0$, the Lagrangian score is $u_j - \lambda c_j$. Between two feasible candidates a and b , the score favors a precisely when

$$u_a - u_b \geq \lambda(c_a - c_b). \quad (21)$$

If detector cost is unchanged by retraining and the compensator has additional cost c_{pre} , the within-detector rule reduces to $V \geq \lambda c_{\text{pre}}$. Otherwise the full measured cost difference is required. Each candidate must also satisfy the original budget constraint.

The relaxed upper bound on the constrained optimum is

$$\max_{j : c_j \leq B} u_j \leq q(\lambda) = \lambda B + \max_{j \in \mathcal{J}} (u_j - \lambda c_j). \quad (22)$$

Indeed, for every feasible j , $u_j \leq u_j + \lambda(B - c_j) \leq q(\lambda)$. Minimizing q over nonnegative multipliers tightens the bound. Finite candidate sets can retain a duality gap, so weighted scores alone need not recover every constrained optimum. For example, consider $(c_j, u_j) \in \{(1, 0), (2, 0.4), (3, 1)\}$ and $B = 2$. The middle point is optimal under the budget, but its unrestricted score exceeds the first only if $\lambda \leq 0.4$ and the third only if $\lambda \geq 0.6$. No nonnegative multiplier can satisfy both inequalities simultaneously. Direct feasible enumeration resolves this case. These values are an analytic illustration, independent of the reported experiments.

B Substitution Conditions

The decrease in Equation (7) requires assumptions on the interaction between capacity and preprocessing. Let $H(\kappa, p)$ be a twice continuously differentiable accuracy surface for a controlled detector family and preprocessing expenditure p . Training follows a common protocol across the detector family. If training is randomized, H denotes the expectation over that randomness as well as evaluation samples. Suppose that p_0 and $p_1 > p_0$ correspond to the baseline and a fixed compensated alternative. The compensation increment is then defined by

$$V_H(\kappa) = H(\kappa, p_1) - H(\kappa, p_0). \quad (23)$$

Differentiation and the fundamental theorem of calculus give

$$V'_H(\kappa) = \int_{p_0}^{p_1} H_{\kappa p}(\kappa, s) ds. \quad (24)$$

Hence $H_{\kappa p} \leq 0$ throughout the relevant rectangle is sufficient for decreasing compensation gains. The finite-difference form follows by a second integration,

$$\begin{aligned} V_H(\kappa') - V_H(\kappa) &= \int_{\kappa}^{\kappa'} \int_{p_0}^{p_1} H_{\kappa p}(r, s) ds dr \\ &\leq 0 \quad (\kappa' > \kappa). \end{aligned} \quad (25)$$

This is a decreasing-differences condition on the joint surface. Concavity in each variable separately does not establish it. Even joint concavity is insufficient to determine the cross-partial sign. For example, $H(\kappa, p) = a\kappa + bp - (\kappa - p)^2$ is jointly concave and has $H_{\kappa p} = 2 > 0$. Thus its compensation increment increases with κ . On any bounded domain, constants and positive rescaling can place this illustrative surface within the accuracy range without changing the sign conclusion.

The fixed transformation experiment samples no continuous preprocessing axis, and release order supplies no calibrated capacity coordinate. Consequently, the observed ordering cannot identify $H_{\kappa p}$. Testing the sufficient condition requires controlled measurements over both axes, together with independent uncertainty estimates. A diminishing accuracy increment also leaves the cost comparison in Equation (21) unresolved.

C Trajectory and Selection Effects

Validation histories describe repeated measurements along a dependent training process. Let $\delta_t = m_t^{\text{enh}} - m_t^{\text{base}}$ be random under repetition of the full experiment. The variance of its trajectory mean is

$$\text{Var}(\bar{\delta}) = \frac{1}{T^2} \left[\sum_{t=1}^T \text{Var}(\delta_t) + 2 \sum_{1 \leq s < t \leq T} \text{Cov}(\delta_s, \delta_t) \right]. \quad (26)$$

The covariance terms preclude an uncertainty calculation that treats epochs as independent replications. Training nonstationarity further prevents interpreting a single trajectory as repeated draws from a common performance distribution. For the observed histories, Equation (13) is a descriptive summary.

Checkpoint selection introduces a separate effect. For a fixed finite candidate set, write the observed validation score as $\hat{u}_t = u_t + \varepsilon_t$, where u_t is its target expected score and $\mathbb{E}[\varepsilon_t] = 0$. Convexity of the maximum implies

$$\mathbb{E} \left[\max_t \hat{u}_t \right] \geq \max_t u_t. \quad (27)$$

Independence across the errors is unnecessary for this inequality. It formalizes selection optimism under the stated fixed-candidate model. Adaptive stopping requires an extension of this fixed-candidate model. Baseline and compensated selection biases can differ, so their subtraction has no guaranteed bias cancellation. Evaluation on an independent test set after checkpoint selection and repeated training address distinct sources of uncertainty.

D Training Settings

All six runs use 640×640 input images, a batch size of 16, seed 0 and a maximum of 100 epochs. Settings are reported as matched within each baseline–compensated pair. Training settings differ across configurations, confounding comparisons of architecture and capacity.

YOLOv5n uses the reported default hyperparameters with an initial learning rate of 0.01, hue-saturation-value jitter, translation, scaling and horizontal flipping. YOLOv8n uses adaptive moment estimation with decoupled weight decay (AdamW) [34] with

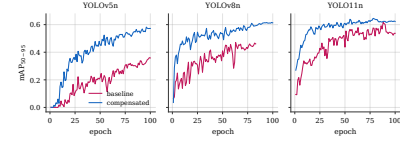


Figure 7: Validation mAP_{50–95} across training for the six recorded baseline and compensated runs.

an initial learning rate of 0.001, cosine decay, mixup [37] and copy-paste probabilities of 0.1, random augmentation (RandAugment) [38] and random erasing [39] at 0.4. YOLO11n uses an initial learning rate of 0.01, automatic optimizer selection, cosine decay, RandAugment and random erasing at 0.4, without mixup or copy-paste. All recipes include mosaic augmentation, disabled for the last ten epochs in the YOLOv8n and YOLO11n recipes.

Early-stopping patience is recorded as 20 epochs for YOLOv5n and YOLOv8n. Available descriptions give inconsistent values of 20 and 30 for YOLO11n, leaving its stopping setting uncertain. The matched-epoch analysis uses 100, 83 and 100 shared epochs, respectively. No additional seeds or held-out test evaluations are reported.

E Compensator Parameter Selection

The fixed parameter vector is $\phi = (c_{\text{clip}}, G, \alpha, D, \sigma_c, \sigma_s) = (2.0, 10 \times 10, 0.3, 5, 75, 75)$ across all six runs. Contrast-limited adaptive histogram equalization (CLAHE) uses ten tiles along each axis. Increasing the tile count decreases the region size for local equalization. A sharpening weight of 0.3 was selected after visual comparisons with 0.1 and 0.5, and a bilateral filter diameter of 5 after comparisons with 9 and 15. Parameter selection used visual and histogram inspection. Detection-based optimization and stage ablations are required to assess optimality and component contributions. Reproduction also requires the original implementation and its handling of numerical edge cases.

F Training Trajectories

Figure 7 contains the six validation histories used in Equation (13). The YOLO11n baseline has a transient peak at epoch 88, which contributes to the difference between best-checkpoint and trajectory summaries. Both summaries use the validation data. Shared training histories and repeated images induce dependence across epochs, as formalized in Appendix C.

Figures 8–10 reproduce the framework’s loss and metric panels. The plotted losses have architecture-specific definitions, which restrict cross-model comparison. In the panel labels, *cls* denotes classification, *obj* objectness, *box* bounding-box regression, *dfl* distribution focal loss (DFL), *val* validation and *lr* learning rate. The suffix *B* in metric labels denotes bounding boxes. The curves document the recorded training behavior.

G Detection Diagnostics

The class confusion matrices in Figure 11 complement Table 1. Figures 12 and 13 show precision–recall curves and the F1 score as a function of confidence threshold. F1 is the harmonic mean of

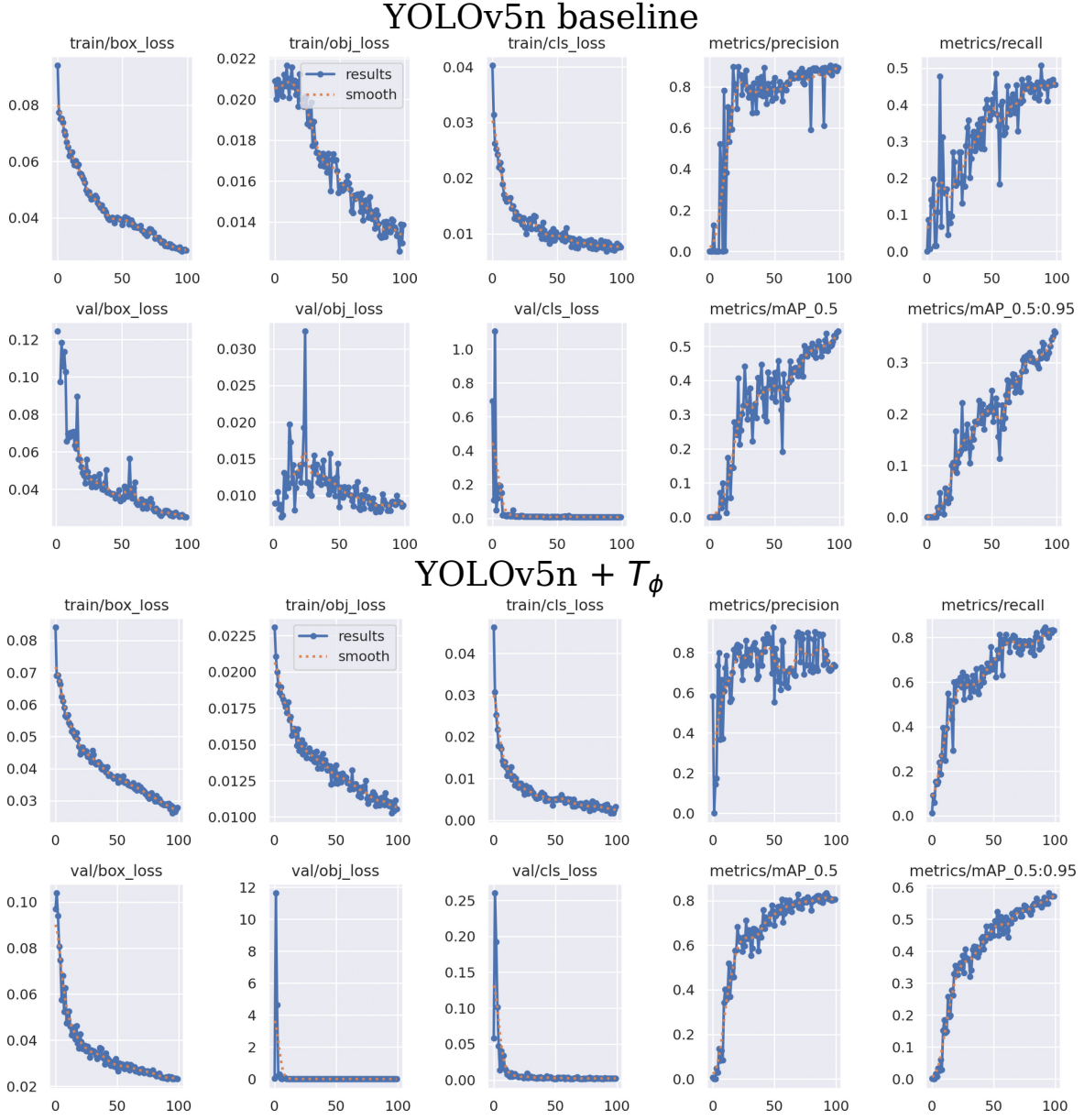


Figure 8: Training losses and validation metrics for YOLOv5n, with the baseline run above the compensated run. All panels represent the recorded single-seed experiments.

precision and recall. All reported diagnostic panels use the validation data. The undocumented operating thresholds for the scalar metrics limit comparisons at a common operating point and interpretation of the underlying detection mechanism.

H Compensation Examples

Figure 14 complements Figure 2 with stage-wise illustrations on tub and gate scenes. The red, green and blue (RGB) histograms show changes in intensity distributions under the fixed pipeline. Detection-based stage ablations are required to attribute performance changes to individual operations.

YOLOv8n baseline

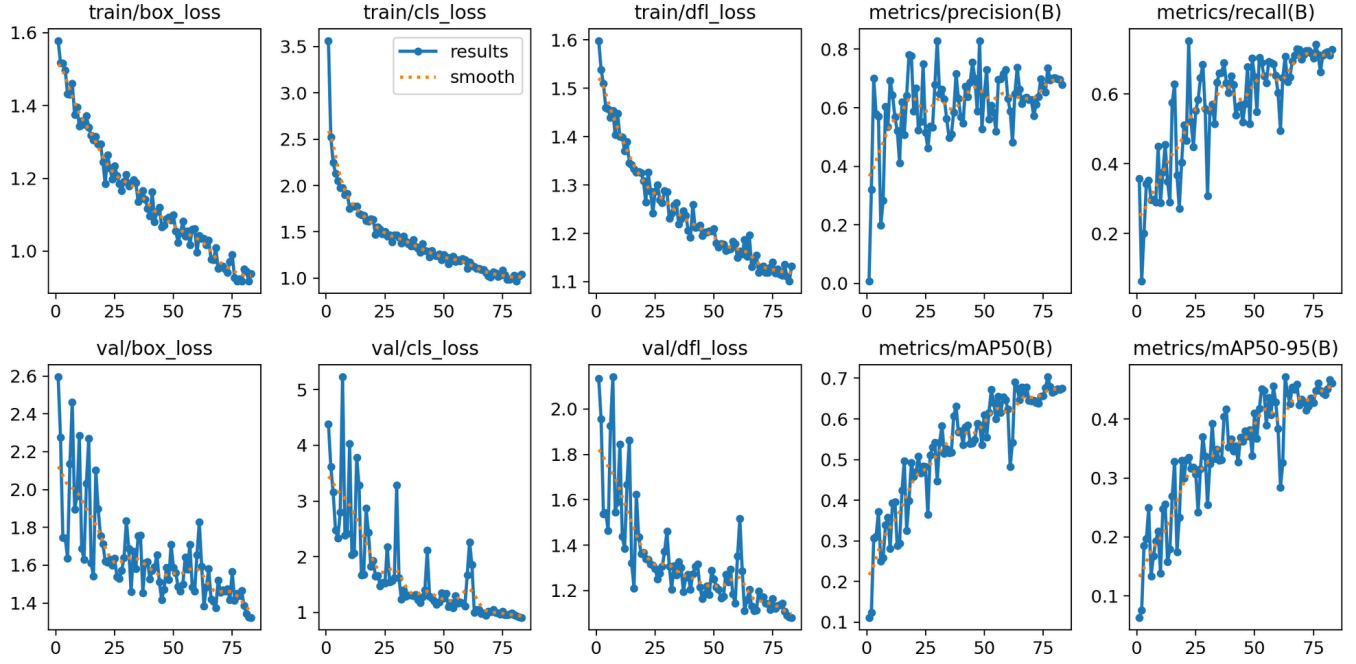
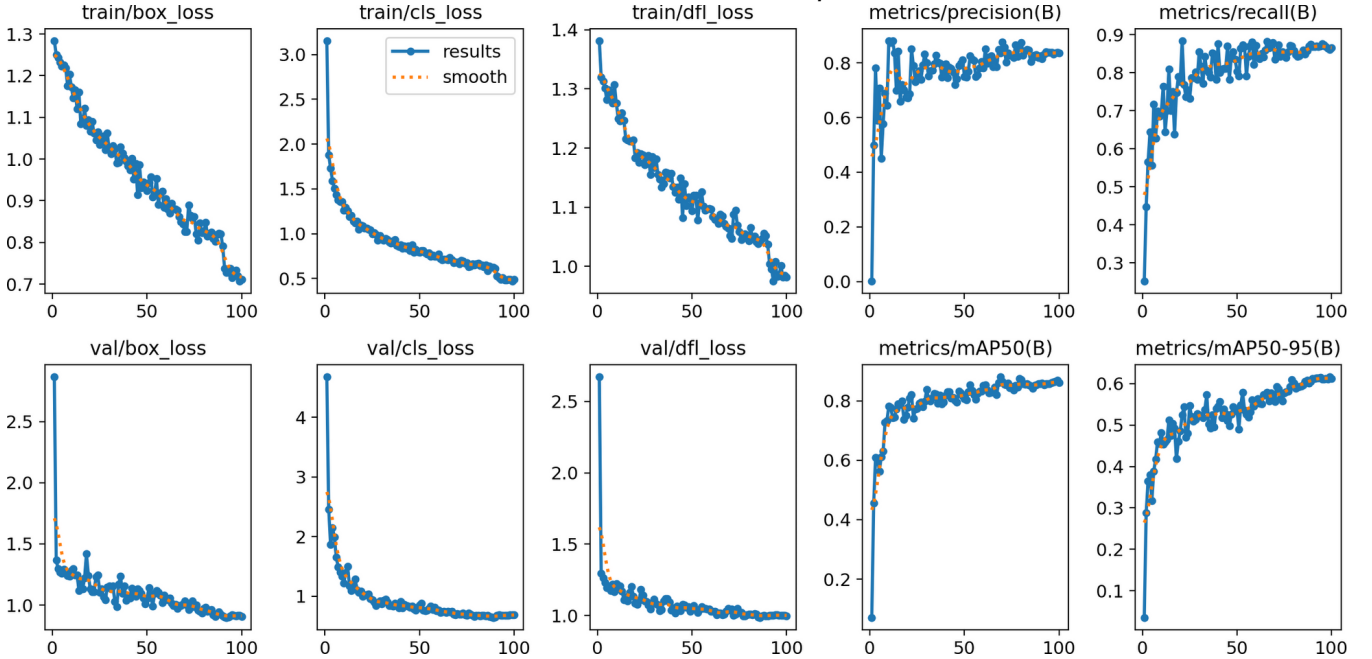
YOLOv8n + T_ϕ 

Figure 9: Training losses and validation metrics for YOLOv8n, with the baseline run above the compensated run. All panels represent the recorded single-seed experiments.

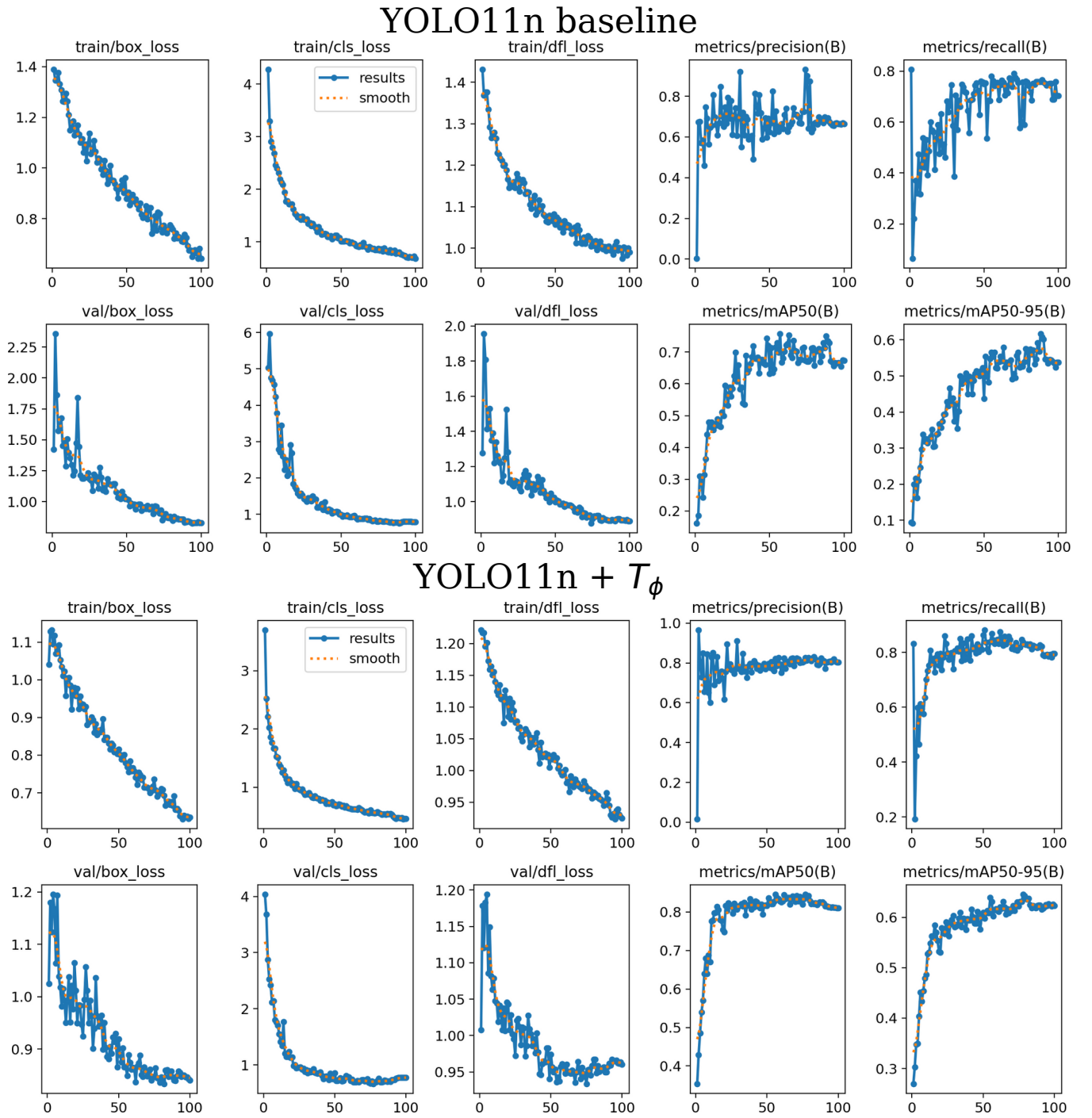


Figure 10: Training losses and validation metrics for YOLO11n, with the baseline run above the compensated run. All panels represent the recorded single-seed experiments.

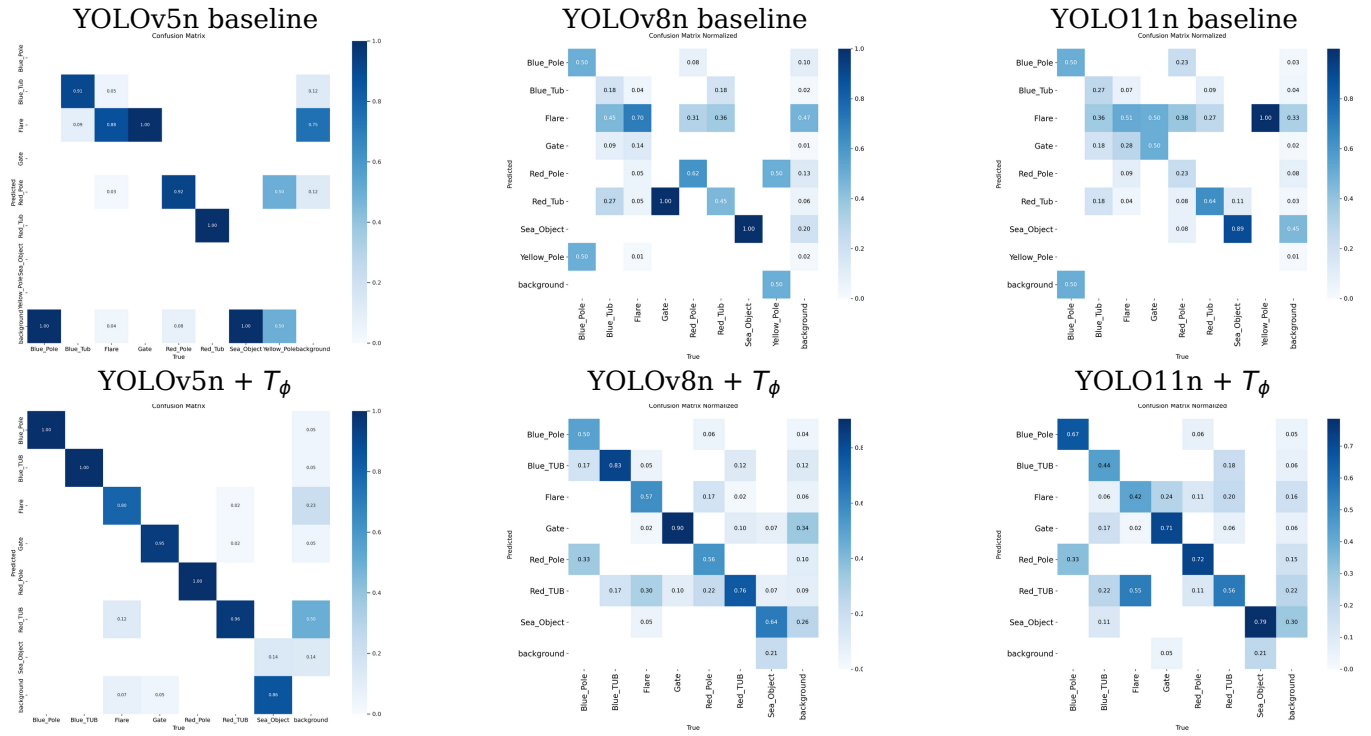


Figure 11: Validation confusion matrices for all six runs, with baseline runs above and compensated runs below.

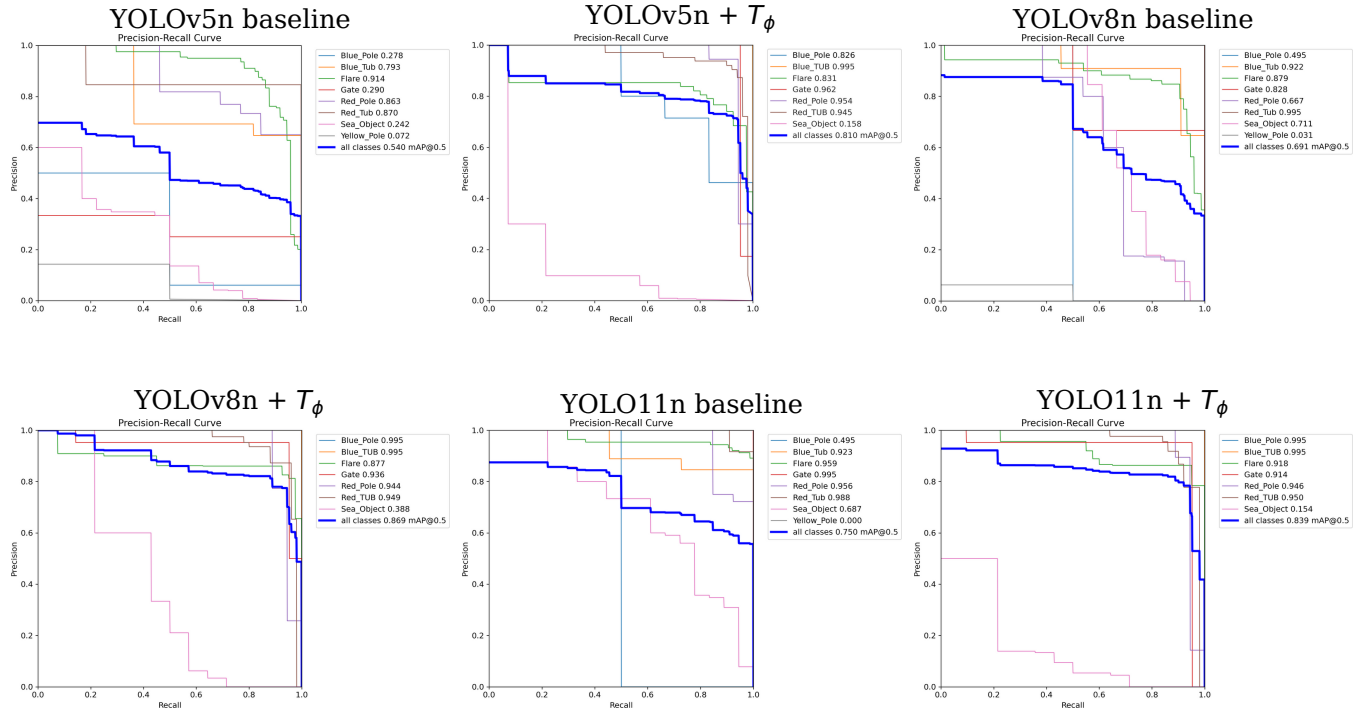


Figure 12: Framework precision-recall curves at each run's best validation checkpoint, with panel headings identifying the architecture and input configuration.

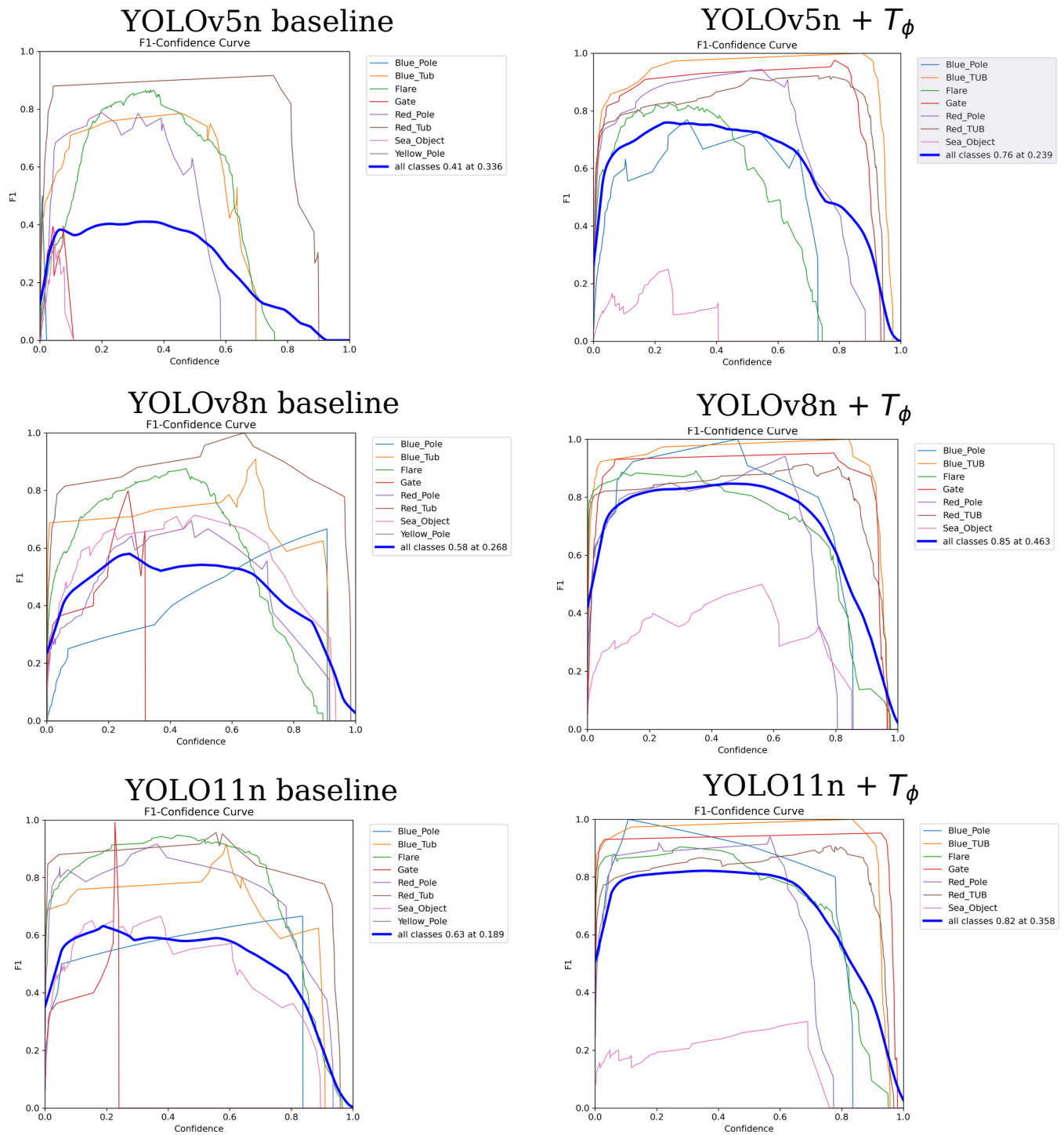


Figure 13: Framework F1-confidence curves at each run’s best validation checkpoint, with one detector per row and baseline and compensated runs in the left and right columns, respectively.

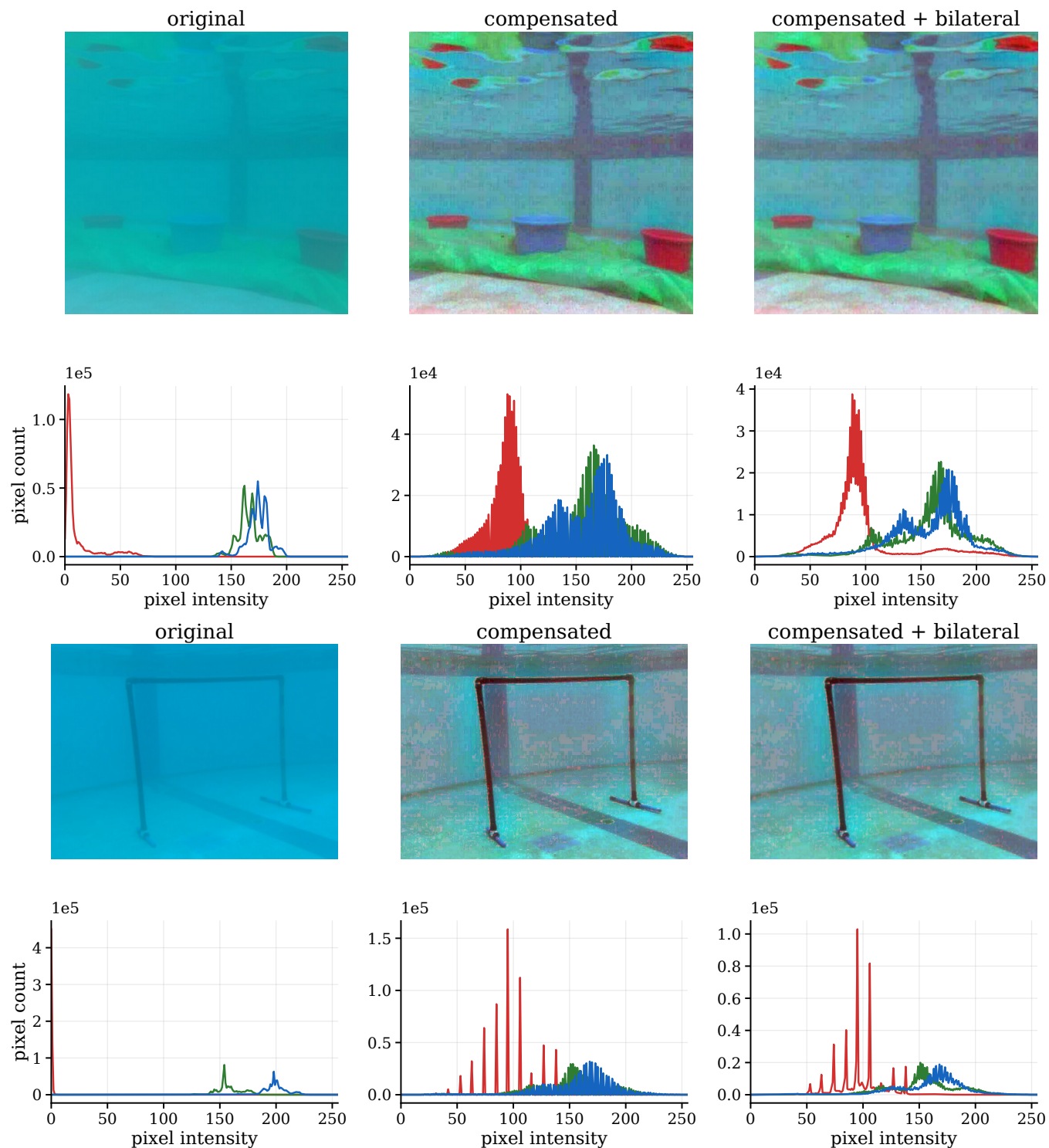


Figure 14: Compensation stages for tub (top) and gate (bottom) scenes. Each row group shows the original image, the output after per-channel processing and the output after bilateral filtering, with red, green and blue intensity histograms below.