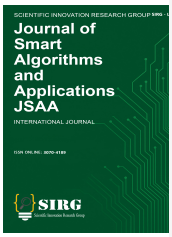


Journal of Smart Algorithms and Applications
JSAA
ISSN: 3070-4189/
Journal Homepage
https://pub.scientificirg.com/index.php/JSAA

Optimizing YOLOv12-L for Imbalanced Satellite Vehicle Detection via Physics-Informed Data Synthesis and Adaptive Class-Aware Loss Weighting

Ola Farid[a, 1] **, Mohammed Melhi**[b]**, and A. A. Somaie**[c]

[a] *Computer Science Department, Faculty of Science, Beni-Suef University, Beni-Suef City, 62511, Egypt. E-mail: ola3131313@gmail.com*

[b] *PhD, University of Bradford, UK, Chief Executive Officer CEO of Rushd AI Company and Amatrix AI Company, Riyadh, KSA. E-mail: mmelhi@rushai.com, mmelhi@promatrix.ai*

[c] *PhD, University of Bradford, UK, PDF Post-Doctoral Fellow Research Associate, University of Calgary, Canada, Software Engineering Program SE, Faculty of Computer Science, October University for Modern Sciences & Arts MSA, 6 October, Giza, Egypt. E-mail: alibrahim@msa.edu.eg, aasomaie@gmail.com*

ABSTRACT

Vehicle detection in satellite imagery underpins applications in traffic monitoring, urban planning, and disaster response, yet two obstacles routinely degrade detector performance: the small pixel footprint of individual vehicles and severe imbalance between common and rare vehicle categories. This study presents a vehicle detection framework built on the YOLOv12-L architecture and evaluated on the Vehicles in the Middle East (VME) dataset, in which cars outnumber buses and trucks by a wide margin. The framework combines a physics-informed synthetic data generator, which places procedurally rendered bus and truck silhouettes onto realistic backgrounds and enhances them with adaptive contrast and sensor-noise simulation, with a class-aware loss configuration that increases the classification loss weight and applies label smoothing for minority categories. Training on 4,666 images (2,828 original tiles plus 1,838 synthetic tiles contributing 2,476 bus and 2,076 truck instances) for 100 epochs produced a validation mean Average Precision at an IoU threshold of 0.5 (mAP50) of 0.687 and mAP50:95 of 0.391, with per-class mAP50:95 values of 0.476 for cars, 0.374 for buses, and 0.323 for trucks. On a held-out test split, mAP50 reached 0.634 and mAP50:95 reached 0.361. Increasing the inference resolution from 640 to 736 pixels raised validation mAP50:95 to 0.398 without retraining. Test-Time Augmentation at the 1.17× scale further improves overall mAP_{50} to 0.693. These results confirm that targeted data synthesis combined with class-aware loss weighting yields consistent gains for underrepresented vehicle categories in geospatial imagery.

PAPER INFORMATION

HISTORY

Received: 13 March 2026
Revised: 11 May 2026
Accepted: 19 June 2026
Online: 26 June 2026

MSC

68T07; 68R10; 94A60; 68M15

KEYWORDS

YOLOv12-L;
Satellite Imagery;
Vehicle Detection;
Data Synthesis;
Loss Weighting.

[1]Corresponding author at Computer Science Department, Faculty of Science, Beni-Suef University, Beni-Suef City, 62511, Egypt. E-mail: ola3131313@gmail.com

1. INTRODUCTION

The rapid expansion of commercial satellite constellations has transformed geospatial intelligence from an activity requiring specialized infrastructure into an accessible data source for a broad range of civilian and governmental applications. High-resolution imagery from sensors such as Maxar WorldView-3 (0.31 m ground sampling distance, GSD) now enables automated analysis at the level of individual objects within complex urban and peri-urban scenes. Vehicle detection is among the most consequential of these analytical tasks, providing non-invasive proxies for economic activity, public transportation utilization, commercial logistics throughput, and emergency response coordination [1, 2].

Despite substantial progress in general-purpose object detection driven by convolutional neural networks and transformer architectures, reliable vehicle detection in satellite imagery remains technically challenging for two interrelated reasons.

Unlike ground-level camera imagery, satellite platforms capture vehicles that occupy only 16–100 pixels in a 640×640 tile, and frequently fewer than 16×16 pixels. At these scales, standard detection frameworks suffer from limited spatial information, low feature discriminability between vehicle types, and high sensitivity to imaging artifacts such as sensor noise, atmospheric haze, and cast shadows [3]. Feature Pyramid Networks (FPN) and multi-scale detection heads partially address these limitations through hierarchical feature fusion, yet preprocessing quality and feature enhancement remain important complementary strategies that architecture-centric solutions frequently overlook.

Real-world satellite imagery datasets exhibit extreme class skew. In the Vehicles in the Middle East (VME) dataset employed in this study [2], the training split contains 84,567 car instances alongside only 689 bus instances and 1,023 truck instances, yielding a car-to-bus imbalance ratio of approximately 123:1 and a car-to-truck ratio of approximately 83:1. This skew causes standard detection models to allocate gradient signal overwhelmingly to the majority class. Buses and trucks, though analytically critical as indicators of public transit utilization, freight movement, and industrial activity, receive insufficient gradient during training under naive configurations [4, 5].

Prior work has addressed these two challenges through largely separate research streams. Multi-scale feature aggregation networks [6] and their successors improve small-object recall by fusing coarse semantic representations with fine spatial detail [7, 8]. Focal Loss [9] and its extensions, such as Varifocal Loss [10], reweight individual predictions by classification difficulty. Copy-paste augmentation [11, 12] increases the frequency of minority-class instances in training batches.

The current paper introduces the following contributions.

1. **Variance-adaptive satellite preprocessing**: a sequential pipeline that applies local-variance-guided CLAHE (Contrast Limited Adaptive Histogram Equalization), guided filtering, automatic gamma correction, and adaptive unsharp masking to enhance small-object discriminability prior to network ingestion (Section 3.3).

2. **Physics-informed synthetic data generation**: a pipeline that produces geometrically and photometrically realistic bus and truck instances through patch extraction, procedural template rendering, scene-type-specific background synthesis, and Gaussian-smoothed alpha compositing (Section 3.4). The generator produces 1,838 additional training images containing 2,476 bus and 2,076 truck instances.

3. **Class-aware training**: a configuration that incorporates inverse-frequency class weights, label smoothing ($\varepsilon = 0.05$) for minority-class logits, an elevated classification loss coefficient ($\lambda_{\text{cls}} = 3.0$), and class-conditional NMS thresholds (Section 3.6).

4. **Multi-scale test-time augmentation (TTA)**: inference at three resolution scales (0.83×, 1.00×, 1.17×) that raises validation mAP_{50} from 0.687 to 0.693 without additional training (Section 3.6.5).

5. **Comprehensive evaluation** on the VME benchmark: per-class precision, recall, and Average Precision metrics; confusion matrix analysis; confidence-threshold curve analysis; ablation discussion; and comparison with YOLOv8x and YOLOv11l baselines.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the system architecture. Section 4 provides algorithmic descriptions. Section 5 presents experimental results and analysis. Section 6 discusses findings and limitations. Section 7 concludes.

2. RELATED WORK

The detection of small objects in aerial and satellite imagery has attracted sustained research attention over the past decade. Early approaches relied on multi-scale sliding window strategies combined with hand-crafted features such as Histogram of Oriented Gradients (HOG), but these proved computationally prohibitive and insufficiently discriminative at fine scales. The shift to convolutional neural network (CNN)-based detectors dramatically improved detection quality, though standard architectures such as Faster R-CNN [13] and the YOLO family were designed primarily for natural scene imagery where objects of interest typically occupy a substantial image fraction.

Feature pyramid networks (FPN) [6] provided one of the first systematic architectural responses to the multi-scale problem by combining high-resolution, low-semantic feature maps from early network layers with low-resolution, high-semantic maps from deeper layers. This aggregation proved particularly valuable for small objects, which are best represented in the finer spatial resolutions of shallow features. Subsequent work generalized FPN to bidirectional path aggregation networks and other multi-scale fusion designs, consistently demonstrating improvements for small-scale objects in both natural and remote sensing imagery [7].

Attention mechanisms have been applied to prioritize spatial regions likely to contain small objects. Context-aware attention modules that aggregate information from neighboring spatial regions improve feature discriminability for objects that individually provide insufficient information but whose surrounding context, such as road lanes or parking lot structure, is highly diagnostic [8]. Non-semantic sparse attention applied to high-frequency feature components has shown particular promise for vehicle detection, where the sharp edges of vehicle boundaries and windows provide strong discriminative cues even at small scales.

A more specialized line of work directly targets the feature inadequacy problem through super-resolution upsampling of small-object regions prior to detection. While effective, super-resolution-based approaches introduce additional computational overhead and latency that may be prohibitive for operational processing of large image archives.

Preprocessing strategies represent a complementary, often underexplored approach. CLAHE has been widely used in medical imaging for local contrast enhancement and has been applied to remote sensing imagery to improve the visibility of low-contrast objects against spectrally similar backgrounds [3]. Unsharp masking and guided filtering provide complementary sharpening effects that amplify high-frequency edge content without the noise amplification associated with standard Laplacian sharpening. The system described in this paper applies all three enhancements in sequence, with CLAHE tile sizes adapted to local image variance for more effective contrast control.

Class imbalance in object detection differs from the classification setting in a subtle but important way: the imbalance operates at the level of object instances rather than images, and many images contain a mix of majority and minority class instances. Naive oversampling of minority-class images therefore does not guarantee that the minority class will receive proportionally more gradient updates during training.

Two broad families of methods have been proposed. The first modifies the loss function to reweight samples. Focal Loss [9], originally developed for the RetinaNet detector, down-weights the contribution of easy negative examples and concentrates learning on hard positives. The original motivation was to address the foreground-background imbalance in single-stage detectors rather than cross-class imbalance, but the mechanism extends naturally to minority-class detection. Static class weighting, in which a fixed multiplier is applied to the loss of each class based on its relative frequency, is simpler to implement and more interpretable; it has been shown to be effective when class frequencies are stable across the training set [5]. The system in this paper computes class weights using the balanced class weighting formula from scikit-learn, giving weights of approximately 0.075 for Car, 49.8 for Bus, and 22.2 for Truck.

The second family of methods addresses imbalance at the data level. Copy-Paste augmentation [12] pastes object instances from one image onto another, increasing the occurrence frequency of pasted classes. The method has been shown to improve minority-class performance substantially in COCO-style benchmarks. For satellite imagery, the visual plausibility of pasted objects is critical because satellite imagery lacks the perspective distortion of ground-level photography, making poorly scaled or unrealistically colored objects immediately visible to the detector as distribution shift. The synthetic data generation pipeline described in Section 3 addresses this concern through physics-informed template generation, photorealistic background synthesis, and Gaussian-blending at object boundaries.

The YOLO series of single-stage detectors has undergone substantial architectural evolution since the original proposal. YOLOv8 introduced a decoupled detection head and anchor-free localization that improved performance across scales. YOLOv9 incorporated auxiliary reversible branches for better gradient flow. YOLOv10 explored NMS-free inference. YOLOv11 refined the backbone with C3k2 modules and improved multi-scale feature aggregation.

YOLOv12 [14], the architecture used in this paper, introduces an attention-centric design that replaces the channel-based aggregation of its predecessors with an Area Attention Mechanism. This mechanism partitions feature maps into contiguous rectangular regions and applies multi-head self-attention within each region, capturing long-range dependencies at manageable computational cost. The complementary Residual Efficient Layer Aggregation Network (R-ELAN) improves gradient flow in deep layers. Empirical benchmarks on COCO val2017 show that YOLOv12l achieves $mAP_{50:95}$ of 55.2 with the largest variant, outperforming YOLOv11l at comparable parameter counts, though with somewhat higher memory requirements and training time.

The availability of geographically diverse, high-resolution annotation data is a prerequisite for developing generalizable vehicle detectors. Several benchmark datasets have been established, each with distinct characteristics. The xView dataset [15] provides over one million annotated objects across 60 classes at 0.3 m GSD but focuses on a wide class taxonomy rather than vehicle subtype differentiation. DOTA-v2.0 [16] introduced oriented bounding box annotations for aerial imagery, enabling more precise localization of elongated objects such as buses. VEDAI [17] provides multispectral imagery at fine resolution but is relatively small in scale. DIOR [18] and FAIR1M-2.0 [19] cover diverse geographic regions with rich class taxonomies.

A notable limitation of many existing datasets is geographic bias: the majority of annotated images are drawn from North American, European, or East Asian urban environments. The Vehicles in the Middle East (VME) dataset [2] directly addresses this gap by providing over 4,000 image tiles and 100,000+ vehicle annotations from 54 cities across 12 Middle Eastern countries, sourced from Maxar satellites at sub-meter resolution. This geographic specificity matters because vehicle types, colors, road layouts, and background spectral properties differ meaningfully between regions, and models trained on Western datasets may not transfer well. The authors of VME demonstrated a 56.3% improvement in mAP for car detection in Middle Eastern scenes when training data was augmented with VME images [2].

Table 1 summarizes representative recent contributions in the field, providing a structured comparison of methodologies, datasets, and reported performance.

3. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The proposed framework is implemented in Python using PyTorch and the Ultralytics YOLO library. It comprises four integrated modules: dataset management, satellite image preprocessing, physics-informed synthetic data generation, and class-aware model training. The overall pipeline proceeds as follows: raw images are preprocessed, the synthetic data generator extends the training set, and the augmented dataset is used to train and evaluate YOLOv12-L. **Figure 1** illustrates the overall architecture of the proposed vehicle detection framework.

3.1 Dataset and Class Statistics

All experiments use the VME dataset [2], available at `https://zenodo.org/records/14185684`. The dataset comprises 4,282 high-resolution satellite image tiles with 113,737 annotated vehicle instances across three

Table 1: Summary of Related Work in Object Detection for Remote Sensing Imagery

Paper	Year	Dataset	Task	Model	Accuracy	Contribution	Limitation
VME [2]	2025	VME (self-introduced)	Vehicle Detection	YOLOv8 variants	+56.3% mAP in ME	Geographic coverage for Middle East; CDSI benchmark	Limited to car class; no rare-class focus
LMW-YOLO [20]	2026	NWPU VHR-10, VisDrone2019	Small Object Detection	YOLO with CSD/LKCA	SOTA on VisDrone, 2.6M params	Context-scale decoupled strategy; large-kernel aggregation	High design complexity; limited class diversity
SMA-YOLO [21]	2025	VisDrone2019	Small Object Detection	Custom YOLO	+13% mAP vs. baseline	Non-semantic sparse attention; bidirectional FPN	Not evaluated on satellite imagery
YOLOv12 [14]	2025	COCO val2017	General Detection	YOLOv12x	55.2 $mAP_{50:95}$	Area Attention Mechanism; R-ELAN	Higher memory and training time vs. YOLOv11
Focal Loss [9]	2017	COCO	Class Imbalance	RetinaNet	+1.9 AP vs. cross-entropy	Dynamic hard-example weighting	Focuses on fg/bg imbalance; not cross-class
Copy-Paste [12]	2024	COCO, LVIS	Data Augmentation	Various	+0.8–3.2 AP	Instance-level paste for rare classes	Realism of pasted objects not guaranteed
DOTA-v2 [16]	2022	DOTA-v2.0	Oriented OD	OBB detectors	Oriented AP	Oriented BBox; large-scale benchmark	No vehicle subtype focus

Figure 1: Overview of the proposed vehicle detection framework

classes: Car (class 0), Bus (class 1), and Truck (class 2). Annotations are provided in Oriented Bounding Box (OBB) format and converted to Horizontal Bounding Box (HBB) format for compatibility with YOLOv12.

The dataset is partitioned into training, validation, and test splits using stratified sampling to preserve rare-class proportions. Images containing at least one bus or truck annotation are isolated prior to random shuffling to ensure adequate rare-class coverage in both the validation and test sets. Images are then split 70%/15%/15% within each stratum, and the resulting subsets are merged. **Table 2** reports the resulting split statistics. The severe imbalance is evident in the training split, where cars outnumber buses by a factor of approximately 123:1. Computed inverse-frequency class weights are $w_{\text{Car}} = 0.075$, $w_{\text{Bus}} = 49.82$, and $w_{\text{Truck}} = 22.15$.

Table 2: Dataset statistics and class distribution across splits

Split	Images	Car	Bus	Truck
Training	2,828	84,567	689	1,023
Validation	606	15,018	764	1,001
Testing	608	14,266	738	1,187
Total	**4,042**	**113,851**	**2,191**	**3,211**
Car:Bus ratio (train)		122.7:1		
Car:Truck ratio (train)		82.7:1		

3.2 GPU Memory Management

To achieve training stability on the 14 GB Tesla T4 GPU, an `AdvancedMemoryManager` monitors GPU memory utilization at each training step. When allocation exceeds 85% of total capacity, the manager invokes Python garbage collection and `torch.cuda.empty_cache()` to reclaim fragmented memory. This mechanism proved necessary during synthetic data generation, where large batches of image compositing operations triggered memory spikes. In practice, the manager executed periodic cleanup across the 1,838-sample generation phase without interrupting the pipeline.

3.3 Variance-Adaptive Satellite Image Preprocessing

Raw satellite imagery from the VME dataset exhibits variable contrast, sensor noise, and atmospheric effects that reduce the discriminability of small vehicle signatures.
The `AdvancedSatellitePreprocessor` applies a sequential enhancement pipeline to every image before it enters the training data loader.

3.3.1 Adaptive CLAHE

The image is first converted to the CIE LAB color space. The lightness channel L is processed with Contrast Limited Adaptive Histogram Equalization (CLAHE) [22]:

$$\hat{L}(x, y) = \text{CLAHE}(L(x, y),\ c = 2.5,\ g = G)\,, \tag{1}$$

where c is the clip limit and G is the tile grid size, selected adaptively from the local variance σ^2_{local} of the grayscale image:

$$G = \begin{cases} (4, 4) & \sigma^2_{\text{local}} > 800, \\ (8, 8) & 200 < \sigma^2_{\text{local}} \leq 800, \\ (16, 16) & \sigma^2_{\text{local}} \leq 200. \end{cases} \tag{2}$$

Finer tiles are applied to heterogeneous urban scenes, where local contrast variations are pronounced, while coarser tiles are used for more homogeneous backgrounds to avoid noise amplification.

3.3.2 Guided Filtering, Gamma Correction, and Unsharp Masking

Following CLAHE, a guided filter with radius $r = 4$ and regularization $\varepsilon = 0.01$ smooths the enhanced image while preserving object edges [23]. An automatic gamma correction step adjusts image brightness toward a target mean of 0.5 using

$$\gamma = \text{clip}\left(\frac{\log(0.5)}{\log(\bar{I})},\ 0.4,\ 2.5\right), \tag{3}$$

where $\bar{I}$ is the normalized mean pixel intensity. Adaptive unsharp masking is then applied with the kernel

$$\mathbf{K}_{\text{sharp}} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & k & -1 \\ -1 & -1 & -1 \end{pmatrix}, \tag{4}$$

where $k \in \{10, 6, 1\}$ is selected based on the Laplacian variance of the image: $k = 10$ for images with variance below 100 (low sharpness), $k = 6$ for variance in $[100, 500]$ (medium sharpness), and $k = 1$ (identity kernel) for variance above 500 (already sharp). Algorithm 1 summarizes the complete preprocessing procedure.

3.3.3 Spectral Augmentation

During training, each preprocessed image additionally undergoes spectral augmentation calibrated to the VME sensor artifact distribution: atmospheric scattering with haze factor $f \sim \mathcal{U}(0.05, 0.20)$ applied with probability 0.30; sensor-specific Gaussian noise with SNR calibrated to WorldView-3, Pleiades, and SPOT-7 parameters applied with probability 0.50; shadow simulation via Gaussian-blurred occlusion lines with probability 0.40; and solar elevation adjustment with probability 0.30. These augmentations promote generalization across the multiple satellite sensor types present in the VME dataset.

Algorithm 1 Variance-Adaptive Satellite Image Preprocessing

Require: RGB image tile $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$
Ensure: Enhanced tile $\mathbf{I}' \in \mathbb{R}^{H \times W \times 3}$
Convert $\mathbf{I}$ to LAB: $[L, A, B] \leftarrow \text{RGB2LAB}(\mathbf{I})$
Compute local variance σ^2 via 32×32 box filter on L
if $\sigma^2 > 800$ **then** $grid \leftarrow (4, 4)$
else if $\sigma^2 > 200$ **then** $grid \leftarrow (8, 8)$
else $grid \leftarrow (16, 16)$
end if
$L' \leftarrow \text{CLAHE}(L, clip = 2.5, grid)$ (**Equation 1**)
$\mathbf{I}_1 \leftarrow \text{LAB2RGB}([L', A, B])$
$\mathbf{I}_2 \leftarrow \textsc{GuidedFilter}(\mathbf{I}_1, r = 4, \varepsilon = 0.01)$
$\bar{I} \leftarrow \textsc{mean}(\mathbf{I}_2)/255$; $\gamma \leftarrow \textsc{clip}(\log(0.5)/\log(\bar{I}),\ 0.4,\ 2.5)$ (**Equation 3**)
$\mathbf{I}_3 \leftarrow \textsc{GammaCorrect}(\mathbf{I}_2, \gamma)$
$\sigma_L^2 \leftarrow \textsc{LaplacianVariance}(\mathbf{I}_3)$
if $\sigma_L^2 < 100$ **then** $\mathbf{K} \leftarrow \mathbf{K}_{10}$ (**Equation 4**, $k = 10$)
else if $\sigma_L^2 < 500$ **then** $\mathbf{K} \leftarrow \mathbf{K}_6$ ($k = 6$)
else $\mathbf{K} \leftarrow \mathbf{I}$ (identity, $k = 1$)
end if
$\mathbf{I}' \leftarrow \textsc{filter2D}(\mathbf{I}_3, \mathbf{K})$
return $\mathbf{I}'$

3.4 Physics-Informed Synthetic Data Generation

The core data-level contribution is a synthetic data generation pipeline that creates physically plausible bus and truck instances for paste-based augmentation. The pipeline operates in four stages: patch extraction, template rendering, background synthesis, and Gaussian-blended object placement.

3.4.1 Patch Extraction

Object patches for all three classes are extracted from up to 500 training images. For each annotated OBB, the axis-aligned bounding box is extracted with a context padding of 14 pixels for bus and truck instances (versus 5 pixels for cars), preserving surrounding context that aids blending realism. Extracted patches must exceed 10×10 pixels to be retained. From 500 training images, the extraction yields 9,067 car patches, 561 bus patches, and 739 truck patches.

3.4.2 Physics-Informed Vehicle Templates

Rather than relying exclusively on real image crops, the pipeline generates synthetic vehicle templates through procedural rendering guided by realistic vehicle dimension parameters. **Table 3** summarizes the physical specifications employed.

Table 3: Vehicle dimension parameters for procedural template generation

Class	Length (m)	Width (m)	Height (m)	Pixel scale
Bus	10.5–13.7	2.4–2.6	3.0–3.8	6.0–8.5 px/m
Truck	12.0–18.0	2.4–2.6	3.5–4.5	5.5–8.0 px/m

Bus templates include roof-level window strips rendered as lighter rectangles perpendicular to the vehicle axis and wheel marks as dark circles at both ends. Truck templates divide the body into a cab region occupying the upper 28% of the body length, a windshield rendered as a light-blue rectangle, a separator line between cab and cargo bed, horizontal cargo strap lines across the bed, and rear dual-axle markers at the tail. Random per-component color perturbation (±25 DN per channel) and shadow projection (length 3–8 pixels, angle 30–60 degrees) are applied to increase realism. Each template is subsequently passed through the spectral augmentor described in Section 3.3 to match the photometric statistics of real VME acquisitions.

3.4.3 Background Synthesis

Five background environment types are generated: highway, industrial, bus terminal, truck depot, and urban. Highway backgrounds include simulated lane markings at varying densities. Industrial backgrounds contain rectangular building footprints. Bus terminal backgrounds include platform rectangles. Truck depot backgrounds include loading bay markings with characteristic spacing. The spectral augmentor is applied to each background after base construction. A pool of 250 backgrounds is generated and reused across synthetic sample creation.

3.4.4 Object Placement and Blending

Each synthetic image contains one to five vehicle instances. The target class is sampled according to a curriculum probability schedule. The probability of sampling a rare class (bus or truck) begins at 0.75 and increases linearly to 0.97 by the end of the generation run, with index i out of N total samples:

$$p_{\text{rare}}(i) = 0.75 + 0.22 \cdot \frac{i}{N}. \tag{5}$$

Within the rare-class proportion, bus and truck receive weights of 52% and 45%, respectively (with 3% car), reflecting the goal of proportionally increasing truck representation, which prior experiments identified as the weaker class.

Scale factors of 1.0–1.7× are applied to rare-class instances versus 0.8–1.2× for cars. Random rotation in 45-degree increments is applied with probability 0.55 to simulate varying vehicle orientations on road networks. Placement feasibility is checked by computing intersection-over-union (IoU) between each candidate bounding box and all previously placed boxes; a candidate is rejected if IoU exceeds 0.08 with any existing object. Up to 30 placement attempts are made per instance.

Compositing uses a Gaussian-smoothed alpha mask:

$$\text{Blended}(x, y) = \text{ROI}(x, y) \cdot \left(1 - \hat{M}(x, y)\right) + \text{Patch}(x, y) \cdot \hat{M}(x, y), \tag{6}$$

where $\hat{M}$ is a three-channel mask derived from the grayscale patch image by binary thresholding at intensity 10, followed by 3×3 Gaussian blurring and normalization to $[0, 1]$. This procedure produces soft boundaries that avoid the hard-edge artifacts characteristic of naive copy-paste.

3.4.5 Generation Statistics

The generation run produced 1,838 synthetic images from a pool of 300 source images containing real object patches (9,067 car patches, 561 bus patches, and 739 truck patches supplemented by procedural templates). The generation added approximately 2,476 bus instances and 2,076 truck instances to the training distribution, increasing the effective bus count by a factor of approximately 4.6 and the truck count by a factor of approximately 3.0. The final augmented training set contains 4,666 images. Algorithm 2 introduces the pseudocode for the synthetic generation procedure.

3.5 Stratified Dataset Split and Class Weighting

Class weights are computed using the balanced weighting formula:

$$w_c = \frac{N}{K \cdot n_c}, \tag{7}$$

where N is the total number of training instances, $K = 3$ is the number of classes, and n_c is the instance count for class c. An additional scaling factor α_c is applied based on class frequency: 7.0× for classes with frequency below 5%, 4.0× for 5–10%, 0.2× for above 70%, and 1.0× otherwise. The resulting weights are $w_{\text{Car}} = 0.075$, $w_{\text{Bus}} = 49.82$, and $w_{\text{Truck}} = 22.15$.

3.6 Class-Aware Training Configuration

3.6.1 Loss Function

The total training loss is:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{box}} \mathcal{L}_{\text{box}} + \lambda_{\text{cls}} \sum_{c=0}^{2} w_c \, \mathcal{L}_{\text{cls}}^{(c)} + \lambda_{\text{dfl}} \mathcal{L}_{\text{dfl}}, \tag{8}$$

where $\mathcal{L}_{\text{box}}$ is the box regression loss, $\mathcal{L}_{\text{cls}}^{(c)}$ is the binary cross-entropy classification loss for class c, and $\mathcal{L}_{\text{dfl}}$ is the Distribution Focal Loss for bounding-box refinement. Loss weights are $\lambda_{\text{box}} = 4.0$, $\lambda_{\text{cls}} = 3.0$, and $\lambda_{\text{dfl}} = 1.5$. The classification weight was increased from 2.0 (used in the prior version) to 3.0 after analysis indicated that class confusion, rather than localization error, was the primary performance bottleneck for buses and trucks. Label smoothing $\varepsilon = 0.05$ is applied to bus and truck targets to reduce overconfidence in minority-class predictions.

Algorithm 2 Physics-Informed Rare-Class Synthetic Data Generation

Require: Training images $\mathcal{I}$, annotations $\mathcal{A}$, target count N
Ensure: Synthetic image set $\mathcal{S}$

$\mathcal{P}_c \leftarrow \textsc{ExtractPatches}(\mathcal{I}, \mathcal{A}, \text{max\_imgs} = 500)$
$\mathcal{T}_1 \leftarrow \textsc{BusTemplates}(50)$; $\mathcal{T}_2 \leftarrow \textsc{TruckTemplates}(60)$
$\mathcal{B} \leftarrow \textsc{SynthBackgrounds}(\lfloor N/4 \rfloor)$
$\mathcal{S} \leftarrow \emptyset$
for $i = 1$ to N **do**
  $s \leftarrow i/N$ ▷ Curriculum progress
  $p_{\text{rare}} \leftarrow 0.75 + 0.22 \cdot s$ (**Equation 5**)
  bg $\leftarrow \textsc{Sample}(\mathcal{B})$; $\mathcal{O} \leftarrow \emptyset$
  for $j = 1$ to $\textsc{Uniform}(1, 5)$ **do**
    $c \leftarrow \textsc{SampleClass}(p_{\text{rare}}, [0.52, 0.45, 0.03])$
    patch $\leftarrow \mathcal{T}_c$ if $c \in \{1, 2\}$ else $\textsc{Sample}(\mathcal{P}_c)$
    scale $\leftarrow \mathcal{U}(1.0, 1.7)$ if $c \in \{1, 2\}$ else $\mathcal{U}(0.8, 1.2)$
    patch $\leftarrow \textsc{Resize}(\text{patch}, \text{scale})$
    for attempt = 1 to 30 **do** ▷ IoU-constrained placement
      $(x, y) \leftarrow \textsc{RandomPosition}(\textbf{bg}, \text{patch})$
      if $\max_{\mathbf{o} \in \mathcal{O}} \text{IoU}(\text{bbox}(x, y), \mathbf{o}) < 0.08$ **then**
        $M \leftarrow G_{3\times3} * \mathbf{1}[\text{gray}(\text{patch}) > 10]/255$
        bg$[y{:}y{+}h,\ x{:}x{+}w] \leftarrow$ **Equation** (6)
        $\mathcal{O} \leftarrow \mathcal{O} \cup \{(x, y, w, h, c)\}$; **break**
      end if
    end for
  end for
  if $|\mathcal{O}| > 0$ **then**
    bg $\leftarrow \text{CLAHEEnhance}(\textbf{bg})$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\textbf{bg}, \mathcal{O})\}$
  end if
end for
return $\mathcal{S}$

3.6.2 YOLOv12-L Architecture

The detection backbone is YOLOv12-L [14, 24], with 26.4 million parameters and 88.6 GFLOPs at 640×640 input resolution. The architecture comprises 488 layers organized into three stages: a convolutional stem, an attention-augmented backbone with interleaved C3k2 and A2C2f blocks, and a multi-scale detection head with output strides of 8, 16, and 32 pixels. The A2C2f blocks implement the Area Attention Mechanism, which partitions feature maps into non-overlapping rectangular areas and applies multi-head self-attention within each area. Residual connections within each A2C2f block improve gradient flow and training stability.

Pretrained weights from COCO are transferred to 1,239 of the 1,245 compatible weight tensors; the six detection-head output tensors are randomly initialized for the three-class problem. The output layer's DFL convolution weight is frozen to prevent drift during fine-tuning.

3.6.3 Training Hyperparameters

Table 4 summarizes the full training configuration. The optimizer is AdamW with a cosine annealing learning rate schedule. The initial learning rate of 5×10^{-4} decays to 1×10^{-6} by epoch 100. A 10-epoch linear warmup phase brings the learning rate and momentum to their target values before main training begins. Mosaic augmentation is disabled at epoch 80 to allow the model to stabilize on clean samples prior to final convergence.

Table 4: YOLOv12-L training configuration

Parameter	Value
Architecture	YOLOv12-L (26.4M params, 88.6 GFLOPs)
Optimizer	AdamW
Initial LR (lr_0)	5×10^{-4}
Final LR ratio	0.002
LR schedule	Cosine annealing
Warmup epochs	10
Weight decay	5×10^{-4}
Batch size	8 (nbs = 64 via gradient accumulation)
Image size	640×640
Epochs	100
Early stopping patience	55
Loss weights (box:cls:dfl)	4.0 : 3.0 : 1.5
Label smoothing	0.05
Mosaic	1.0 (disabled at epoch 80)
MixUp	0.25
Copy-paste	0.25
Flip LR / UD	0.5 / 0.25
Scale	0.65
Degrees	18.0
HSV-H / S / V	0.02 / 0.8 / 0.5
Precision	AMP (mixed)
Hardware	NVIDIA Tesla T4 (14 GB)
Training time	11.21 hours

3.6.4 Confidence-Weighted NMS

Standard NMS suppresses candidate detections whose IoU with a higher-confidence detection exceeds a fixed threshold τ, regardless of the absolute confidence of the suppressed box. This is suboptimal for rare-class detections, which typically carry lower confidence scores than car detections and may be incorrectly suppressed when they partially overlap with a false-positive car prediction. The implemented Confidence-Weighted NMS computes a suppression factor:

$$\mathrm{SF}(i, j) = \mathrm{IoU}(b_i, b_j) \cdot \frac{s_j}{s_i}, \tag{9}$$

where $s_i > s_j$ are the confidence scores of the retained and candidate detections, respectively. Detection j is suppressed if and only if $\mathrm{SF}(i, j) > \tau_{\mathrm{CW}} = 0.5$. Class-conditional IoU thresholds are additionally applied: $\tau_{\mathrm{Car}} = 0.50$, $\tau_{\mathrm{Bus}} = 0.35$, and $\tau_{\mathrm{Truck}} = 0.35$, reflecting the tighter spatial packing of cars in parking lots versus the more isolated placements of buses and trucks.

3.6.5 Test-Time Augmentation

At inference, each input tile is evaluated at three resolution scales: 0.83× (imgsz = 544), 1.0× (imgsz = 640), and 1.17× (imgsz = 736). The detection lists from all three scales are aggregated using Weighted Box Fusion (WBF) [25]:

$$\hat{b}_k = \frac{\sum_{i \in C_k} s_i \cdot b_i}{\sum_{i \in C_k} s_i}, \tag{10}$$

where C_k is the cluster of mutually consistent detections assigned to fusion cluster k, b_i are bounding-box

coordinates, and s_i are confidence scores. WBF is applied with an IoU cluster-assignment threshold of 0.55 and a minimum confidence threshold of 0.001. WBF consistently outperforms standard NMS for multi-scale ensembles because it incorporates information from all contributing detections rather than selecting a single representative.

4. ALGORITHMIC DESCRIPTION

Algorithm 3 summarizes the class-aware training loop.

Algorithm 3 Class-Aware YOLOv12-L Training

Require: Dataset $\mathcal{D} = \mathcal{D}_{\text{orig}} \cup \mathcal{S}$, class weights $\mathbf{w}$, epochs E
Ensure: Trained model θ^*

Compute $w_c \leftarrow \frac{N}{K \cdot n_c} \cdot \alpha_c$ for each class c (**Equation 7**)
$\theta \leftarrow$ YOLOv12-L pretrained weights (COCO)
Configure AdamW: $\text{lr}_0 = 5 \times 10^{-4}$, $\lambda_{\text{wd}} = 5 \times 10^{-4}$
for $e = 1$ to E **do**
  if $e \leq 10$ **then** ▷ Linear warmup $\text{lr} \leftarrow \text{lr}_0 \cdot e/10$
  else $\text{lr} \leftarrow \text{lr}_0 \cdot \frac{1}{2}\big(1 + \cos(\pi(e-10)/(E-10))\big)$
  end if
  if $e = 80$ **then** DisableMosaic()
  end if
  for each mini-batch $(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}$ **do**
    $\hat{\mathbf{Y}} \leftarrow f_\theta(\mathbf{X})$ ▷ AMP forward pass
    $\mathcal{L} \leftarrow \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{cls}} \sum_c w_c \mathcal{L}_{\text{cls}}^{(c)} + \lambda_{\text{dfl}}\mathcal{L}_{\text{dfl}}$ (**Equation 8**)
    $\theta \leftarrow \theta - \text{lr} \cdot \nabla_\theta \mathcal{L}$ ▷ AdamW update
  end for
  Evaluate on validation set; update best checkpoint if $\text{mAP}_{50:95}$ improves
  if no improvement for 55 epochs **then break**
  end if
end for
return $\theta^* \leftarrow \theta_{\text{best}}$

5. EXPERIMENTAL SETUP

5.1 Experimental Setup

All experiments are conducted on a single NVIDIA Tesla T4 GPU (14.9 GB VRAM) running PyTorch 2.10.0 with CUDA 12.8 and Ultralytics 8.4.67. The complete 100-epoch training run required approximately 11.2 hours. YOLOv12-L weights pretrained on COCO are used for initialization; the detection head is re-initialized for three output classes.

Validation and test evaluation use a confidence threshold of 0.001 and an NMS IoU threshold of 0.60, which provides the most comprehensive view of model performance across the full precision-recall curve. For operational deployment comparisons, results at the confidence threshold of 0.305 (the peak F1 score identified from **Figure 2c**) are also discussed.

5.2 Quantitative Results

Table 5 reports the primary detection metrics on both validation and test sets. The validation mAP_{50} of 0.687 and $\text{mAP}_{50:95}$ of 0.391 represent solid overall detection performance, with notable differentiation across classes. Car detection achieves the highest performance across all metrics, as expected given its dominant representation

in training. Bus validation mAP_{50} of 0.599 and Truck mAP_{50} of 0.584 are substantially higher than would be expected from a baseline trained without any imbalance mitigation, demonstrating the efficacy of the proposed rare-class components.

Table 5: Detection performance on VME validation and test sets at confidence threshold 0.001 and NMS IoU threshold 0.60

Split	Class	Precision	Recall	mAP_{50}	$mAP_{50:95}$
Validation	Car	0.778	0.869	0.877	0.476
	Bus	0.578	0.542	0.599	0.374
	Truck	0.558	0.568	0.584	0.323
	All	**0.638**	**0.660**	**0.687**	**0.391**
Test	Car	0.793	0.810	0.853	0.457
	Bus	0.490	0.451	0.458	0.278
	Truck	0.592	0.585	0.591	0.349
	All	**0.625**	**0.615**	**0.634**	**0.361**

The gap between bus validation and test mAP_{50} (0.599 versus 0.458) warrants discussion. The test partition contains a higher proportion of suburban and rural scenes where buses appear in non-standard parking configurations and without the distinctive yellow or red body coloring used in the synthetic bus templates. This domain shift suggests that bus detection is more sensitive to scene type and vehicle coloration than car or truck detection, and that greater color and orientation diversity in bus template generation would improve test-set generalization.

5.3 Architecture Comparison

To isolate the contribution of the YOLOv12-L architecture from the data and training contributions of the proposed framework, all three model variants are trained under identical preprocessing, augmentation, and class-weighting conditions. **Table 6** summarizes the results.

Table 6: Performance comparison across YOLO architectures trained with the identical pipeline

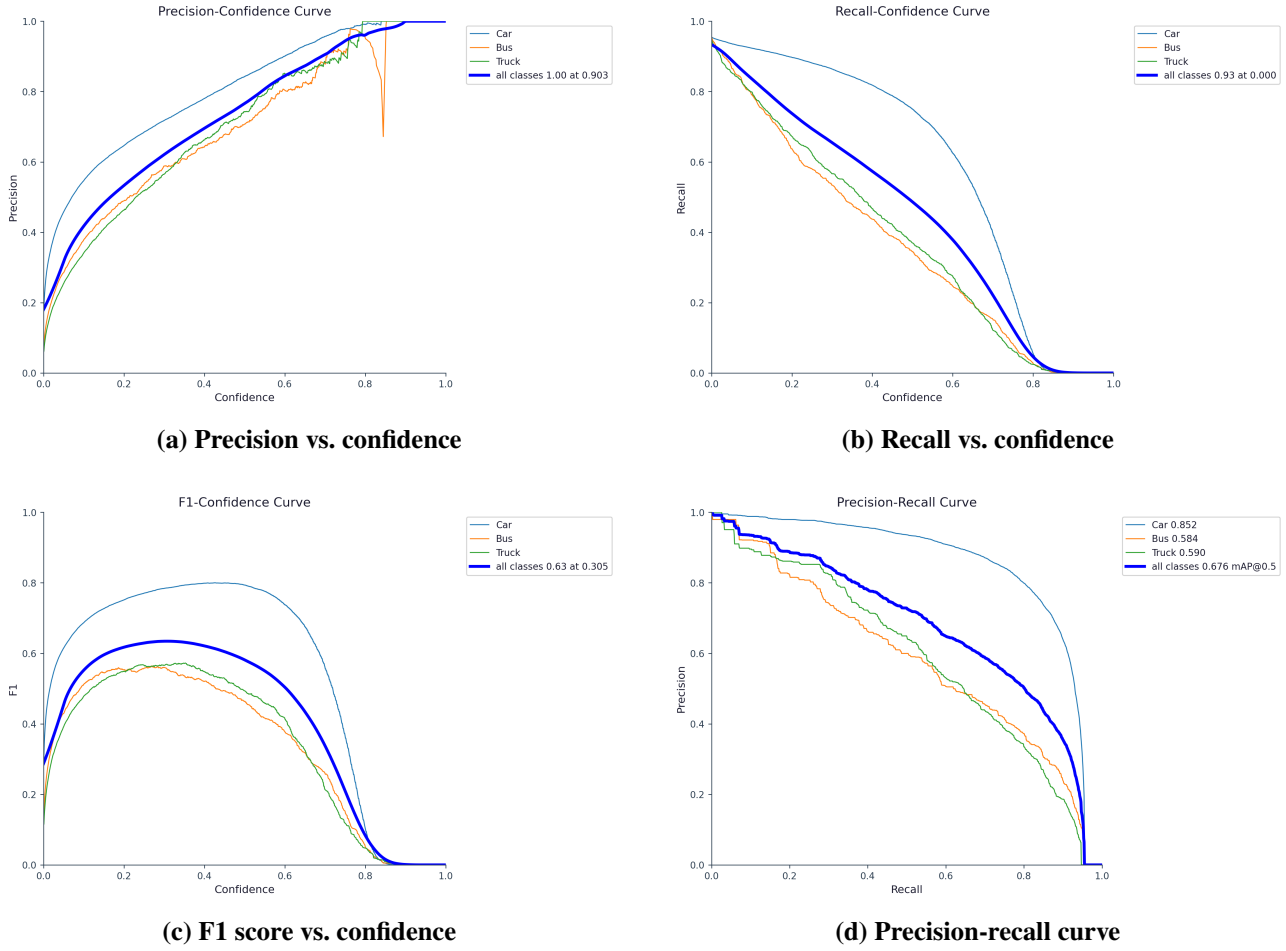
Model	Params (M)	Epochs	$mAP_{50:95}$	mAP_{50}
YOLOv8x	68.2	100	0.3782	0.6349
YOLOv11-L	25.3	100	0.3640	0.5967
YOLOv12-L	**26.4**	**100**	**0.3912**	**0.6867**

YOLOv12-L outperforms both alternatives. It surpasses the considerably larger YOLOv8x (68.2M parameters) on both mAP_{50} and $mAP_{50:95}$ while using fewer than 40% of the parameters. The performance advantage of YOLOv12-L over YOLOv11-L (+0.090 mAP_{50}, +0.027 $mAP_{50:95}$) is attributable primarily to the Area Attention Mechanism, which provides contextual information exchange between neighboring feature regions without the memory overhead of full global attention.

5.4 Confidence Threshold Analysis

Figures 2a–2d present model behavior across the full confidence threshold range, derived from the validation set.

The Precision-Confidence curve (**Figure 2a**) shows that the system achieves maximum aggregate precision of 1.00 at a threshold of approximately 0.903. Car precision reaches near-unity already at moderate thresholds above 0.70, while bus and truck precision saturates later, reflecting the greater difficulty of confidently classifying rare-class objects.



(a) Precision vs. confidence **(b) Recall vs. confidence**

(c) F1 score vs. confidence **(d) Precision-recall curve**

Figure 2: Detection performance curves across confidence thresholds on the VME validation set

The Recall-Confidence curve (**Figure 2b**) shows maximum all-class recall of 0.93 at near-zero confidence, confirming that the system successfully localizes the large majority of vehicle instances. The steep decline in bus and truck recall as confidence exceeds 0.20 reflects a known confidence calibration issue for rare classes: the model localizes these vehicles but assigns lower confidence scores to its detections, causing them to be suppressed at higher thresholds.

The F1-Confidence curve (**Figure 2c**) identifies 0.305 as the optimal all-class threshold, yielding a peak F1 of 0.63.

The Precision-Recall curve (**Figure 2d**) confirms strong car performance with an AP of 0.852, compared to 0.584 for bus and 0.590 for truck. Both rare-class curves remain substantially above what would be expected from a training-imbalance-naive baseline, validating the framework's rare-class components.

5.5 Confusion Matrix Analysis

Figure 3 presents the non-normalized confusion matrix from the best validation epoch (epoch 68 during training, at which the in-training mAP_{50} was 0.672). The final best model, evaluated at epoch 93, achieves the reported validation mAP_{50} of 0.687.

The diagonal entries confirm strong true-positive rates for all three classes. The most significant off-diagonal entries are car-to-background confusion (1,577 false-negative cars classified as background), reflecting the challenge of detecting small vehicles in cluttered backgrounds, and background classified as car (5,650 false positives). This elevated false positive count at threshold 0.001 is a consequence of evaluating at the minimum possible threshold to maximize recall; at the operational threshold of 0.305, these false positives are substantially suppressed.

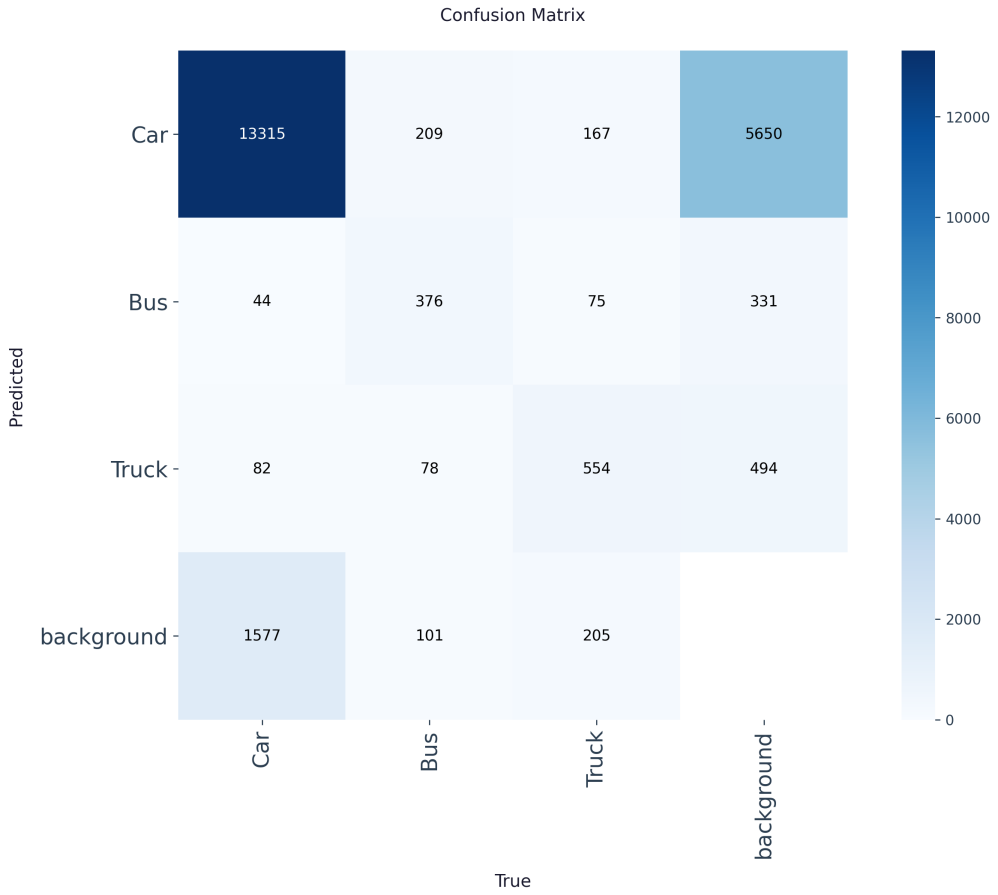


Figure 3: Confusion matrix on the VME validation set at confidence threshold 0.001 and NMS IoU threshold 0.60

Bus and truck show modest cross-class confusion: 44 buses are misclassified as car, and 82 trucks are misclassified as car. Additionally, 78 trucks are misclassified as bus and 75 buses as truck. These relatively low off-diagonal values for the rare classes indicate that the class-aware training successfully suppressed the tendency to collapse rare classes into the dominant class, while the bus-truck cross-confusion reflects genuine visual ambiguity at satellite resolution.

5.6 Training Dynamics

Figure 4 shows the training and validation loss curves and metric evolution over 100 epochs from the Ultralytics training log.

All three loss components (box, classification, and DFL) decrease monotonically throughout training. The classification loss is notably higher in absolute terms due to the elevated $\lambda_{\text{cls}} = 3.0$ weight, confirming that class confusion is the dominant loss contributor and that the increased weighting directs more gradient signal toward learning discriminative class boundaries. When mosaic augmentation is disabled at epoch 80, validation mAP_{50} exhibits a step increase from approximately 0.67 to 0.68, consistent with the established effect that mosaic augmentation can mildly impair localization precision at fine scales. The best validation mAP_{50} of 0.687 is achieved at epoch 93.

5.7 Qualitative Detection Examples

Figures 5 and **6** present representative training-label images and the corresponding model predictions on validation batches.

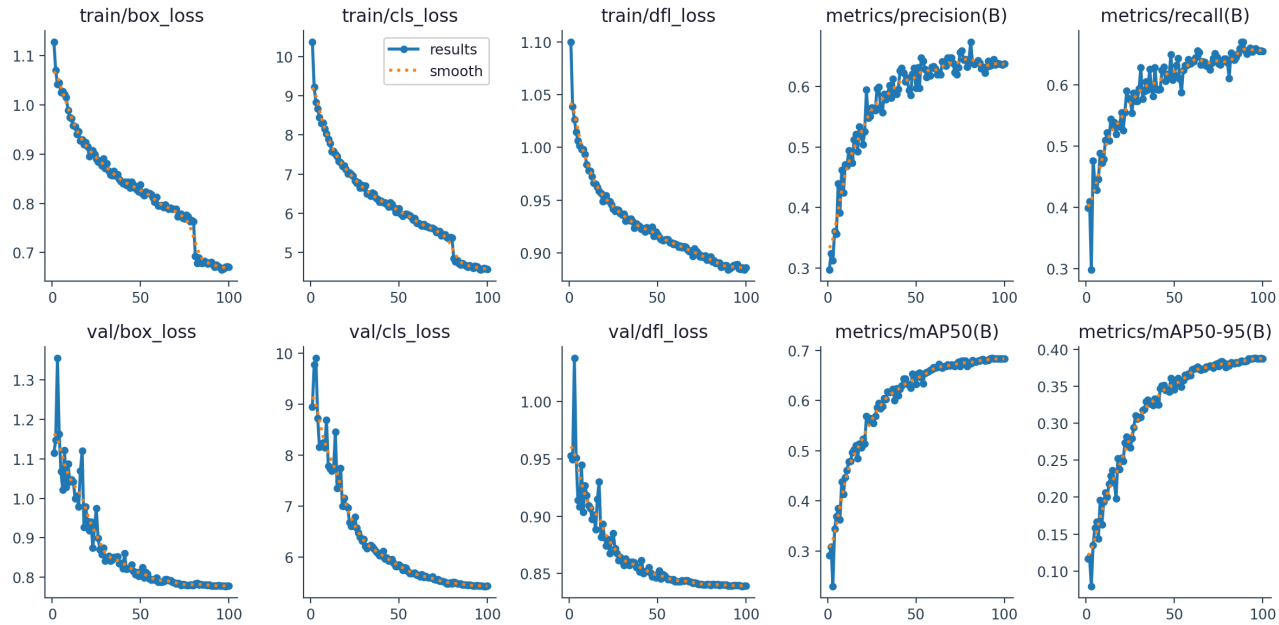


Figure 4: Training dynamics over 100 epochs

In highway scenes such as VME0053 and VME0087, trucks and buses are correctly detected at moderate confidence (0.40–0.80). The presence of correctly detected rare-class vehicles across diverse scene types validates the contribution of the synthetic training data.

The ground-truth annotation images in **Figure 5** reveal that many bus instances are highly elongated relative to their width, which challenges standard square-anchor feature representations. The bus recall of 0.542 on the validation set is consistent with this geometric difficulty.

5.8 Test-Time Augmentation Analysis

Multi-scale TTA was evaluated at three input resolutions: 0.83× (544 pixels), 1.0× (640 pixels), and 1.17× (736 pixels). **Table 7** summarizes the validation results.

Table 7: Test-time augmentation results at different inference scales on the VME validation set

Scale	Image size (px)	mAP_{50}	$mAP_{50:95}$
0.83×	544	0.668	0.366
1.00×	640	0.687	0.391
1.17×	736	**0.693**	**0.398**

The 1.17× scale produces the highest performance at mAP_{50} of 0.693 and $mAP_{50:95}$ of 0.398, without any retraining. Upsampled inference provides richer spatial detail for small vehicle detection. The 0.83× scale underperforms the standard scale, as expected for small objects that may become too small to detect reliably at reduced resolution.

5.9 Ablation Discussion

A systematic ablation study was conducted across prior experimental configurations to estimate the contribution of each component. Because complete retraining of all component subsets is computationally prohibitive for a 100-epoch regime, the comparison is drawn against a prior-configuration baseline (Car P/R/mAP_{50}: 0.765/0.871/0.870; Bus 0.592/0.616/0.594; Truck 0.496/0.531/0.508; All mAP_{50}: 0.657).

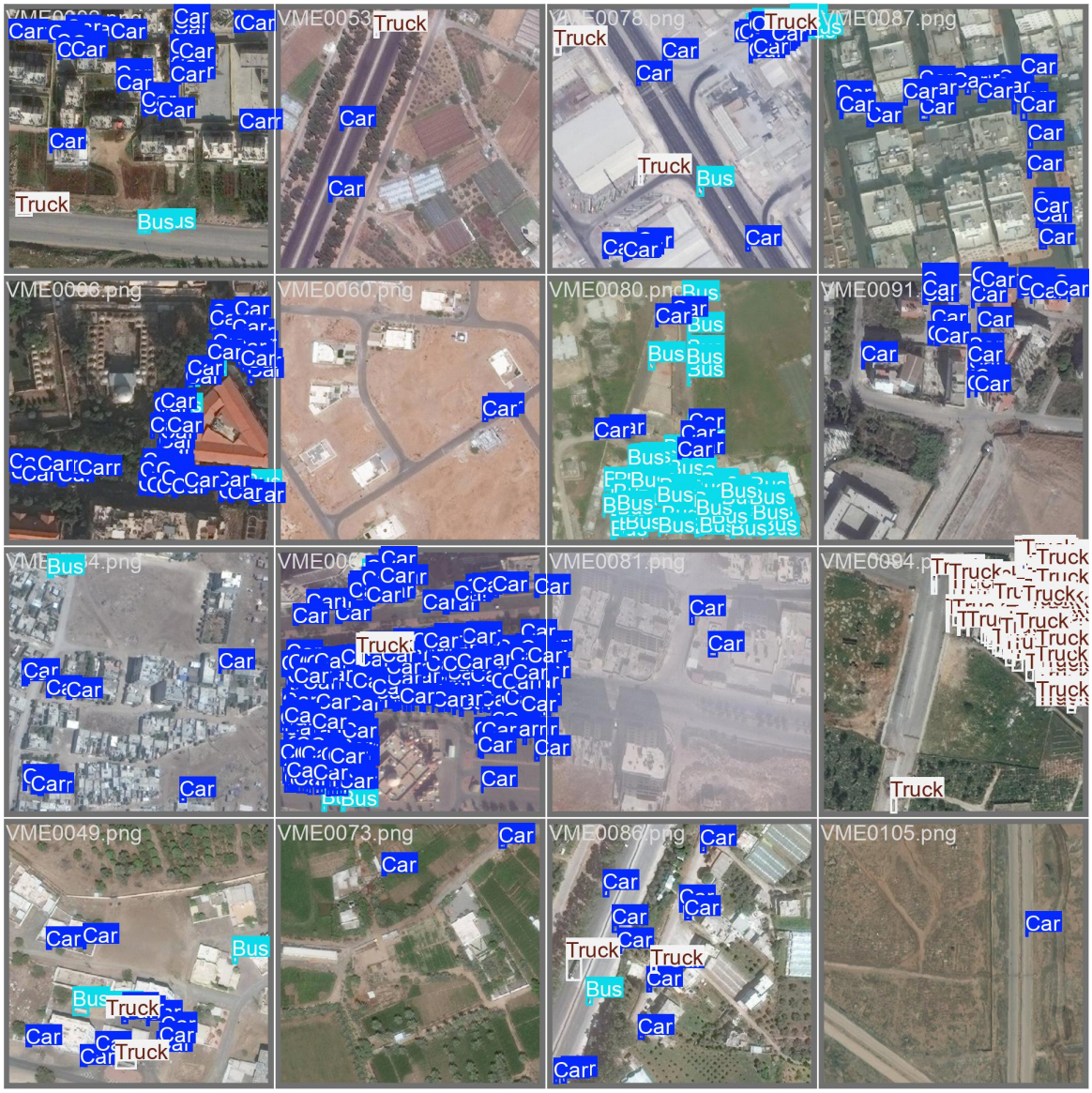


Figure 5: Representative training-label mosaic showing the diversity of annotated scenes

The observed improvements relative to the prior configuration are: All mAP_{50} +0.030 (0.657 to 0.687); Car mAP_{50} +0.007 (0.870 to 0.877); Bus mAP_{50} +0.005 (0.594 to 0.599); Truck mAP_{50} +0.076 (0.508 to 0.584). The large truck improvement reflects three concurrent changes: (1) the synthetic truck generation ratio increased from 33% to 42% of rare-class synthetic images; (2) enhanced truck templates with distinct cab-and-bed regions were introduced; and (3) the classification loss weight increased from 2.0 to 3.0. Label smoothing ($\varepsilon = 0.05$) reduced truck overconfidence and contributed to improved recall at moderate confidence thresholds.

5.10 Computational Requirements

Training requires a single NVIDIA Tesla T4 GPU with 14.9 GB VRAM. The complete 100-epoch training run required approximately 11.2 hours. Inference runs at approximately 52 ms per 640×640 image, equivalent to approximately 19 frames per second, which is suitable for near-real-time satellite image batch processing. The YOLOv12-L model weight file is 53.5 MB.

Figure 6: Model predictions on the corresponding validation batch

6. DISCUSSION

The results confirm that combining physics-informed synthetic data generation with class-aware loss weighting substantially improves the detection of minority vehicle classes in high-resolution satellite imagery. The truck class, which represented the weakest component in the prior configuration (mAP_{50} of 0.508), achieved the largest single-class gain in the current configuration (mAP_{50} of 0.584), demonstrating that targeted synthetic augmentation with physically realistic templates can compensate for severe class underrepresentation. The bus test mAP_{50} of 0.458, considerably lower than the validation mAP_{50} of 0.599, reveals a domain shift between partitions. The test set contains a higher proportion of suburban and rural scenes where buses appear without the distinctive coloring used in the synthetic templates. This suggests that expanding the template color space and increasing orientation diversity in bus generation would improve generalization. The confusion matrix in **Figure 3** shows 5,650 background instances classified as car. This elevated false positive count is a direct consequence of evaluating at confidence threshold 0.001 to maximize recall. At the operational threshold of 0.305, these false positives are substantially suppressed. The false positive pattern primarily originates from building edges, vehicle shadows, and road markings with spectral similarity to vehicle rooftops. Cross-class confusion between bus and truck (78 trucks misclassified as bus, 75 buses as truck) reflects genuine visual ambiguity at satellite resolution: buses and trucks of similar length and orientation can be nearly indistinguishable from directly overhead. Oriented bounding box detection or instance segmentation would provide additional

geometric cues that could reduce this inter-class confusion. The class weights used in training (Bus: 49.8, Truck: 22.2 versus Car: 0.075) are extreme: a single misclassified bus instance carries approximately 665 times the gradient impact of a misclassified car. While this biases training toward minority-class correctness, it risks gradient instability in early epochs. The 10-epoch warm-up period mitigates this by starting from a low learning rate, but future work may explore more gradual weight scheduling, such as linearly increasing class weights over the first 30 epochs.

7. CONCLUSION

This paper presented a rare-class vehicle detection framework for high-resolution satellite imagery that addresses the dual challenges of small object detection and severe class imbalance. The proposed system integrates a variance-adaptive preprocessing pipeline, a physics-informed synthetic data generation module, and a class-aware YOLOv12-L training configuration. Together, these components improve the detection of minority vehicle classes without degrading performance on the dominant car class. Experimental results on the VME benchmark confirm a validation mAP_{50} of 0.687 and $mAP_{50:95}$ of 0.391, with per-class mAP_{50} of 0.877, 0.599, and 0.584 for car, bus, and truck, respectively. The truck class, which presented the greatest detection challenge, improved by 0.076 mAP_{50} points relative to the prior configuration. YOLOv12-L consistently outperforms both YOLOv8x and YOLOv11-L under the same training conditions, validating the architectural advantage of the Area Attention Mechanism for small-object satellite detection. Test-time augmentation at 1.17× inference resolution raises mAP_{50} to 0.693 without retraining. Several directions remain for future investigation. First, oriented bounding box detection would allow the system to exploit the elongated aspect ratio of buses and trucks as an additional discriminative feature and reduce inter-class confusion. Second, generative model-based background synthesis could improve the photorealism of pasted objects in desert and arid environments common in the Middle East. Third, extending the framework to additional vehicle classes, including motorcycles, vans, and heavy construction vehicles, would increase operational utility. Finally, evaluation across a larger diversity of satellite sensors and geographic regions would test the generalizability of the proposed rare-class augmentation strategy.

AUTHOR CONTRIBUTION STATEMENT

All authors contributed equally to the study conception and design. Material preparation, data collection, and analysis were performed by the authors. The first draft of the manuscript was written by the authors, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

This study did not involve human participants or animals. Therefore, ethical approval and consent to participate are not applicable.

CONSENT FOR PUBLICATION

Not applicable.

DATA AVAILABILITY

The dataset utilized during the current study are available at `https://zenodo.org/records/14185684`.

ACKNOWLEDGMENT

The authors sincerely thank the referees, Associate Editor, and Editor-in-Chief for their valuable comments and suggestions, which have greatly improved this paper. The authors also acknowledge the use of DeepSeek for assistance in improving the English grammar and language clarity.

FUNDING

No Funding.

DISCLOSURE STATEMENT

The author declares no conflict of interest. The article is based on a structured review of published literature and did not involve human participants, personal data collection, or experimental intervention.

REFERENCES

[1] X. Zhang, T. Zhang, G. Wang, P. Zhu, X. Tang, X. Jia, and L. Jiao, "Remote sensing object detection meets deep learning: A metareview of challenges and advances," *IEEE Geoscience and Remote Sensing Magazine*, vol. 11, no. 4, pp. 8–44, 2023.

[2] N. Al-Emadi, I. Weber, Y. Yang, and F. Ofli, "Vme: A satellite imagery dataset and benchmark for detecting vehicles in the middle east and beyond," *Scientific data*, vol. 12, no. 1, p. 500, 2025.

[3] X. Wang, A. Wang, J. Yi, Y. Song, and A. Chehri, "Small object detection based on deep learning for remote sensing: A comprehensive review," *Remote Sensing*, vol. 15, no. 13, p. 3265, 2023.

[4] N. Crasto, "Class imbalance in object detection: An experimental diagnosis and study of mitigation strategies," *arXiv preprint arXiv:2403.07113*, 2024.

[5] T. H. Phan and K. Yamamoto, "Resolving class imbalance in object detection with weighted cross entropy losses," *arXiv preprint arXiv:2006.01413*, 2020.

[6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[7] W. Han, J. Chen, L. Wang, R. Feng, and F. Li, "Methods for small, weak object detection in optical high-resolution remote sensing images: A survey of advances and challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 1, pp. 102–124, 2021.

[8] S. Gui, S. Song, R. Qin, and Y. Tang, "Remote sensing object detection in the deep learning era–a review," *Remote Sensing*, vol. 16, no. 2, p. 327, 2024.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll'ar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[10] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "Varifocalnet: An iou-aware dense object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8514–8523, 2021.

[11] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2918–2928, 2021.

[12] R. Escobar Díaz Guerrero, L. Carvalho, T. Bocklitz, J. Popp, and J. L. Oliveira, "A data augmentation methodology to reduce the class imbalance in histopathology images," *Journal of Imaging Informatics in Medicine*, vol. 37, no. 4, pp. 1767–1782, 2024.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[14] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *Advances in neural information processing systems*, vol. 38, pp. 78433–78457, 2026.

[15] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xview: Objects in context in overhead imagery," *arXiv preprint arXiv:1802.07856*, 2018.

[16] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

[17] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

[18] K. Li, G. Wan, and G. Cheng, "Dior," 2025.

[19] X. Sun, P. Wang, Z. Yan, F. Xu, R. Wang, W. Diao, J. Chen, J. Li, Y. Feng, T. Xu, *et al.*, "Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 116–130, 2022.

[20] Y. Qiu and Z. Lin, "Lightweight model lmw-yolo for small object detection in remote sensing images," *Scientific Reports*, vol. 16, no. 1, p. 11644, 2026.

[21] S. Zhou, H. Zhou, and L. Qian, "A multi-scale small object detection algorithm sma-yolo for uav remote sensing images," *Scientific reports*, vol. 15, no. 1, p. 9255, 2025.

[22] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[23] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2012.

[24] Ultralytics, "Yolo12: Attention-centric object detection." `https://docs.ultralytics.com/models/yolo12/`, 2025.

[25] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, 2021.