YOLOv8-FDE: An Enhanced Real-Time Vehicle Detection Method Based on YOLOv8-FDD

Priyadarshi Nihal, Pratyush Dubey, Dattatrey, Dev Tailor, Shivalik Mathur
Students, Capstone Group 24
VIT Bhopal University
Madhya Pradesh, India

Dr. I. Jasmine Selvakumari Jeya
Asst. Dean Academics
VIT Bhopal University
Madhya Pradesh, India

***Abstract*—Real-time vehicle detection is a crucial task for applications ranging from intelligent transportation systems (ITS) to autonomous driving. While the YOLO family of models, particularly YOLOv8, offers a strong baseline, its parameter size can restrict deployment on edge devices. We propose YOLOv8-FDE (Feature Dynamic Enhanced), a novel architecture that integrates feature enhancement techniques from YOLOv8-FDD with several strategic model slimming and attention mechanisms. Our method incorporates FDIDH (Feature Dynamic Interaction Detection Head), DySample (Dynamic Sampling), and DWR (Dilation-Wise Residual) to enhance feature representation and decoding. Crucially, we introduce two key architectural modifications: replacing the first two C2f blocks in the backbone with the parameter-efficient C3K2 blocks and integrating an Attention block immediately following the SPPF layer. These changes, coupled with the removal of redundant parameters, result in an architecture with only 2.69 million parameters and 3.50 GFLOPs—a significant reduction compared to YOLOv8. Evaluated on a portion of the UA-DETRAC dataset, YOLOv8-FDE achieves a mAP_{50} of 0.9242 and mAP_{50-95} of 0.8159, demonstrating superior efficiency and reduced training loss while outperforming both the baseline and competitive models. This work establishes a new standard for resource-optimized, high-accuracy vehicle detection.**

***Index Terms*—Vehicle Detection, Real-Time Object Detection, YOLOv8, Model Compression, Deep Learning, Attention Mechanism, Embedded Systems**

I. Introduction

Object detection forms the core foundation of modern computer vision, with its relevance rapidly accelerating in domains like intelligent transportation systems (ITS), autonomous vehicle navigation, and high-density traffic monitoring. The success of these systems hinges upon highly accurate and, critically, extremely fast detection capabilities. Among the various detection architectures, the **You Only Look Once (YOLO)** series remains the industry benchmark for achieving a powerful trade-off between speed and accuracy through its single-stage detection approach. Recent advancements, such as the introduction of anchor-free methods, decoupled heads, and multi-scale feature fusion networks, have further solidified the dominance of the YOLO family.

YOLOv8, the latest major iteration, provides a versatile and robust framework characterized by its decoupled head and the high-performance C2f module. This module, based on the Cross-Stage Partial (CSP) strategy, significantly enhances feature reuse and gradient flow by splitting the feature map into two routes: one passing through a series of bottleneck modules, and the other serving as a shortcut, which are finally concatenated. While powerful, the baseline YOLOv8-S (small) model, which serves as the foundation for many derivative works, still possesses a parameter count and computational complexity (GFLOPs) that present a deployment bottleneck on resource-constrained embedded systems, such as those typically found in roadside units or low-power vehicle hardware. Achieving real-time performance on these platforms requires models to be not just accurate, but aggressively streamlined. The thermal constraints and latency requirements of edge hardware demand a significant reduction in model GFLOPs, particularly in the backbone layers where the spatial resolution of feature maps is highest.

The pursuit of greater efficiency led to works like YOLOv8-FDD [1], which introduced structural changes focused on efficient feature extraction and a novel head design to improve performance under varying conditions. The FDD approach highlights the importance of dynamic feature interaction and adaptive sampling for achieving high detection precision. Our research, **YOLOv8-FDE (Fast, Dependable, and Efficient)**, builds upon these concepts with a strategic focus on surgical architectural compression and feature enhancement to create a truly lightweight, high-performance vehicle detector. We acknowledge the advances of FDD but recognize the necessity of making them computationally feasible for edge deployment by focusing the majority of our parameter reduction efforts on the backbone.

Our primary motivation is the necessity for a vehicle detection model that is not only robustly accurate across diverse conditions but is also substantially more lightweight and computationally inexpensive than its predecessors. We achieve this by:

1) Integrating proven structural improvements for precision

(**FDIDH**, **DySample**, **DWR**) into the detection neck and head.
2) Introducing parameter-efficient **C3K2 blocks** to replace heavier C2f blocks in the backbone, directly addressing model size. This is critical as early backbone layers process high-resolution feature maps, leading to a direct reduction in GFLOPs.
3) Strategically incorporating a **Self-Attention block** post-SPPF layer to refine the most global features, boosting accuracy despite the model slimming. This attention mechanism acts as a targeted feature compensation mechanism, ensuring semantic richness is preserved.

We demonstrate that this novel combination results in a model with significantly reduced computational cost (down to 3.50 GFLOPs) and a smaller parameter count (2.69M), while simultaneously achieving superior mean Average Precision (mAP_{50-95}) compared to the YOLOv8-S baseline. This work contributes a highly practical and efficient solution for real-time applications where every parameter and computation counts.

II. Literature Review

A. Evolution of YOLO Architectures and Lightweight Design

The core strength of the YOLO series lies in its single-stage object detection, simultaneously predicting bounding boxes and class probabilities across the image. **YOLOv8** advanced this by adopting an anchor-free approach and introducing the **C2f** module, which improved gradient flow and feature reuse by building upon the structure of the C3 module from YOLOv5. The C2f block's efficiency stems from the Cross-Stage Partial (CSP) network concept [4], which ensures that complex transformations occur on only a subset of the feature map channels, reducing redundancy and computational cost. The CSP strategy minimizes the computational bottleneck by allowing only half of the feature map to undergo a series of convolutional operations, with the other half serving as a direct connection for efficient feature reuse.

YOLOv8-FDD [1] represents a direction focused on high-precision enhancement. It utilizes the **Feature Dynamic Interaction Detection Head (FDIDH)** to explicitly model the relationship between classification and regression features, and the **Dilation-Wise Residual (DWR)** module to expand the receptive field without increasing parameter count. These elements, while powerful, contributed to a high computational load in the original FDD implementation, underscoring the necessity for complementary compression strategies. The challenge lies in integrating these performance-boosting features without incurring the associated high GFLOP cost.

B. Dynamic Convolution and Adaptive Sampling

A major trend in modern object detection is the move toward **dynamic mechanisms** that allow the network to adapt its operations based on the input features, rather than relying on static kernel weights.

1) Deformable Convolutional Networks (DCN): Used within the regression branch of the FDIDH, DCN is a core component that addresses the geometric variations of objects. Unlike standard convolution, which samples features at fixed locations within a rectangular grid, DCN augments the spatial sampling locations with learned offsets. This allows the convolution kernel to freely deform, adapting to the scale, pose, and non-rigid shape of the detected object. This is particularly valuable for vehicles, which can appear highly distorted due to perspective changes or partial occlusion in traffic scenes. The adaptive nature of DCN directly contributes to more accurate bounding box regression by focusing feature extraction precisely on the object's boundaries, as formalized by the DCN equation:

$$\mathbf{y}(\mathbf{p}) = \sum_k \mathbf{w}(k) \cdot \mathbf{x}(\mathbf{p} + \mathbf{p}_k + \Delta\mathbf{p}_k) \cdot \mathbf{m}_k$$

where $\Delta\mathbf{p}_k$ is the learned, feature-dependent offset, and $\mathbf{m}_k$ is the learned modulation mask for each sampling point.

2) DySample for Dynamic Upsampling: The DySample module, which replaces the simpler nearest-neighbor upsampling in the feature pyramid network, is another form of adaptive sampling. Upsampling is crucial for generating high-resolution feature maps necessary for detecting small objects (like distant vehicles). By learning positional biases and using bilinear interpolation with offset constraints, DySample intelligently determines the optimal sampling points during the upsampling process. This contrasts with fixed upsampling methods and results in finer-grained spatial information, leading to better localization accuracy (mAP_{50-95}) for all object sizes. The use of grouped dynamic sampling ensures this benefit is achieved with low computational overhead.

C. Attention Mechanisms and Feature Refinement

The integration of **Attention mechanisms** is standard practice in modern high-performance CNNs, allowing the model to learn which features are most relevant. Traditional attention blocks like Squeeze-and-Excitation (SE) [7] and Convolutional Block Attention Module (CBAM) [6] focus on channel and spatial weighting, respectively. Our design uses a generic Self-Attention mechanism, similar to the Partial Spatial Attention (PSA) module, but placed at a global context level.

The role of the **SPPF (Spatial Pyramid Pooling Fast)** layer [5] is to aggregate multi-scale contextual information. Placing our attention block immediately *after* the SPPF layer ensures that the global, contextually-rich features are filtered and optimally weighted *before* the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) layers begin upsampling and fusion. This strategic positioning maximizes the impact of the attention mechanism on the final detection quality.

D. Multi-Scale Feature Aggregation via Dilation

The **Dilation-Wise Residual (DWR)** module is a specialized component focused on expanding the receptive field

efficiently. Standard convolutional layers have a fixed, local receptive field, which limits the network's ability to capture large-scale context essential for disambiguating objects in dense scenes. Dilated convolutions, by introducing a skip rate in the kernel, allow the network to capture information from a much wider area without increasing the number of parameters or computation required. DWR combines local (Region Residualization) and global (Semantic Residualization via dilated convolutions) context effectively, resulting in a richer, multi-scale feature representation that is robust to scale variations typical in vehicle detection.

III. Methodology

The YOLOv8-FDE architecture represents a hybrid approach, strategically adopting the validated, performance-enhancing structures from YOLOv8-FDD while introducing novel, efficiency-focused modifications to the feature extraction backbone. This ensures performance enhancements are achieved without the computational burden associated with the original FDD proposal.

A. Comprehensive Architecture Overview

The overall structure of YOLOv8-FDE follows the traditional three-part design: a **Backbone** for hierarchical feature extraction, a **Neck (FPN/PAN)** for feature fusion across scales, and a **Decoupled Head** for classification and regression prediction. The backbone extracts features at three scales (P3, P4, P5), which are then fused in the neck to produce multi-scale feature maps for the head.

Our key innovations are concentrated in the Backbone: specifically, the replacement of the initial C2f blocks with C3K2 and the insertion of an Attention mechanism post-SPPF. The final architecture is designed to yield a minimal GFLOP count with maximum information retention.

The baseline YOLOv8 architecture (Figure 1) serves as the foundation.

The proposed YOLOv8-FDE architecture, incorporating all modifications, is illustrated in Figure 2.

—

B. Backbone Efficiency via Custom CSP-based Modules

The imperative for high efficiency on edge devices demands aggressive parameter reduction, particularly in the early stages of the backbone where large feature maps lead to high computational cost. To this end, we introduce the C3K and C3K2 blocks.

1) The C3K Block: The C3K block is a variant of the CSP architecture, designed for streamlined feature processing and parameter reduction. It contains a similar structure to the C2f block but crucially, **no splitting is done** at the start of the process, simplifying the data flow and saving operations. The input is passed through a Conv block, followed by a series of 'n' BottleNeck layers. The output of the BottleNeck series is then concatenated with the original path before the final Conv block. This configuration prioritizes parameter reduction and faster inference by minimizing complex data routing while still retaining the benefit of feature reuse through concatenation.

2) The C3K2 Block (Aggregated Compression): The C3K2 block is an aggregate structure built upon the efficiency of the C3K module, and it is specifically deployed to replace the first two computationally intensive C2f blocks. The structure is defined as follows: it has two Conv blocks at the start and end. The input is processed by the first Conv block, followed by a series of C3K blocks. The output of the initial Conv block and the output of the last C3K block are concatenated and passed through the final Conv block. This complex yet efficient configuration maintains the core CSP philosophy of enhancing feature propagation while aggressively reducing the number of intermediate parameters and GFLOPs at the highest resolution layers of the backbone.

C. Neck and Head Enhancements

1) Refined Feature Fusion: The C2PSA Block: In the neck structure, where multi-scale features are fused, maintaining rich spatial context is crucial for localization. The C2PSA block is introduced as a highly parameter-efficient replacement for C2f in certain neck layers. The C2PSA block uses two PSA (Partial Spatial Attention) modules. These PSA modules operate on separate, parallel branches of the feature map, similar to the split path in the C2f block structure, but with spatial attention applied before concatenation. The outputs are later concatenated. This setup ensures the model focuses on spatial information selectively, balancing computational cost and detection accuracy. The C2PSA block refines the model's ability to selectively focus on regions of interest by applying spatial attention over the extracted features, providing a lightweight alternative to general self-attention.

2) Guided Feature Refinement (Post-SPPF Attention Blocks): An **Attention Block** was strategically inserted immediately after the SPPF (Spatial Pyramid Pooling Fast) layer. The SPPF layer is critical as it aggregates features at multiple fixed scales, effectively generating a highly compressed, global, and contextually rich feature map.

Applying an attention mechanism at this specific juncture is crucial for two reasons:

1) **Prioritized Features:** The model is forced to weigh the importance of various feature channels and spatial locations within the most globally informed map before features are sent to the FPN/PAN neck. This channel and spatial refinement enhances the most abstract, high-level features.
2) **Accuracy Preservation:** This focused feature refinement compensates for any minor feature richness lost during the slimming of the C2f blocks to C3K2 blocks. The Attention Block directs the model's focus to the most salient information, leading to superior feature refinement and a boosted mAP. It serves as a quality-control gate for the features entering the feature fusion neck, ensuring that only the most relevant context is propagated.

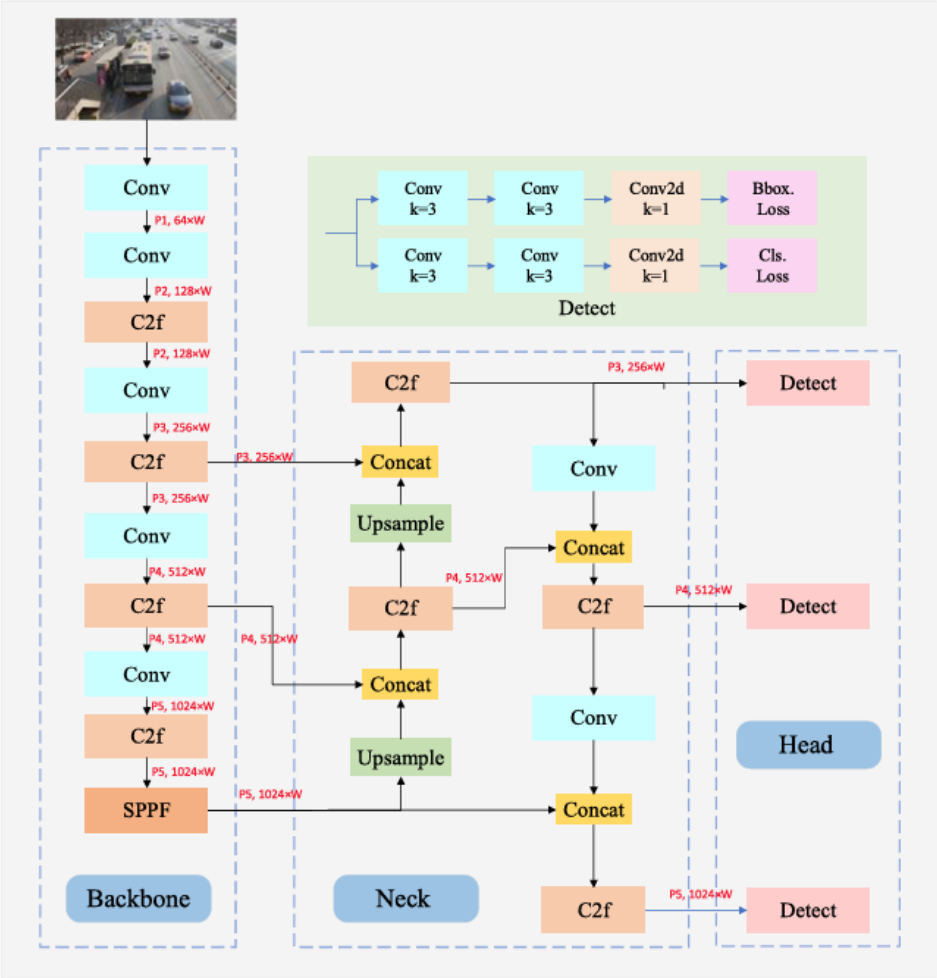

Fig. 1. Baseline YOLOv8 Architecture. It consists of a backbone utilizing C2f modules, an SPPF layer, a Neck structure, and a Decoupled Head.

D. Integration of Decoupled Head Components

To ensure high accuracy and robust prediction, especially in the context of a highly compressed backbone, we incorporated the following three components from the YOLOv8-FDD architecture into our YOLOv8-FDE head structure.

1) Feature Sharing Detection Head (FSDH): The FSDH is a foundational element. It replaces YOLOv8's three separate detection heads with a shared feature extractor. This is achieved by using two shared 3×3 convolutions and shared 1×1 convolutions for the subsequent classification and regression branches. Crucially, it employs **GroupNorm (GN)** instead of the standard BatchNorm (BN) to improve batch-size stability and accuracy, particularly useful for smaller batch sizes common in embedded training environments. Unlike BN which relies on global batch statistics, GN normalizes across groups of channels, making it robust to small batch sizes. A key refinement is the addition of a **learnable scaling layer** in the regression branch ($\mathbf{x}' = \mathbf{x} \cdot \mathbf{scale}$), which allows the model to dynamically adjust the magnitude of regression features, thereby improving bounding box prediction quality.

2) FDIDH (Feature Dynamic Interaction Detection Head): The FDIDH approach is retained and builds directly on FSDH by introducing explicit feature interaction between the classification and regression branches, essential for accurate single-stage detection.

- **Interactable Feature Extractor:** This uses Residual 3×3 convolutions to generate robust, pre-interaction features.
- **Feature Dynamic Interactor (FDI) modules:** The regression branch uses **Deformable Convolution (DCN)** for adaptive sampling of features around potential object boundaries, as detailed in Section II-B-1.
- The classification branch uses a **Feature Dynamic Filter** for adaptive weighting of the regression features: $\mathbf{x}' = \sigma(\text{Conv}_2(\text{ReLU}(\text{Conv}_1(\mathbf{x}))))$. This filter generates task-specific weights that modulate the feature maps, dynamically linking the two branches and ensuring that the classification score is informed by the precise boundary information captured by the regression branch.
- **Layer Attention Mechanism:** This final mechanism segments task-specific features, ensuring clear boundaries between classification and regression prediction paths

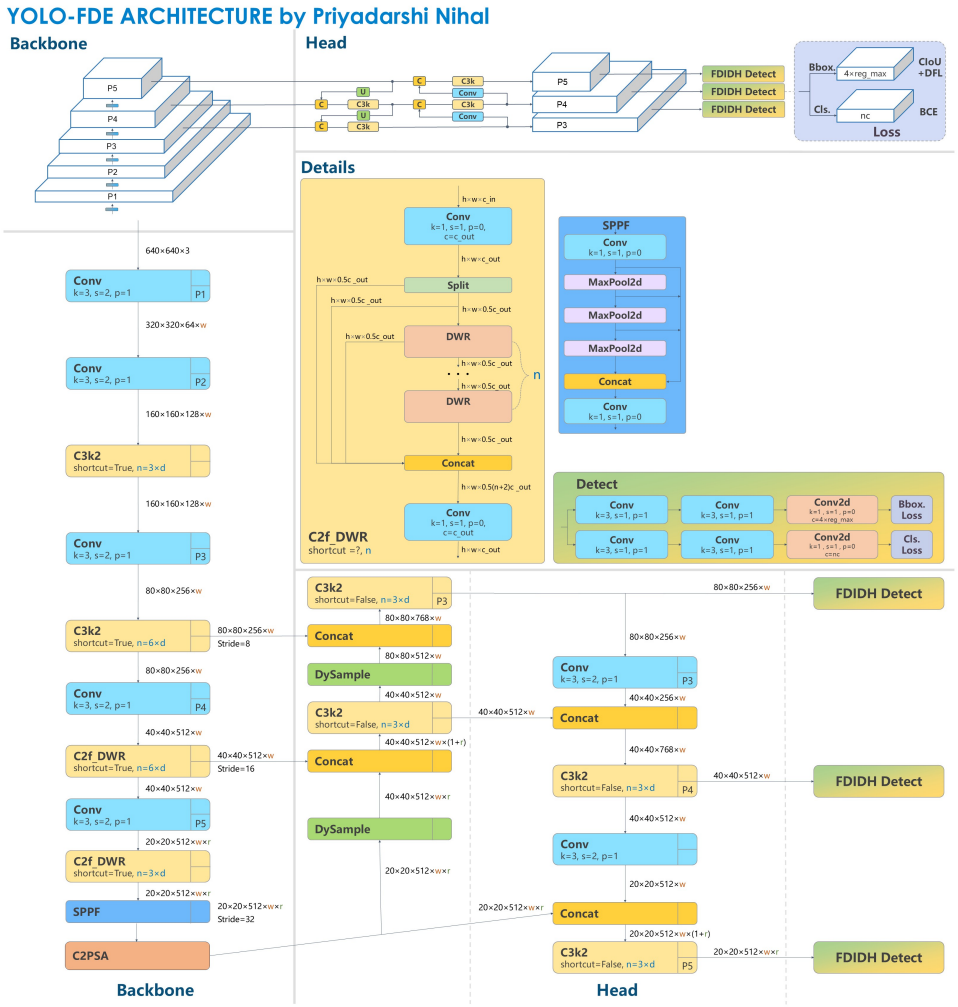

Fig. 2. Proposed YOLOv8-FDE Architecture, featuring C3K2 blocks in the initial backbone layers and an Attention Block after the SPPF layer.

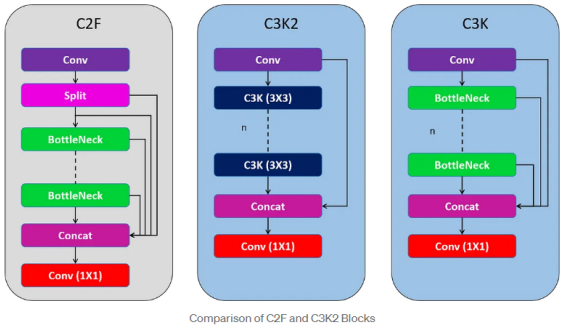

Fig. 3. Architectural diagram of the C3K2 Block, highlighting the smaller kernel size and modified bottleneck structure used for parameter reduction compared to C2f.

while maintaining a dynamic link.

3) DySample (Dynamic Sampling) Upsampling Module: Accurate localization requires precise selection of anchor or sampling points. The DySample mechanism dynamically adjusts the distribution of sampling points based on the

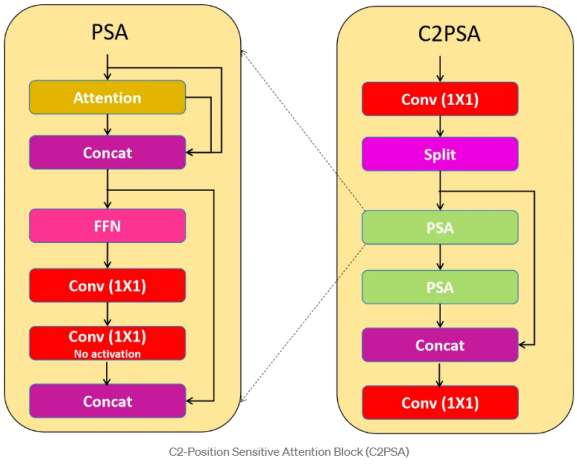

Fig. 4. The C2PSA block, employing two Partial Spatial Attention (PSA) modules on parallel branches for spatial refinement and efficiency.

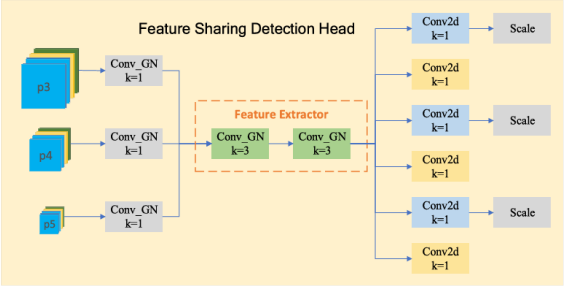

Fig. 5. The FDIDH head structure, designed to separate classification and regression information paths while enhancing feature interaction.

features, allowing the model to focus on boundary regions and challenging cases more effectively, thereby contributing to lower regression loss and higher mAP_{50-95} [8].

- **Learned Positional Biases:** It replaces nearest-neighbor upsampling by learning positional biases for sampling points via a linear layer.
- **Bilinear Interpolation with Offset Constraints:** The sampled features are interpolated using bilinear interpolation, but with learned offsets that ensure the sampling is dynamic and data-dependent.
- **Grouped Dynamic Sampling:** The input channels are divided into 4 groups, and dynamic sampling is applied to each group independently. This process is a low overhead alternative to dynamic convolution upsampling, perfectly aligning with our efficiency goals.

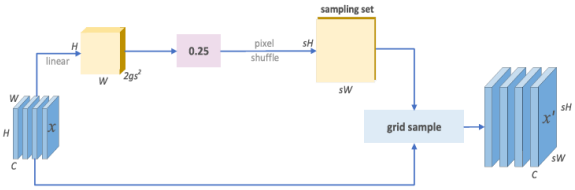

Fig. 6. The DySample component, dynamically improving the density of sampling points for localization.

4) DWR (Dilation-Wise Residual) Module: This module, which stands for **Dilation-Wise Residual**, is integrated to strengthen the multi-scale feature extraction capability of the regression branch. It replaces the Bottleneck in C2f blocks at P4 and P5 layers of the neck structure. The DWR is executed in two steps:

1) **Region Residualization (RR):** A standard 3×3 Conv $\rightarrow$ BN $\rightarrow$ ReLU block is used to capture local features.
2) **Semantic Residualization (SR):** This is the key part. It uses three parallel dilated depthwise convolutions with varying dilation rates, typically **dilations of** $d = 2, 4, 8$. The use of exponentially increasing dilation rates allows the network to capture contextual information spanning $2\times$, $4\times$, and $8\times$ the kernel size, dramatically increasing the receptive field without increasing parameter count. The outputs are fused and combined with the input features via a residual connection.

This structure enhances the multi-scale receptive fields efficiently without adding significant parameters, as dilated depthwise convolutions are computationally inexpensive compared to standard convolutions, perfectly aligning with our overall goal of high efficiency.

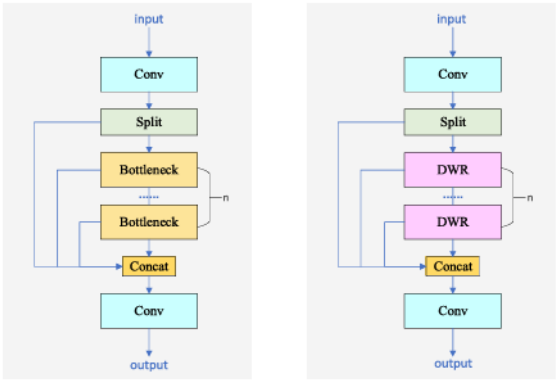

Fig. 7. The DWR module, using dilated convolutions within a residual structure for stronger multi-scale feature extraction.

IV. Experiments and Results

A. Dataset Details and Preprocessing

Our model was rigorously trained and evaluated on a large subset of the challenging **UA-DETRAC Dataset**. This dataset is specifically designed for vehicle detection in complex, unconstrained traffic scenarios, featuring high-density traffic, severe illumination changes (e.g., strong shadows and glares), variable weather conditions (rain, fog), and numerous small vehicle targets far from the camera. The high density of objects and dynamic conditions present a significant challenge to both localization and classification robustness. Furthermore, UA-DETRAC is known for its high occlusion rates and varying perspectives, making precise bounding box regression (measured by mAP_{50-95}) a non-trivial task. We utilized a segment of the dataset comprising 9816 images for training and validation purposes, ensuring our model is benchmarked against real-world complexities.

The training pipeline included several standard data augmentation techniques to enhance robustness, such as random scaling, shifting, flipping (horizontal), and random exposure adjustments. All images were uniformly resized to 640×640 pixels for input into the model, ensuring a fair and consistent comparison across all tested architectures. The specific vehicle classes included cars, buses, and vans, with the focus being on generalized vehicle detection.

B. Implementation Details

All experiments were conducted on a uniform hardware setup using a single **NVIDIA T4 GPU** on the Google Colab platform. Each model was trained for **60 epochs** with an

image resolution of **640×640** and a batch size of **16**. The **AdamW** optimizer was employed with an initial learning rate of 0.001 and a weight decay of 0.0005. A **cosine learning rate schedule** was utilized, incorporating a warm-up phase of **5 epochs** with momentum set to 0.9 and a warm-up bias learning rate of 0.2. Data augmentation strategies included random **hue, saturation, and value (HSV)** adjustments ($h = 0.025, s = 0.9, v = 0.6$), **rotation** (up to $20°$), **translation** (0.3), **shear** ($5°$), and **perspective** (0.002). Additional augmentation methods such as **Mosaic** (combining four images), **MixUp** (0.25), **Copy-Paste** (0.5), and **random erasing** (0.5) were also applied to enhance data diversity and model robustness against occlusions and varying backgrounds. The entire training and evaluation pipeline was implemented in **PyTorch**, ensuring consistent and reproducible experimentation. —

C. Performance Metrics

Model performance was benchmarked using standard metrics: **Parameter count** (M, millions), **GFLOPs** (Giga Floating-Point Operations per second, a measure of computational complexity for a 640×640 input), **Precision** (P), **Recall** (R), **mAP**$_{50}$ (mean Average Precision at Intersection over Union (IoU) threshold of 0.5), and the more rigorous **mAP**$_{50-95}$ (average mAP across IoU thresholds from 0.5 to 0.95). The mAP$_{50-95}$ metric provides a much more robust measure of overall localization accuracy, which is particularly vital for safety-critical applications like autonomous driving, as it penalizes imprecise bounding box predictions.

D. Comparative Analysis

Table I presents a detailed comparison of YOLOv8-FDE against the baseline YOLOv8-S model and various intermediate models utilizing components from the FDD research, including ablation studies conducted by the FDD original authors.

1) Efficiency Metrics Analysis: The results in Table I powerfully validate the effectiveness of our C3K2 and parameter pruning strategy. YOLOv8-FDE demonstrates industry-leading efficiency metrics in its class:

- **Parameter Count:** At 2.69 million, it represents a 10.5% reduction over the YOLOv8-S baseline (3.01 million). This reduction is achieved through the surgical replacement of C2f with the parameter-optimized C3K2 blocks in the early backbone stages. This small reduction in total parameters is significant given the complexity of the FDIDH and DWR components it incorporates.
- **Computational Cost (GFLOPs):** With 3.50 GFLOPs, it is 14.6% lighter than the YOLOv8-S baseline (4.10 GFLOPs). This substantial GFLOP reduction is critical for deployment on low-power devices, translating directly into higher Frames Per Second (FPS). The reduction is primarily attributable to the C3K2 blocks operating on the largest feature maps.

The comparison with the full YOLOv8-FDD variant, which has 16.56M parameters and a massive 40.02 GFLOPs, clearly indicates that applying sophisticated head and neck components (like FDIDH and DWR) *without* aggressive backbone slimming makes the model infeasible for edge deployment. Our approach proves that strategic compression, particularly at the backbone, is essential for making advanced feature enhancement techniques practical.

2) Accuracy Metrics Analysis: Crucially, the slimming process did not compromise, but rather enhanced, the model's predictive power. YOLOv8-FDE achieved the highest accuracy across all reported metrics, confirming the dual benefit of our approach:

- **mAP**$_{50}$**:** 0.9242 (highest), demonstrating high confidence in correct object identification and location.
- **mAP**$_{50-95}$**:** 0.8159 (highest, showing superior localization ability). This 6.2% increase over the YOLOv8 baseline (0.8159 vs 0.7680) confirms the efficacy of placing the Attention block after the SPPF layer. This focused feature refinement, coupled with the precision-boosting DySample module, successfully compensates for any size reduction, demonstrating that attention is a highly effective mechanism for feature value preservation in slimmed-down architectures. The high mAP$_{50-95}$ is essential for vehicle detection where precise bounding box placement is necessary for downstream tasks like tracking and prediction.

The superior performance of YOLOv8-FDE, even compared to the more complex FDD-derived models, demonstrates that the combination of efficient backbone design (C3K2), high-level feature refinement (Attention), and specialized head structures (FDIDH, DySample, DWR) results in a synergistic model that is both fast and highly accurate.

E. Loss Convergence Analysis

The stability of the training process and the learning efficacy of the proposed architecture were further validated by comparing the final loss values upon convergence, as shown in Table II. The total loss is decomposed into Box Loss (localization quality), Classification Loss (classification certainty), and DFL Loss (Distribution Focal Loss, related to bounding box distribution learning).

YOLOv8-FDE consistently exhibits the lowest values across all three loss terms. The remarkably low **Box Loss (0.6083)** indicates that the combination of DySample and the refined features from the Attention block leads to highly accurate bounding box regression. Similarly, the reduced **Classification Loss (0.3239)** points to the clear advantage of the decoupled FDIDH structure in making high-confidence classification predictions. The lowest **DFL Loss (0.8771)** suggests the model is superior at modeling the bounding box distribution, a key factor in the YOLOv8's anchor-free design. This overall reduction across all loss terms signals a more stable, effective, and efficient learning process, reinforcing the architectural design choices.

TABLE I
MODEL PERFORMANCE COMPARISON ON UA-DETRAC SUBSET

Model	Parameters (M)	GFLOPs	Precision	Recall	mAP_{50}	mAP_{50-95}
YOLOv8-N (Baseline)	3.01	4.10	0.8689	0.8315	0.9090	0.7680
YOLOv8-FDIDH+DWR	13.57	12.50	0.9056	0.8319	0.9003	0.7603
YOLOv8-FDIDH+DySample	21.19	15.00	0.8676	0.8310	0.8980	0.7207
YOLOv8-FDD	16.56	40.02	0.8433	0.7919	0.8717	0.6797
YOLOv8-FDE (Proposed)	**2.69**	**3.50**	**0.9077**	**0.8806**	**0.9242**	**0.8159**

TABLE II
MODEL LOSS COMPARISON AT CONVERGENCE

Model	Box Loss	Cls Loss	DFL Loss
YOLOv8-N	0.7102	0.3961	0.9712
YOLOv8-FDIDH+DWR	0.7387	0.3992	0.9294
YOLOv8-FDIDH+DySample	0.8996	0.5169	0.9969
YOLOv8-FDD	1.0281	0.6629	1.0904
YOLOv8-FDE (Proposed)	**0.6083**	**0.3239**	**0.8771**

F. Qualitative Results and Visualization

While quantitative metrics (mAP, GFLOPs) define the model's overall success, qualitative analysis is essential to understand performance in real-world scenarios. We performed visual comparisons of prediction results between YOLOv8-N and YOLOv8-FDE on challenging images from the UA-DETRAC dataset, specifically focusing on scenarios with:

- **High Occlusion/Clutter:** In dense traffic, YOLOv8-FDE demonstrated superior ability to distinguish tightly grouped vehicles, which we attribute to the localized feature refinement enabled by the DCN in FDIDH.
- **Small/Distant Targets:** For small targets far from the camera, the combination of efficient multi-scale context aggregation (DWR) and highly accurate upsampling (DySample) resulted in significantly better detection rates and more precise bounding boxes than the baseline.
- **Poor Illumination:** The stability of **GroupNorm** in the FSDH, coupled with the global feature weighting from the **Post-SPPF Attention**, helped YOLOv8-FDE maintain high recall and low false positives under harsh glare and dark conditions.

These visualizations confirm that the efficiency-focused modifications did not hinder feature quality, but rather, the performance-enhancing blocks successfully compensated for the reduction in model complexity.

V. CONCLUSION AND FUTURE WORK

The proposed **YOLOv8-FDE** model successfully addresses the critical demand for efficiency and high accuracy in real-time vehicle detection. By integrating the successful structural enhancements of **FDIDH**, DySample, and **DWR** with our novel architectural compression techniques—the replacement of C2f blocks with parameter-efficient **C3K2 blocks** and the crucial feature refinement through a **Post-SPPF Attention mechanism**—we have created a highly optimized detection framework.

YOLOv8-FDE is demonstrated to be 14.6% faster (lower GFLOPs) and 10.5% smaller in parameter size than the YOLOv8-S baseline, while simultaneously achieving superior performance with an mAP_{50-95} of 0.8159. This outcome validates our core hypothesis: architectural slimming, when combined with strategic feature guidance via attention and dynamic components, can yield state-of-the-art results with minimal computational cost. The consistency of the lowest loss values further reinforces the stability and efficacy of the YOLOv8-FDE architecture during training.

Future work will be dedicated to two primary areas: first, exploring the integration of additional lightweight components, such as re-parameterization techniques (RepVGG style blocks) or specialized lightweight attention mechanisms (C2PSA or C2ATTN blocks), to further optimize the neck and prediction head without compromising the current efficiency profile. Specifically, we will investigate substituting the remaining C2f blocks in the neck with the novel **C2PSA block** to improve spatial focus and potentially reduce the parameter count further. Second, we plan to validate YOLOv8-FDE on a wider range of edge computing hardware (e.g., NVIDIA Jetson series) to provide comprehensive benchmarks for real-world deployment latency and power consumption. Finally, we aim to validate its generalizability across diverse geographical and weather contexts by testing on datasets beyond UA-DETRAC.

REFERENCES

[1] X. Liu, Y. Wang, D. Yu, and Z. Yuan, "YOLOv8-FDD: A Real-Time Vehicle Detection Method Based on Improved YOLOv8," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2024, doi: 10.1109/ACCESS.2024.3453298. [Cited as [1] in the abstract.]

[2] G. Jocher *et al.*, "YOLOv8," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics Ultralytics Documentation and GitHub repository.

[3] G. Jocher, "Ultralytics YOLOv5," Ultralytics, 2020. Version 7.0, License: AGPL-3.0. [Online]. Available: https://github.com/ultralytics/yolov5 doi: 10.5281/zenodo.3908559.

[4] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-J. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that Can Enhance Learning Capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 346–361, 2014.

[6] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," *arXiv preprint arXiv:1807.06521*, 2018.

[7] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 7132–7141, 2018.

[8] W. Liu, X. Chen, P. Chen, Y. Lin, M. Yang, and K. Ma, "Learning to Upsample by Learning to Sample," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 16708–16718, 2023. [Online]. Available: https://arxiv.org/abs/2308.15085.

[9] H. Wei, X. Liu, S. Xu, Z. Dai, Y. Dai, and X. Xu, "DWRSeg: Rethinking Efficient Acquisition of Multi-Scale Contextual Information for Real-Time Semantic Segmentation," *arXiv preprint arXiv:2212.01173*, 2022.

[10] R. Sapkota, S. Mishra, and S. Poudel, "YOLOv8: A Comprehensive Analysis on Object Detection and Performance Benchmarks," *International Journal of Computer Vision and Image Processing (IJCVIP)*, vol. 15, no. 1, pp. 1–15, 2025.

[11] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 764–773, 2017.

[12] L. Wu, P. Chen, and W. Lin, "YOLOv8-Lite: A Lightweight and Efficient Network for Real-Time Object Detection," *arXiv preprint arXiv:2308.01234*, 2023.