fire

MDPI

Article

HIFE-YOLO: HierarchicalInformation Flow Enhancement for Lightweight Fire and Smoke Detection

Shaokai Feng [1], Shaoxian Feng [2], Junzhe Hu [1] and Qinggang Su [3,*]

[1] School of Artificial Intelligence and Data Science, Shanghai Dianji University, Shanghai 201306, China
[2] School of Future Technology, China University of Geosciences (Wuhan), Wuhan 430074, China
[3] School of Aeronautics, Shanghai Dianji University, Shanghai 201306, China
* Correspondence: suqg@sdju.edu.cn

Abstract

Fire and smoke detection requires compact models that preserve weak local evidence while using sufficient context to distinguish diffuse targets. This paper proposes HIFE-YOLO, a YOLO11n-based detector that coordinates multiscale Ghost feature generation, retention-path-assisted neck downsampling, and complementary task residuals before the detection towers. The fused model contains 2.303 million parameters and has an estimated computational cost of 6.062 GFLOPs for a 640 × 640 input, representing reductions of 10.84% and 4.00%, respectively, relative to YOLO11n. Under a common D-Fire training protocol of 100 epochs, three paired training seeds yield an mAP50 of 73.25 ± 0.36% and an mAP50–95 of 41.38 ± 0.44%, compared with 71.95 ± 0.23% and 40.16 ± 0.19% for the baseline. Fire and smoke AP50 for relatively small objects improve by 1.41 and 4.49 percentage points, respectively. On the independent fire_merged held-out view, frozen-weight testing improves overall mAP50 and recall by 1.74 and 5.67 percentage points, respectively, while precision and the HPWREN smoke AP do not improve. The parameter and storage savings are accompanied by slightly higher latency and peak GPU allocation in the measured desktop pipeline. The relatively small-object tests and the BoWFire fire-presence test show clear gains, and all conclusions remain limited to the D-Fire training protocol and the recorded seeds.

Keywords: fire detection; smoke detection; YOLO11n; lightweight object detection; multiscale features; information flow enhancement; task-aware detection

1. Introduction

Visual fire and smoke detection can cover large areas using fixed cameras, unmanned aerial vehicles, or existing surveillance platforms, providing an important complement to point sensors. Compared with scenes containing clearly developed flames, early fire sources are often small and have irregular boundaries, while smoke exhibits low contrast, translucency, and extensive spatial diffusion. A recent comprehensive review identifies dataset coverage, false-alarm control, small-object recognition, and computational constraints as common challenges in early fire detection [1]. Another technical review surveys methods for early wildfire detection across sensing, imaging, and intelligent analysis technologies [2]. YOLO unifies object detection within a single-stage prediction framework, providing a foundation for subsequent lightweight real-time detectors [3]. YOLOv10 further explores end-to-end detection and simplified post-processing, advancing the balance between accuracy and deployment complexity in real-time detection [4]. D-Fire incorporates Fire, Smoke,

Check for updates

Academic Editor: Lizhong Yang

Received: 6 August 2026
Revised: 14 October 2026
Accepted: 21 October 2026
Published: 1 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

and background images into a unified object detection task, providing a public basis for model research under resource constraints [5].

Fire and smoke differ in their visual properties. Flames typically exhibit high brightness and fragmented boundaries, making detection dependent on local edges and shape. Smoke has more diffuse contours and is easily confused with clouds, haze, and industrial emissions, requiring broader contextual information. FPN constructs multiscale feature representations through top-down pathways and lateral connections [6]. PANet strengthens bottom-up path aggregation to shorten the transmission path from low-level localization information to high-level features [7]. NAS-FPN incorporates cross-scale fusion structures into neural architecture search [8]. EfficientDet combines a weighted bidirectional feature pyramid with compound scaling to balance accuracy and efficiency [9]. For fixed-camera smoke monitoring, SmokeyNet combines spatial and temporal information from the FIgLib image collection acquired in Southern California [10]. Falcao et al. fuse CNN and vision Transformer predictions through a learned meta-classifier to identify smoke plumes amid visually similar phenomena such as low clouds and fog [11]. These methods provide background for organizing fire and smoke features in terms of local appearance, contextual evidence, and acquisition conditions.

Visual fire detection research also encompasses residual features, lightweight networks, image enhancement, and edge deployment. Dewi et al. combine YOLOv11 with histogram equalization for forest and land fire detection [12]. Dunnings and Breckon investigate low-complexity CNN architectures for non-temporal fire recognition and superpixel localization [13]. Among general-purpose visual backbones, ResNet improves gradient propagation in deep networks through residual learning [14]. MobileNetV2 reduces the computational cost of mobile networks using inverted residuals and linear bottlenecks [15]. MobileNetV3 further reduces network overhead through lightweight architecture search and attention design [16]. SENet enhances discriminative responses through channel recalibration [17]. CBAM connects channel attention and spatial attention in sequence [18]. ECA-Net establishes efficient channel interactions using local one-dimensional convolution [19]. Coordinate Attention encodes positional information into channel attention [20]. These studies provide research background and methodological context; their results correspond to their respective initialization, data splits, optimization, training schedules, and hardware settings. Parameter counts and GFLOPs describe model size and estimated computational cost.

This paper is motivated by the preservation of weak local evidence and more effective feature utilization in compact fire and smoke detectors. We organize HIFE-YOLO around three consecutive stages of the YOLO11n information flow: feature generation, downsampling transmission, and prediction-input adaptation. GhostNet introduces a lightweight approach that generates redundant features through inexpensive operations [21]. GSConv demonstrates the use of mixed convolution and channel shuffling in lightweight detection necks [22]. YOLOX illustrates the role of decoupled prediction in modern single-stage detectors [23]. TOOD addresses differences between classification scores and localization quality through task alignment [24]. Within the fixed prediction scales and training protocol of YOLO11n, these ideas motivate a coordinated design with distinct architectural roles for the three modules.

HIFE-YOLO arranges three units with distinct responsibilities along the data flow of the detection graph. MSGhostC3k2 forms multiscale deep features; RetentionGSConv introduces a fixed retention path during neck scale transitions, aiming to mitigate local-response attenuation during downsampling; CTD provides bounded task-related residual adjustments before the original bounding-box and classification towers. The design preserves the prediction scales and backbone width. Under the common training protocol, the complete

architecture combines fewer parameters and a smaller checkpoint with higher overall AP than the baseline in all three paired training seeds.

The main contributions of this paper are as follows:

1. We propose a coordinated YOLO11n architecture that assigns low-cost multiscale feature generation, a fixed downsampling retention path, and bounded task-specific corrections to distinct stages of feature generation, scale transition, and prediction-input adaptation.
2. We clarify the responsibilities and resource requirements of each module through tensor dimensions, parameter and computational cost estimates at the insertion points, independent and incremental ablations, and sensitivity experiments on the residual amplitude limit.
3. We evaluate detection accuracy and execution behavior using three paired seeds, class-specific small-object metrics, screened external tests, and a paired desktop latency benchmark.

1.1. Related Work

1.1.1. Visual Fire and Smoke Detection

Visual fire detection encompasses handcrafted appearance features, learned image representations, and temporal sequence analysis. Chino et al. combine pixel color and texture analysis in BoWFire for fire detection in still images [25]. The public data and background images of D-Fire provide a reproducible fire and smoke object detection task [5]. Subsequent research combines spatial and temporal patterns to address dynamic changes in fire scenes [26]. Goncalves et al. evaluate YOLO models on ground and aerial smoke and wildfire images [27]. Mamadaliev et al. construct ESFD-YOLOv8n using residual structures, a GELAN neck module incorporating C2f, and WIoUv3 [28]. Alkhammash compares YOLOv9, YOLOv10, and YOLO11 for joint fire and smoke detection [29].

Acquisition conditions and annotation tasks also influence visual wildfire research. Shamsoshoara et al. establish FLAME using UAV images of prescribed pile burns in Arizona, defining fire-presence classification and segmentation tasks [30]. Dewangan et al. release FIgLib and use spatiotemporal camera-image features in SmokeyNet to identify wildland smoke [10]. More recently, Seidel et al. use vision–language models to generate structured descriptions of UAV fire scenes, including forest type, fire status, and fire type [31]. These studies provide complementary perspectives on aerial imaging, fixed cameras, and scene interpretation; each image collection in this paper is evaluated using the units and annotations specified in the corresponding section.

1.1.2. Lightweight Features and Neck Design

Low-complexity fire networks and lightweight detection components control resource use at different levels. Dunnings and Breckon evaluate simplified CNN architectures for single-image fire recognition and region localization [13]. GhostNet generates additional features through inexpensive transformations [21], whereas GSConv combines standard convolution, depthwise convolution, and channel shuffling in compact detection necks [22]. For fire and smoke detection, SA-YOLO assigns attention according to network stage [32]. HIFE-YOLO coordinates feature generation, neck scale transitions, and prediction-input adaptation, with comparisons conducted under the common YOLO11n protocol used in this paper.

1.1.3. Task-Aware Prediction

Classification and localization objectives in object detection correspond to class prediction and bounding-box regression, respectively. YOLOX uses decoupled prediction towers

to model classification and localization branches separately [23]. TOOD addresses differences between classification scores and localization quality through task alignment [24]. CTD preserves the original YOLO11 prediction towers and loss, generating bounded residuals for detail and context from the input features at each pyramid level. Its final projections are zero-initialized, so CTD initially preserves the source predictions when all other parameters and inference states remain unchanged.

2. HIFE-YOLO Architecture

2.1. YOLO11n Baseline Detector

We use Ultralytics YOLO11n as the baseline detector. Given a 640×640 input, the backbone progressively reduces spatial resolution through successive 3×3 stride-2 convolutions and uses C3k2 for feature extraction at each stage. P3, P4, and P5 correspond to input strides of 8, 16, and 32, respectively. At the backbone output, SPPF aggregates information from multiple receptive fields, while C2PSA further models high-level spatial and channel relationships, producing three feature levels for the neck.

The YOLO11n neck consists of a top-down FPN pathway and a bottom-up PAN pathway. FPN successively upsamples P5, concatenates it with lateral P4 and P3 features, and fuses the resulting features through C3k2. PAN uses two stride-2 convolutions to propagate the fused P3 features back to P4 and P5, performing concatenation and C3k2 aggregation at each level. The three output scales enter decoupled detection heads: the bounding-box towers produce $4 \times$ reg_max distributional regression logits, and the classification towers produce n_c class logits, followed by decoding at each scale. Figure 1 shows the complete connections among the backbone, neck, and prediction stage.

Under the common training protocol of this study, YOLO11n contains 2.583 M parameters and has an estimated computational cost of 6.314 GFLOPs. The proposed modifications preserve this backbone hierarchy, neck topology, and three-scale prediction framework to support controlled comparison of the resulting architectures.

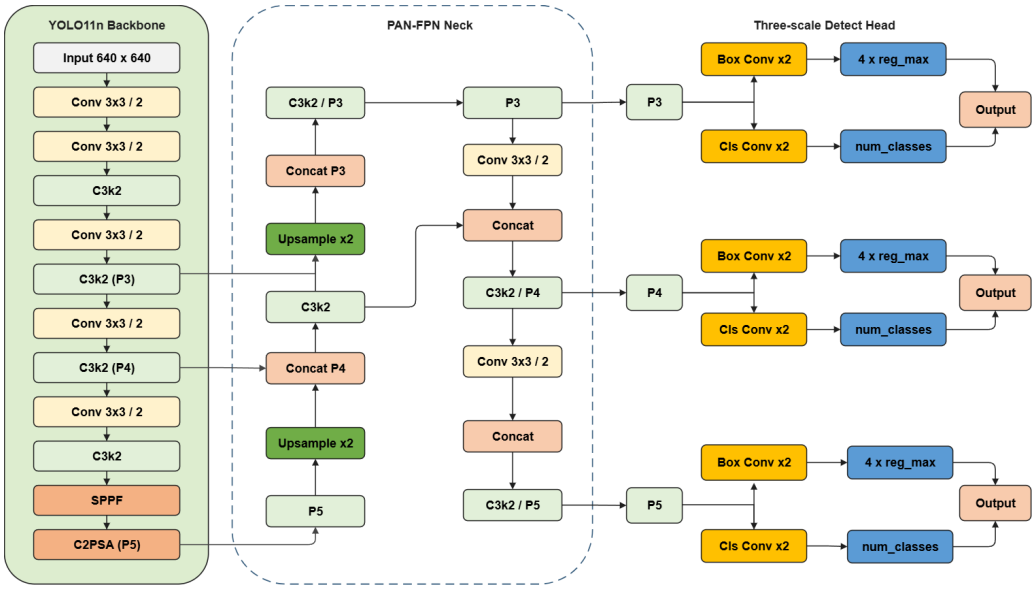

Figure 1. Overall architecture of the YOLO11n baseline detector. The backbone outputs P3/P4/P5 features at strides of 8, 16, and 32. The PAN–FPN neck performs bidirectional multiscale aggregation, and each scale generates detection outputs through separate bounding-box and classification towers.

2.2. Improved HIFE-YOLO Architecture

HIFE-YOLO preserves the backbone hierarchy, bidirectional PAN–FPN aggregation pathways, and P3, P4, and P5 three-scale prediction structure of YOLO11n. MSGhostC3k2

replaces the high-level C3k2 before SPPF in the backbone and the fusion C3k2 that produces P5 at the end of the bottom-up neck pathway. RetentionGSConv replaces the two stride-2 downsampling units at P3→P4 and P4→P5 in the neck. CTD operates independently at all three scales and generates task-related inputs before the original bounding-box and classification towers. All other upsampling, concatenation, loss functions, and post-processing remain unchanged.

The three units have consecutive but distinct responsibilities in the network: MSGhostC3k2 forms multiscale features, RetentionGSConv transmits information during scale reduction, and CTD allocates features between prediction tasks. The architecture preserves the baseline prediction scales and backbone width. The complete model contains 2.303 M parameters and requires 6.062 GFLOPs, compared with 2.583 M parameters and 6.314 GFLOPs for the YOLO11n baseline.

2.3. MSGhostC3k2

Figure 2 shows the three-level structure of MSGhostC3k2. In Figure 2a, the input $\mathbf{x}$ is first mapped to $2C_h$ channels by a 1×1 convolution and then split along the channel dimension into equal-width features $\mathbf{y}_0$ and $\mathbf{y}_1$. The feature $\mathbf{y}_0$ forms a direct path, while $\mathbf{y}_1$ passes sequentially through n MSGhostBottlenecks. The output of each stage is retained and included in the final concatenation:

$$[\mathbf{y}_0, \mathbf{y}_1] = \mathrm{Split}(C_s(\mathbf{x})), \qquad \mathbf{y}_{i+1} = B_i(\mathbf{y}_i), \quad i = 1, \ldots, n, \tag{1}$$

$$\mathbf{o} = C_o(\mathrm{Concat}[\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{n+1}]), \tag{2}$$

where C_s and C_o are both 1×1 convolutions. The shared collection line indicates that $\mathbf{y}_0, \mathbf{y}_1$ and the outputs of all bottlenecks participate in Concat. This progressive collection scheme jointly fuses the shallow direct-path features and subsequent transformed features at the output.

Figure 2b shows the ith MSGhostBottleneck. For an input $\mathbf{u}$ with C channels, the first MSGhostConv reduces the channel count to $C/2$, and the second restores it to C. Because the input and output dimensions within this module are identical, the identity and transformation branches can be added elementwise:

$$B_i(\mathbf{u}) = \mathbf{u} + M_2(M_1(\mathbf{u})), \qquad M_1 : C \to C/2, \quad M_2 : C/2 \to C. \tag{3}$$

Figure 2c expands MSGhostConv M. Let its target output channel count be C_o. The primary convolution first generates $C_h = C_o/2$ intrinsic features:

$$\mathbf{y} = C_p(\mathbf{x}). \tag{4}$$

The 3×3, 5×5, and 7×7 depthwise convolution branches then extract low-cost responses from different neighborhoods of $\mathbf{y}$:

$$\mathbf{z}_k = DW_k(\mathbf{y}), \qquad k \in \{3, 5, 7\}. \tag{5}$$

The gating branch applies global average pooling to $\mathbf{y}$ and generates scale coefficients for the current sample through a three-dimensional linear mapping followed by Softmax:

$$\boldsymbol{\alpha} = \mathrm{softmax}(W\,\mathrm{GAP}(\mathbf{y}) + \mathbf{b}) = [\alpha_3, \alpha_5, \alpha_7], \qquad \sum_k \alpha_k = 1. \tag{6}$$

The three branches are summed using the corresponding coefficients to form complementary features $\mathbf{z}$ with the same channel count as $\mathbf{y}$:

$$\mathbf{z} = \sum_{k \in \{3,5,7\}} \alpha_k \mathbf{z}_k. \tag{7}$$

Finally, the primary features $\mathbf{y}$ and complementary features $\mathbf{z}$ are concatenated along the channel dimension, yielding $M(\mathbf{x}) = \text{Concat}(\mathbf{y}, \mathbf{z})$ and restoring C_o output channels. The design predicts three sample-level coefficients. Depthwise kernels of different sizes cover receptive fields ranging from local edges to broader context, allowing multiscale responses to contribute to feature generation according to the input content.

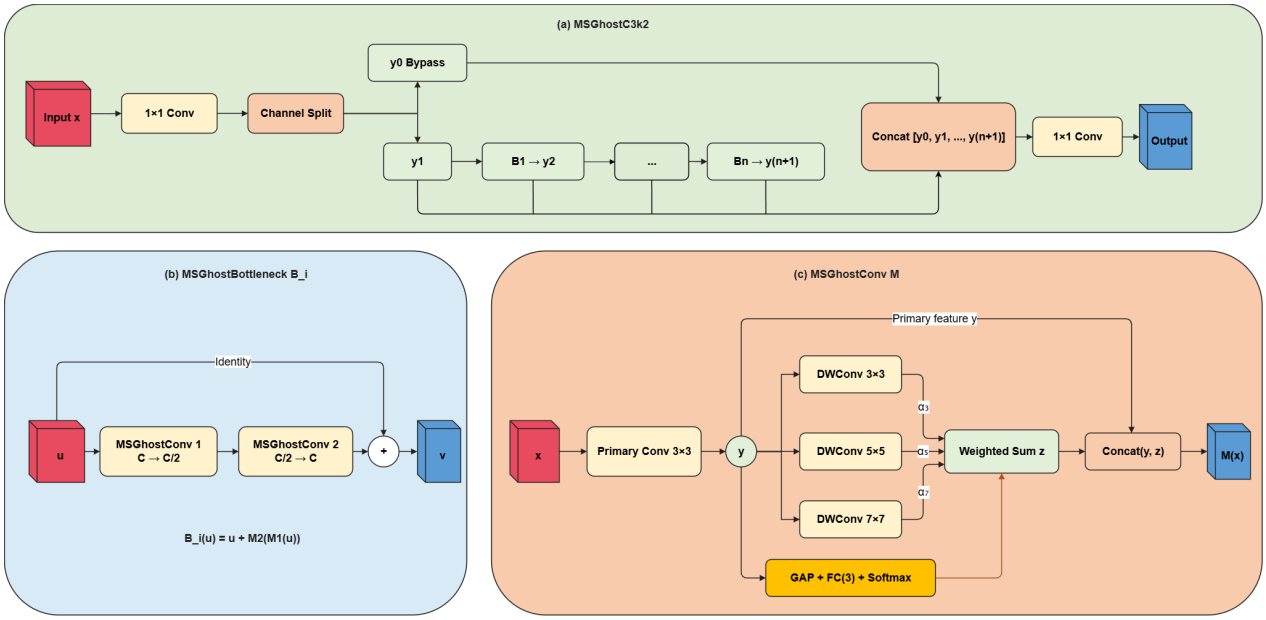

Figure 2. Internal structure of MSGhostC3k2: (**a**) channel splitting, sequential bottlenecks, and multilevel feature collection; (**b**) an MSGhostBottleneck consisting of two MSGhostConv units and an identity branch; (**c**) an MSGhostConv consisting of primary features, three-scale depthwise convolution branches, and sample-level weighted fusion.

2.4. RetentionGSConv

Figure 3 shows the structure of RetentionGSConv. For stride-2 neck transformations, RetentionGSConv consists of a learnable transformation path and a fixed retention path. For an input $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, the module requires equal input and output channel counts, namely $C_{in} = C_{out} = C$. The transformation path $G(\mathbf{x})$ successively applies a 3×3 stride-2 convolution, a 5×5 depthwise convolution, concatenation, and channel shuffling. The retention path $P(\mathbf{x})$ uses 3×3 stride-2 max pooling. Both paths output $H/2 \times W/2 \times C$ features, which are fused using a fixed ratio $\rho = 0.25$:

$$\mathbf{y} = \frac{G(\mathbf{x}) + \rho P(\mathbf{x})}{\sqrt{1 + \rho^2}}. \tag{8}$$

The normalization term moderates amplitude changes caused by the additional path. The fixed max-pooling path is parameter-free and directly transmits the maximum downsampled responses to the fusion point, forming the output together with the learnable transformation.

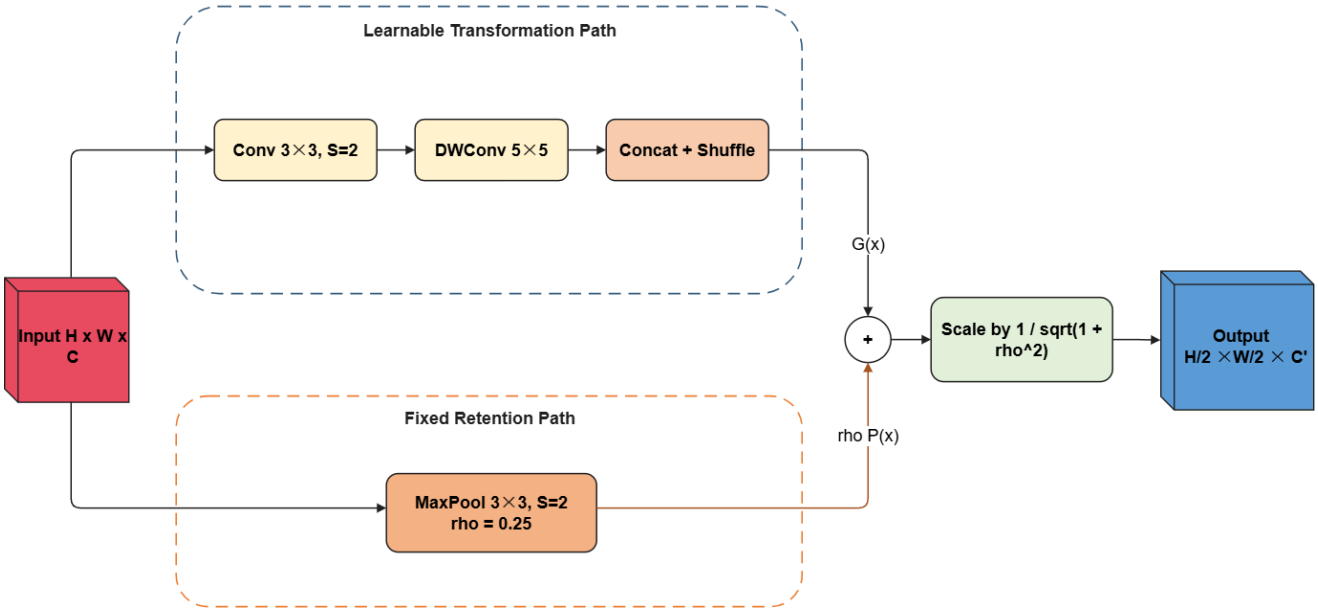

Figure 3. Structure of RetentionGSConv. Input features pass through a learnable GSConv transformation path and a fixed max-pooling retention path. Both paths retain identical output dimensions and channel counts and are combined through weighting and normalization according to Equation (8).

2.5. CTD

Figure 4 illustrates the CTD structure at each pyramid level. CTD operates independently at each pyramid level l. A 1×1 projection first compresses the source features $\mathbf{f}_l$ into four channels:

$$\mathbf{h}_l = R_l(\mathbf{f}_l). \tag{9}$$

Local detail and contextual features are defined as

$$\mathbf{d}_l = DW_{3,l}(\mathbf{h}_l) - \mathrm{AvgPool}_3(\mathbf{h}_l), \qquad \mathbf{c}_l = DW_{3,l}^{\delta=2}(\mathbf{h}_l). \tag{10}$$

The former encodes local contrast through a detail-oriented residual, while the latter expands the contextual range through dilated convolution at the same spatial resolution. The selector concatenates $|\mathbf{d}_l|$ and $\mathbf{c}_l$, generates a single-channel spatial map through a 1×1 convolution and Sigmoid, and broadcasts the map across the four hidden channels to estimate the proportion of complementary information exchanged within the level:

$$\mathbf{s}_l = \sigma(S_l([|\mathbf{d}_l|, \mathbf{c}_l])). \tag{11}$$

The classification and localization encodings are

$$\mathbf{u}_l^{cls} = \mathbf{c}_l + \mathbf{s}_l \odot \mathbf{d}_l, \qquad \mathbf{u}_l^{box} = \mathbf{d}_l + \mathbf{s}_l \odot \mathbf{c}_l. \tag{12}$$

After group normalization and SiLU, two 1×1 projections restore the original channel counts and form bounded residuals:

$$\mathbf{r}_l^t = \eta \tanh\left(E_l^t(\mathrm{SiLU}(GN_l^t(\mathbf{u}_l^t)))\right), \quad t \in \{cls, box\}, \quad \eta = 0.2. \tag{13}$$

The resulting residuals are added separately to the identity features $\mathbf{f}_l$ to form classification and regression features, which are then passed to the original YOLO11 classification and bounding-box towers. P3, P4, and P5 each use their own CTD parameters and within-level features. Both final projections are zero-initialized, making the residuals zero when

the module is introduced. With all other source parameters and inference states unchanged, the initial predictions are identical to those of the source model.

The four channels form a learned correction bottleneck. The identity paths fully preserve the 64, 128, and 256 channels of P3, P4, and P5, respectively, while CTD encodes compact task-specific adjustments. All levels share the same hidden-channel width, establishing a common adaptation budget. The three adapters add 6563 parameters in total. The main experiments use the four-channel configuration. To assess the hidden-channel width, the two- and eight-channel variants are also trained and evaluated under the same D-Fire 100-epoch protocol with seed 20260716. Detection performance and parameter counts for all three widths are reported in Section 3.6.

The setting $\eta = 0.2$ restricts each residual element to a feature-amplitude range of $|r_{l,i}^t| \leq 0.2$. Zero initialization makes the adapter start from an identity mapping when all other states of the source detector remain unchanged. Section 3.6 presents a single-seed sensitivity comparison for $\eta = 0.1, 0.2, 0.3$; the main experiments retain the original 0.2 setting.

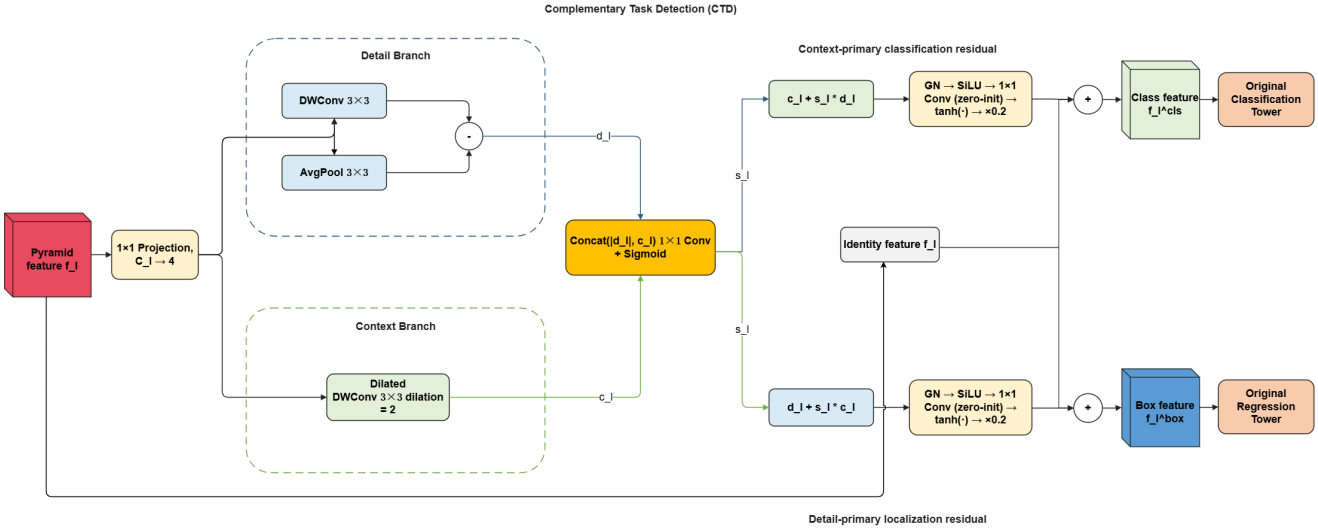

Figure 4. Structure of CTD. Each pyramid level independently generates local detail features $\mathbf{d}_l$, dilated contextual features $\mathbf{c}_l$, and a single-channel selection map $\mathbf{s}_l$. The classification and localization encodings pass through zero-initialized bounded residual projections and are injected into the original classification and bounding-box towers, respectively.

2.6. Complete Model Architecture

Embedding these three technical units into the fixed YOLO11n data flow produces the complete HIFE-YOLO model. The two MSGhostC3k2 modules are located at the high-level backbone and neck P5 output. The two RetentionGSConv modules occupy the scale-transition positions in the PAN pathway, and the three CTD modules are placed before the original detection towers at P3, P4, and P5. Figure 5 summarizes the connections among the input, backbone, neck, module insertion points, and prediction stage.

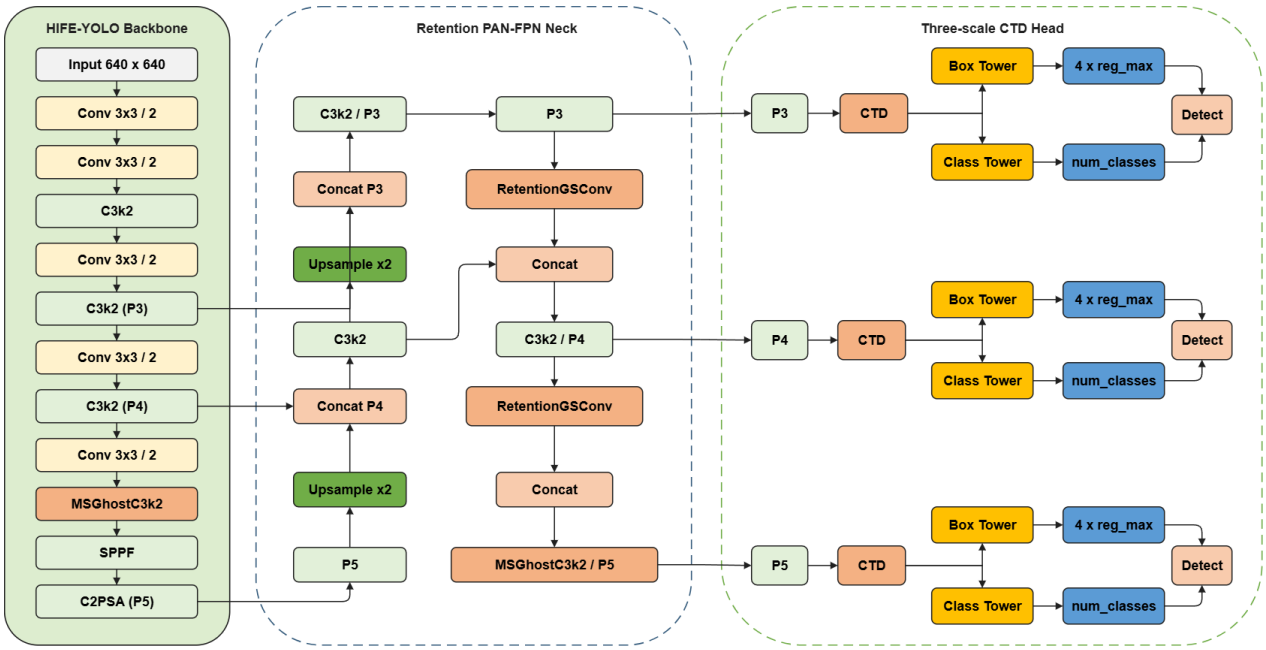

Figure 5. Overall architecture of HIFE-YOLO. MSGhostC3k2 is deployed at the high-level backbone and the final neck P5 fusion stage; RetentionGSConv performs the two downsampling operations in the PAN pathway; three CTD modules connect to the original decoupled detection towers at P3/P4/P5, respectively.

2.7. Module Dimensions and Computational Costs

Table 1 reports the cost at each modified position for a batch size of one and an RGB input of 640 × 640. The three CTD rows account for the added adapters, with the original prediction towers retained separately. Parameter counts are obtained directly from model tensors after convolution and normalization fusion; the unfused baseline and complete models contain 2,590,230 and 2,310,053 parameters, respectively. All models use the same Ultralytics/THOP GFLOPs convention: multiply–accumulate operations are counted at a stride-scale input, multiplied by two, and scaled to the target resolution. Estimated arithmetic costs and measured pipeline latency are reported separately.

Table 1. Dimensions and standalone costs of the modified modules. Tensor shapes are $C \times H \times W$.

Position	Module	Input	Output	Parameters	GFLOPs
Backbone P5	MSGhostC3k2	256 × 20 × 20	256 × 20 × 20	246,726	0.196826
Neck P5 fusion	MSGhostC3k2	384 × 20 × 20	256 × 20 × 20	279,494	0.223040
PAN P3→P4	RetentionGSConv	64 × 80 × 80	64 × 40 × 40	19,296	0.061542
PAN P4→P5	RetentionGSConv	128 × 40 × 40	128 × 20 × 20	75,456	0.060262
P3 adapter	CTD	64 × 80 × 80	64 × 80 × 80	993	0.010854
P4 adapter	CTD	128 × 40 × 40	128 × 40 × 40	1889	0.005171
P5 adapter	CTD	256 × 20 × 20	256 × 20 × 20	3681	0.002522

CTD outputs separate bounding-box and classification tensors, both with the output shape shown in the table. CTD rows count only the added adapters, and their GFLOPs are convolutional computation estimates. The complete detection head, including the original prediction towers, contains 436,217 parameters and has an estimated cost of 1.718797 GFLOPs.

Replacing the two C3k2 modules with MSGhostC3k2 saves 196,724 fused parameters and an estimated 0.156851 GFLOPs. Replacing the two stride-two convolutions with RetentionGSConv saves a further 89,760 parameters and 0.114125 GFLOPs. CTD adds 6563 parameters and 0.018547 GFLOPs. The net changes after replacement are −279,921 parameters and −0.252429 GFLOPs; Table 1 lists the absolute standalone cost of each module.

3. Experiments and Results

3.1. Datasets and Their Evaluation Roles

D-Fire provides the primary training, validation, and testing tasks for fire and smoke detection [5]. We use the fixed training, validation, and test directories supplied with the downloaded archive. The original test directory contains 4306 images, four of which are rejected by the evaluation loader, leaving an effective test set of 4302 images used throughout the paper. Empty annotation files define background images. Table 2 reports the adopted data counts; Figure 6 shows representative validation images. Checks of exact content and file basenames across the D-Fire splits record zero duplicates within the checked scope. These checks concern exact duplicates and do not constitute a comprehensive perceptual-near-duplicate audit.

Table 2. Statistics of the D-Fire data splits.

Split	Images	Objects	Fire Instances	Smoke Instances	Background Images
Training	14,122	17,432	9638	7794	6458
Validation	3099	3932	2176	1756	1375
Test	4302	5186	2875	2311	2005

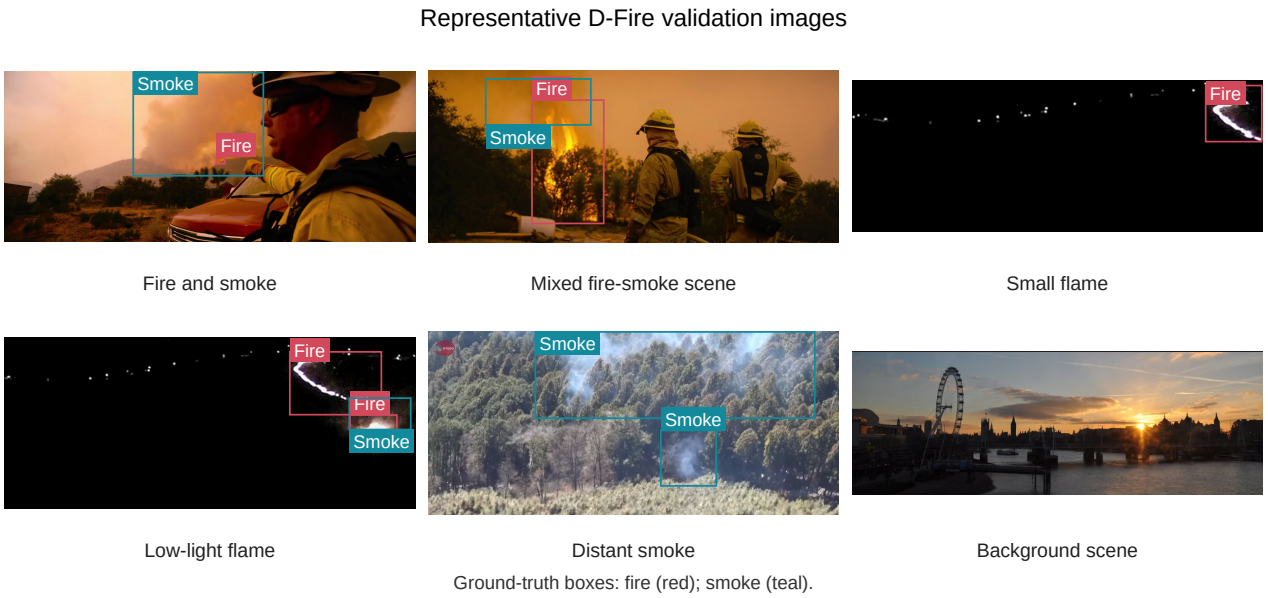

Figure 6. Representative D-Fire validation images with ground-truth boxes, including fire, diffuse smoke, small objects, low-light conditions, and background scenes. Red boxes denote fire, and teal boxes denote smoke.

The supplementary datasets serve different evaluation roles. For `fire_merged`, 30 visually reviewed candidates depicting the same scenes or adjacent frames are excluded from its 601 test images, leaving 571 images with unique image hashes, which form the screened held-out transfer view used in this evaluation. This candidate screening does not constitute an exhaustive perceptual-near-duplicate audit. The source was used for data survey and overlap screening only; it was not used for training, model selection, or threshold selection. The larger `merged` directory contains D-Fire, which is already represented by the D-Fire splits used for training and testing; the external results table therefore uses the screened `fire_merged` view described above.

BoWFire [25] provides a supplementary fire-presence test on still images; the downloaded evaluation archive contains 119 positive and 107 negative images. The HPWREN

bounding-box archive provides 744 smoke-positive images of 640 × 480, with one smoke box per image, of which 272 occupy less than 1% of the image area. The FIgLib/SmokeyNet study introduces fixed-camera smoke monitoring with HPWREN [10]; the 744 bounding-box images used here come from the annotation repository listed in the Data Availability Statement. Both complete datasets are used for image-presence and bounding-box evaluation, respectively. All external tests use frozen baseline/HIFE-YOLO checkpoints with seed 20260716 and the fixed settings described below. Source screening and sensitivity results are reported alongside the corresponding analyses.

3.2. Experimental Environment and Training Protocol

Tables 3 and 4 list the recorded environment and common protocol, respectively.

The main comparisons and module ablations use seed 20260716. Two paired repetitions of the two architectures use seeds 20260921 and 20260922. The training-seed analysis therefore contains three baseline/HIFE pairs. The best checkpoints are selected according to the recorded validation procedure.

Paired initialization copies shape-compatible tensors from the baseline model to the modified architecture while retaining the initial states of the added components for the corresponding seed. For the audited MSGhost+Retention/CTD model pair, the maximum absolute difference between shared source tensors is zero. When all other source parameters and inference states remain unchanged, the zero-initialized CTD output projections preserve the source predictions upon insertion. The original training uses four data-loader workers, while revision runs adjust the worker count according to the recorded local execution settings. The data, model-selection rules, and 100-epoch training schedule are retained.

Table 3. Recorded software stack and revision workstation configuration.

Item	Configuration
CPU (revision workstation)	AMD Ryzen 5 7500F; 6 cores/12 threads
GPU	NVIDIA GeForce RTX 3070; 8 GiB
Installed memory (revision workstation)	16 GiB
Operating system	Windows 11 Pro, 64-bit; build 26200
Python/PyTorch	3.13.0/2.7.1+cu118
torchvision/CUDA build	0.22.1+cu118/11.8
Ultralytics	8.4.41
cuDNN/profiling tool (revision audit)	90100/ultralytics-thop 2.0.18
Recorded NVIDIA driver	610.88 (archived evaluation record)

CPU and installed memory describe the verified revision workstation; software, GPU, and driver fields use the recorded evaluation environment.

Table 4. Common training and detection evaluation settings.

Item	Setting
Input/batch size/epochs	640 × 640/8/100
Initialization/precision	No pretrained backbone; AMP training; FP32 final evaluation
Optimizer	AdamW; initial learning rate 0.001; weight decay 0.0005
Learning rate schedule	Cosine decay; final learning rate multiplier 0.01; 3 warm-up epochs
Loss weights	Bounding box 7.5; classification 0.5; DFL 1.5
Augmentation	HSV $(0.015, 0.7, 0.4)$; translation 0.1; scaling 0.5; flipping 0.5
Mosaic/other mixing	Mosaic 1.0, disabled for the final 10 epochs; MixUp/CutMix 0
Reproducibility	Deterministic mode; recorded seeds and frozen best.pt
Evaluation	Batch size 8; confidence floor 0.001; NMS IoU 0.7; at most 300 boxes

3.3. Evaluation Metrics

At a selected confidence threshold, true positives (TP), false positives (FP), and false negatives (FN) are defined through one-to-one matching with the correct class:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F_1 = \frac{2PR}{P + R}. \tag{14}$$

For a fixed matching IoU of τ, sweeping the confidence threshold produces a precision–recall curve whose interpolated area is

$$AP_\tau = \int_0^1 p_{\tau,\text{interp}}(r)\, dr, \qquad mAP_\tau = \frac{AP_\tau^{Fire} + AP_\tau^{Smoke}}{2}. \tag{15}$$

AP50 uses $\tau = 0.50$; AP50–95 averages over ten IoU thresholds from 0.50 to 0.95 at intervals of 0.05 [33]. AP integrates the confidence-swept curve. Except for metrics labeled as fixed, P/R in the main detector, ablation, and external transfer tables use the Ultralytics summary at the point of maximum mean F1 on the evaluation curve. Fixed-operating-point error statistics, HPWREN P/R, and image-presence tests use a confidence threshold of 0.25. Bounding-box matching for fixed-operating-point error statistics uses an IoU of 0.5, and NMS uses an IoU of 0.7. AP50–95 uses the ten matching IoU thresholds defined above.

FPPI is the number of unmatched predicted boxes divided by the number of images. The background false-alarm rate is the proportion of images with empty ground truth that contain at least one predicted box. Relatively small-object evaluation selects objects whose annotated box area divided by image area is below 0.01. A dedicated COCO-style 101-point evaluator ignores objects outside the area range and allows at most 300 detections; the overall detection tables use the Ultralytics evaluator. Means and sample standard deviations across training seeds are reported together, with $n - 1$ as the standard deviation denominator. Absolute AP gains are expressed in percentage points. For paired-seed summaries, relative gains are the mean paired AP gain divided by the corresponding baseline mean, multiplied by 100%; single-run relative gains use the corresponding single-run baseline. Reported gains are distinguished from differences obtained by subtracting rounded display values.

3.4. Model Comparison Under a Common Protocol

The complete model achieves 73.37% mAP50 and 41.24% mAP50–95 in the original paired run (Table 5), improving upon YOLO11n by 1.67 and 1.23 percentage points, respectively (approximately 2.33% and 3.07% relative gains), while reducing the fused parameter count by 10.84%. Compared with YOLOv8n, HIFE-YOLO has a similar mAP50 (73.37% versus 73.39%), a higher mAP50–95 (41.24% versus 40.75%), and lower parameter and estimated computational costs. The s-series models achieve higher AP with larger parameter counts and computational costs. Table 5 groups the n- and s-scale results under the common training stack.

Published D-Fire results provide methodological context. Goncalves et al. report mAP50 values of 75.9% and 77.8% for YOLOv5n and YOLOv8n, respectively [27]; Alkhammash reports a YOLO11 comparison under the settings of that study [29]. Table 5 uses the data splits, initialization, augmentation, and optimization protocol specified in this paper.

Table 5. Comparison under the same stack on the effective D-Fire test set (seed 20260716). P/R use the evaluator's summary operating point.

Model	Precision	Recall	mAP50	mAP50-95	Parameters (M)	GFLOPs
YOLOv5n	0.7252	0.6423	0.7094	0.3927	2.503	7.066
YOLOv8n	0.7380	0.6698	0.7339	0.4075	3.006	8.086
YOLOv10n	0.6996	0.6221	0.6861	0.3814	2.266	6.529
YOLO11n	0.7198	0.6638	0.7170	0.4001	2.583	6.314
HIFE-YOLO	0.7363	0.6699	0.7337	0.4124	2.303	6.062
YOLOv5s	0.7594	0.6825	0.7488	0.4222	9.112	23.830
YOLOv8s	0.7661	0.6952	0.7599	0.4317	11.126	28.440
YOLO11s	0.7589	0.6821	0.7509	0.4242	9.414	21.304

3.5. Variability across Training Seeds

Table 6 presents three paired comparisons, summarized as mean ± sample standard deviation. The baseline mAP50/mAP50–95 values are 71.95 ± 0.23%/40.16 ± 0.19%, compared with 73.25 ± 0.36%/41.38 ± 0.44% for HIFE-YOLO. The mean paired absolute gains are 1.29 and 1.22 percentage points, corresponding to approximately 1.79% and 3.04% relative gains, respectively. Both overall AP metrics improve for every paired seed. The mAP50 gains for the individual seeds are 1.67, 0.81, and 1.39 percentage points, indicating that the magnitude of the gain varies across runs.

Table 6. Results for all paired training seeds on D-Fire, including mean ± sample standard deviation for overall AP. Values are percentages.

Model	Seed	mAP50	mAP50–95	Fire AP50	Smoke AP50
YOLO11n	20260716	71.70	40.01	65.42	77.97
HIFE-YOLO	20260716	73.37	41.24	66.73	80.01
YOLO11n	20260921	72.03	40.37	66.24	77.81
HIFE-YOLO	20260921	72.84	41.03	66.53	79.15
YOLO11n	20260922	72.14	40.10	65.52	78.76
HIFE-YOLO	20260922	73.53	41.88	66.49	79.53
YOLO11n	Summary ($n = 3$)	71.95 ± 0.23	40.16 ± 0.19	—	—
HIFE-YOLO	Summary ($n = 3$)	73.25 ± 0.36	41.38 ± 0.44	—	—

Summary rows report mean ± sample standard deviation ($n = 3$, denominator $n - 1$) for overall AP only. Individual-run values and aggregate statistics are displayed to two decimal places; recombining rounded values may produce small differences. The reported mean paired gains are 1.29 and 1.22 percentage points. Relative gains use these paired gains and the corresponding baseline means.

3.6. Module Ablations and CTD Sensitivity

Table 7 summarizes the independent and incremental module comparisons for seed 20260716. In this run, MSGhost alone increases mAP50–95 by 0.60 percentage points while reducing the parameter count by 196,724. Retention alone improves mAP50 and mAP50–95 by 1.34 and 1.00 percentage points, respectively, and CTD alone by 0.31 and 0.21 percentage points. Adding CTD to MSGhost+Retention introduces 6563 parameters and yields observed mAP50/mAP50–95 gains of 0.83/0.41 percentage points. The complete model achieves 73.37% mAP50 and 41.24% mAP50–95, exceeding the baseline and all tested individual-module variants on both metrics and supporting the effectiveness of the coordinated configuration under the recorded conditions. Module effects depend on the combination architecture, so independent-ablation differences are not simply added. These single-seed ablations do not support statistical-significance claims or establish a super-additive interaction.

Table 7. Common single-seed ablations (seed 20260716). AP values are percentages; parameter counts use the fused convention.

Variant	mAP50	mAP50–95	Parameters (M)	GFLOPs
YOLO11n	71.70	40.01	2.582542	6.314086
MSGhost only	72.62	40.61	2.385818	6.157235
Retention only	73.04	41.01	2.492782	6.199962
CTD only	72.01	40.22	2.589105	6.332634
MSGhost + Retention	72.54	40.83	2.296058	6.043110
HIFE-YOLO	73.37	41.24	2.302621	6.061658

Table 8 reports sensitivity to the residual amplitude limit for seed 20260716: the mAP50 range is 0.069 percentage points, and the mAP50–95 range is 0.219 percentage points. All settings use a four-channel width and have the same 2.303 M parameters and estimated 6.062 GFLOPs. The setting $\eta = 0.3$ achieves the highest mAP50–95 among the three tested settings; the main experiments retain the original $\eta = 0.2$ configuration.

Table 8. Sensitivity to the CTD residual amplitude limit under the same D-Fire 100-epoch protocol. Values are percentages.

η	mAP50	mAP50–95
0.1	73.42	41.25
0.2 (original)	73.37	41.24
0.3	73.44	41.46

To examine the choice of CTD hidden-channel width, models with widths of 2 and 8 are additionally trained using the same D-Fire 100-epoch protocol, input size, optimizer, and random seed as the original four-channel configuration. Table 9 reports the detection metrics and fused parameter counts for the three widths. The main experiments use the four-channel configuration.

Table 9. Sensitivity to CTD hidden-channel width under the same D-Fire 100-epoch protocol. The random seed is 20260716, values are percentages, and parameter counts use the fused convention.

CTD Width	Precision	Recall	mAP50	mAP50–95	Parameters (M)
2	73.45	65.81	72.36	40.63	2.299789
4 (original)	73.63	66.99	73.37	41.24	2.302621
8	73.70	67.35	73.48	41.19	2.308285

Relative to width four, width two reduces the fused parameter count by 2832 but decreases mAP50 and mAP50–95 by 1.01 and 0.62 percentage points, respectively. Width eight adds 5,664 parameters, increases mAP50 by only 0.11 percentage points, and decreases mAP50–95 by 0.05 percentage points. Thus, under the current test protocol, width four offers a practical compromise between detection accuracy and model complexity: the narrower configuration loses detection accuracy, while the wider configuration increases the parameter count without consistently improving overall performance.

Activation measurements use a fixed set of 32 validation images, comprising 12 fire, 12 smoke, and 8 background images. The retention-path-to-output RMS ratios of the two downsampling modules are 0.281 and 0.232. The RMS ratios of CTD bounding-box/classification residuals to the input are 0.075/0.069 at P3, 0.086/0.089 at P4, and 0.097/0.104 at P5. These measurements describe the relative response amplitudes of the added paths on the fixed samples and provide descriptive activation diagnostics.

RMS ratios are not direct measures of task-relevant information retention and do not, by themselves, establish a causal mechanism for changes in detection performance.

3.7. Small-Object and Challenging-Scene Evaluation

To evaluate the ability of HIFE-YOLO to detect weak, small objects, we first conduct relatively small-object evaluation on D-Fire, selecting targets whose annotated box area occupies less than 1% of the image area. This definition classifies objects by their relative scale; the evaluation subset contains 1859 fire boxes and 499 smoke boxes. Table 10 shows that fire AP50 increases from 59.19% to 60.60% and smoke AP50 from 56.96% to 61.45%, representing gains of 1.41 and 4.49 percentage points, respectively. The corresponding AP50–95 gains are 0.67 and 2.64 percentage points. These results indicate improved detection and localization of relatively small objects, with a more pronounced improvement for smoke.

Table 10. D-Fire relatively small-object AP obtained using a dedicated 101-point evaluator. Annotated boxes occupy less than 1% of the image area; values are percentages.

Class	Boxes	Baseline AP50	HIFE AP50	Baseline AP50–95	HIFE AP50–95
Fire	1859	59.19	60.60	28.22	28.89
Smoke	499	56.96	61.45	24.39	27.03

We then select BoWFire and HPWREN as supplementary test sets to examine fire recognition and smoke detection in different scenes (Figure 7). The two datasets have different evaluation emphases: BoWFire includes both fire and non-fire images and is suitable for assessing the balance between fire detection and background false alarms; HPWREN targets fixed-camera wildland smoke monitoring and is suitable for assessing recognition and spatial localization of weak smoke. Both tests use frozen checkpoints trained on D-Fire with random seed 20260716, without additional training or fine-tuning.

BoWFire is selected primarily to examine flame morphology and background interference. Flames have irregular boundaries and fragmented local textures, and their color and brightness can be confused with visually similar regions in non-fire backgrounds. Evaluating the detection rate on fire images alone is therefore insufficient; false alarms on non-fire images must also be examined. The BoWFire evaluation set used here contains 119 fire images and 107 non-fire images and is evaluated at the image level for fire presence. At a confidence threshold of 0.25, YOLO11n and HIFE-YOLO detect 111 and 114 fire images, respectively, increasing the hit rate from 93.28% to 95.80%. Fire-class false alarms on non-fire images decrease from 11 to 10 images, reducing the false-alarm rate from 10.28% to 9.35%. In this BoWFire fire-presence evaluation, HIFE-YOLO improves the fire-image hit rate and yields a slightly lower image-level fire-class FAR. This dataset-specific result does not imply lower background FAR across all evaluation sets.

HPWREN is selected primarily to examine the weak visual features of distant smoke. Smoke typically exhibits low contrast, translucency, and diffuse boundaries. At long distances, it may occupy only a small image region, challenging both contextual recognition and boundary localization. The HPWREN set used here contains 744 smoke-positive images, each with one annotated smoke box; 272 boxes occupy less than 1% of the image area. This dataset therefore complements the evaluation of weak, small smoke targets in wildland surveillance, while also covering smoke targets at different scales.

On all 744 HPWREN images, YOLO11n/HIFE-YOLO achieve AP50 values of 63.28%/60.17% and AP50–95 values of 29.66%/29.24%. At the fixed operating point of confidence 0.25 and matching IoU 0.5, precision increases from 76.46% to 79.41%, while recall decreases from 52.82% to 46.64%; the corresponding counts are $TP = 393/347$, $FP = 121/90$, and $FN = 351/397$. These results reveal a trade-off between precision

and detection coverage: HIFE-YOLO produces fewer unmatched smoke predictions but more missed detections, and its overall AP does not exceed the baseline. This difference may relate to different smoke-scale distributions and boundary annotation conventions in D-Fire and HPWREN; HIFE-YOLO is trained for joint fire and smoke detection and has not undergone specialized training or fine-tuning for distant, weak, small smoke targets.

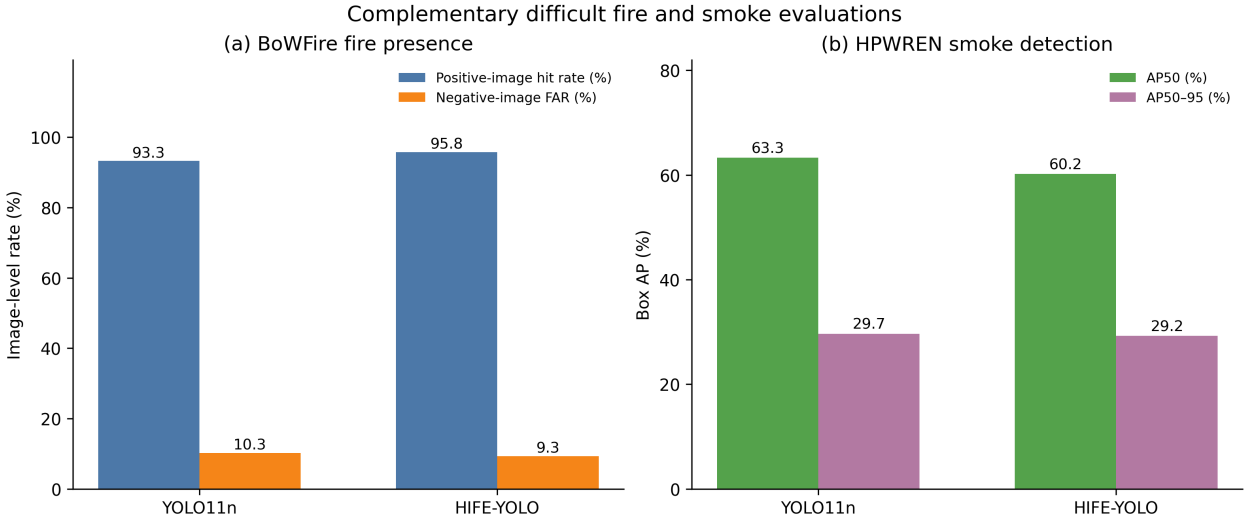

Figure 7. Supplementary fire and smoke tests using frozen checkpoints with seed 20260716. (**a**) BoWFire image-level fire hit rate and fire-class false-alarm rate (FAR) on non-fire images, comprising 119 fire images and 107 non-fire images. (**b**) HPWREN smoke-box AP on all 744 smoke-positive images. The two panels use image-level and box-level evaluation metrics, respectively.

To examine the effects of duplicate images and source overlap on HPWREN results, we additionally report two screened versions. The original dataset contains seven pairs of images with identical content but conflicting annotated boxes. Excluding all 14 images leaves 730 images, on which YOLO11n/HIFE-YOLO achieve AP50 values of 64.39%/61.17%. Further excluding 121 visually reviewed candidates with scene overlap with D-Fire leaves 609 images, yielding AP50 values of 64.82%/59.57% and AP50–95 values of 28.45%/26.71%. The removal of identical-content pairs and the manual screening of scene-overlap candidates address different concerns; the latter is a partial screening rather than a comprehensive near-duplicate audit. The Supplementary Materials collates the reported results for the three screening views. Unreported metrics are marked NR, which does not denote zero. Overall, HIFE-YOLO improves D-Fire relatively small-object evaluation and BoWFire fire-presence testing, whereas the HPWREN results show that detection coverage for distant smoke still requires improvement.

3.8. Cross-Dataset Transfer

To evaluate transfer beyond the training distribution, we use the publicly compiled `fire_merged` dataset as an independent held-out dataset. The final HIFE-YOLO and YOLO11n baseline are both trained exclusively on D-Fire; `fire_merged` is not used for training, model selection, threshold selection, or parameter adjustment. Table 11 therefore reports frozen-weight cross-dataset evaluation on data not used in training. An image-level SHA-256 audit finds no exact hash matches with the D-Fire training images. Manual inspection of most `fire_merged` images shows marked differences from D-Fire in scene composition, camera viewpoints, and visual appearance. The original test view contains 601 images, of which 571 with unique image hashes remain after screening and form the main evaluation view. Exact-hash checking and manual inspection provide distinct

forms of review; a comprehensive perceptual-near-duplicate audit of all samples has not been completed.

Table 11 presents overall metrics and class-specific AP50 on the screened `fire_merged` view. Relative to YOLO11n, HIFE-YOLO improves Recall, mAP50, and mAP50–95 by 5.67, 1.74, and 0.17 percentage points, respectively, indicating higher detection sensitivity and slightly better overall localization quality on the independent held-out data. P/R use the operating point at which the evaluator's mean F1 is maximized; all class metrics in the table are computed using the original annotations provided by the data source.

Table 11. Cross-dataset transfer results on the screened independent `fire_merged` held-out set. All values are percentages.

Test View	Model	P	R	mAP50	mAP50–95	Fire AP50	Smoke AP50
`fire_merged` (571)	Baseline	55.23	59.74	44.49	19.49	56.64	32.34
`fire_merged` (571)	HIFE-YOLO	49.66	65.41	46.23	19.65	60.21	32.23

P/R use the operating point of maximum mean F1 in the evaluator. The screened view contains 571 unique image hashes; all matching and metric computations use the original annotations provided by the data source.

In the screened view, HIFE-YOLO achieves a Precision of 49.66%, below the baseline's 55.23%, while Recall increases from 59.74% to 65.41%. At a fixed confidence threshold of 0.25, the baseline/HIFE-YOLO counts are $TP = 434/450$, $FP = 479/514$, and $FN = 591/575$. The HIFE model produces appreciably more correct detections and fewer missed detections, but also appreciably more false detections. Reviews of some unmatched predictions record suspected small flames with missing corresponding annotation boxes. The decrease in Precision may therefore reflect both annotation completeness and greater model sensitivity to weak, small targets.

As a supplementary assessment of screening effects, the baseline/HIFE-YOLO mAP50 values on the original 601 `fire_merged/test` images are 44.49%/46.23%, and the mAP50–95 values are 19.49%/19.65%. HIFE-YOLO maintains higher mAP50 both before and after screening, indicating a degree of cross-dataset generalization on `fire_merged`, which was not used in training. Its advantages are mainly reflected in higher recall, increased detection coverage at a relaxed IoU threshold, and enhanced detection of subtle targets.

3.9. Fixed-Operating-Point Errors and Measured Efficiency

Table 12 reports results at the fixed D-Fire operating point. HIFE-YOLO reduces unmatched predicted boxes from 1770 to 1616 and box-level FPPI from 0.4114 to 0.3756, while correct matches increase from 3593 to 3605. Both models produce at least one predicted box on 65 of the 2005 background images, giving the same image-level background FAR of 3.24%. Thus, unmatched-box counts decrease, but the proportion of background images with a false alarm does not improve. FPPI uses all 4302 evaluation images as its denominator, whereas background FAR uses the 2005 background images.

Table 12. D-Fire error statistics at confidence 0.25 and matching IoU 0.5. P/R/F1 are percentages.

Model	P	R	F1	FPPI	Background FAR
YOLO11n	67.00	69.28	68.12	0.4114	65/2005 (3.24%)
HIFE-YOLO	69.05	69.51	69.28	0.3756	65/2005 (3.24%)

The paired desktop timing protocol uses the same 200 decoded real images. After 50 warm-up runs per model, the model execution order is shuffled for each image using a deterministic random procedure. Table 13 reports FP32 inference results with a batch size of

one on an RTX 3070. Timing starts from the decoded image input and covers preprocessing, host–device transfers, the network forward pass, NMS, and result construction. FPS is the reciprocal of mean latency; median and 95th-percentile latency are reported separately.

Table 13. Desktop inference pipeline measurements and model storage.

Model	Mean ms	Median ms	p95 ms	FPS	Peak GPU MiB	File MB
YOLO11n	8.763	8.401	11.370	114.11	65.9	5.44
HIFE-YOLO	9.116	8.953	11.570	109.88	71.9	4.93

Timing uses a paired benchmark of 200 images. Peak GPU allocation and checkpoint file sizes are taken from separately recorded pipeline and storage audits on the same device.

HIFE-YOLO contains 2.303 M fused parameters and has an estimated cost of 6.062 GFLOPs, reductions of 10.84% and 4.00%, respectively, relative to YOLO11n. Its checkpoint decreases from 5.44 to 4.93 MB, giving a more compact model and storage footprint. In the measured RTX 3070 FP32 desktop pipeline with batch size one, its mean latency is 9.116 ms and peak GPU allocation is 71.9 MiB, both slightly higher than the baseline values of 8.763 ms and 65.9 MiB; the corresponding throughput is 109.88 FPS. The lightweight advantage therefore concerns parameter count and checkpoint storage, with a modest increase in the measured pipeline overhead. Multibranch convolutions and additional paths may introduce extra memory accesses, operator launches, and synchronization costs, but the specific causes require further profiling.

3.10. Convergence and Descriptive Diagnostics

Figure 8 shows the 100-epoch validation trajectories of the original four variants, while Figure 9 enlarges epochs 60–100. Both figures distinguish the raw trajectories from five-epoch moving averages. Over the final five epochs, the original complete model has a mean/standard deviation of 0.41954/0.00009 for mAP50–95, compared with 0.41286/0.00024 for YOLO11n. These statistics describe late-training fluctuations in the recorded individual runs; Table 6 reports variability across paired training seeds.

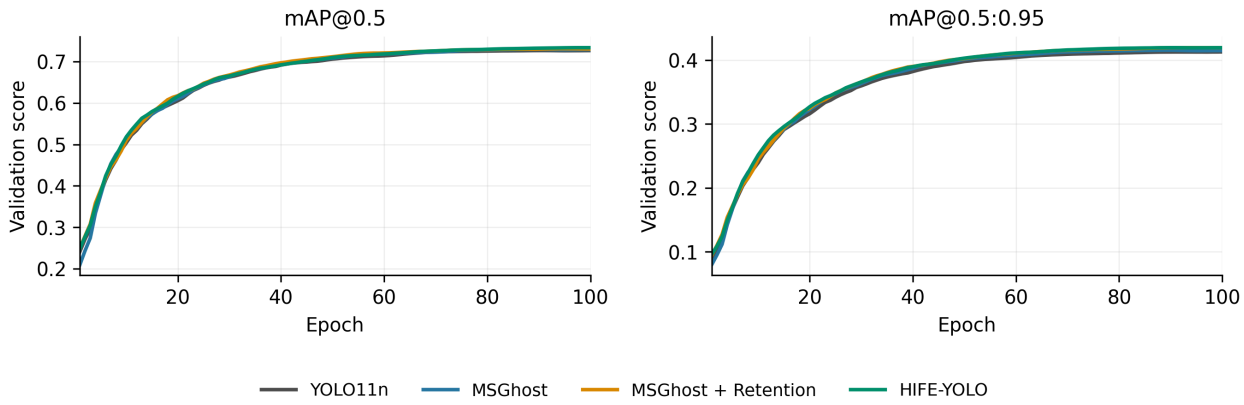

Figure 8. Validation mAP trajectories of the original four incremental variants over 100 training epochs. Five-epoch moving averages improve readability.

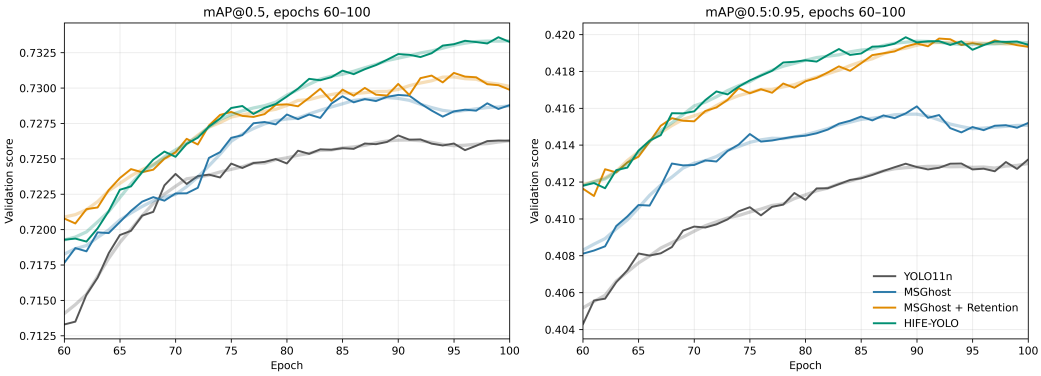

Figure 9. Enlarged validation trajectories for epochs 60–100, retaining both raw values and five-epoch smoothing to show late-training differences.

The loss curves in Figure 10 show a rapid decrease early in training followed by slower changes under the fixed schedule. The validation PR curves (Figure 11) show detection performance across confidence thresholds, while Figure 12 presents ground-truth boxes and predictions from the two models on selected validation images.

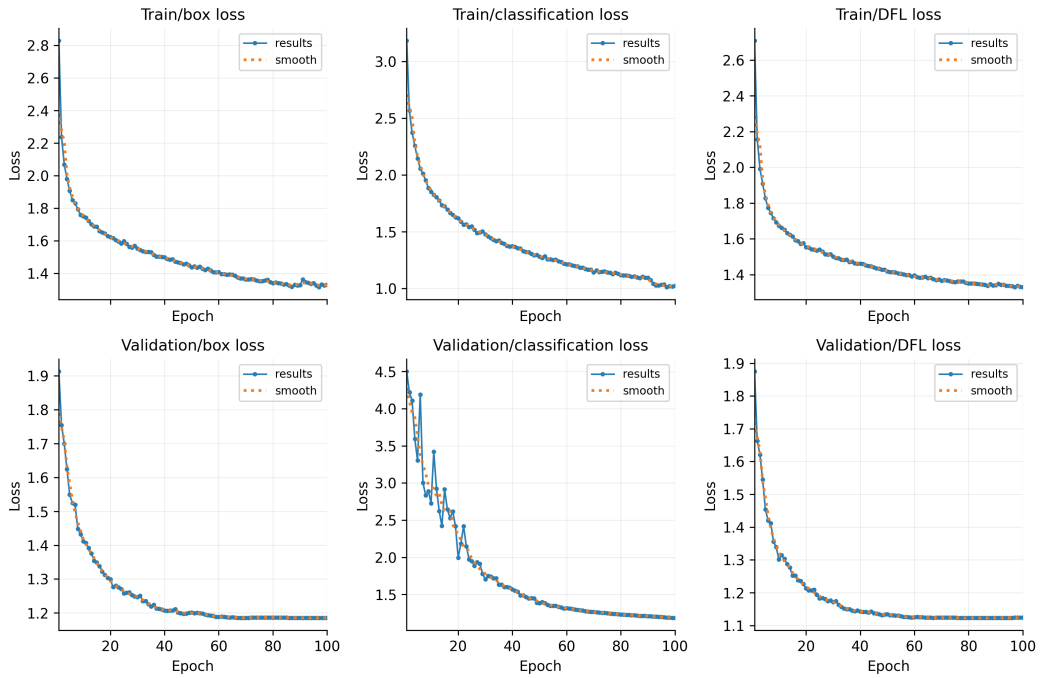

Figure 10. Training and validation bounding-box, classification, and DFL losses of HIFE-YOLO. Per-epoch values are distinguished from five-epoch moving averages.

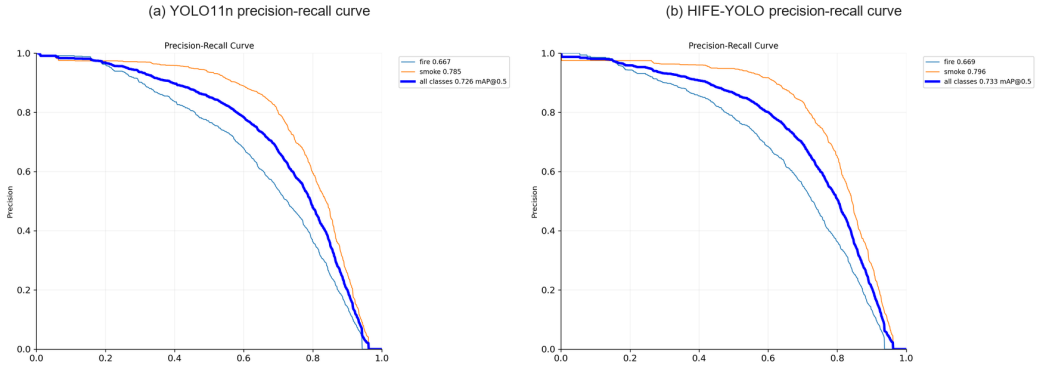

Figure 11. Validation precision–recall curves: (**a**) YOLO11n; (**b**) HIFE-YOLO.

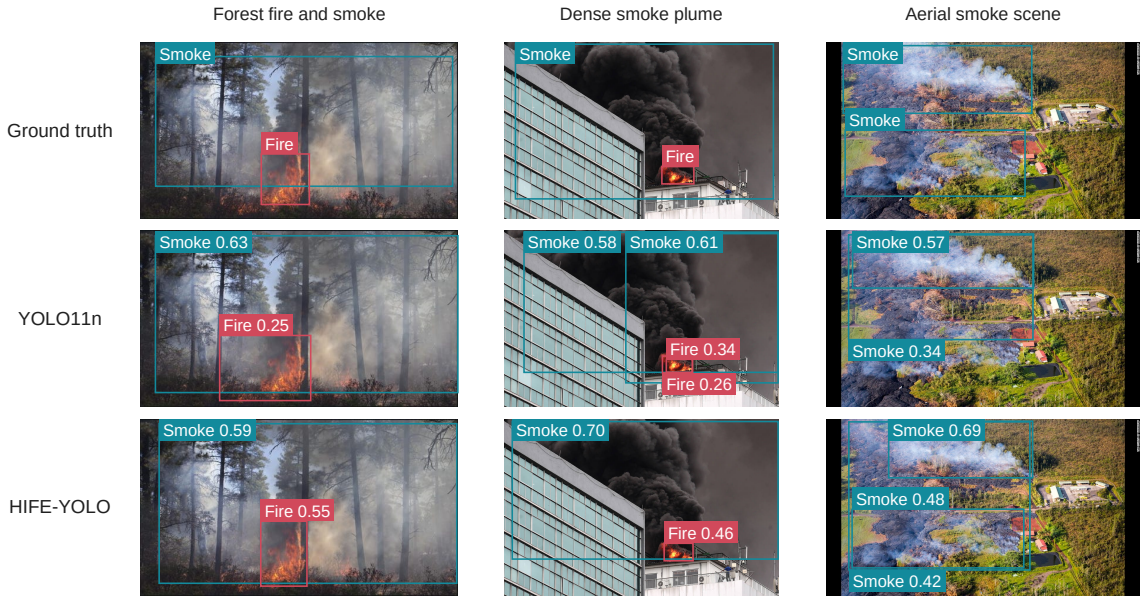

Figure 12. Selected validation examples: from top to bottom, ground truth, baseline predictions, and HIFE-YOLO predictions. Red boxes denote fire, and teal boxes denote smoke; numbers beside predicted class labels are confidence scores.

4. Conclusions

HIFE-YOLO coordinates efficient multiscale feature generation, retention-path-assisted downsampling, and bounded task adaptation within the original YOLO11n three-scale graph. The complete model reduces the fused parameter count from 2,582,542 to 2,302,621 (10.84%) and the estimated computational cost from 6.314086 to 6.061658 GFLOPs (4.00%). Across three paired seeds, D-Fire mAP50 improves from $71.95 \pm 0.23\%$ to $73.25 \pm 0.36\%$ and mAP50–95 from $40.16 \pm 0.19\%$ to $41.38 \pm 0.44\%$, i.e., mean paired gains of 1.29 and 1.22 percentage points (approximately 1.79% and 3.04% relative gains), and both metrics improve in every paired run. Class-specific relatively small-object tests show gains for fire (AP50 59.19% to 60.60%, +1.41 percentage points) and smoke (AP50 56.96% to 61.45%, +4.49 percentage points).

On the independent `fire_merged` held-out view, frozen-weight transfer raises recall by 5.67 percentage points and mAP50 by 1.74 percentage points at a lower precision. BoWFire reports image-level fire hit and false-alarm rates (93.28% to 95.80% and 10.28% to 9.35%), whereas on HPWREN, the AP50 of the proposed model is below that of the baseline (60.17% versus 63.28%) and more distant smoke is missed; the advantages observed in the relatively small-object and fire-presence views of this study therefore do not extend to all external conditions. The RTX 3070 benchmark measures mean end-to-end pipeline latencies of 8.763 and 9.116 ms (114.11 versus 109.88 FPS) and peak GPU allocations of 65.9 and 71.9 MiB for YOLO11n and HIFE-YOLO, respectively. The proposed model therefore combines a lower parameter count, estimated arithmetic cost, and checkpoint size with slightly higher latency and peak GPU allocation in the measured desktop pipeline. The results above are limited to the D-Fire dataset, the training protocol of Table 4 and the recorded seeds; operational early wildfire detection and embedded deployment still require validation on real edge devices and under field conditions.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/fire9110478/s1. The supplementary file contains HPWREN screening-view results and fixed-operating-point error counts: `Supplementary_Results_EN.pdf`.

Author Contributions: Conceptualization, methodology, software, validation, data curation, visualization, and writing—Original draft preparation: S.F. (Shaokai Feng); dataset collection, data cleaning and annotation, architecture diagram preparation, and training and validation of selected candidate modules: S.F. (Shaoxian Feng); manuscript formatting, proofreading, and layout consistency checks: J.H.; supervision, result review, writing—Review and editing, and project administration: Q.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study uses public image datasets and does not involve human participants or animal experiments.

Informed Consent Statement: Not applicable.

Data Availability Statement: D-Fire is available from its official repository: D-Fire repository (accessed on 20 October 2026). The source of HPWREN annotations is recorded at https://github.com/aiformankind/wildfire-smoke-dataset (accessed on 20 October 2026); FIRESENSE is archived at https://zenodo.org/records/836749 (accessed on 20 October 2026). Frozen evaluation manifests, overlap-screening records, model configurations, and per-run results are retained with the revision materials and are available from the corresponding author upon reasonable request. Local external sources remain subject to the original data-use terms. The model configurations, training scripts, and evaluation scripts used in this study are available from the corresponding author upon reasonable request. Following formal acceptance, the authors may choose to provide a public code repository address.

Acknowledgments: The authors thank Shanghai Dianji University for providing the research environment and computing facilities.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Elhanashi, A.; Essahraui, S.; Dini, P.; Saponara, S. Early Fire and Smoke Detection Using Deep Learning: A Comprehensive Review of Models, Datasets, and Challenges. *Appl. Sci.* **2025**, *15*, 10255. [CrossRef]
2. Honary, R.; Shelton, J.; Kavehpour, P. A Review of Technologies for the Early Detection of Wildfires. *ASME Open J. Eng.* **2025**, *4*, 040803. [CrossRef]
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
4. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, 37, 107984–108011. [CrossRef]
5. de Venancio, P.V.A.B.; Lisboa, A.C.; Barbosa, A.V. An Automatic Fire Detection System Based on Deep Convolutional Neural Networks for Low-Power, Resource-Constrained Devices. *Neural Comput. Appl.* **2022**, *34*, 15349–15368. [CrossRef]
6. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
7. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768. [CrossRef]
8. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7029–7038. [CrossRef]
9. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10778–10787. [CrossRef]
10. Dewangan, A.; Pande, Y.; Braun, H.-W.; Vernon, F.; Perez, I.; Altintas, I.; Cottrell, G.W.; Nguyen, M.H. FIgLib & SmokeyNet: Dataset and Deep Learning Model for Real-Time Wildland Fire Smoke Detection. *Remote Sens.* **2022**, *14*, 1007. [CrossRef]
11. Falcão, G.; Fernandes, A.M.; Garcia, N.; Aidos, H.; Tomás, P. Stacking Deep Learning Models for Early Detection of Wildfire Smoke Plumes. In Proceedings of the 31st European Signal Processing Conference, Helsinki, Finland, 4–8 September 2023; pp. 1370–1374. [CrossRef]
12. Dewi, C.; Santoso, M.V.V.; Chernovita, H.P.; Mailoa, E.; Philemon, S.A.; Chen, A.P.S. Integration of YOLOv11 and Histogram Equalization for Fire and Smoke-Based Detection of Forest and Land Fires. *Comput. Mater. Contin.* **2025**, *84*, 5361–5379. [CrossRef]

13. Dunnings, A.J.; Breckon, T.P. Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection. In Proceedings of the 25th IEEE International Conference on Image Processing, Athens, Greece, 7–10 October 2018; pp. 1558–1562. [CrossRef]
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
15. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. [CrossRef]
16. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
17. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
18. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision–ECCV 2018*; Springer: Cham, Switzerland, 2018; pp. 3–19. [CrossRef]
19. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11531–11539. [CrossRef]
20. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13708–13717. [CrossRef]
21. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589. [CrossRef]
22. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-Neck by GSConv: A Lightweight-Design for Real-Time Detector Architectures. *J. Real.-Time Image Process.* **2024**, *21*, 62. [CrossRef]
23. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
24. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 3490–3499. [CrossRef]
25. Chino, D.Y.T.; Avalhais, L.P.S.; Rodrigues, J.F., Jr.; Traina, A.J.M. BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. In Proceedings of the 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 95–102. [CrossRef]
26. de Venancio, P.V.A.B.; Campos, R.J.; Rezende, T.M.; Lisboa, A.C.; Barbosa, A.V. A Hybrid Method for Fire Detection Based on Spatial and Temporal Patterns. *Neural Comput. Appl.* **2023**, *35*, 9349–9361. [CrossRef]
27. Goncalves, L.A.O.; Ghali, R.; Akhloufi, M.A. YOLO-Based Models for Smoke and Wildfire Detection in Ground and Aerial Images. *Fire* **2024**, *7*, 140. [CrossRef]
28. Mamadaliev, D.; Touko, P.L.M.; Kim, J.-H.; Kim, S.-C. ESFD-YOLOv8n: Early Smoke and Fire Detection Method Based on an Improved YOLOv8n Model. *Fire* **2024**, *7*, 303. [CrossRef]
29. Alkhammash, E.H. A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection. *Fire* **2025**, *8*, 26. [CrossRef]
30. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial Imagery Pile Burn Detection Using Deep Learning: The FLAME Dataset. *Comput. Netw.* **2021**, *193*, 108001. [CrossRef]
31. Seidel, L.; Gehringer, S.; Raczok, T.; Ivens, S.-N.; Eckardt, B.; Maerz, M. Advancing Early Wildfire Detection: Integration of Vision Language Models with Unmanned Aerial Vehicle Remote Sensing for Enhanced Situational Awareness. *Drones* **2025**, *9*, 347. [CrossRef]
32. Kim, J.; Kim, G. SA-YOLO: Stage-Aware Attention for Real-Time Fire and Smoke Detection on Edge Devices. *IEEE Access* **2026**, *14*, 42270–42284. [CrossRef]
33. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision–ECCV 2014*; Springer: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.