

Article

Multi-Scale Feature Fusion Method for Component Detection in Construction Drawings

Hua Zhang ^{1,2}, Yubei Tang ³, Jingfeng Yuan ¹ , Lulu Liang ⁴, Yanghao Yuan ³, Yong Chen ⁴, Fan Yang ² and Baili Zhang ^{3,5,*} 

¹ Southeast University National and Local Joint Engineering Research Center for Intelligent Construction and Maintenance, Nanjing 210000, China; howard@seu.edu.cn (H.Z.); jingfeng-yuan@seu.edu.cn (J.Y.)

² Jiangsu Dongyin Intelligent Engineering Technology Research Institute, Nanjing 210000, China; yangfan@seudongyin.com

³ School of Computer Science and Engineering, Southeast University, Nanjing 210000, China; 220246347@seu.edu.cn (Y.T.); 220222139@seu.edu.cn (Y.Y.)

⁴ Nanjing Longyuan Microelectronics Technology Co., Ltd., Nanjing 210000, China; 2240000044@student.must.edu.mo (L.L.); 230259971@seu.edu.cn (Y.C.)

⁵ Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications, Ministry of Education, Nanjing 210000, China

* Correspondence: 101010783@seu.edu.cn; Tel.: +86-137-0516-8696 (B.Z.)

Abstract

In the intelligent recognition of construction drawings, architectural components significantly vary in size, and small objects are easily overlooked. To address these two challenges, this paper proposes a multi-scale feature fusion method based on channel and spatial attention mechanisms. Specifically, the method constructs a hybrid feature pyramid structure (ArchFuse-Net) to enhance multi-scale feature representation. Subsequently, a small-object perception module is introduced to improve the recall rate of small components. Finally, the bounding box regression loss function and post-processing mechanisms are refined to boost detection accuracy and localization precision. Experimental results demonstrate that the proposed method effectively improves overall detection accuracy across varying sizes and significantly boosts the recall of small objects, providing a robust solution for architectural drawing analysis.

Keywords: engineering drawings; multi-scale; small objects; object detection

1. Introduction

Object detection in construction drawings faces two major challenges: component size variation and the frequent omission of small objects. As illustrated in Figure 1, component sizes in architectural drawings significantly vary. These elements range from small items, such as pile foundations and minor equipment, to large structures, such as tower cranes. This scale disparity causes uneven performance in conventional object detection methods across different target sizes. Specifically, small objects often lose critical features during the image downsampling process. This loss directly degrades their detection accuracy. Meanwhile, detection models tend to focus on extracting the salient features of large components. As a result, they neglect the fine-grained details of small objects. This imbalance ultimately leads to unstable overall detection accuracy.



Academic Editor: Antonio Caggiano

Received: 26 March 2026

Revised: 11 May 2026

Accepted: 17 May 2026

Published: 20 May 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.

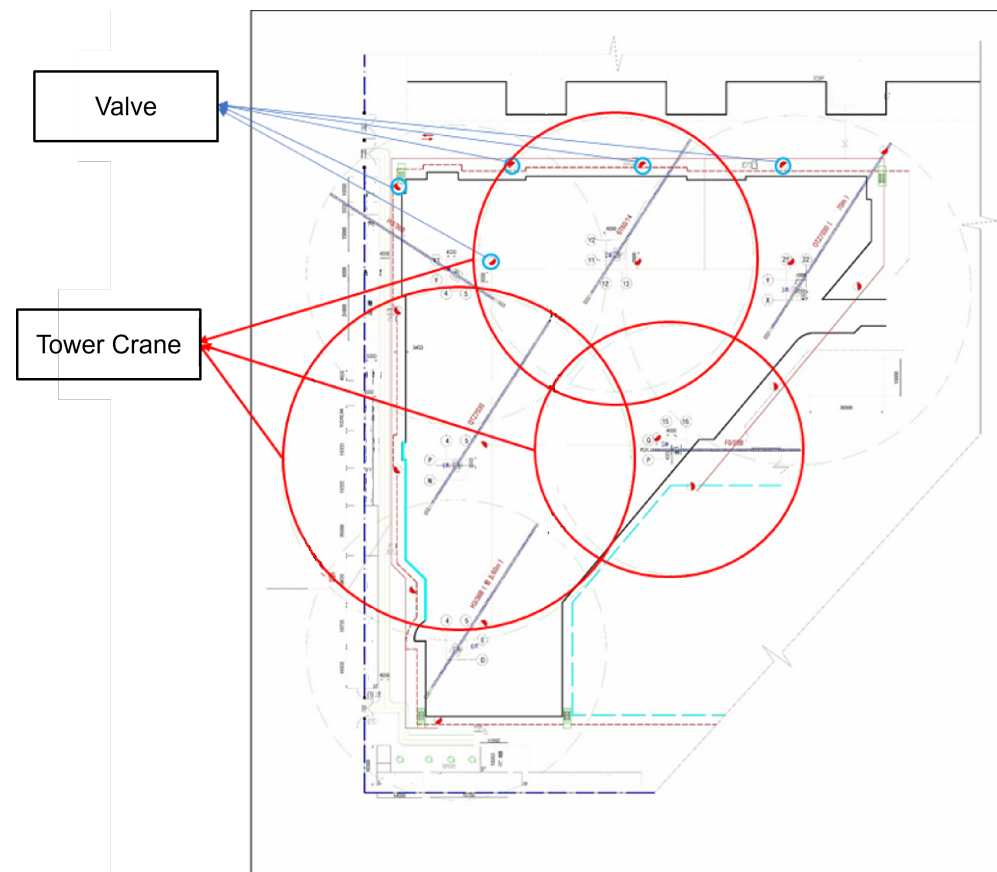


Figure 1. Illustration of component size variation.

To effectively address the large size variations in components and the difficulty of recognizing small elements, this paper proposes an attention-based multi-scale feature fusion network. This network, termed Architectural Multi-Scale Fusion Network (ArchFuse-Net), enhances the model's representational capability for multi-scale and small-object targets.

The significance of the proposed ArchFuse-Net extends beyond improving the efficiency of automated compliance review for construction site layout plans. These plans often feature extreme scale disparities, ranging from miniature components to large-scale equipment like tower cranes. ArchFuse-Net effectively resolves these scale challenges. As a result, this attention-based multi-scale fusion paradigm provides a robust and generalizable reference. It can be applied to broader image recognition tasks with dense small-object distributions and complex spatial hierarchies.

2. Related Work

The multi-scale feature fusion network aims to fully integrate feature information from different depth layers. This integration enhances the object detection model's ability to perceive multi-scale targets. In Convolutional Neural Networks (CNNs), shallow feature maps possess higher spatial resolution and smaller receptive fields. These properties make them more effective in preserving pixel-level details. Therefore, shallow features are critical for small-object detection. In contrast, deep feature maps have larger receptive fields. They encode rich semantic information and global context. This broad context facilitates the recognition of the overall structure of large-scale objects. Effectively fusing these two types of features preserves spatial details and enhances semantic representation. Consequently, it significantly improves detection performance in complex scenarios. This fusion strategy is grounded in the complementarity of hierarchical features. It constitutes the theoretical foundation of multi-scale feature processing methods.

At present, mainstream multi-scale feature fusion architectures follow two technical paradigms. The first paradigm is cross-scale interaction systems based on parallel branches. The second paradigm is hierarchical enhancement mechanisms based on serial skip connections. Parallel systems achieve information complementation by processing feature maps of different resolutions synchronously through multiple branches. In contrast, serial systems enhance semantic transmission efficiency in feature pyramids by leveraging cross-layer connections. Both approaches are founded on the collaborative utilization of features with different receptive fields. However, they substantially differ in their implementation. Parallel architectures emphasize horizontal fusion in the feature space. Meanwhile, serial designs focus on vertical information flow across hierarchical layers.

In recent years, deep learning models for object recognition have continuously evolved [1]. Advanced multi-scale feature fusion and attention mechanisms have achieved remarkable success in resolving complex scale variations across various specialized domains. For example, recent studies have effectively applied these techniques to medical image segmentation [2], drone-based small object detection [3], and sonar target recognition [4]. However, directly migrating these frameworks to architectural construction drawings often yields sub-optimal results. Construction site layout plans feature rigid geometric constraints and extreme scale imbalances. For instance, massive tower cranes and miniature valve components frequently coexist in the same drawing. General fusion strategies often wash out the shallow texture information of small objects in deep networks. Inspired by the aforementioned domain-specific advancements, the proposed ArchFuse-Net is specifically designed to tackle these extreme scale variations. It achieves this through a hybrid pyramid structure and a dedicated small-object perception module.

2.1. Networks Based on Parallel Branches

The Inception module innovatively introduced a multi-branch parallel convolutional architecture, which simultaneously extracts complementary features through convolutional kernels of different scales [5]. Its design breakthrough lies in the following aspects: the 1×1 convolution kernel performs channel-wise dimensionality reduction to control computational complexity; the 3×3 and 5×5 convolution kernels capture mid-level and high-level spatial features, respectively; and the max-pooling layer preserves salient activation responses. The outputs of all branches are concatenated along the channel dimension to form a composite feature representation. This width-expansion strategy effectively enhances the nonlinear representational capacity of the model, and its network structure is illustrated in Figure 2 [6].

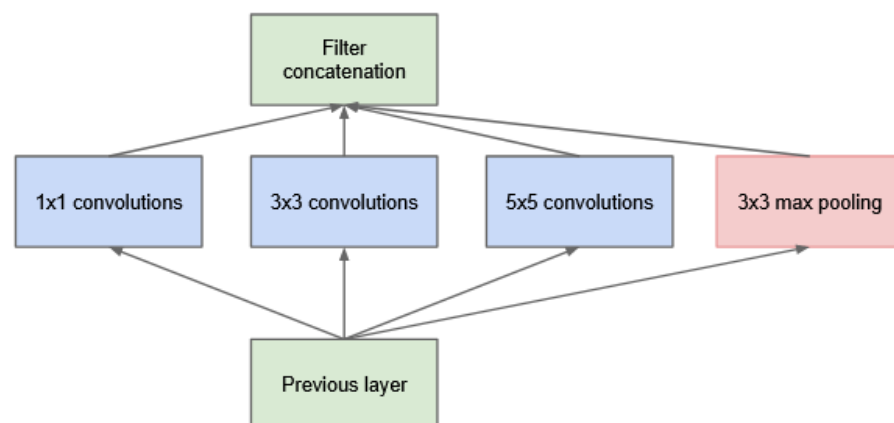


Figure 2. Inception architecture.

SPPNet inherits the concept of multi-scale processing while achieving architectural innovation [7]. Its core component, the Spatial Pyramid Pooling (SPP) layer, eliminates the constraint of input image size through hierarchical pooling operations: under a fixed network architecture, this layer can adaptively generate fixed-dimensional feature vectors [8]. In contrast to the parallel feature extraction mechanism of Inception, SPPNet focuses on optimizing scale compatibility at the post-processing stage.

In the field of image semantic segmentation, there are two typical solutions for multi-scale feature extraction. The Atrous Spatial Pyramid Pooling (ASPP) module proposed in DeepLabv2 model achieves feature fusion through a parallel multi-branch structure, with its core design centered on constructing atrous convolution layers with different receptive fields. This module employs a spatial pyramid pooling mechanism, where the sampling density is adjusted by modifying the rate parameter, thereby obtaining multi-scale features [9]. Such a design not only preserves the resolution of feature maps but also effectively expands the receptive field, significantly improving the model's accuracy in recognizing object boundaries.

The Pyramid Pooling Module (PPM) provides another approach. This structure processes input features through hierarchical pooling operations and fuses concatenated feature vectors from sub-regions at different scales. Compared with the ASPP module, the PPM requires no additional parameters; instead, it captures semantic features at multiple levels via multi-stage pooling operations. This method effectively integrates global context information while maintaining computational efficiency, which is of particular value for segmentation tasks in complex scenes.

2.2. Networks Based on Serial Skip Connections

Mainstream fusion schemes include architectures such as the Feature Pyramid Network (FPN) and U-Net. FPN constructs a feature pyramid through a dual-pathway structure: the bottom-up pathway extracts semantic information, while the top-down pathway restores spatial details [10]. U-Net adopts an encoder–decoder architecture with skip connections to achieve feature fusion. The encoder extracts semantic features through convolution and pooling operations, the decoder restores spatial resolution via upsampling, and the skip connections integrate multi-level features [11].

High-Resolution Network (HRNet) enhances fine-grained information extraction by maintaining high-resolution feature processing throughout the entire network. It employs four parallel branches to handle features at different scales while introducing supervisory mechanisms to accelerate convergence [12]. Path Aggregation Network (PANet) addresses the limitations of FPN by introducing an additional bottom-up pathway to transmit shallow features and by reducing the number of convolutional layers [13]. This bidirectional pathway design strengthens multi-scale feature fusion.

BiFPN, proposed by the Google Brain team, further improves parameter efficiency [14]. This architecture integrates shallow and deep features through bidirectional connections to construct multi-scale feature pyramids. By removing redundant nodes and repeatedly stacking modules, it reduces computational cost while enhancing fusion performance. Recursive-FPN, on the other hand, introduces an iterative feature processing mechanism, achieving breakthroughs across multiple computer vision tasks [15].

The proposed ArchFuse-Net is a hybrid network that combines parallel and serial architectures. It builds upon serial skip connections to transmit semantic information. Additionally, it introduces attention-based horizontal enhancements similar to parallel branches. This approach effectively integrates shallow spatial details with deep semantic features. Consequently, this hybrid design resolves the representational bottleneck of small-scale components in complex construction site layout plans.

3. Multi-Scale Feature Fusion Method

3.1. Multi-Scale Feature Fusion Network

Typical detector model architectures consist of a Backbone, a Neck, and a Head. The Backbone generates a Feature Pyramid; the Neck, located between the Backbone and the Head, is responsible for further processing and integrating the features extracted by the Backbone; the Head then predicts object positions and categories based on object scales. The ArchFuse-Net proposed in this chapter can be integrated into detector models with a Backbone–Neck–Head structure, with optimization primarily focused on the Neck component. The overall integration process is illustrated in Figure 3.

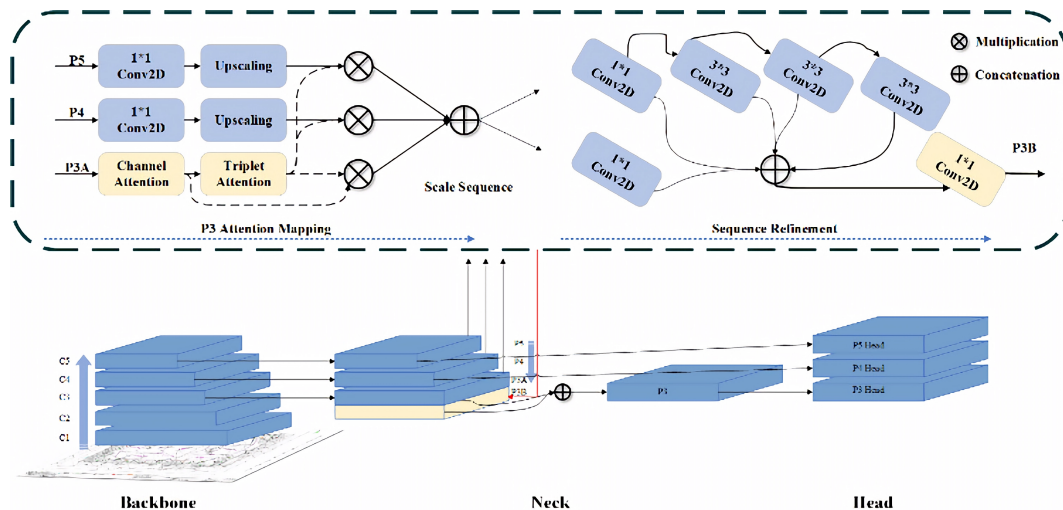


Figure 3. Schematic integration of ArchFuse-Net.

In conventional Neck modules, only a simple upsampling and concatenation process is included, a design that lacks sufficient theoretical foundation and practical effectiveness for small-object recognition. ArchFuse-Net is designed as a two-stage module: it generates a scale sequence from the feature pyramid involved in Head computation, refines it, and connects it to the dynamic head P_3 .

The formation of the scale sequence itself requires more sophisticated processing, for which an attention mechanism is introduced. In this chapter, the attention module is redesigned with a focus on small-object detection.

In the feature maps, P_3 retains the most informative cues for small objects. Therefore, to improve the feature representation of small objects, it is effective to construct a dedicated attention map within P_3 . Accordingly, the attention-enhanced scale-sequence features, denoted as S_i , are formally defined as:

$$S_i = M_s(P'_3) \otimes \text{Align}(P_i, \text{size}(P'_3)), \quad i \in \{3, 4, 5, \dots, L\} \quad (1)$$

where P_i denotes the i -th pyramid feature generated from the Backbone, and L denotes the total number of pyramid levels. The symbol $\otimes$ represents element-wise (Hadamard) multiplication. Since the spatial resolution of P_i decreases by a factor of 2 at each sequential level, direct multiplication is mathematically invalid due to dimension mismatch. Therefore, the $\text{Align}(\cdot)$ function is introduced to denote the spatial upsampling operation required to match the exact spatial dimensions of the P'_3 feature map before multiplication. Specifically, P_4 is upsampled by a factor of $2\times$, and P_5 is upsampled by $4\times$.

The P'_3 is the intermediate feature map refined by the channel attention map M_c :

$$P'_3 = M_c(P_3) \otimes P_3 \quad (2)$$

where M_c and M_s denote the channel attention map and the spatial attention map, respectively. First, the channel attention map for P_3 is generated as follows:

$$\begin{aligned} M_c(P_3) &= \sigma\left(\text{MLP}(\text{AvgPool}(P_3)) + \text{MLP}(\text{MaxPool}(P_3))\right) \\ &= \sigma\left(W_1(\text{SiLU}(W_0(P_{avg}^c))) + W_1(\text{SiLU}(W_0(P_{max}^c)))\right) \end{aligned} \quad (3)$$

AvgPool and MaxPool denote average pooling and max pooling, respectively; MLP denotes a linear multilayer perceptron. The symbols σ and SiLU denote activation functions. The channel attention map generated in this manner is applied to the input P_3 feature map, and the spatial attention map is then produced based on the refined feature map P'_3 :

$$\begin{aligned} M_s(P'_3) &= \sigma\left(f^{7 \times 7}([\text{AvgPool}(P'_3); \text{MaxPool}(P'_3)])\right) \\ &= \sigma\left(f^{7 \times 7}([P_{avg}^s; P_{max}^s])\right) \end{aligned} \quad (4)$$

At this point, the attention maps have been generated based on the P_3 feature map. Next, they are applied to all P_i feature maps involved in the Head. Compared with applying attention mechanisms to all network layers, this strategy substantially improves small-object detection performance. It not only increases accuracy but also achieves a lightweight design by reducing unnecessary computation, thereby ensuring the versatility of the module. Meanwhile, the attention-map generation in ArchFuse-Net is performed only on the P_3 feature map, because spatial information about small objects is more likely to appear in the P_3 map, the latter having relatively high resolution.

The spatial attention map derived from P'_3 is applied not only to P'_3 , but also to the other spatially aligned feature maps P_i . Consequently, the final scale-sequence feature maps constitute the set $\{S_i\}$. This approach concentrates attention on small objects. By applying the P_3 -derived attention map to all P_i feature maps participating in the Head, it not only bridges the semantic gap among feature layers but also enhances inference in the Head by preserving spatial information of small objects that might otherwise be lost at low resolutions.

3.2. Small-Object Perception Enhancement Method

To mitigate the missed-detection issue for small-sized components in drawings (smaller than 32×32 pixels), a Small-scale Recognition Augmentation (SRA) scheme is designed, comprising the following core components:

- **Shallow Feature Enhancement**
High-resolution feature preservation: An Extended Residual Block is added after the stage of the Backbone (at 1/4 resolution), where 3×3 Depthwise Separable Convolutions replace standard convolutions to retain fine textures while reducing computation.
- **Dynamic Sample Balancing Strategy**
Consistent Mixed Cropping (CMC): During data augmentation, regions densely populated with small objects are mixed with large-scale background regions at a ratio of 1:3, thereby enforcing the model's attention to local details. Specifically, pile-foundation-dense patches (512×512 pixels) are randomly stitched with tower-crane background patches (2048×2048 pixels) to generate training samples.

3.3. Boundary Localization Optimization Strategy

To address boundary ambiguities caused by densely arranged components of different sizes, the following boundary localization optimization strategies are employed:

3.3.1. Multi-Task Joint Learning

Semantic segmentation auxiliary branch: A lightweight Segmentation Head is appended to the output of ArchFuse-Net. Combined with the multi-scale feature fusion of ArchFuse-Net, it produces pixel-level component masks. For example, features from layers P2–P5 are upsampled and concatenated to output a segmentation map at 1/4 resolution, thereby assisting in distinguishing components of different shapes.

The boundary-sensitive loss adopts a composite loss, as shown in:

$$\mathcal{L} = \lambda_{det}\mathcal{L}_{EIou} + \lambda_{seg}\mathcal{L}_{Dice} + \lambda_{edge}\mathcal{L}_{Laplacian} \quad (5)$$

where λ_{det} , λ_{seg} , and λ_{edge} are empirical weighting coefficients. To balance the multi-task learning process without overwhelming the primary detection task, they are set to 1.0, 0.5, and 0.1, respectively. Since standard construction drawing datasets primarily provide bounding box annotations, the segmentation branch is supervised using pseudo-masks generated by filling the interior of the ground-truth bounding boxes. The $\mathcal{L}_{EIou}$ loss improves regression accuracy through center-distance penalties and aspect-ratio constraints. The $\mathcal{L}_{Laplacian}$ loss enforces alignment between predicted boundaries and high-frequency gradients of the annotations via Laplacian edge detection. Compared with CIoU, the improved EIou employs a stepwise computation strategy for handling the size parameters of the target box, decomposing the aspect-ratio factor into independent variables for dynamic adjustment. The loss comprises three core modules: overlap loss, center-distance loss, and width-height loss. The newly introduced width and height loss terms significantly accelerate convergence. The penalty term is formulated in:

$$\begin{aligned} \mathcal{L}_{EIou} &= \mathcal{L}_{IoU} + \mathcal{L}_{dis} + \mathcal{L}_{asp} \\ &= 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{C_w^2} + \frac{\rho^2(h, h^{gt})}{C_h^2} \end{aligned} \quad (6)$$

where $\rho(\cdot)$ denotes the Euclidean distance, and C_w and C_h are the width and height of the smallest enclosing rectangle covering the two boxes. EIou further incorporates a focal reweighting toward high-quality boxes, yielding $\mathcal{L}_{Focal-EIoU}$. From a gradient perspective, high-quality and low-quality boxes are separated, with the penalty defined in:

$$\mathcal{L}_{Focal-EIoU} = IoU^\gamma \mathcal{L}_{EIou} \quad (7)$$

where IoU follows the standard intersection-over-union definition $IoU = |A \cap B| / |A \cup B|$, and γ is a fixed hyperparameter that controls the suppression strength of outliers during training, which is empirically set to 0.5 in our experiments.

3.3.2. Post-Processing Optimization

Non-Maximum Suppression (NMS): NMS is a key post-processing technique for improving output precision. The algorithm filters redundant predicted boxes via an IoU-thresholding scheme and finally retains the candidate boxes with the highest confidence scores as the output, as illustrated in Figure 4.

Figure 4a presents the detection results for three double-leaf doors, among which only the central box satisfies the accuracy requirement. The remaining detection boxes exhibit localization bias and must be filtered by the Non-Maximum Suppression (NMS) algorithm. The final output is shown in Figure 4b, where the retained box achieves the optimal position match and target coverage.

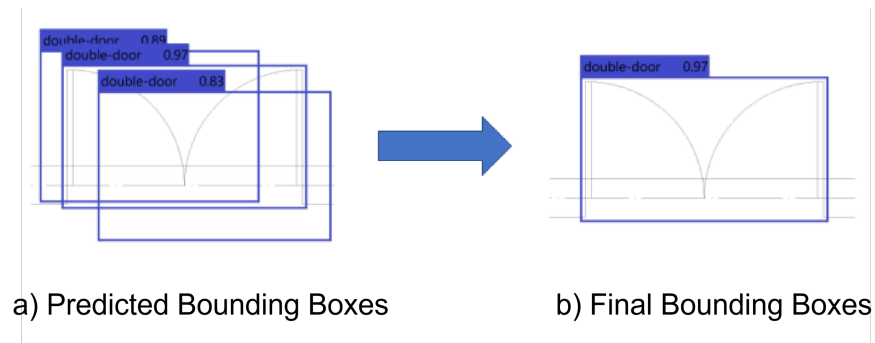


Figure 4. Prediction bounding-box mechanism.

Unlike the training phase, ground-truth annotations are inaccessible during inference. Therefore, the NMS algorithm iteratively selects the optimal boxes based purely on their predicted confidence scores. Let $\mathcal{B} = \{b_1, b_2, \dots, b_n\}$ be the set of candidate boxes and $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ be their corresponding confidence scores. The execution steps are mathematically formalized as follows:

1. Select the box b_m with the maximum confidence score s_m from $\mathcal{B}$, and add it to the final detection set $\mathcal{D}$.
2. Compute the Intersection over Union (IoU) between b_m and all remaining boxes in $\mathcal{B}$.
3. Remove any box b_i from $\mathcal{B}$ if $\text{IoU}(b_m, b_i) \geq \tau$, where τ is a predefined IoU threshold.
4. Repeat steps 1 to 3 until the candidate set $\mathcal{B}$ becomes empty.

By employing this iterative suppression strategy, only the bounding boxes with the highest local confidence are retained, thereby effectively eliminating duplicate predictions.

4. Experiments and Results

To evaluate the effectiveness of the proposed multi-scale feature fusion method for architectural component detection, this chapter conducts analyses from three aspects: experimental configuration, evaluation metrics, and comparative experiments. Moreover, comprehensive comparative evaluations across different model architectures are performed to verify the overall performance gains and robustness of the proposed method in practical detection settings.

YOLOv7 and YOLOv8, two mainstream object detection frameworks, are selected as baseline models for comparative evaluation. YOLOv7 serves as a representative P5-series model that balances accuracy and speed; its highest-output detection head corresponds to a feature map with a downsampling stride of 32. YOLOv8, a lightweight architecture proposed in recent years, offers favorable deployment adaptability and likewise belongs to the P5 series. To further assess performance under deeper feature hierarchies, YOLOv7-W6 is additionally introduced as a representative model whose feature outputs are extended to the sixth level (i.e., a downsampling stride of 64), thus belonging to the P6 series. This configuration affords a larger receptive field and stronger representational capacity, making it suitable for high-resolution inputs and large-object detection tasks.

ArchFuse-Net is integrated into YOLOv7, YOLOv7-X, YOLOv7-W6, YOLOv7n, and YOLOv8s, respectively, to evaluate its adaptability and performance gains across different model scales and feature depths. All models are trained via transfer learning using the officially released pretrained weights to enhance feature representation and accelerate convergence. Hyperparameters during training strictly follow the default settings of YOLOv7 and YOLOv8—including learning rate, batch size, optimizer selection, and data augmentation—to ensure fair comparisons. All training and testing are conducted in the experimental environment described in Chapter 4.

To ensure experimental reliability and to validate the generality of the proposed method, the dataset choice is adjusted given that BuildCAD800 is relatively small and cannot reflect performance at scale. This chapter therefore adopts the FloorPlanCAD dataset, which contains more than 15,000 floor plans spanning residential to commercial buildings; the annotations cover multi-scale detection targets. Tests are also performed on the BuildCAD800 dataset introduced in Chapter 4.

4.1. Implementation Details

To ensure experimental transparency and reproducibility, all models were implemented using the PyTorch (version 2.0.1) deep learning framework. Training and testing were conducted on a Linux server equipped with two NVIDIA GeForce RTX 3090 GPUs (24GB VRAM per GPU).

During the training phase, we strictly adhered to the default training protocols of the baseline frameworks (YOLOv7 and YOLOv8) to ensure fair comparisons. All models were trained for 300 epochs using the Stochastic Gradient Descent (SGD) optimizer. The initial learning rate was set to 0.01, with a momentum of 0.937 and a weight decay of 0.0005. A cosine annealing learning rate schedule was applied to ensure stable convergence. The batch size was set to 32, which fully utilized the dual-GPU memory capacity. The input images were resized to 640×640 pixels for standard models, and 1280×1280 pixels for the P6-level YOLOv7-W6 model. For data augmentation, in addition to the proposed Consistent Mixed Cropping (CMC) strategy, standard techniques including Mosaic, random horizontal flipping, and HSV color space adjustments were employed. The dataset was randomly partitioned into training (80%), validation (10%), and testing (10%) subsets.

4.2. Design of Evaluation Metrics

The three types of Average Intersection over Union (IoU) metrics were measured as follows. In addition to the average AP calculated over the IoU range from 0.50 to 0.95, AP50 and AP75 were computed at IoU thresholds of 0.50 and 0.75, respectively. Furthermore, the AP values for small, medium, and large objects were evaluated based on object size categories, denoted as AP_S , AP_M , and AP_L , as detailed in Table 1.

Table 1. Definitions of Evaluation Metrics.

Metric	Definition
AP	Mean Average Precision over IoU thresholds from 0.50 to 0.95 with a step size of 0.05
AP50	Average Precision at IoU = 0.50
AP75	Average Precision at IoU = 0.75
AP_S	Average Precision for small objects ($area \leq 32^2$ pixels)
AP_M	Average Precision for medium objects ($32^2 \leq area \leq 96^2$ pixels)
AP_L	Average Precision for large objects ($area > 96^2$ pixels)

Comprehensive Detection Performance Metric (AP@[0.50:0.95]): The average precision (AP) is calculated over the IoU threshold range from 0.50 to 0.95 (with a step size of 0.05), representing the mean AP across 10 IoU thresholds (0.50, 0.55, ..., 0.95) for all object categories and sizes. This metric reflects the model's overall performance under varying localization accuracy requirements.

IoU-based Submetrics: AP50 is the average precision at an IoU threshold of 0.50, used to assess the model's detection capability under relaxed localization constraints.

AP75 is the average precision at an IoU threshold of 0.75, used to reflect the model's detection accuracy under stricter localization constraints (i.e., the bounding box must overlap significantly).

Object Size-based Submetrics: AP_S represents the AP value for small objects, calculated for objects with an area smaller than 32^2 pixels. AP_M represents the AP value for medium objects, calculated for objects with an area ranging from 32^2 to 96^2 pixels. AP_L represents the AP value for large objects, calculated for objects with an area larger than 96^2 pixels.

4.3. Experimental Results and Analysis

Table 2 and Figure 5 present the performance comparison between the baseline models and the ArchFuse-Net module proposed in this chapter for the object detection task. The comparison covers four P5-series models (YOLOv7, YOLOv7-X, YOLOv7n, YOLOv8s) and one P6-level model (YOLOv7-W6).

Table 2. Performance comparison of Baselines (YOLOv7/YOLOv8) and ArchFuse-Net on FloorPlanCAD.

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv7	51.2	69.7	55.5	35.2	56	66.7
YOLOv7-X	52.9	71.1	57.5	36.9	57.7	68.6
YOLOv7-W6	54.6	72.3	59.5	40.1	59	68.6
YOLOv7n	37.4	52.9	40.3	18.6	41	53.5
YOLOv8s	44.9	62.1	48.3	25.9	49.9	61
ArchFuse-Net+YOLOv7	52.4	70.9	58	37.2	57.2	65.5
ArchFuse-Net+YOLOv7-X	55.3	73.1	60.3	36.1	60.1	72
ArchFuse-Net+YOLOv7-W6	55.8	73.9	61.9	39.9	61	67.4
ArchFuse-Net+YOLOv7n	37.4	54.1	41.1	22.2	43.4	50.5
ArchFuse-Net+YOLOv8s	45.3	63	51.5	28.9	51.5	58.5

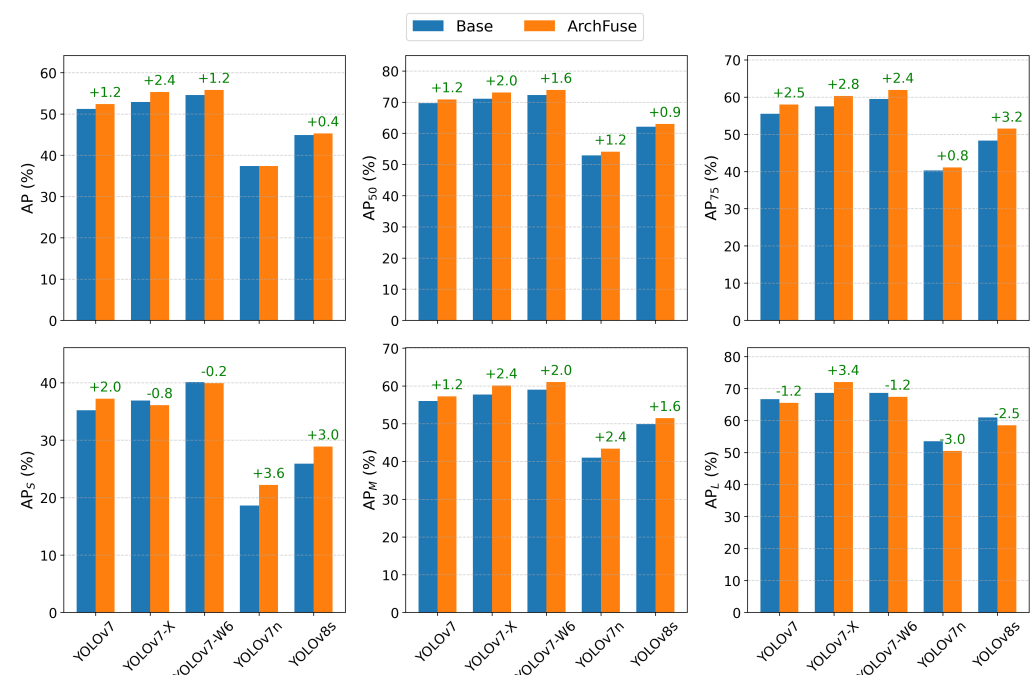


Figure 5. Visualization of performance comparison of Baselines (YOLOv7/YOLOv8) and ArchFuse-Net on FloorPlanCAD.

Across all metrics, ArchFuse-Net delivers performance gains of varying magnitudes across different model architectures, validating its effectiveness for multi-scale object detection. In terms of overall detection performance, ArchFuse-Net increases the AP of YOLOv7-X from 52.9 to 55.3 (+2.4%), that of YOLOv7-W6 from 54.6 to 55.8 (+1.2%), and yields a 1.2% improvement for the vanilla YOLOv7. In addition, for the lightweight YOLOv8s, the overall AP improves by 0.4%. These results indicate that ArchFuse-Net

enhances detection capability across models of different scales, exhibiting particularly good scalability in models whose Backbones have limited representational capacity.

For small-object detection (AP_S), ArchFuse-Net also brings notable improvements: the AP_S of YOLOv7 increases from 35.2 to 37.2 (+2.0%), YOLOv7n from 18.6 to 22.2 (+3.6%), and YOLOv8s from 25.9 to 28.9 (+3.0%). Although the AP_S of YOLOv7-W6 decreases slightly (from 40.1 to 39.9), both the strict-precision metric (AP_{75}) and large-object accuracy (AP_L) improve, suggesting that the module remains robust in complex scenarios.

It is worth noting that while the accuracy for small and medium objects significantly improves, the detection performance for large objects (AP_L) experiences a slight degradation in certain models (e.g., YOLOv7 and YOLOv8s). This phenomenon is a reasonable trade-off: as the attention mechanism forces the network to focus more heavily on the shallow features of densely packed small components, it introduces minor perturbations to the global receptive field required for large-scale structures. Nevertheless, collectively, these results demonstrate that ArchFuse-Net substantially compensates for the critical perceptual blind spots of conventional detectors when handling miniature components, making it particularly suitable for construction drawings characterized by intricate structural details. In addition to the detection accuracy, the computational efficiency of the models is also evaluated, as summarized in Table 3.

Table 3. Efficiency performance comparison of Baselines (YOLOv7/YOLOv8) and ArchFuse-Net on FloorPlanCAD.

Model	Size	#Params (M)	FLOPs (G)	Batch32 (ms)	Batch1 (fps)
YOLOv7	640	36.9	104.7	7.1	77
YOLOv7-X	640	71.3	189.9	8.9	63
YOLOv7-W6	1280	70.4	89.9	13.7	47
YOLOv7n	640	3.2	8.7	1.6	83
YOLOv8s	640	11.2	28.6	2.8	87
ArchFuse-Net+YOLOv7	640	39.5 (+2.6)	125.9 (+21.2)	7.7 (+0.6)	69 (−8)
ArchFuse-Net+YOLOv7-X	640	74.2 (+2.9)	219.5 (+29.6)	9.9 (+1.0)	58 (−5)
ArchFuse-Net+YOLOv7-W6	1280	84.4 (+14.0)	123.4 (+33.5)	19.6 (+5.9)	35 (−12)
ArchFuse-Net+YOLOv7n	640	3.3 (+0.1)	9.9 (+1.4)	1.9 (+0.3)	68 (−15)
ArchFuse-Net+YOLOv8s	640	11.6 (+0.4)	33.3 (+4.7)	3.3 (+0.5)	66 (−21)

In terms of concrete performance, ArchFuse-Net yields AP gains of +1.2%, +2.4%, and +1.2% on YOLOv7, YOLOv7-X, and YOLOv7-W6, respectively. For lightweight models, ArchFuse-Net achieves a +0.4% AP improvement on YOLOv8s, while YOLOv7n maintains the same AP but exhibits increases in AP_{50} and AP_{75} . These results indicate that ArchFuse-Net is not only compatible with lightweight models but also delivers clear advantages in larger architectures, with particularly notable improvements on YOLOv7-X. Compared with the original baselines, the feature enhancement strategy of ArchFuse-Net effectively improves overall detection performance, demonstrating strong generality and transferability.

Based on the above experimental results, we conclude that ArchFuse-Net consistently improves detection performance across models of different scales and architectures. In particular, it exhibits higher adaptability and performance potential for multi-scale detection tasks.

4.4. Ablation Study

YOLOv7 was used as the reference detector for the ablation study because it serves as a representative baseline in the main comparative experiments. More importantly, using a fixed detector architecture allows the proposed components to be introduced progressively while keeping the backbone, detection head, data split, training protocol, and evaluation setting consistent. This design isolates the effect of each component and avoids attributing performance changes to architectural differences among different detector

families. The cross-architecture adaptability of ArchFuse-Net is assessed separately in the comparative experiments.

To further clarify the contribution of each component in the proposed framework, a component-wise ablation study was conducted under this fixed YOLOv7 setting. All variants were trained and evaluated on the same dataset split with identical input resolution, optimizer, learning rate schedule, batch size, training epochs, and pretrained weights. Starting from the original YOLOv7 baseline, the proposed components were progressively introduced, including the multi-scale feature fusion neck, the P3-guided attention refinement module, the small-object enhancement strategy, the localization-oriented loss adaptation, and the post-processing calibration. The post-processing calibration was evaluated at the inference stage without retraining the model, so that its contribution could be separated from the training-stage components.

The ablation settings are defined as follows. A0 denotes the original YOLOv7 baseline. A1 introduces only the multi-scale feature fusion neck. A2 further incorporates the P3-guided channel-spatial attention refinement. A3 adds the small-object enhancement strategy, including shallow-feature reinforcement and consistent mixed cropping. A4 introduces the localization-oriented loss adaptation. Finally, A5 denotes the complete ArchFuse-Net configuration with post-processing calibration.

As shown in Tables 4 and 5, the multi-scale feature fusion neck provides the initial improvement over the baseline. Compared with A0, A1 improves AP from 51.2 to 51.7 and AP₇₅ from 55.5 to 56.2, indicating that cross-level feature aggregation enhances the basic representation of construction components with different scales. However, the improvement is moderate, suggesting that feature fusion alone is insufficient to fully address the extreme scale imbalance in construction drawings.

Table 4. Ablation settings of the proposed components on YOLOv7.

Model	Fusion Neck	P3 Attention	Small-Object Enhancement	Loss Adaptation	Post-Processing
A0	–	–	–	–	–
A1	✓	–	–	–	–
A2	✓	✓	–	–	–
A3	✓	✓	✓	–	–
A4	✓	✓	✓	✓	–
A5	✓	✓	✓	✓	✓

Table 5. Detection performance of the ablation variants on YOLOv7.

Model	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
A0	51.2	69.7	55.5	35.2	56.0	66.7
A1	51.7	70.1	56.2	35.7	56.6	66.4
A2	52.1	70.5	56.8	36.8	56.9	66.0
A3	52.3	70.7	57.1	37.4	57.1	65.8
A4	52.4	70.8	57.9	37.3	57.2	65.6
A5	52.4	70.9	58.0	37.2	57.2	65.5

The introduction of the P3-guided attention refinement module in A2 leads to a more evident improvement in small-object detection. AP_S increases from 35.7 in A1 to 36.8 in A2, while the overall AP further increases to 52.1. This result supports the motivation of using the high-resolution P3 feature map to guide attention generation, because small components in construction drawings are more likely to retain discriminative spatial cues in shallow feature maps.

After adding the small-object enhancement strategy in A3, APS further increases to 37.4, which is the highest small-object AP among all variants. This indicates that the proposed small-object-oriented training strategy effectively increases the model's exposure to densely distributed small components and helps reduce missed detections. The improvement from A1 to A3 shows that the major gain in small-object detection mainly comes from the combination of P3-guided attention and small-object enhancement.

The localization-oriented loss adaptation in A4 mainly improves the stricter localization metric. Specifically, AP75 increases from 57.1 in A3 to 57.9 in A4, while the overall AP remains stable. This suggests that the modified localization objective is more beneficial for improving bounding-box accuracy under stricter IoU requirements than for increasing coarse detection recall.

The post-processing calibration in A5 produces only a marginal change. Compared with A4, AP remains unchanged, AP50 increases slightly from 70.8 to 70.9, and AP75 increases from 57.9 to 58.0. This indicates that the main performance gains are obtained from feature representation and training-stage optimization rather than inference-time threshold adjustment.

It should also be noted that the complete model slightly reduces APL from 66.7 to 65.5. This suggests that the small-object-oriented design may introduce a mild trade-off for large-object detection. Therefore, ArchFuse-Net should not be interpreted as a universally superior replacement for all object scales, but rather as a task-specific enhancement for construction drawings where small and densely distributed components are critical.

To further examine the computational cost introduced by the proposed components, the efficiency of the main ablation variants was also compared, as shown in Table 6.

Table 6. Efficiency comparison of the ablation variants on YOLOv7.

Variant	#Params (M)	FLOPs (G)	Batch32 (ms)	Batch1 (FPS)
A0	36.9	104.7	7.1	77
A1	38.0	114.2	7.3	74
A2	39.2	123.8	7.6	70
A3	39.2	123.8	7.6	70
A4	39.5	125.9	7.7	69
A5	39.5	125.9	7.7	69

The efficiency results show that ArchFuse-Net introduces additional computational cost. Compared with the YOLOv7 baseline, the complete A5 model increases the parameter count from 36.9 M to 39.5 M and FLOPs from 104.7 G to 125.9 G. The inference speed decreases from 77 FPS to 69 FPS. These results confirm that the proposed method should be viewed as an accuracy–complexity trade-off. It is more suitable for scenarios where improved small-object detection and localization accuracy are prioritized over maximum inference speed.

Overall, the ablation study demonstrates that the multi-scale fusion neck provides the basic performance improvement, while the P3-guided attention module and the small-object enhancement strategy contribute most clearly to small-object detection. The loss adaptation mainly improves localization-sensitive metrics, and the post-processing calibration provides only marginal additional gains. These findings support the design motivation of ArchFuse-Net for construction drawing detection under extreme scale imbalance.

5. Conclusions and Future Work

This paper targets the challenges of pronounced specification span among architectural components and frequent missed detections of small objects, and proposes a multi-scale co-operative detection and small-object discrimination optimization approach. First, a hybrid

Feature Pyramid Network (ArchFuse-Net) is constructed, in which a tri-directional feature interaction architecture—comprising top-down, bottom-up, and fuse-split paths—enables cross-hierarchical feature enhancement to improve the model’s perception of multi-scale targets. A Small-scale Recognition Augmentation (SRA) module is designed by integrating shallow-feature reinforcement, a context attention mechanism, and a hard-sample reweighting strategy, effectively improving the recall of tiny components. A joint segmentation–detection framework is further developed together with an improved EIoU loss function to simultaneously boost boundary localization accuracy and semantic discrimination capability. Through a quantitative evaluation protocol, comprehensive comparative analyses, and a systematic component-wise ablation study, the contribution of the proposed framework is systematically assessed. Experimental results demonstrate that the proposed multi-scale fusion method effectively improves overall average precision and significantly boosts the recall and localization accuracy of miniature components. These performance gains involve practical engineering trade-offs. The small-object-oriented design introduces a measurable increase in computational complexity, including higher parameter counts, FLOPs, and reduced inference frame rates. Additionally, it may cause a slight performance variation in large-object detection for certain architectures. Consequently, ArchFuse-Net serves as a targeted and robust engineering solution. It provides practical deployability for real-world compliance review applications where the accurate perception of tiny, dense architectural details is prioritized over maximum inference speed.

Author Contributions: Conceptualization, H.Z.; methodology, H.Z., Y.Y. and B.Z.; software, Y.T. and Y.C.; validation, Y.C. and F.Y.; formal analysis, J.Y.; investigation, J.Y.; data curation, L.L.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.T.; visualization, L.L.; supervision, B.Z.; project administration, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Hainan Province Science and Technology Special Fund (Grant No. ZDYF2023GXJS150), the National Key R&D Program of China (Grant No. 2023YFC3806004) and the National Natural Science Foundation of China (Grant No. 62373104).

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to commercial confidentiality agreements with the collaborating enterprises.

Conflicts of Interest: Author Lulu Liang was employed by the company Nanjing Longyuan Micro-electronics Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Tsirtsakis, P.; Zacharis, G.; Maraslidis, G.S.; Fragulis, G.F. Deep learning for object recognition: A comprehensive review of models and algorithms. *Int. J. Cogn. Comput. Eng.* **2025**, *6*, 298–312. [[CrossRef](#)]
2. Xu, Z.; Tian, B.; Liu, S.; Wang, X.; Yuan, D.; Gu, J.; Chen, J.; Lukasiewicz, T.; Leung, V.C.M. Collaborative attention guided multi-scale feature fusion network for medical image segmentation. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 1857–1871. [[CrossRef](#)]
3. Han, G.; Guo, C.; Li, Z.; Zhao, H. Small object detection based on microscale perception and enhancement-location feature pyramid. *IEEE Trans. Cogn. Dev. Syst.* **2024**, *16*, 1982–1996. [[CrossRef](#)]
4. Yang, N.; Li, G.; Wang, S.; Wei, Z.; Ren, H.; Zhang, X.; Pei, Y. SS-YOLO: A lightweight deep learning model focused on side-scan sonar target detection. *J. Mar. Sci. Eng.* **2025**, *13*, 66.
5. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
6. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.

7. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
8. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
9. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
10. Lin, T.Y.; Dollár, 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. 2117–2125.
11. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
12. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [[CrossRef](#)] [[PubMed](#)]
13. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
14. 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, 13–19 June 2020; pp. 10781–10790.
15. Qiao, S.; Chen, L.C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10213–10224.

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.