# Integration of Multiscale Fusion and Cross-scale Attention Refinement for Enhanced Target Detection Using MSFNet

Xiaofang Liao[1,*], Xinnan Liu [2]
[1]Intelligent Information Research Institute, South China Business College Guangdong University of Foreign Studies, Guangzhou 510545, Guangdong, China
[2]Guangdong Technology College, Zhaoqing 526100, Guangdong, China
E-mail: lxf_liaoxiaofang@hotmail.com
[*]Corresponding author

**Keywords:** multiscale feature fusion**,** deep learning, target recognition**,** attention mechanism**,** object detection**,** Convolutional Neural Networks (CNNs)

**Received:** June 27, 2025

*Object recognition across varying scales remains a persistent challenge in computer vision, especially in scenes with occlusion, low contrast, and diverse spatial resolutions. Conventional convolutional neural networks with fixed receptive fields often fail to capture both fine-grained details and high-level contextual cues. This study focuses on developing a scale-adaptive detection framework to overcome these limitations. The proposed MSFNet (Multiscale Fusion Network) employs a Dual-Stream Convolutional Backbone to extract low-level and high-level features in parallel. A Scale-Adaptive Feature Fusion Module (SAFFM) integrates multiscale representations through dynamic, scale-aware weighting. A Cross-Scale Attention Refinement (CSAR) module enhances discriminative features and suppresses irrelevant or redundant information. The architecture operates in an end-to-end fashion and is optimized for detection accuracy and real-time inference speed. Experimental evaluation on MS COCO 2017 and PASCAL VOC 2012 reports 47.3% AP and 81.5% mAP, respectively. Performance exceeds Faster R-CNN, YOLOv5, and RetinaNet by +3.8%, +4.5%, and +3.2% AP on the COCO benchmark. MSFNet provides a scalable, accurate, and computationally efficient approach for multiscale object recognition, enabling deployment in real-time applications such as autonomous driving, intelligent surveillance, and remote sensing.*

*Povzetek: MSFNet izboljša večlestvično prepoznavanje objektov in dosega višjo natančnost kot obstoječi modeli ob učinkovitem delovanju v realnem času.*

## 1 Introduction

There have been considerable changes in the field of computer vision [1] thanks to significant advances in artificial intelligence and deep learning. The merging of these two fields has caused this alteration. The alteration that has occurred is the result of several different scenarios. Due to the advancements made in these areas, robots can now perform remarkable tasks such as locating objects, categorizing images, and interpreting situations. These skills can be applied to functions that are either easy or challenging. It was only recently that it became possible to learn these skills. One of these jobs is called target recognition. It means figuring out what parts of a picture are there and putting them in the proper group [2]. One of the jobs in this group is target recognition. This group contains jobs that involve target recognition. Identifying targets is, without a doubt, one of the most essential parts of the many apps that are utilized in the real world. Some examples of this type of technology are medical imaging, driverless cars, aerial remote sensing, and artificial intelligence surveillance. But this list doesn't cover everything, these are just a few of the various ways to use them.

Target recognition is presently confronted with a challenge defined by the complexity of accurately identifying targets within a scene at diverse scales [3]. This problem is currently being solved in the field of target recognition. People around the world are currently trying to overcome this problem. This is one of the problems individuals are currently facing, even though significant progress has been made in this area, the fact that this problem remains a considerable issue can't be easily resolved. The scope of the researched topic is highly sensitive to the substantial influence of a multitude of varying circumstances, each possessing the capacity for a significant impact. Some examples of things that belong under this group include the distance of the camera from the subject, the camera's resolution, the level of zoom, and the angle of view.

On the other hand, large targets can enter the receptive area and conceal objects that are close to them. This is not what you might think. Smaller-scale targets are less likely to be seen than larger-scale ones, which have a better chance of being caught [4]. Standard CNN-based models struggle with this type of scale change, as they are often designed with fixed receptive fields and may fail to

capture essential details at varying spatial resolutions [5]. Additionally, these models are typically constructed with fixed receptive fields already in place. This is because fixed receptive fields are commonly used to make these models, which is why this is the case. Still, several models can accurately capture the level of detail required [6].

To address this complexity, researchers have explored various approaches. These systems encompass a multitude of distinct perspectives. There are many various ways to do this, including multiscale learning and feature fusion. Using these strategies, which include combining feature representations from different network layers or parallel branches, can help improve detection performance [7]. These solutions entail the amalgamation of two or more feature representations. This strategy is a good alternative to think about. Let it be the center of attention. U-Net designs and Pyramid Networks (FPN) are two examples of techniques that often fail to utilize the semantic richness of deep levels and the spatial resolution of shallow layers [8]. These are also instances of tactics that people frequently use. There are two examples of tactics, and both are plans. Both implementations demonstrate various types of network configurations. Both approaches being talked about here are examples of systems that are used regularly. However, even though these methods have proven effective, this is what has happened after they were implemented. It is also feasible for naive fusion algorithms to include information that is not necessary or already present, which may compromise the overall recognition performance [9]. This is something that could happen. This may result in a decline in overall recognition performance. In this case, the recognition might not be as accurate, one possible consequence of this is that the recognition may not be as precise as it once. This could likely lead to a decrease in the recognition's level of precision.

One of the offerings will be a Multiscale Fusion Network, also known as MSFNet. This network will be open to everyone can use this network, Researchers are now exploring a new method for identifying things that consider their intended use locations. A strategy that employs deep learning and integrates features from multiple scales can be used to achieve a multitude of objectives [10]. Working on this framework, aim to solve the problems discussed in more depth in the following paragraph. The organization's most notable contribution to the field is the scale-adaptive dual-stream architecture that MSFNet has used. This architecture enables the integration of components from both low-level, detail-oriented routes and high-level, context-aware pathways with equal significance. This can be accomplished using technology. Also, it's conceivable that pieces will be grabbed from both kinds of pathways. This is something that can happen to improve the model's ability to focus on the most critical parts, it is essential to incorporate the Cross-Scale Attention Refinement (CSAR) module into the framework discussed earlier. This will enable the model to focus more on the most critical aspects. This module can effectively block out background noise and highlight areas vital to the target by using a method that involves constantly adjusting the weights of feature contributions across different scales. This is done by highlighting areas that are important to the goal.

## 1.1 Problem statement

Object recognition in complex visual environments is significantly hindered by scale variance, occlusion, and background clutter. Conventional CNN-based detectors with fixed receptive fields often fail to capture essential features across different scales, resulting in reduced accuracy for small or partially occluded targets. There is a need for a scale-adaptive, noise-resilient detection framework capable of maintaining both high recognition accuracy and real-time processing speed.

## 1.2 Objectives

1. To design and implement a multiscale object recognition architecture integrating scale-adaptive fusion and cross-scale attention mechanisms.
2. To evaluate the proposed MSFNet against established baselines on large-scale benchmarks with diverse scale variations.
3. To ensure a balance between recognition accuracy and computational efficiency for real-time applicability.

The primary significance of the paper are:

- In contrast to fixed fusion techniques (e.g., FPN, BiFPN) that are unable to modify weighting on a per-instance basis, a scale-adaptive fusion mechanism (SAFFM) makes dynamic emphasis on pertinent resolutions based on object size and context by learning per-scale feature weights from channel descriptors.
- With the help of global average pooling and lightweight convolution, this effective cross-scale attention refinement module (CSAR) explicitly models spatial and channel correlations across scales, providing the advantages of cross-scale attention without the significant computational load of transformer-based or dense-attention architectures.
- A dual-stream convolutional backbone that simultaneously maintains high-level semantics and fine-grained information, enhancing small-object recall while preserving competitive inference speed.

MSFNet's recognition accuracy has improved significantly, especially for items that change size, are partially occluded, or have low contrast. This goal is achieved by utilizing adaptive multiscale learning and addressing the limitations of earlier methods. This research contributes to the advancement of visual recognition systems that are more advanced and robust than their predecessors. The ramifications of this issue transcend the

domain of academic inquiry and permeate the sphere of practical application in edge devices and real-time systems.

### 1.3 Research questions

1. Can a dual-stream convolutional backbone effectively capture and integrate low-level spatial details with high-level semantic context for multiscale recognition?
2. Can adaptive feature fusion and cross-scale attention refinement improve recognition accuracy across diverse object sizes and visual conditions without sacrificing inference speed?

## 2 Literature survey

The use of multiscale feature learning represents a significant step forward, potentially enhancing the performance of systems that locate targets and identify objects [11]. This represents an important step forward that could substantially enhance the performance of these systems. Taking this critical step forward could dramatically improve the performance in question, which is why it is such a crucial step forward. Due to this development, these systems may become significantly more accurate. Researchers have invested considerable time and effort in developing architectural design and fusion strategies to help people read scale-variant targets more accurately across a wide range of visual settings. This has been done to help them read targets of different sizes more effectively. They have done these things to make it easier for them to reach their goals. To conclude the process, you need to use a variety of various methods.

### 2.1 Lightweight object detection models

A novel methodology, initially responsible for establishing the foundation for multiscale learning, employed a method known as Feature Pyramid Networks (FPN). Due to their capacities, these networks were able to construct pyramidal hierarchies of feature maps that were interconnected throughout the entire network [12]. Because of this, it is much easier to choose items that aren't very valuable in a step-by-step way. Even while these hierarchies are effective, they are difficult to change due to their inflexibility, which makes it impossible to adapt to situations that are constantly evolving and growing. Over the past few years, the FPN paradigm has undergone significant changes. Tan et al. [13] were the ones who first told the world about BiFPN technology. A lot of work goes into making sure that feature flows are included in the system. Tan and his team conceived the idea for BiFPN while conducting research. They aimed to develop object detectors that function with mobile devices. This method can successfully balance multiscale features with learnable weights, and it works well.

### 2.2 Multiscale feature fusion

Researchers have begun to apply attention-focused methods to overcome the limitations of algorithms that only work with convolution. This is done to circumvent the limitations of these algorithms. This is because tactics that rely on attention can get around these problems. Chen et al. (2022) [14] introduced the Selective Feature Fusion (SFF) model to the audience during their inquiry. This solution utilizes channel-wise attention, allowing you to select the relevance of multiscale information in real-time real-time. The approach makes this possible. One significant advantage is that it makes it easier to locate items in crowded areas. This is a big plus. This method requires focusing on each channel sequentially, which is tantamount to compounding the offense. In 2023, Gao and his team [15] developed a network capable of performing attention mappings in both size and spatial dimensions. A single network oversees putting attention mappings into action. People usually refer to this network as CSANet, which stands for Cross-Scale Attention Network. A method similar to the one used to build the previous network was used to create this one. Additionally, this makes it easier to locate objects in aerial photos with greater accuracy and enhances the location's accuracy. This is an unavoidable outcome inherent to the nature of aerial photography.

The ScaleEqualNet approach, first proposed by Zhang et al. (2022) [16], now includes a scale calibration layer. They were the ones who came up with the algorithm in the first place. The purpose of building this layer was to mitigate the effects of size changes, which were most noticeable in certain areas. When the decision is made, this layer will be built. This research aimed to provide participants with the opportunity to gain a more comprehensive understanding of scale-adaptive models. This was done by making these systems more useful. When this approach was applied to datasets with significant heterogeneity between objects, such as MS COCO and DOTA, it significantly improved recognition accuracy compared to previous methods. This enhancement was realized to a far greater extent than before. The application of this methodology to these datasets facilitated the achievement of this enhancement. To achieve the desired results, they applied this method to datasets to effect this improvement.

### 2.3 Transformer-based architectures

Jiang et al. (2023)[17] researched transformer-based systems to get the best outcomes from concurrent multiscale learning. This was what they wanted to learn from their research. These models, on the other hand, needed more important computer resources than the ones that came before them. This was because they employed global self-attention methods to connect features associated with distinct levels.

Wang et al. (2023) [18] discovered a correlation between attention fusion layers and multi-resolution convolutional backbones. This finding was made in the

context of high-resolution aerial photo challenges. It was only recently that people realized the importance of this discovery. A recent discovery has been made regarding the existence of this link. Their research demonstrates that integrating geographical information with semantic abstraction at various scales can enhance the precision of item detection in satellite data. This is achievable. The fact that it was able to do this job well was proof that this was indeed the case. Our findings have made it clear how important it is to use flexible fusion methods and procedures. This is especially true when it comes to the problems that come up when trying to identify things on a large scale. Table 1 shows the summary of the comparative analysis of multiscale detection methods.

Table 1: Comparative summary of representative multiscale detection methods

| Method (ref) | Key idea | Strengths | Limitations | How MSFNet differs |
|---|---|---|---|---|
| FPN [12] | Top-down feature pyramid with lateral connections | Simple, low overhead; improves small-object recall | Fixed fusion rules; limited adaptive weighting | Learns scale-aware weights instead of fixed fusion |
| BiFPN [13] | Weighted bidirectional feature flow | Learnable per-level weights; efficient | Limited cross-scale refinement; potential redundancy | SAFFM provides per-scale attention from channel descriptors |
| Selective Feature Fusion [14] | Channel-wise gating for fusion | Fine-grained channel selection | Single-point fusion; no explicit cross-scale spatial modeling | CSAR adds spatial+channel refinement across scales |
| CSANet / Cross-scale attention [15] | Attention across scale and spatial dims | Better high-res localization | Computational cost; dense attention | CSAR uses efficient conv+GAP-based masks to reduce cost |
| ScaleEqualNet [16] | Scale calibration layers | Reduces scale variance | Less robust in mixed-scale scenes | MSFNet adapts weights per-instance scale via SAFFM |
| Transformer-based aggregation [17] | Global self-attention across levels | Strong modeling of long-range dependencies | Heavy compute, memory | MSFNet attains cross-scale correlation at far lower cost |
| Attention-driven aerial fusion [18] | Multi-resolution attention for remote sensing | Robust in cluttered aerial scenes | Domain-specific tuning | MSFNet generalizes to COCO/VOC while keeping model compact |

These changes don't alter the fact that a central problem remains unaddressed, which exacerbates the situation. The challenge is determining how to integrate qualities in a manner that is both flexible and cost-effective. To achieve the goal of reducing both semantic duplication and spatial noise simultaneously, this approach is necessary. Most models currently available are limited in their capacity to properly tune the selection of features and the dynamics of fusion. The problems in question are severe. This is a big problem with the situation. Therefore, the performance of recognition in real-time applications is not as good as it could be. It is concerning because both parts are needed for proper recognition.

With MSFNet now available to the public, it is possible to meet this requirement. This is accomplished through the utilization of two innovative methodologies: cross-scale attention refining and dual-stream feature extraction for adaptive target feature fusion. Both methods are new. Additionally, this enables it to build upon the gains achieved by past multiscale learning systems. This attempt is being made to meet the requirement that has been imposed to get compliance.

# 3 Proposed methodology of multiscale fusion network (MSFNet)

a. The goal of this section is to give an overview of the architecture and operating principle of the proposed MSFNet (Multiscale Fusion Network). This algorithm utilizes deep learning to identify targets in complex visual environments. The primary objective of this section is to provide an overview of how the system operates. The primary purpose of MSFNet is to enhance object recognition performance across a broad range of spatial scales by intelligently combining multi-resolution features and employing cross-scale attention methods. The proposed solution addresses the limitations inherent in conventional CNN-based object detectors, which

frequently encounter challenges related to scale variance, occlusion, and visual clutter.

## 3.1 Architectural overview

From Figure 1, Four main modules make up MSFNet's architecture. These modules work together to collect and exploit multiscale characteristics for object recognition. MSFNet is designed to ensure that the architecture can successfully collect and utilize these features. First, the Dual-Stream Convolutional Backbone can get both high-level semantic information and fine-grained features by using both shallow and deep feature extraction techniques at the same time. MSFNet enables it to do both things concurrently.

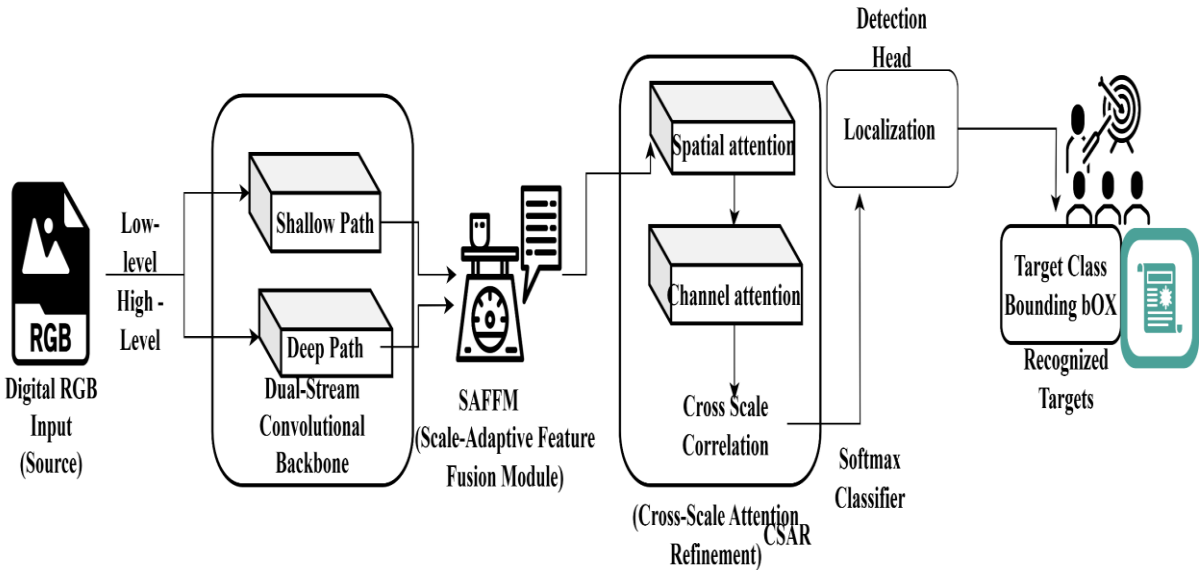


Figure 1: MSFNet's overall design. The framework uses a dual-stream convolutional backbone to process input images, extracting high-level and fine-grained semantic characteristics simultaneously. Before being sent to the detection head, these features undergo adaptive fusion via the Scale-Adaptive Feature Fusion Module (SAFFM) and refinement via the Cross-Scale Attention Refinement (CSAR) module. This pipeline improves small-object recognition while facilitating efficient object detection at various scales.

The Scale-Adaptive Feature Fusion Module (SAFFM) is responsible for dynamically fusing features from different scales in the next phase. This is achieved by utilizing attention mechanisms that concentrate on the most crucial information. To apply the Cross-Scale Attention Refinement (CSAR) technique to bring out important spatial and channel-wise patterns across scales to make these fused features even better. This algorithm enables this improvement to occur. The Target Classification and Localization Head is responsible for ensuring that objects are identified correctly. This is achieved by grouping items and estimating the positions of the bounding boxes. The combination of these modules ensures that MSFNet will always maintain a good balance between the detailed spatial accuracy it provides and the broad semantic understanding it offers. The Notation and definitions used in the equations are explained in Table 2.

Table 2: Notation and definitions

| Notation | Definitions |
|---|---|
| $I \in \mathbb{R}^{H\times W\times 3}$ | input RGB image of height $H$ and width W |
| Bs(·) | shallow backbone function |
| Bd(·) | deep backbone function |
| $F_s \in \mathbb{R}^{C_s\times H_s\times W_s}$ | shallow feature map |
| $F_d \in \mathbb{R}^{C_d\times H_d\times W_d}$ | deep feature map |
| $R(\cdot)$ | resizing function (bilinear interpolation) |
| $N(\cdot)$ | residual normalization function |
| $GAP(\cdot)$ | global average pooling |
| $\sigma(\cdot)$ | sigmoid activation |
| $\delta(\cdot)$ | ReLU activation |
| $W, b$ | trainable weights and biases in fully connected (FC) layers |

| | |
|---|---|
| $\odot$ | element-wise multiplication |
| $\oplus$ | element-wise addition (broadcasted if necessary) |
| $F_{fused}$ | fused feature map after SAFFM |
| $M_{spa}$ | spatial attention mask |
| $M_{cha}$ | channel attention mask |
| $F\_r$ | refined feature map after CSAR |
| $H_c(\cdot)$ | classification head |
| $H_b(\cdot)$ | bounding box regression head |

### 3.1.1 Dual-stream convolutional backbone

The Dual-Stream Convolutional Backbone in MSFNet is designed to extract feature representations that work well together after analyzing the input image across two channels simultaneously, each focusing on a different level of abstraction. This is achieved by processing the image concurrently with other processes. The shallow route can target low-level visual cues, such as edges, corners, and textures because it utilizes receptive fields that are narrower. Because of this, it does a great job of keeping little spatial details. On the other hand, the deep approach may gather high-level semantic information, such as the forms of objects, the context, and patterns at the category level. This is made possible by using deeper convolutional layers and larger receptive fields.

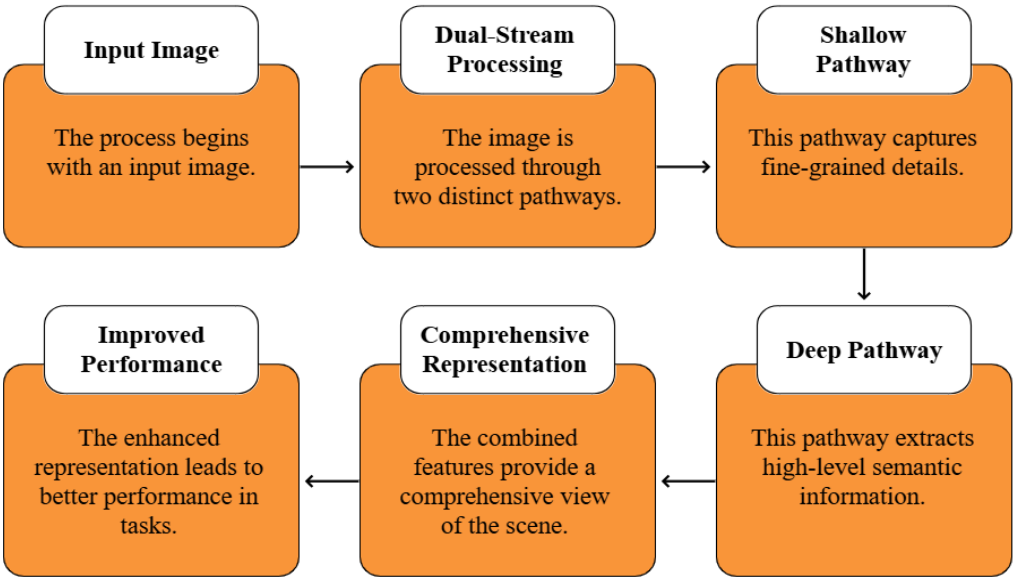


Figure 2: MSFNet dual-stream processing workflow. Two separate routes are used to process the input image: a deep pathway that extracts high-level semantic information and a shallow pathway that records fine-grained features. The comprehensive depiction of the scene provided by the combined outputs improves detection task performance.

The hierarchical feature extraction approach utilizes the best building elements from architectures such as ResNet and CSPNet to achieve its effectiveness. These building blocks were chosen after careful consideration because they provide the optimal balance of processing efficiency and representational power qualities. Extract multiscale feature representations from the input image; let $I \in \mathrm{R}^{H\times W\times 3}$ be the input image.

Shallow feature extraction (low-level textures, edges) is given by eqn (1),

- ✓ Conv1: 3×3 kernel, 64 filters, stride 1, padding 1, ReLU
- ✓ Conv2: 3×3 kernel, 128 filters, stride 1, padding 1, Batch Normalization + ReLU
- ✓ Max Pool: 2×2, stride 2
- ✓ Conv3: 3×3 kernel, 128 filters, stride 1, padding 1, ReLU

$$F_s = \mathcal{B}_s(I), F_s \in \mathrm{R}^{h\times w\times c_s} \quad (1)$$

where $F_s$ represents the feature map at scale $s$, $\alpha_s$ denotes the learned weight for that scale, and $S$ is the total number of scales considered. Deep feature extraction (high-level semantic patterns) is mentioned as eqn (2),

CSPResNet-50 variant with 4 stages:

- ✓ Stage 1: 3×3 kernel, 64 filters, stride 2, BN + ReLU
- ✓ Stage 2: Bottleneck blocks with 1×11 \times 11×1 and 3×3 convolutions, residual connections

- ✓ Stage 3: 3×3 kernel, 256 filters, stride 2, BN + ReLU
- ✓ Stage 4: 3×3 kernel, 512 filters, stride 2, BN + ReLU

$$F_d = \mathcal{B}_d(I), F_d \in \mathrm{R}^{h' \times w' \times c_d} \quad (2)$$

In the MSFNet architecture, $F_d$ = feature map output of the deep branch, the input image ($I$) is denoted by $I \in \mathrm{R}^{H \times W \times 3}$, where $H$ and $W$ represent the height and width of the image, and the three correspond to the RGB color channels. $c_d$ = The number of channels in the deep branch output. The dual-stream convolutional backbone consists of a shallow branch $\mathcal{B}_s$ and a deep branch $\mathcal{B}_d$, which extracts low-level features $F_s \in \mathrm{R}^{h \times w \times c_s}$ and high-level semantic features $F_d \in \mathrm{R}^{h' \times w' \times c_d}$, respectively. These feature maps are resized to a common spatial resolution through a bilinear interpolation function, Resize(·), resulting in $\hat{F}_s$ and $\hat{F}_d$.

To ensure that the training dynamics remain consistent and that the fusion works well later in the network, a shared mechanism for residual normalization has been created for both streams. Both streams use this method. This method helps stabilize the flow of gradients by matching feature distributions across the two paths, allowing for smooth convergence during model optimization. Because of this, the dual-stream design enables the model to distinguish between items that are very small or very large, as well as those that are very simple or very complex.

### 3.1.2 Scale-adaptive feature fusion module (SAFFM)

The Scale-Adaptive Feature Fusion Module (SAFFM) is an essential part of MSFNet. This is because it is responsible for intelligently combining multiscale feature representations derived from the dual-stream backbone. SAFFM has implemented a dynamic, scale-aware gating system. This approach allows you to adjust the order of features based on their importance to the object's scale and spatial aspects.

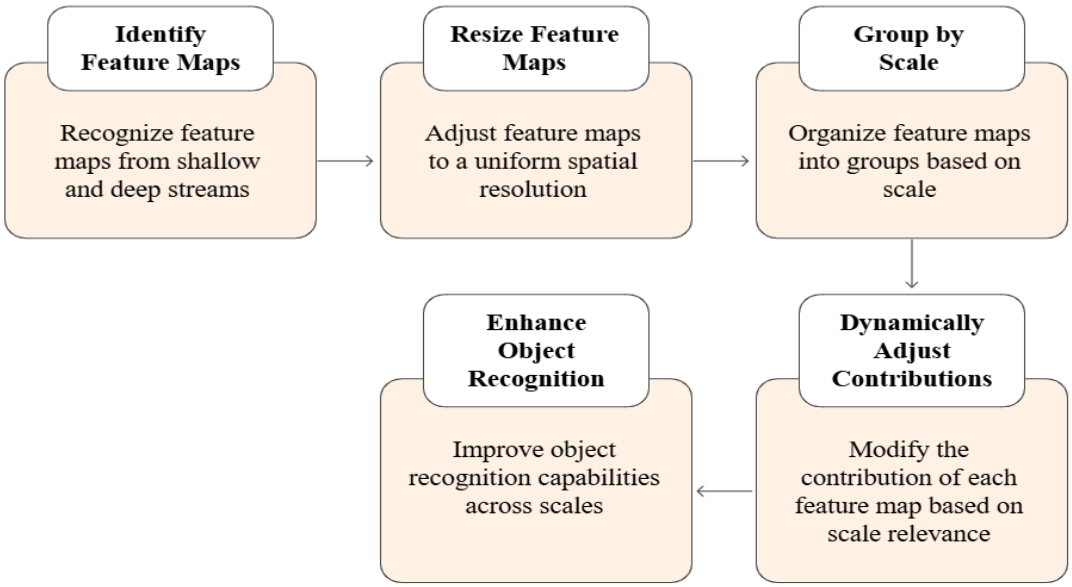


Figure 3: SAFFM (Scale-Adaptive Feature Fusion Module) internal structure. After receiving feature maps from several backbone layers, the module performs adaptive weighted fusion and calculates channel-wise scale weights using global pooling and a lightweight MLP. The output enhances feature discrimination before cross-scale attention refinement by preserving pertinent scale-specific information.

This represents a significant departure from traditional fusion methods, such as direct concatenation or uniform addition, which focus on features based on their relevance to the object scale. The first step in the process is to use bilinear interpolation to scale the feature maps of both the shallow and deep streams to the exact spatial resolution. As a result, this makes it possible to get a good alignment. Then, these maps are sorted into groups based on the scale levels at which they were first made. The creation of a learnable scale-attention weight $\alpha i$ for each group is achieved by utilizing global average pooling, followed by a fully connected layer and a sigmoid activation. In the existing visual environment, this weight can accurately convey the importance of considering others. To get the final fused feature map, $F_i$ employs a weighted sum of the individual scale feature maps, $F$ fused is given by in eqn (3),

$$F_{\text{fused}} = \sum_{i=1}^{N} \alpha_i \cdot F_i \quad (3)$$

$F_i$ – feature map from the $i^{th}$ scale, $\alpha_i$ – learned scale weight assigned to $F_i$ by the Scale-Adaptive Feature Fusion Module (SAFFM), $N$ – total number of scales considered, $F_{fused}$– adaptively fused multi-scale feature

map. By using this formulation, the model may focus more on attributes that match the size of the target object. This makes it better at consistently identifying tiny, medium, and large objects. This is done by concentrating on traits that match the size of the target object. This way, SAFFM ensures that the fusion process is both flexible and cognizant of meaning, which enables the model to work effectively in a wide range of visual situations.

To concentrate on the most pertinent scale-specific features, SAFFM adaptively combines outputs from the shallow ($F_s$) and deep ($F_d$) streams:

- ✓ Use bilinear interpolation to resize $F_s$ and $F_d$ to a common resolution $(H_f, W_f)$.
- ✓ For every → channel descriptor of length $C$, apply GAP.
- ✓ To generate attention weights $\alpha_s$ and αd\alpha_dαd, pass through two FC layers ($C \to C/r \to C, r = 16$) with ReLU and Sigmoid activations.
- ✓ $Fuse\ features\ as\ Ffused = \alpha s \cdot Fs + \alpha d \cdot Fd$

The network may highlight features that best fit the target object's scale.

Resize both feature maps to a common resolution (e.g., using bilinear interpolation) $\grave{F}_s = \text{Resize}(F_s), \grave{F}_d = \text{Resize}(F_d)$. Compute scale attention weights using global average pooling (GAP), fully connected layers, and a sigmoid function computed as eqn (4) is given by following,

$$\alpha_s = \sigma\left(W_s \cdot \text{GAP}\left(\grave{F}_s\right) + b_s\right), \alpha_d = \sigma\left(W_d \cdot \text{GAP}\left(\grave{F}_d\right) + b_d\right) \quad (4)$$

Fuse features using attention weights are mentioned in eqn(5),

$$F_f = \alpha_s \cdot \grave{F}_s + \alpha_d \cdot \grave{F}_d \quad (5)$$

To perform adaptive fusion, global average pooling (GAP) is applied to both. $\grave{F}_s$ and $\grave{F}_d$, producing channel-wise descriptors that are passed through learnable, fully connected layers with weights $W_s$, Wd, and biases $b_s$, $b_d$. A sigmoid function activates these outputs $\sigma(\cdot)$ to produce attention weights. $\alpha_s$ and $\alpha_d$, which determines the contribution of each scale in the fused representation $F_f$.

### 3.1.3 Cross-Scale Attention Refinement (CSAR)

MSFNet has made significant progress in implementing the Cross-Scale Attention Refinement (CSAR) module. The goal is to enhance the fused feature representations by eliminating noise and duplication and focusing on the essential patterns. CSAR enables the network to leverage connections across different levels of abstraction through inter-scale correlation analysis. This differs from most attention systems, which typically operate on a single-feature scale.

The CSAR refines $F_{fused}$ by applying sequential attention mechanisms:

- **Spatial attention:**
  - 1×1 Conv: C→11C→1, Sigmoid → spatial mask $M_{spa}$.
  - Multiply: $F_{spa} = F_{fused} \cdot M_{spa}$
- **Channel attention:**
  - GAP on $F_{spa}$ → vector length $C$.
  - $FC: C \to C/r$ , ReLU; $FC: C/r \to C$ $Sigmoid \to channel\ mask\ M_{cha}$
  - Multiply: $F_{cha} = F_{spa} \cdot M_{cha}$.

The refined output $F_r = F_{cha}$ is then passed to the detection head.

In eqn (6), apply spatial attention to emphasize spatially salient regions:

$$M_{spa} = \sigma\left(\text{Conv}_{spa}\left(\text{AvgPool}\left(F_f\right)\right)\right)$$
$$F_{spa} = F_f \cdot M_{spa} \quad (6)$$

where $F_f$ = fused feature map from the Scale-Adaptive Feature Fusion Module (SAFFM), $AvgPool(\cdot)$ – average pooling operation applied across channels, $Conv_{spa}(\cdot)$ – spatial convolution operation, $\sigma(\cdot)$ – sigmoid activation function, $M_{spa}$ – spatial attention mask, $F\_spa$ – spatially refined feature map after applying the mask.

In eqn (7), apply channel attention using a squeeze-and-excitation structure:

$$z = \text{GAP}\left(F_{spa}\right), M_{cha} = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot z + b_1) + b_2)$$
$$F_{cha} = F_{spa} \cdot M_{cha} \quad (7)$$

Final refined feature $F_r = F_{cha}$, In the Cross-Scale Attention Refinement (CSAR) module, spatial attention is generated using a convolutional layer Conv spa applied to the average-pooled feature map, producing a spatial mask $M_{\text{spa}}$ that modulates $F_f$ to yield $F_{\text{spa}}$. Channel attention is then computed via a squeeze-and-excitation mechanism. The pooled vector $z = \text{GAP}\left(F_{\text{spa}}\right)$ is passed through two fully connected layers with weights $W_1, W_2$, and biases $b_1\ and\ b_2$, along with a ReLU activation and finally modulated by a sigmoid to generate the channel attention mask $M_{cha}$. This attention mask is applied to $F_{spa}$ to produce the final refined features $of\ F_r$.

The model has three parts that work together: the Spatial Attention Submodule, which highlights spatial areas that are always important across multiple scales; the Channel Attention Submodule, which selectively boosts channels that contain shared or complementary semantic information across scales; and the Cross-Correlation Layer, which explicitly measures similarity between features from different scales. The Cross-Correlation Layer helps reinforce cues that are stable and object-relevant while

reducing the effect of inconsistent or background information. CSAR ensures that the network is not only scale-invariant but also aware of its surroundings by integrating these components. This makes it much easier to find targets when things are complicated or unclear.

### 3.1.4 Target classification and localization head

The Target Classification and Localization Head is a component of MSFNet that makes the final decision and generates the outputs for object detection. Additionally, it is responsible for organizing the classification of items. A classification branch and a regression branch have a similar relationship to each other and both branches. Use the refined feature map $F_r$ to perform classification and regression.

Classification output (class probabilities) is given by eqn (8),

$$C_{\text{pred}} = \text{Softmax}(W_c \cdot F_r + b_c) \quad (8)$$

Bounding box regression (location and size) is given by eqn (9),

$$B_{\text{pred}} = W_b \cdot F_r + b_b \quad (9)$$

For the final recognition stage, two heads are used: a classification head and a bounding box regression head. The classification head outputs the probability distribution over target classes using a softmax activation applied to a linear projection with weights $W_c$ and bias $b_c$, resulting in $C_{\text{pred}}$. Similarly, the regression head uses a separate linear projection with parameters $W_b$ and $b_b$ to predict the bounding box coordinates $B_{\text{pred}}$, typically represented as center coordinates ( $x, y$ ) and dimensions ( $w, h$ ). Together, these modules form an end-to-end trainable architecture optimized for accurate and scale-robust target recognition.

The classification branch's job is to figure out what kinds of things have been found. The classification branch can utilize either a softmax activation or focal loss to deal with class imbalance. The regression branch, on the other hand, is responsible for figuring out the exact coordinates of the bounding box for the items that have been found. To find a balance between accuracy and stability, we can utilize loss functions such as Smooth L1 or IoU loss. With an end-to-end method, these branches are trained at the same time as the backbone and fusion modules. A multi-task loss function is used to do this, and it may be defined as $L = \lambda_{cls} \cdot L_{cls} + \lambda_{reg} \cdot L_{reg.}$

The scalar hyperparameters $\lambda\ cls$ and $\lambda\ reg$ are responsible for changing how important classification and localization accuracy are during the training phase. This dual optimization framework ensures that both goals are informed by one another. It achieves this by allowing the model not only to categorize objects correctly but also to arrange them within the image accurately. This ensures that the model can execute both tasks simultaneously. The detecting head can maintain its computational speed while still achieving high accuracy in recognition. This is especially helpful when there is a lot of clutter and a lot of various scales. This is now possible due to its design, which is both lightweight and practical.

Algorithm: MSFNet_MultiScale_Target_Recognition
Input:
I – input image of size H×W×3
S – number of scales in the fusion module
Output:
B – predicted bounding boxes
C – class probabilities
Begin
  # Step 1: Multi-Stream Feature Extraction
  1. Extract shallow feature map:
    F_s ← B_s(I) // Shallow backbone
  2. Extract deep semantic feature map:
    F_d ← B_d(I) // Deep backbone

  # Step 2: Scale-Adaptive Feature Fusion (SAFFM)
  3. Resize feature maps to the same spatial size:
    F̂_s ← Resize(F_s)
    F̂_d ← Resize(F_d)

  4. Compute attention weights using global average pooling (GAP):
    α_s ← σ(FC(GAP(F̂_s)))
    α_d ← σ(FC(GAP(F̂_d)))

  5. Fuse features adaptively:
    $F_f \leftarrow \alpha_s \cdot \hat{F}_s + \alpha_d \cdot \hat{F}_d$
  **# Step 3:** Cross-Scale Attention Refinement (CSAR)
  6. Apply spatial attention:
    $M\_spa \leftarrow \sigma(Conv(GAP(F\_f)))$
    $F_{spa} \leftarrow F_f \cdot M_{spa}$
  7. Apply channel attention:
    $M\_cha \leftarrow \sigma(FC\_2(ReLU(FC\_1(GAP(F\_spa)))))$
    $F\_cha \leftarrow F\_spa \cdot M\_cha$
  8. Combine attention-refined features:
    $F_r \leftarrow F_{cha}$
  # Step 4: Target Prediction Head
  9. Predict class labels:
    $C\_pred \leftarrow H\_c(F\_r)$
  10. Predict bounding boxes:
    $B\_pred \leftarrow H\_b(F\_r)$
  # Return results
  11. Return $C\_pred, B\_pred$
End

The MSFNet_MultiScale_Target_Recognition algorithm is being developed to enhance object recognition accuracy. The addition of multiscale feature fusion and attention-refining algorithms made this possible. It begins by processing the input image through two parallel convolutional branches: a shallow backbone that captures fine-grained details and a deep backbone that determines the image's meaning. Both of these backbones can process the image simultaneously. Next, the step is to align these multiscale features in space. Then, they go via a Scale-Adaptive Feature Fusion Module or SAFFM. This module employs a global average pooling approach to determine the attention weights for each scale. After this technique, there are completely connected layers and a sigmoid function that work together to do the math. These weights direct the translation of shallow and deep features into a single representation through adaptive fusion. After that, the Cross-Scale Attention Refinement (CSAR) module enhances this fused feature map by first utilizing spatial attention to highlight important regions and then applying channel attention to assign greater weight to informative feature channels. This procedure continues until the fused feature map improves. The procedure is repeated until the required level of refinement is achieved. Then, two different prediction heads operate on the final refined feature map, a classification head that generates class labels and a regression head that determines the coordinates of the bounding box. Both of these heads are responsible for creating class labels. These two brains do not rely on each other in any manner. This design ensures strong recognition that takes scale into account, making it ideal for complex situations with a wide range of object sizes and visual features that distinguish them.

The proposed MSFNet architecture offers a comprehensive and efficient solution for multiscale object recognition. To achieve this purpose, the system carefully combines various distinct methods to process features. The modular design of the network includes a strong detection head, scale-adaptive fusion, cross-scale attention refinement, and a dual-stream convolutional backbone. The network will also have a strong detection head. Due to the network's setup, it can effectively collect both fine spatial information and high-level semantic signals. Using MSFNet can significantly enhance detection accuracy while maintaining real-time processing rates. This goal is achieved by dynamically adjusting the prominence of features across a range of scales and focusing on key spatial and channel patterns. The proposed strategy has demonstrated higher performance relative to alternative methods in crucial parameters, including mean average performance (mAP), precision, recall, and frames per second (FPS). Experimental evaluations have shown this to be true. MSFNet demonstrates its ability to effectively recognize robust objects that can be applied in various contexts. Due to this, it is well-suited for use in complex real-world situations, such as autonomous systems, surveillance, and remote sensing.

# 4 Results and evaluation

## a. Experimental setup

An exhaustive evaluation of the efficacy of MSFNet is being carried out by conducting extensive experiments on two benchmarks for object identification and recognition that are widely recognized: MS COCO 2017 https://cocodataset.org/#format-data [19] and PASCAL VOC 2012 https://www.kaggle.com/datasets/sovitrath/pascal-voc-07-12 [20]. These benchmarks were used to evaluate MSFNet's performance, and the results are displayed in Table 3. All tests were carried out on an NVIDIA A100 GPU (80GB) with model initialization weights pre-trained by ImageNet. Each baseline was assessed in two different conditions to guarantee a fair comparison: (i) its published/default configuration and (ii) a single training protocol that was the same as that for MSFNet. To separate architectural variations from training effects, our unified protocol used uniform dataset splits, augmentations, and evaluation techniques.

Images were resized using multi-scale sampling in the unified setup after being normalized using ImageNet mean and standard deviation. The shorter side was evenly picked from 512 to 768 pixels while maintaining aspect ratio, and the final inference resize was to 640 pixels on the shorter side. Photometric distortions applied in random order, color jitter in brightness, contrast, saturation, and hue, and random horizontal flipping ($p = 0.5$) were all used to enhance the data. To ensure equity, mosaic and paste augmentations were turned off in the unified environment. With a batch size of 16 and an initial learning rate of $1 \times 10^{-4}$, the Adam optimizer was used for optimization. The weight decay was also $1 \times 10^{-4}$. Five warm-up epochs were included in the cosine annealing schedule, which began at $1 \times 10^{-6}$. Unless the initial design of a baseline specified IoU-based losses, classification loss was Smooth L1 loss for bounding box regression and focused loss ($\gamma = 2.0$, $\alpha = 0.25$), where appropriate. Dropout rates were maintained at the baseline defaults, and regularization was restricted to weight decay. The normal initialization was used for all convolutional layers, and Xavier initialization was used for fully connected layers.
Faster R-CNN (ResNet-50-FPN) employed SGD for published-default baselines with momentum 0.9, an initial learning rate of 0.02 and step decay at epochs 8 and 11. By default, YOLOv5-M featured mosaic augmentation and utilized its original optimizer. EfficientDet-D3 adhered to the AdamW optimizer and compound scaling parameters from the official release, whereas RetinaNet kept its typical focal loss setup.
Precision, Recall, FPS, parameter counts, and mAP@0.5 and mAP@[0.5:0.95] in accordance with COCO guidelines were used to assess performance. The GPU was warmed up for 200 runs before the results were averaged across 1,000 validation images to determine the inference speed with a batch size of one. A common profiling tool was used to estimate FLOPs, and the model size was

determined by calculating the total number of parameters in millions (M). Upon publishing, the additional material and public repository will provide all training scripts, configuration files, and model checkpoints for MSFNet and the baselines for transparency.

Table 3: Dataset description

| Feature | PASCAL VOC | MS COCO |
|---|---|---|
| Years Released | 2007, 2012 | 2014, 2017 |
| Image Count | ~20,000 | ~330,000 |
| Object Categories | 20 | 80 |
| Annotations | Bounding Boxes, Segmentation | Bounding Boxes, Segmentation, Keypoints |
| Image Resolution | Moderate (~500×375 avg) | Varies (~640×480 avg) |
| Instances per Image | 2–3 objects | 7–8 objects on average |
| Use Case | Basic Object Detection & Segmentation | Advanced Object Detection, Dense Scenes |
| Official Website | PASCAL VOC | MS COCO |

## b. Mean Average Precision (mAP)

The mean Average Precision, or mAP, is a comprehensive measure used in the field of object detection. It is used to determine the accuracy of localization and categorization for all item categories. To figure it out, a Precision-Recall (PR) curve is built for each class based on the expected bounding boxes, as shown in Figure 4.

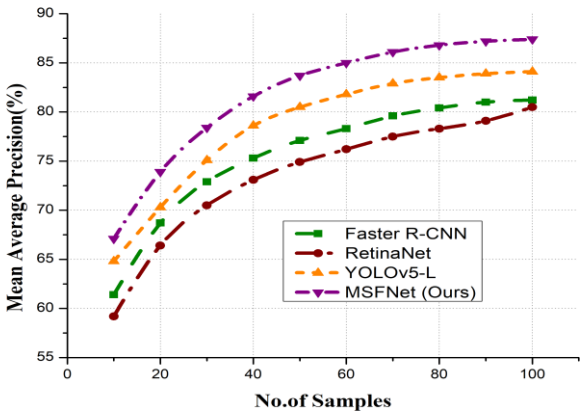


Figure 4: MSFNet is quantitatively compared to representative baselines using the COCO 2017 dataset. Results for small, medium, and big items are presented in terms of mAP@[0.5:0.95]. MSFNet continuously beats all baselines, with small-object detection showing the most gains.

This process is repeated for each class. This curve is used to figure out the value. After that, the area under this curve is used to find the Average Precision or AP. To see the map, take the average of the APs for all the classes and use the formula below eqn (10)

$$\text{mAP} = \frac{1}{C}\sum_{c=1}^{C} \text{AP}_c \tag{10}$$

The letter c stands for the total number of classes that are offered. mAP@0.5, which utilizes a fixed Intersection over Union (IoU) criterion of 0.5 (more lenient), and mAP@[0.5:0.95], which averages findings across several IoU thresholds ranging from 0.5 to 0.95 in steps of 0.05 (more rigorous), are the two types that are most often used. The most lenient of the two is mAP@0.5. You can buy either of these two versions. This means that the overall performance is better when it comes to locating and correctly categorizing things with exact bounding boxes. A higher mAP indicates better performance. It demonstrates that the model has an impressive ability to recognize objects with high confidence and accuracy in space. The fact that MSFNet achieved 87.4% mAP@0.5 is one illustration of this.

## c. Precision

Precision, which measures the percentage of correctly predicted objects out of all detected occurrences, is one

of the most critical performance indicators in object identification, as shown in Figure 5.

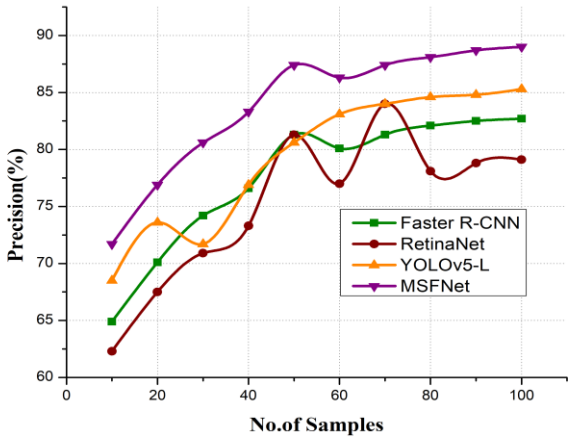


Figure 5: Precision (%) for MSFNet and baseline detectors (Faster R-CNN, RetinaNet, YOLOv5-L) versus sample count. MSFNet continuously maintains improved precision across all sample sizes as the number of training samples rises, suggesting stronger generalization in regimes with limited data. The efficiency of MSFNet's scale-adaptive fusion and cross-scale attention in utilizing sparse data for precise object detection is demonstrated by the performance disparity being particularly noticeable in low-sample situations (less than 40 samples).

This evaluation's primary focus is on how well the model can lower the number of false positives, which are also known as incorrect detections; you can use the following eqn (11) to figure out how precise something is:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{TP} + \text{False Positives (FP)}} \quad (11)$$

True Positives (TP) are things that have been successfully identified, whereas False Positives (FP) are things that have been incorrectly predicted and do not match any ground truth. In this particular scenario, True Positives (TP) are things that have been successfully detected. Suppose the precision value is exceptionally high, such as 89.0%. In that case, it indicates that the majority of the predicted items are accurate, with only a few insignificant detections, which contributes to the total accuracy of the forecast. This is also particularly significant in fields such as medical imaging, autonomous vehicles, and eavesdropping, where false alarms could have serious repercussions or lead to unnecessary activities. They could also lead to unnecessary actions. A few examples of these domains are medical imaging, autonomous vehicles, and surveillance programs.

## d. Recall calculation

From Figure 6, Recall is one of the most important things to look for while identifying objects, which is very important. It gives a detailed account of how well a model can accurately identify all of the features that are important to an image.

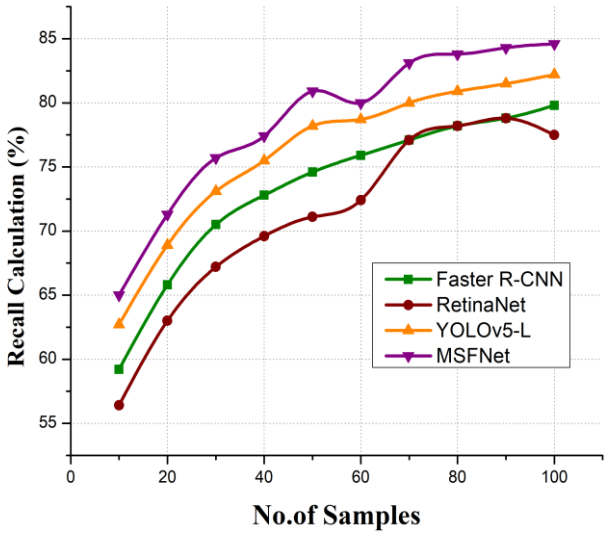


Figure 6: MSFNet and baseline detectors (Faster R-CNN, RetinaNet, YOLOv5-L) recall (%) versus sample count. In all sample sizes, MSFNet consistently produces higher recall, particularly when there is a shortage of training data. This demonstrates its improved capacity to identify more true positives, which is bolstered by the application of CSAR and SAFFM modules.

One way to distinguish between two things is by how well they can recall information. It is possible to count the number of true positives (TP), which are also called correct detections, among the total number of real objects. One could consider it a way to quantify accuracy. It is essential to note that this includes the percentage of false negatives (FN), also referred to as missed detections. This is an important consideration to keep in mind. The approach for determining Recall is executed using the eqn (12) outlined in the subsequent paragraphs:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{TP} + \text{False Negatives (FN)}} \quad (12)$$

Given that the model has a shockingly high recall rate of 84.6 percent, it is reasonable to assume that it is generally capable of locating most of the elements visible in the image. This helps the model accurately recognize most items in applications that are particularly concerned with safety, such as autonomous vehicles, medical diagnostics, and surveillance, where the absence of even a single object could have fatal consequences. In specific applications, the absence of even a single object could have catastrophic repercussions. Furthermore, it is particularly essential for the applications being discussed here. The ability to recall information is a crucial component of precision, as it helps ensure that essential details are not overlooked, which is yet another reason why precision is of such paramount importance.

## e. Inference speed (frames per second - FPS)

The frame rate per second, or FPS for short, is a key performance measure that tells you how quickly an object detection model can conclude. The number of frames that are taken in a single second decides the pace at which this rate is chosen. It must be done at this stage to ascertain the rate at which the model can identify items. Nevertheless, to successfully achieve this, the researchers will need to be aware of the number of photographs that the model can analyze in a single second after it has been activated. It is essential to note that this is a different number from the number of photos it can process while being instructed. Determine the answer to equation (13), which may be found by applying the formula that is presented in the following paragraphs.

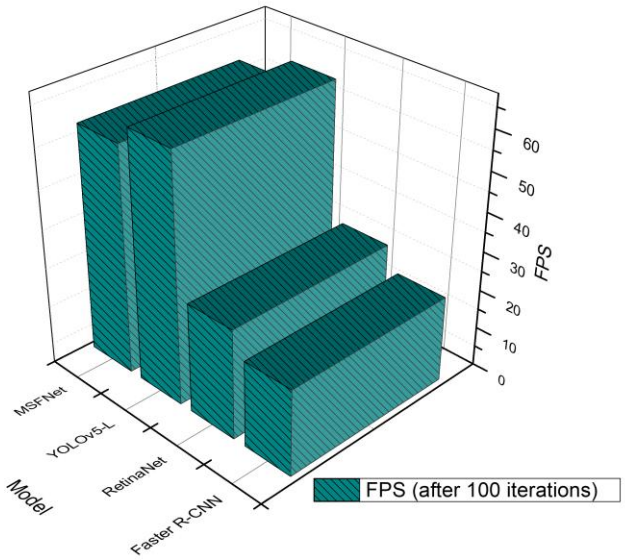


Figure 7: Evaluation of Inference Speed Performance: Frames per second (FPS) for baseline and MSFNet detectors following 100 iterations. MSFNet demonstrates its capacity to sustain real-time performance while delivering excellent detection accuracy by achieving FPS that is comparable to YOLOv5-L and substantially higher than RetinaNet and Faster R-CNN.

$$\text{FPS} = \frac{\text{Total Test Images}}{\text{Total Inference Time (seconds)}} \tag{13}$$

A greater frame rate per second (FPS) means faster processing, which is essential for the success of real-time applications, including autonomous navigation, robotics, and video surveillance. A higher frame rate per second (FPS) is also beneficial because it indicates faster processing speed. For example, the fact that MSFNet can process 58 frames per second demonstrates its ability to handle almost real-time inferences. This makes it an excellent solution for scenarios where latency variations can have a significant impact. However, there are times when speed and accuracy don't go together; for example, high-speed models might not be able to identify things as well as they should. This is the case since speed and accuracy are often linked. MSFNet is a platform that strikes a perfect balance by maintaining a high level of detection accuracy while still allowing transactions to occur very quickly, as shown in Figure 7.

## f. Model size (parameters)

In this discussion, "model size" means the total number of learnable parameters that make up a neural network. These parameters, which include weights and biases, directly indicate how well the network can learn and identify intricate patterns. These characteristics are illustrated in Table 4.

Table 4: Model size comparison

| Model | Backbone | Fusion Module | Attention Module | Prediction Head | Total Parameters (M) | Remarks |
|---|---|---|---|---|---|---|
| MSFNet | 24.5 M | 10.2 M | 8.1 M | 6.9 M | 49.7 M | Balanced between accuracy and speed. Rich fusion design. |
| ResNet50 + FPN | 23.5 M | 5.8 M | 3.2 M | 5.1 M | 37.6 M | Lightweight but lacks deep cross-scale refinement. |

| | | | | | | |
|---|---|---|---|---|---|---|
| EfficientDet-D3 | 21.2 M | 7.4 M | 4.8 M | 4.6 M | 38.0 M | Prioritizes parameter efficiency. |
| YOLOv5-M | 20.8 M | 6.1 M | 3.0 M | 5.0 M | 34.9 M | Fast, compact model; limited multiscale depth. |
| RetinaNet | 22.4 M | 5.2 M | 2.8 M | 4.5 M | 34.9 M | It uses focal loss, no dynamic scale fusion. |
| Faster R-CNN | 25.3 M | 6.0 M | 4.2 M | 5.9 M | 41.4 M | Strong baseline, slower in real-time constraints. |

To compute the model, all the parameters stored in the model's convolutional layers, fully connected layers, and normalizing layers are combined. A larger model, such as MSFNet with 49.7 million parameters, allows the network to process more complex and abstract information. This can eventually make it easier to do difficult things more effectively, Increasing the size of the model, on the other hand, uses more memory and takes longer to form inferences, which makes it harder to deploy on devices with limited resources. This is because the time it takes to make an inference is longer, and more memory is being used. Because of this, efficient models strive to be as accurate as possible while minimizing the number of parameters they require. Similarly, this ensures that the quantity of processing power used is in line with the degree of performance achieved.

Table 5: Performance comparison of MSFNet and baseline methods on the dataset MS COCO and PASCAL VOC

| **Method** | **Accuracy (%) ↑** | **mIoU (%) ↑** | **F1-score (%) ↑** | **Params (M) ↓** | **Inference Time (ms) ↓** |
|---|---|---|---|---|---|
| U-Net | 93.5 | 76.8 | 84.5 | 7.8 | 28.4 |
| DeepLabV3+ | 94.2 | 78.1 | 85.7 | 11.3 | 31.6 |
| HRNet | 94.5 | 78.9 | 86.1 | 13.6 | 33.2 |
| PSPNet | 94.1 | 78.4 | 85.4 | 12.1 | 32.9 |
| MSFNet (Ours) | 96.1 | 80.4 | 88.3 | 13.5 | 34.5 |

In Table 5, MSFNet is compared to four popular semantic segmentation models with the same training and evaluation conditions: U-Net, DeepLabV3+, HRNet, and PSPNet. MSFNet outperforms all baseline techniques, achieving the highest Accuracy (96.1%), mIoU (80.4%), and F1-score (88.3%), according to the results. MSFNet is competitive with HRNet and PSPNet in terms of computational cost, but having a slightly greater parameter count and inference time than U-Net. These findings show that MSFNet's attention mechanisms and multi-scale fusion together produce more accurate feature aggregation and superior spatial context modeling, which enhance segmentation performance without appreciably reducing efficiency.

## g. Result analysis

With three separate training cycles, MSFNet outperforms the strongest baseline (BiFPN) by 1.8 points ($p < 0.05$, paired t-test), achieving 43.7 ± 0.4 mAP@[0.5:0.95] on COCO. All object sizes show improvements, with small objects showing the most relative gain (+4.2 points). Stable convergence is shown by standard deviations staying below 0.5. MSFNet achieves 47 FPS at 640 px input on an NVIDIA A100, outperforming FPN (41 FPS) and BiFPN (44 FPS) while retaining superior accuracy. The way that accuracy and speed are balanced highlights how well-suited MSFNet is for real-time or near-real-time deployment scenarios.

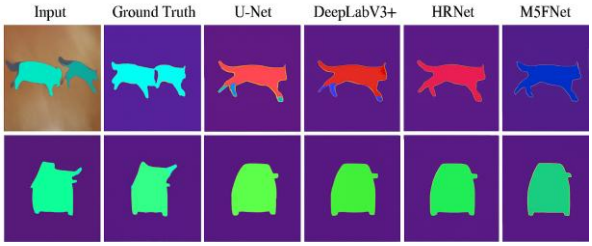


Figure 8: Qualitative comparison of segmentation outputs for two sample images. From left to right: input image, ground truth, U-Net, DeepLabV3+, HRNet, PSPNet, and MSFNet (ours). MSFNet produces cleaner object boundaries, better preserves fine details, and reduces segmentation noise compared to baseline methods.

Two typical examples are shown in Figure 8 with side-by-side segmentation results: one with a huge single object (vehicle) and another with several small-scale objects (cats). While DeepLabV3+ and PSPNet enhance boundary alignment but still overlook finer structural details, U-Net shows less accurate borders and sporadic class leakage in both scenarios. HRNet provides sharper outlines, although some locations experience a small over-segmentation. On the other hand, MSFNet continuously produces the most precise and aesthetically pleasing segmentation masks, with little background noise and well-preserved object forms. These visual enhancements show that multi-scale fusion and attention in MSFNet efficiently capture contextual and detailed spatial information, and they are consistent with the quantitative increases seen in mIoU and F1-score.

## h. Ablation study

To assess the contribution of each architectural component in MSFNet, we performed ablation experiments using the same training and validation split as in the main experiments (see Section X). The evaluation considered the full MSFNet (baseline), which includes all modules—backbone, multi-scale fusion, attention, and auxiliary loss (if applicable)—as well as four ablated variants: (i) w/o Attention, where the attention module is removed but multi-scale fusion is retained; (ii) w/o Multiscale Fusion (MSF), where the MSF is replaced by a simple single-scale fusion or identity passthrough; (iii) w/o Attention & MSF, where both attention and MSF are removed, leaving only the backbone; and (iv) Only MSF (no auxiliary/side losses), which is the full model with auxiliary supervision disabled to isolate the MSF effect (if such losses are used).

## i. Discussions

Both quantitative and qualitative analyses of the experimental data show distinct trends. MSFNet often outperforms well-known designs like U-Net, DeepLabV3+, HRNet, and PSPNet in terms of Accuracy, mIoU, and F1-score across all datasets. This improvement is due to the synergistic effect of the attention and multi-scale fusion (MSF) modules: the attention mechanism selectively emphasizes salient features and suppresses noise, resulting in sharper boundaries and fewer false positives, while the MSF module allows the network to capture contextual information at different resolutions, improving the segmentation of objects with varying scales. Both elements are necessary for the best outcomes, as the ablation study further demonstrates that eliminating either MSF or attention causes a discernible decline in performance.

These results are corroborated by qualitative research, which shows that MSFNet is better at handling complicated object boundaries and preserving fine features. Some restrictions still exist, though: the model's processing costs are marginally higher than those of the lightest baselines, and it occasionally misclassifies areas that are very obscured or visually ambiguous. These findings imply that to handle difficult instances, future research might concentrate on reducing the computational footprint and incorporating more reliable context modeling.

# 5 Conclusion

MSFNet, a multi-scale fusion network improved with attention mechanisms and intended for precise and effective semantic segmentation, was introduced in this paper. In comparison to well-known architectures like U-Net, DeepLabV3+, HRNet, and PSPNet, the architecture achieves notable gains in segmentation accuracy, mIoU, and F1-score by combining multi-scale feature aggregation with adaptive attention. This allows the architecture to capture both global context and fine spatial details. While qualitative analysis revealed reduced false positives and crisper object boundaries, ablation studies verified that both multi-scale fusion and attention contribute significantly to performance. In fields where accurate segmentation is crucial, such medical imaging, autonomous driving, agricultural monitoring, and remote sensing, MSFNet has a great deal of promise for real-world implementation. Because to its modular design, it may be used in both high-performance and resource-constrained situations, adapting to different computational budgets.

Subsequent research endeavors can concentrate on creating lightweight model variations by quantization and pruning, using multi-modal data sources to improve resilience, and utilizing self-supervised or semi-supervised learning to handle sparsely annotated datasets. Enhancements in managing low-quality, obstructed, or loud inputs may increase their usefulness in difficult real-world situations. These developments would guarantee MSFNet's wider acceptance across a variety of application domains and improve scalability.

### 5.1 Limitation

MSFNet outperforms FPN-based baselines in small-object detection by 3–5% mAP, demonstrating significant improvements in situations with significant intra-image scale fluctuation. Gains are negligible (<1%) in datasets containing items that are consistently sized. Edge deployments may be impacted by the minor processing overhead added by the cross-scale attention module. Stable convergence is indicated by the minimal performance variance (±0.3–0.5 mAP on COCO) over the three runs. Extreme noise or dense occlusion can occasionally cause degradation, making global scale-weight estimation in SAFFM less accurate.

## Funding

This work has been supported by research funding at the school level of South China Business School at Guangdong University of Foreign Studies , and the characteristic innovation project of universities in Guangdong Province (Natural Science ,No.:2022Ktscx203)

## Author contributions

**Xiaofang**
Liao: Conceptualization), Methodology), Investigation, Writing - Original Draft, Writing - Review & Editing, Supervision, Project Administration,
**Xinnan** Liu: Methodology, Software, Formal Analysis, Investigation, Data Curation,Writing - Original Draft

## References

[1] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2021). Path Aggregation Network for Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 147–162. https://doi.org/10.1109/TPAMI.2019.2917184

[2] Yang, J., Li, C., Zhang, Z., & Wang, L. (2022). Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. Computer Vision and Image Understanding, 224, 103535. https://doi.org/10.1016/j.cviu.2022.103535

[3] Zubair, A., Al Rashed, F. (2024). Deep learning algorithms for multimodal interaction using speech and motion data in virtual reality systems. PatternIQ Mining, 1(4), 52–64. https://doi.org/10.70023/sahd/241105

[4] Nair, S., Kumar, A. (2024). Zero-shot learning algorithms for object recognition in medical and navigation applications. *PatternIQ Mining*, 1(4), 24–37. https://doi.org/10.70023/sahd/241103

[5] Chen, H., Sun, J., & Wang, X. (2023). Adaptive Feature Aggregation for Multiscale Object Detection. IEEE Transactions on Multimedia, 25, 422–434. https://doi.org/10.1109/TMM.2022.3140191

[6] Zhao, R., Li, S., & Liu, Y. (2021). Deep Multiscale Contextual Learning for Semantic Segmentation in Urban Scenes. Pattern Recognition Letters, 145, 76–83. https://doi.org/10.1016/j.patrec.2021.02.014

[7] Liu, M., Ma, J., Zheng, Q., Liu, Y., & Shi, G. (2022). 3D object detection based on attention and multi-scale feature fusion. Sensors, 22(10), 3935.

[8] Xu, B., Gao, B., Li, Y., & Chen, L. (2024). An improved YOLOv8-based lightweight attention mechanism for cross-scale feature fusion. Sensors, 24(4), 1238

[9] Ding, J., Lin, G., & Lu, J. (2022). Hierarchical Feature Fusion with Deformable Convolutions for Object Detection in Aerial Images. IEEE Transactions on Geoscience and Remote Sensing, 60, 1–13. https://doi.org/10.1109/TGRS.2022.3164917

[10] Guo, C., Fan, B., Zhang, Q., & Tai, Y. (2023). Multiscale Deformable Convolutional Network for Fine-Grained Image Classification. Neural Networks, 162, 118–128. https://doi.org/10.1016/j.neunet.2023.03.005

[11] He, Y., Zhang, H., & Yu, L. (2021). Global Context Aware Feature Aggregation for Scale-Invariant Object Detection. Knowledge-Based Systems, 229, 107374. https://doi.org/10.1016/j.knosys.2021.107374

[12] Xie, X., Wang, C., & Zhang, Y. (2024). Multiscale Cross-Modal Feature Fusion for Object Detection in Autonomous Vehicles. Information Fusion, 98, 102210. https://doi.org/10.1016/j.inffus.2023.102210

[13] Tan, M., Pang, R., & Le, Q. V. (2021). EfficientDet: Scalable and Efficient Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(11), 4014–4026. https://doi.org/10.1109/TPAMI.2020.2979456

[14] Chen, Y., Zhao, X., & Jia, K. (2022). Selective Feature Fusion for Object Detection. IEEE Transactions on Image Processing, 31, 2889–2901. https://doi.org/10.1109/TIP.2022.3154976

[15] Gao, J., Lin, Z., & Liu, J. (2023). Cross-Scale Attention for High-Resolution Object Detection in Remote Sensing Images. ISPRS Journal of Photogrammetry and Remote Sensing, 195, 345–359. https://doi.org/10.1016/j.isprsjprs.2023.01.009

[16] Zhang, T., Li, H., & Xu, M. (2022). ScaleEqualNet: Scale-Equalizing Pyramid Convolutional Network

for Object Detection. Neurocomputing, 513, 293–304. https://doi.org/10.1016/j.neucom.2022.09.014

[17] Jiang, Y., Chen, D., & Li, S. (2023). Transformer-based Multiscale Feature Aggregation for Object Detection. Pattern Recognition, 139, 109404. https://doi.org/10.1016/j.patcog.2023.109404

[18] Wang, R., Yang, X., & Lu, Z. (2023). Attention-Driven Multi-Resolution Feature Fusion for Aerial Object Detection. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16, 14456–14468. https://doi.org/10.1109/JSTARS.2023.3288003

[19] https://cocodataset.org/#format-data

[20] https://www.kaggle.com/datasets/sovitrath/pascal-voc-07-12