mathematics

MDPI

Article

FSDN-DETR: Enhancing Fuzzy Systems Adapter with DeNoising Anchor Boxes for Transfer Learning in Small Object Detection

Zhijie Li [1], Jiahui Zhang [1], Yingjie Zhang [1], Dawei Yan [1], Xing Zhang [1], Marcin Woźniak [2,*] and Wei Dong [1,*]

[1] College of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; jiahuiedu@xauat.edu.cn (J.Z.)
[2] Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland
* Correspondence: marcin.wozniak@polsl.pl (M.W.); dongwei156@xauat.edu.cn (W.D.)

Abstract: The advancement of Transformer models in computer vision has rapidly spurred numerous Transformer-based object detection approaches, such as DEtection TRansformer. Although DETR's self-attention mechanism effectively captures the global context, it struggles with fine-grained detail detection, limiting its efficacy in small object detection where noise can easily obscure or confuse small targets. To address these issues, we propose **F**uzzy **S**ystem **DN**N-**DETR** involving two key modules: Fuzzy Adapter Transformer Encoder and Fuzzy Denoising Transformer Decoder. The fuzzy Adapter Transformer Encoder utilizes adaptive fuzzy membership functions and rule-based smoothing to preserve critical details, such as edges and textures, while mitigating the loss of fine details in global feature processing. Meanwhile, the Fuzzy Denoising Transformer Decoder effectively reduces noise interference and enhances fine-grained feature capture, eliminating redundant computations in irrelevant regions. This approach achieves a balance between computational efficiency for medium-resolution images and the accuracy required for small object detection. Our architecture also employs adapter modules to reduce re-training costs, and a two-stage fine-tuning strategy adapts fuzzy modules to specific domains before harmonizing the model with task-specific adjustments. Experiments on the COCO and AI-TOD-V2 datasets show that FSDN-DETR achieves an approximately 20% improvement in average precision for very small objects, surpassing state-of-the-art models and demonstrating robustness and reliability for small object detection in complex environments.

Keywords: object detection; transformer; transfer learning; DEtection TRansformer; fuzzy system; adapter

MSC: 68T07; 68T45

check for updates

Academic Editor: Jonathan Blackledge

Received: 11 December 2024
Revised: 5 January 2025
Accepted: 6 January 2025
Published: 17 January 2025
Corrected: 4 November 2025

Citation: Li, Z.; Zhang, J.; Zhang, Y.; Yan, D.; Zhang, X.; Woźniak, M.; Dong, W. FSDN-DETR: Enhancing Fuzzy Systems Adapter with DeNoising Anchor Boxes for Transfer Learning in Small Object Detection. *Mathematics* **2025**, *13*, 287. https://doi.org/10.3390/math13020287

Copyright: © 2025 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Small object detection remains challenging in computer vision as Convolutional Neural Networks (CNNs) [1–4]—the leading approach for extracting spatial features—depend on fixed components like anchor boxes and Non-Maximum Suppression (NMS), which limit flexibility in dynamic environments, especially for small objects. Recently, Transformer-based models like the DEtection TRansformer (DETR) [5–8] have introduced a new paradigm by utilizing self-attention to capture global context, spurring diverse adaptations for object detection [9–12].

Despite advancements, small object detection remains challenging, particularly since small objects in medium-resolution images occupy only a few pixels and are highly susceptible to surrounding noise, leading to reduced detection accuracy [13]. While approaches like super-resolution or the direct use of high-resolution images can improve pixel detail for small objects, they also significantly increase computational demands, especially with high-resolution feature maps, where costs grow exponentially [14–16]. These high-resolution approaches often result in excessive computational redundancy, particularly in irrelevant regions [17]. Furthermore, although DETR-based models are widely applied, they still struggle to capture fine-grained details, making them sensitive to noise and complex backgrounds, underscoring the need for more effective solutions in small object detection [12].

Detecting small objects in images presents a significant challenge due to their limited pixel information, making them highly susceptible to noise and background interference. The difficulty is further compounded in complex and noisy environments, where small objects are often obscured or confused with irrelevant details. Addressing this issue requires a method capable of capturing fine-grained features while minimizing the impact of noise. To this end, fuzzy logic offers an effective solution by handling uncertainty and imprecision in data, particularly in object characteristics such as small shifts in position, scale variations, and label ambiguities. By assigning a degree of fuzziness to each feature, fuzzy logic enables the model to capture fine spatial features and better preserve small object details. Motivated by these strengths, we propose **F**uzzy **S**ystem **DN**N-**DETR** (FSDN-DETR), an enhanced model that integrates fuzzy logic into the DETR framework. In contrast to existing DETR-based models, which rely solely on global attention mechanisms, our approach introduces a fuzzy-based approach to improve local feature sensitivity and robustness in challenging environments.

FSDN-DETR consists of two main components: the Fuzzy Adapter Transformer Encoder (FATE) and the Fuzzy Denoising Transformer Decoder (FDTD). FATE introduces the Fuzzy Attention Cross-Domain Module (FACM), which adjusts input features within the pre-trained Vision Transformer (ViT) encoder using fuzzy rules and membership functions. This module enhances the model's ability to capture small variations in object position, shape, and size, providing flexibility in handling the uncertainty of object boundaries. FDTD, on the other hand, integrates a fuzzy attention mechanism that dynamically adapts attention weights based on noise characteristics such as positional shifts and scaling distortions, enabling the model to focus more effectively on relevant object features while suppressing noise.

To further optimize training, we employ a two-stage fine-tuning strategy. In the first stage, only the fuzzy modules (FACM and FDTD) are fine-tuned while the pre-trained ViT encoder and decoder are frozen. In the second stage, the entire model is unfrozen, allowing it to adapt more efficiently to domain-specific features. This approach reduces computational overhead while ensuring that the model benefits from both pre-training and domain-specific fine-tuning. Our experimental results, illustrated in Figure 1, demonstrate that FSDN-DETR outperforms existing methods in small object detection, especially when trained with limited data, highlighting its robustness and efficiency in noisy environments.

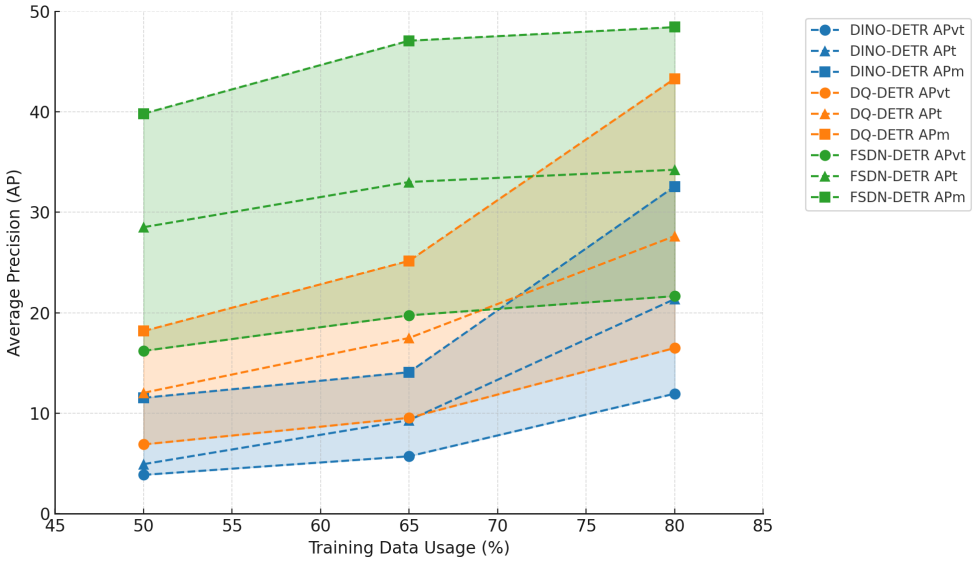


Figure 1. Small Object Detection Performance and Transfer Learning Capabilities of Different Models. Pre-trained on COCO and Fine-tuned with Varying Percentages of AI-TOD-V2 Data. The graph illustrates the Average Precision (AP) values across different training data usage percentages, comparing models DINO-DETR, DQ-DETR, and FSDN-DETR in terms of AP for very tiny, tiny, and medium object detection categories.

Our work makes key contributions in four main areas:

1. **Innovative Fuzzy Logic Integration:** Unlike traditional DETR models, which struggle with fine-detail preservation, we integrate fuzzy logic into the DETR framework to enhance small object detection. This approach captures fine-grained spatial features and reduces uncertainty, improving detection accuracy in complex environments.
2. **Improved Cross-Domain Adaptability:** The Fuzzy Attention Cross-Domain Module (FACM) dynamically adjusts the ViT encoder, enhancing domain-specific feature learning and boosting robustness for cross-domain tasks.
3. **Two-Stage Fine-Tuning Strategy:** We propose a two-stage fine-tuning strategy that optimizes fuzzy logic integration, improving cross-domain performance with minimal computational cost.
4. **Superior Small Object Detection Performance:** Extensive experiments show that FSDN-DETR outperforms existing models, especially in noisy and cluttered environments. It significantly improves fine-grained feature preservation and small object detection accuracy compared to DETR-like models.

2. Related Work

In this section, we elaborate on the advancements in three key areas: (1) Deep learning-based object detection, (2) Deep Neural Fuzzy Systems (DNFS), and (3) Adapter-Based Approaches in Transfer Learning. We discuss the progress and contributions made within each of these domains, highlighting their evolution and significance in the field.

2.1. Deep Learning-Based Object Detection

Deep learning has revolutionized the field of object detection [18], shifting from traditional hand-crafted feature methods to more complex, data-driven approaches. In particular, CNNs and Vision Transformers (ViTs) have significantly advanced the accuracy and efficiency of object detection, as discussed in the following sections on CNN-based and ViT-based methods.

2.1.1. CNN-Based Object Detection

In recent years, deep learning has emerged as a dominant approach in visual tasks. A notable example of this is CNN-based object detection, which is typically classified into two main categories: (1) two-stage detectors [19–22] and (2) single-stage detectors [23–25]. Two-stage detectors first generate region proposals, which are then classified and regressed. In contrast, single-stage detectors integrate both tasks into a single network pass, thereby achieving faster processing times.

The basic idea of two-stage detectors is based on the method of candidate region. Early advancements were marked by Region-based Convolutional Neural Networks (R-CNN) [19], which utilized a selective search for proposal generation and CNNs for feature extraction, yielding notable accuracy improvements but suffering from inefficient processing due to redundant feature computations. This inefficiency was mitigated by SPPNet [20], which introduced spatial pyramid pooling to eliminate repeated feature extractions, significantly enhancing detection speed. Fast R-CNN and its successor [21,26] refined this approach by integrating proposal feature extraction, classification, and cost-free proposal generation through the Region Proposal Network (RPN), achieving notable gains in both speed and accuracy. Subsequent enhancements focused on improving efficiency [27,28], which streamlined computational processes for improved detection speed. Further breakthroughs included the introduction of Feature Pyramid Networks (FPN) [22], which utilized a top-down architecture with lateral connections to better leverage multi-scale features, thereby enhancing detection performance across varying object sizes. As the demand for real-time and computationally efficient models continued to grow, research gradually shifted towards the development of single-stage object detectors. These models, which bypass the proposal generation step by performing object classification and bounding box regression in a single pass, have driven significant advances in detection speed and efficiency, further propelling progress in the field.

One-stage object detectors are characterized by their unified framework, which performs detection based on a regression approach. Typical algorithms, including YOLO [23], SSD [24], and RetinaNet [25], have achieved significant improvements in inference speed while maintaining competitive accuracy, making them particularly suitable for real-time applications such as video processing [29–31]. The YOLO model, introduced in 2015, was the first to demonstrate the feasibility of such an approach, achieving high speed and moderate accuracy by simultaneously predicting bounding boxes and class probabilities. However, early versions of YOLO faced challenges in detecting small objects due to their reliance on coarse grid-based feature maps. Subsequent models, including YOLOv2 and YOLOv3 [32,33], addressed these limitations with improved resolution and multi-scale strategies, enhancing accuracy, especially for larger objects. Similarly, the Single Shot MultiBox Detector (SSD) advanced one-stage detection by leveraging multi-scale feature maps across different layers of the network, significantly improving detection accuracy. Despite these advancements, one-stage detectors, such as YOLO and SSD, continued to lag behind two-stage detectors in small object localization, primarily due to their lack of fine-grained feature extraction. RetinaNet [25], proposed in 2017, introduced the focal loss function to mitigate class imbalance, further bridging the gap between one-stage and two-stage models by enhancing accuracy without sacrificing speed. However, challenges remain in small object detection, as one-stage models still struggle to accurately localize small objects.

To this end, our proposed model addresses these limitations by introducing a novel feature extraction strategy and enhancing multi-resolution capabilities, which significantly improve small object detection performance. By refining the network architecture to better capture fine-grained details, our model achieves superior localization accuracy

and robustness in small object detection, outperforming existing one-stage detectors and offering a more effective solution for real-time applications requiring high precision.

2.1.2. ViT-Based Object Detection

Recent advances in Transformer-based models have enhanced object detection, leveraging self-attention mechanisms to capture global dependencies and improve accuracy. Originally designed for Natural Language Processing, these models have proven effective in vision tasks, but small object detection remains a challenge due to issues like occlusion, low resolution, and noise. The DETR [9] addressed these by introducing an end-to-end framework that eliminates traditional post-processing methods like NMS and anchor boxes. However, DETR faced slow convergence and struggled with fine-grained details.

Building upon DETR's framework, Deformable DETR [5] introduced a deformable attention mechanism that targets specific sampling points, allowing the model to focus on relevant parts of the image and improve both computational efficiency and spatial precision. This modification greatly enhances the performance in detecting objects at various scales, including smaller ones. Additionally, Efficient DETR [6] further refines the attention mechanism by selecting top K positions from encoder predictions, optimizing decoder queries, and improving overall efficiency.

Another significant improvement is DN-DETR [7], which addresses the issue of slow training times in Transformer-based models by incorporating a denoising approach, which introduces noise into the ground-truth labels and bounding boxes during training. This accelerates learning, enhances robustness, and improves the model's practical applicability. Building on this, DINO-DETR [8] further advances the detection process by integrating contrastive denoising training, hybrid query selection, and dual forward prediction. To tackle the detection of small objects, particularly in dense or cluttered environments, DQ-DETR [12] introduced a dynamic query selection mechanism. By adjusting the number and position of queries based on the complexity of each scene, DQ-DETR improves the detection of small objects and reduces false positives, enhancing recall in dense environments. However, this approach only partially addresses the challenges of small object detection, and further improvements are needed to fully overcome issues such as occlusion and noise in real-world scenarios.

These advancements demonstrate the strong synergy between ViT architectures and DETR-based models, leading to significant improvements in object detection in terms of accuracy, and robustness. However, despite these successes, small object detection remains a persistent challenge due to issues such as occlusion, low resolution, and noise. While existing Transformer-based models, including DETR and its derivatives, achieve remarkable performance in general object detection, they still face difficulties in capturing the fine-grained details of small objects in complex environments. Our model overcomes these limitations by integrating fuzzy logic with the Transformer backbone, improving the sensitivity to small, occluded, and noisy objects. This integration boosts the model's robustness and adaptability, particularly in dense or cluttered settings, offering a more reliable solution for small object detection in real-world applications.

2.2. Deep Neural Fuzzy System

The DNFS represents a significant advancement by combining deep neural networks (DNNs) with fuzzy systems, enhancing the handling of uncertainty and improving interpretability [34–37]. Fuzzy systems, including fuzzy logic and neuro-fuzzy models, have gained widespread adoption due to their effectiveness in environments where traditional binary logic is impractical. They use fuzzy IF-THEN rules to represent knowledge, which is particularly useful in uncertain or ambiguous settings. A key example is the Adaptive

Neuro-Fuzzy Inference System (ANFIS) [38], introduced in 1993, which combines fuzzy inference with neural networks to manage uncertainty more effectively.

Building on this, Talpur et al. [34] introduced a novel DNFS by integrating DNNs with fuzzy systems, allowing the system to utilize fuzzy rules for improved decision-making. Further advancements were made by Ali et al. [39], who introduced a Fuzzy Multilayer Perceptron for skin lesion detection, demonstrating the effectiveness of fuzzy activation functions in handling complex tasks. The integration of fuzzy logic with deep learning models, such as CNNs, has enhanced model interpretability and performance, particularly in image processing tasks. By incorporating fuzzy rules, these hybrid models address issues of uncertainty, improving feature extraction and classification, as seen in video emotion recognition and image classification.

The integration of fuzzy systems with ViTs has further advanced the field, with Liu et al. [40] proposing a Fuzzy Transformer Fusion Network for medical image segmentation, which outperformed state-of-the-art algorithms. This synergy has led to the development of Fuzzy-ViT [41], an advanced DNFS designed to leverage large-scale visual data for cross-domain transfer learning, particularly in uncertain and complex environments such as medical applications.

Our research explores the integration of DNFS with a Transformer-based object detection model to improve their performance and generalization. By embedding fuzzy systems into DETR, we enhance their ability to handle uncertainty, improve detection accuracy, and enable better generalization in noisy, complex environments.

2.3. Adapter-Based Approaches in Transfer Learning

The rapid evolution of model architectures, coupled with the escalating demands on scale and training costs, has introduced significant challenges in Transformer-based tasks. As model size grows, the associated training difficulties become a substantial barrier to the efficient deployment and optimization of these architectures. Recent advancements have introduced several strategies to mitigate these challenges, including updating only newly incorporated parameters or those specific to the model's input, selectively modifying a small subset of existing parameters, and employing low-rank factorization techniques to optimize the weights subjected to updates. These approaches have been integrated into recent research to develop unified parameter-efficient training frameworks that substantially alleviate computational burdens [42,43].

Among these strategies, Adapter-based methods have gained considerable attention in both computer vision [44,45] and natural language processing [45,46] due to their ability to introduce task-specific parameters without requiring full model retraining, thus preserving computational efficiency. In contrast to prompt-based techniques, which incorporate trainable parameters into the input, Adapters enable localized fine-tuning of pre-trained models, offering a more efficient solution for task adaptation [47–49]. Recently, parameter-efficient techniques have been extended to models like CLIP, with a focus on improving image-text alignment [50,51].

Building on this foundation, Chen et al. introduced Adapters in the ViT, demonstrating their utility in fine-tuning large pre-trained models for downstream tasks [52]. This extension addresses the challenge of applying large models to specific tasks without retraining the entire model, improving both adaptability and computational efficiency. In 2023, the SAM-Adapter further advanced this concept by integrating domain-specific information into the SAM model for image segmentation, overcoming the limitations of conventional fine-tuning [53]. Despite these advancements, the issue of efficiently adapting large models to complex tasks such as small object detection remains. Existing Adapter

methods have yet to fully address the challenge of knowledge sharing across tasks in multi-task scenarios, which is crucial for improving performance on low-resource datasets.

Our research tackles this gap by applying Adapter-based techniques to small object detection in challenging environments. We demonstrate that enabling Adapters to share information across tasks not only enhances performance but also reduces the number of trainable parameters, making them a more efficient solution for complex visual tasks. This extension of Adapter-based methods to small object detection highlights their potential for overcoming existing limitations and provides a pathway for further advancing the adaptability and efficiency of large-scale pre-trained models.

3. Methodology

This section delves into the methodology behind the FSDN-DETR model, designed to address the challenges of object detection in complex and noisy environments. The model's robustness and adaptability are achieved through the integration of fuzzy logic systems and DeNoising Anchor Boxes within the DN-DETR architecture. To provide a comprehensive understanding of our approach, we will detail the model's key components, starting with an overview, followed by an in-depth discussion of the FATE, the FDTD, the loss function, and the training strategy.

3.1. Overview

We propose the FSDN-DETR to address the challenges of object detection in complex and noisy environments. This model introduces a novel integration of fuzzy logic systems with DeNoising Anchor Boxes, strategically embedded into the DN-DETR architecture via an adapter module. The aim is to enhance the model's robustness and adaptability, especially when processing uncertain and ambiguous features in the data. The FSDN-DETR framework primarily comprises three key components: Linear Projection Flattened Patches, the Fuzzy Adapter Transformer Encoder (FATE), and the Fuzzy Denoising Transformer Decoder (FDTD), as illustrated in Figure 2.

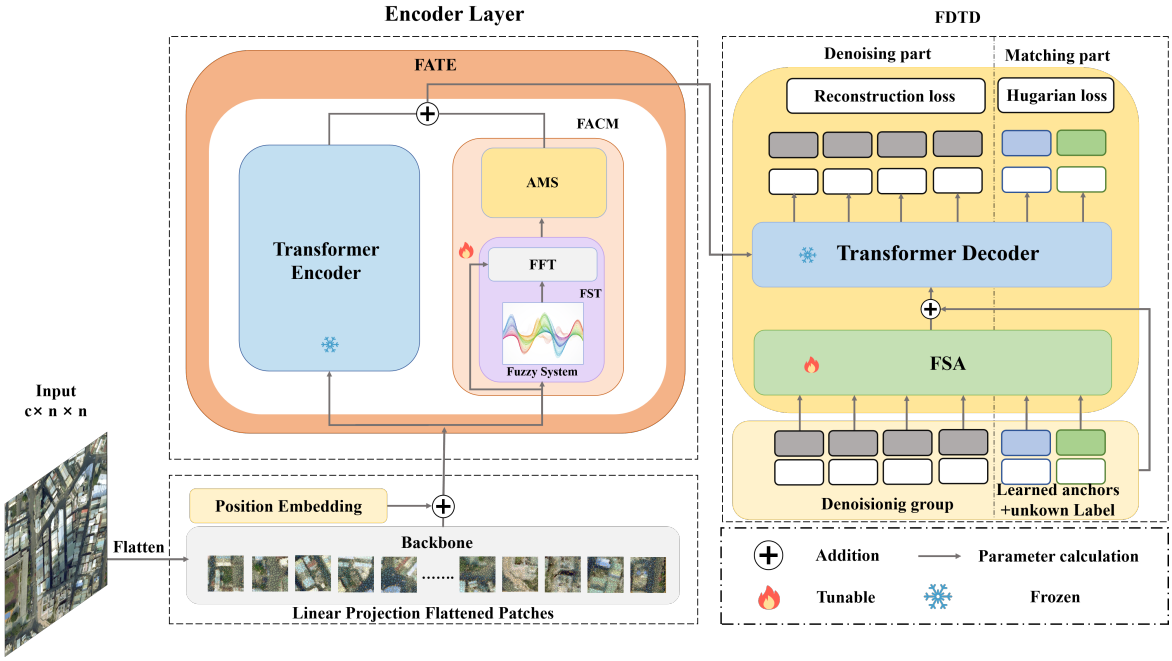


Figure 2. Architecture of the FSDN-DETR Framework. The model leverages the FATE to enhance feature representation and adaptability, while the FDTD mitigates noise and refines object detection results. This integration aims to improve the robustness of object detection under complex and noisy environmental conditions.

The process begins with the input image, represented as $I \in \mathbb{R}^{C \times H \times W}$, where C, H, and W denote the number of channels, height, and width, respectively. This image is first passed through a linear projection layer, which divides it into a sequence of flattened patches, denoted as $X_p \in \mathbb{R}^{N \times D}$, where $N = \frac{H \times W}{p^2}$ is the total number of patches and D is the dimensionality of the patch embedding. These patch embeddings are then used to form a patch-based representation, which serves as the foundation for subsequent transformations, facilitating the efficient processing of visual information.

Next, the patch embeddings are fed into the FATE module, where the features undergo two parallel processes. The Transformer Encoder processes the input features $F_e \in \mathbb{R}^{N \times D}$, refining them with self-attention mechanisms. At the same time, the FACM transforms the same feature set into a fuzzy domain, producing fuzzy-enhanced features $F_f \in \mathbb{R}^{N \times D_f}$, where D_f represents the dimensionality of the fuzzy-enhanced features. The dual-path processing within FATE combines the precise feature extraction of the Transformer with the adaptive, uncertainty-handling capabilities of fuzzy logic, resulting in a richer and more robust feature representation.

The refined feature maps F_e and F_f are then passed to the FDTD module, where they are concatenated or selectively combined before being processed. The FDTD module incorporates DeNoising Anchor Boxes into the decoding process, applying these to the combined feature map $F_d \in \mathbb{R}^{N \times D_d}$, where D_d is the dimensionality after denoising. This crucial step mitigates noise and enhances the clarity of object detection. The final output, Y_{output}, consists of the predicted object classes and bounding boxes, providing improved accuracy and robustness, particularly in environments with high levels of noise and ambiguity.

To optimize the performance of FSDN-DETR, we employ a gradually fine-tuned training strategy. During pre-training, specific components, such as the Transformer Encoder within the FATE module are frozen, while the Fuzzy Attention Cross-Domain Module is fine-tuned. Similarly, in the FDTD module, the Transformer Decoder is frozen, and the focus is placed on fine-tuning the Fuzzy System Attention (FSA) module. This targeted approach ensures that the model achieves superior performance across various object detection tasks, enabling quick adaptation to new, specialized domains with minimal retraining.

3.2. Fuzzy Adapter Transformer Encoder

The FATE module, depicted in Figure 3, is a key component introduced after patch embedding in the ViT, designed to direct input features to both the Transformer Encoder and the FACM for robust cross-domain transfer learning. By incorporating the Fuzzy System Transitioner (FST) and the Attention Mechanism Smoother (AMS) within the FACM, FATE leverages fuzzy logic to address uncertainties and complexities inherent in specialized domains. Building on established methods [41], this design enables fine-tuning of the FACM while keeping the Transformer Encoder frozen. The FACM acts as an adapter, allowing the Transformer Encoder to remain frozen during fine-tuning, with updates focused on the FACM to tailor the model to the target domain. This strategy enhances the model's adaptability while retaining the pre-trained Encoder's feature extraction strength, making it an effective approach for adapting general vision models to specialized domains. The following sections will further explore the Fuzzy System in FATE and the Adapter Structure by FATE.

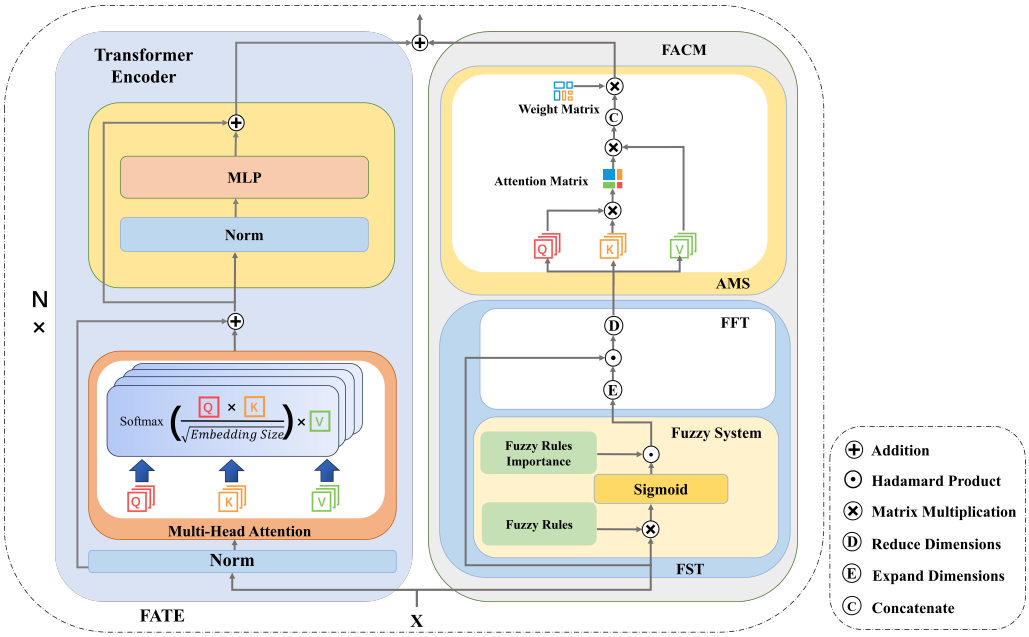


Figure 3. FATE Module Architecture. FATE is introduced after the patch embedding stage in the Vision Transformer and routes features to both the Transformer Encoder and the FACM. The FST and AMS are integrated to handle uncertainties, facilitating robust cross-domain learning.

3.2.1. Adapter Structure by FATE

The FATE module integrates the FACM with the Transformer Encoder via an adapter structure, enabling efficient transfer learning. This design allows the parameters of the Transformer Encoder to remain frozen during fine-tuning, while the FACM is independently adjusted to better suit the specific characteristics of the target domain. This approach leverages the robust feature extraction capabilities of the pre-trained Transformer Encoder while allowing the FACM to focus on domain-specific nuances for enhanced adaptability.

Let x represent the input features, which are passed through the Transformer Encoder and FACM. The output from the Transformer Encoder can be denoted as $T(x)$, while the output from the FACM is denoted as $F(x)$.

The adapter structure allows the FACM to be inserted into the Transformer Encoder without modifying the core architecture. This design is crucial for transfer learning, where the main backbone (the Transformer Encoder) is pre-trained on large-scale general datasets. During domain-specific tasks, the FACM is fine-tuned, which is computationally more efficient than retraining the entire model. This mechanism not only reduces computational costs but also enhances the model's performance by effectively transferring knowledge to specialized domains.

The Transformer Encoder is designed to capture complex dependencies in input data through several key components. At its core, the Multi-Head Attention (MHA) mechanism enables the model to focus on different parts of the input sequence by computing attention scores across multiple heads, capturing relationships between tokens through weighted sums of query, key, and value representations. This allows the model to learn diverse features from various input subspaces. Following MHA, Layer Normalization stabilizes the training process by normalizing activations across features, addressing issues such as vanishing or exploding gradients. The output is then refined through a Multi-Layer Perceptron (MLP) that applies non-linear transformations to produce higher-level representations. The final output of the Transformer Encoder is given by the following:

$$T(x) = \text{Encoder}(x; \theta_T), \tag{1}$$

where θ_T represents the parameters of the Transformer Encoder, which remain fixed during the fine-tuning process.

The FATE incorporates the FACM through an adapter mechanism, enabling it to adapt to domain-specific characteristics. The output of the FACM is expressed as follows:

$$F(x) = \text{FACM}(x; \theta_F), \tag{2}$$

where θ_F denotes the learnable parameters of the FACM, which are fine-tuned to align with the specific requirements of the target domain. This allows the FACM to enhance the model's attention mechanism by incorporating fuzzy logic principles, which effectively address domain-specific uncertainties and ambiguities in feature extraction.

The final output of FATE is obtained by combining the outputs of the Transformer Encoder and the FACM, weighted by learnable factors α and β:

$$Y = \alpha \cdot T(x) + \beta \cdot F(x), \tag{3}$$

where α and β are parameters that control the contributions of the Transformer Encoder and FACM, respectively. These factors are optimized during fine-tuning to strike a balance between the generalization capabilities of the Transformer Encoder and the domain-specific adaptability of the FACM. This weighted combination enables FATE to effectively leverage both components, enhancing its overall performance for specialized tasks.

3.2.2. Fuzzy System in FATE

The Fuzzy System, embedded within the FACM in the FATE structure, plays a crucial role in handling the complexities and ambiguities inherent in domain-specific datasets. This system operates by transforming the feature space from the ViT backbone into a fuzzy domain, thereby enriching the model's ability to interpret uncertain and imprecise data. Within the Fuzzy System, two key components are used: the FST and the AMS. The FST facilitates the transformation of features into a fuzzy representation, allowing the model to capture and process uncertainty. Meanwhile, the AMS smooths the attention mechanism, ensuring that the model focuses more effectively on the most relevant features, further enhancing its cross-domain transfer learning capabilities.

FST: The transformation process starts within the FST module, using a learnable fuzzy rule base to adapt general features to domain-specific tasks. This rule base is mathematically represented as follows:

$$w_r \in \mathbb{R}^{N_r \times D}, \tag{4}$$

where N_r denotes the number of fuzzy rules, and each row of w_r corresponds to a distinct fuzzy rule. This structure allows each rule to be applied to features, ensuring that the feature space is transformed in alignment with the specific requirements of the domain.

A Gaussian activation function $\phi(w_r^i, X)$ is employed to compute the similarity between input features and fuzzy rules, capturing spatial and semantic relevance. The membership degrees M are then aggregated across all fuzzy rules:

$$M = \sum_{i=1}^{N_r} \phi(w_r^i, X). \tag{5}$$

To normalize M within the range $[0, 1]$, a sigmoid function ρ is applied:

$$M_{\text{norm}} = \rho(M) = \frac{1}{1 + e^{-M}}. \tag{6}$$

In the Fuzzy Feature Transformer (FFT), the fuzzy system plays a pivotal role in managing the uncertainty and complexity inherent in specialized datasets. By leveraging a fuzzy rule base and membership function calculations, the FFT enhances the model's ability to interpret uncertain or imprecise data, which is especially valuable in domains where feature ambiguity or noise is prevalent. To account for the varying significance of different fuzzy rules, learnable weights $\alpha \in \mathbb{R}^{N_r}$ are introduced. These weights enable the model to adjust the influence of each rule, resulting in refined membership degrees M_{adj}, which are computed as follows:

$$M_{\text{adj}} = M_{\text{norm}} \odot \alpha, \tag{7}$$

where $\odot$ denotes the Hadamard product. This adjustment ensures that the model focuses on the most diagnostically relevant fuzzy rules.

AMS: The next stage, occurring within the AMS module, involves enriching the feature representations by integrating fuzzy logic interpretations into the input features. The adjusted membership values M_{adj} are expanded to match the dimensions of the input feature map, denoted x. The expansion is mathematically represented as follows:

$$M_{\text{exp}} = \text{Exp}(M_{\text{adj}}, D), \tag{8}$$

where $\text{Exp}(\cdot)$ is the dimension-expansion function. This alignment ensures that each feature in x is appropriately scaled by its corresponding membership value. The transformed features are then generated by element-wise multiplication:

$$F_t = x \odot M_{\text{exp}}. \tag{9}$$

This operation embeds fuzzy logic into the feature space, resulting in a fuzzy-enhanced feature representation, denoted F_t. Finally, the dimensions of F_t are reduced to preserve the structural integrity of the original feature map:

$$x_t = \text{Red}(F_t), \tag{10}$$

where $\text{Red}(\cdot)$ represents the dimensionality-reduction function. This process ensures that the data's structural consistency is preserved while maintaining the fuzzy logic insights, making x_t more effective for handling complex, domain-specific challenges.

The refined fuzzy-transformed features x_t are then passed through an attention mechanism to further enhance their relevance for classification tasks. This attention mechanism is similar to those used in Transformer architectures and is crucial for identifying the most significant features in the data. The queries (Q), keys (K), and values (V) are split into multiple attention heads, enabling the model to capture intricate relationships within the data. The splitting operation is expressed as follows:

$$\{Q_h, K_h, V_h\} = \text{SIH}(Q, K, V, N_H), \tag{11}$$

where $\text{SIH}(\cdot)$ denotes the splitting function and N_H represents the number of attention heads. Within each attention head, the dot product of Q_h and K_h is computed and scaled by $\sqrt{d_k}$, where d_k is the dimensionality of the keys. This scaled product is passed through a Softmax function to produce the attention matrix A_h:

$$A_h = \text{Softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right). \tag{12}$$

This attention matrix helps the model focus on the most relevant features by guiding it to prioritize significant aspects of the data. The final step involves applying the attention matrix to the values V_h to compute a weighted sum, which results in the refined feature set V_h':

$$V_h' = A_h V_h, \quad h \in \{1, 2, \ldots, N_H\}. \tag{13}$$

The outputs from all attention heads are concatenated and passed through a linear transformation with a weight matrix W_O, aggregating the information into a coherent representation:

$$X_{ts} = \text{Concat}(V_1', V_2', \ldots, V_{N_H}')W_O. \tag{14}$$

The final output, X_{ts}, represents the smoothed and refined feature map after the fuzzy transformation, optimized for domain-specific classification tasks. In this process, the operation Concat concatenates the transformed feature matrices $V_1', V_2', \ldots, V_{N_H}'$ along the feature dimension, enhancing the model's ability to capture a wider range of domain-specific characteristics. The concatenated features are then projected using W_O, further refining the representation for the specific task. This dual-path feature aggregation strategy improves the model's capacity to handle complex, uncertain data, particularly in environments where high precision is critical.

3.3. Fuzzy Denoising Transformer Decoder

The FDTD module, as shown in Figure 4, integrates an FSA mechanism within the transformer decoder to enhance object detection in noisy environments. It uses fuzzy rules and membership functions to assess noise impact and adaptively adjust attention scores based on the level of distortion in the input queries. This allows the model to prioritize cleaner, more reliable features, improving detection robustness. The effectiveness of this approach depends on the interplay between key components, including the fuzzy rule base, attention mechanism, transformer architecture, and loss function, all of which contribute to enhanced noise handling and prediction accuracy.

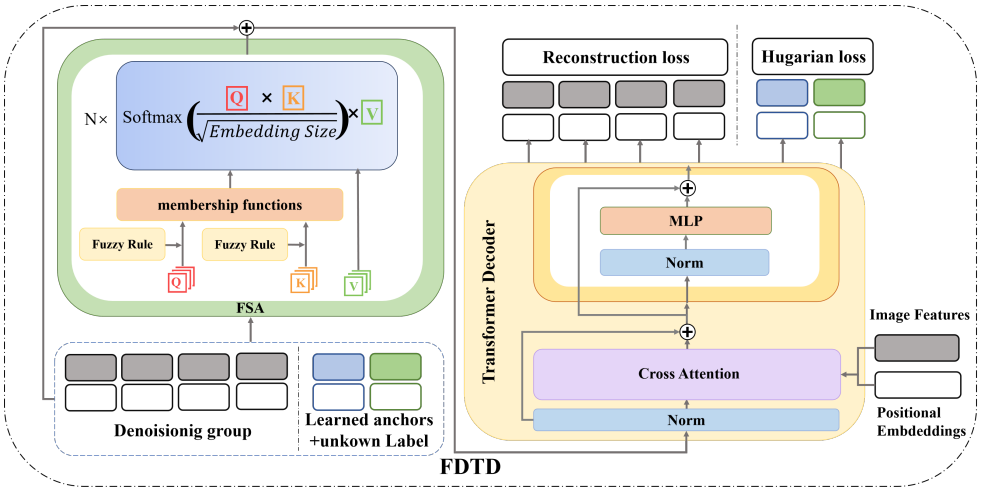


Figure 4. FDTD Module Architecture. The FDTD integrates FSA in the transformer decoder using adaptive attention scores based on noise levels to prioritize less noisy information, improving detection robustness in noisy environments.

3.3.1. Adapter Structure by FDTD

The FDTD module integrates the FSA with the transformer decoder through an adapter structure, facilitating the seamless interaction between fuzzy logic adjustments and the robust query-based object detection capabilities of the DN-DETR decoder [7]. This decoder explicitly formulates queries as box coordinates and processes inputs in two parts: the

matching part, which uses learnable anchors to approximate ground truth box-label pairs via bipartite graph matching, and the denoising part, which reconstructs noisy ground truth objects by introducing grouped variations.

The FSA enhances the decoder's performance in noisy environments by dynamically adjusting attention scores. Specifically, fuzzy attention scores A_{fuzzy} are computed as a weighted sum of membership degrees:

$$A_{\text{fuzzy}} = \sum_{k=1}^{N_r} \alpha_k M_k, \tag{15}$$

where N_r is the total number of fuzzy rules, α_k denotes the activation strength of the k-th rule, and M_k is its corresponding membership degree. These scores refine the Transformer attention weights $A_{\text{transformer}}$ through element-wise multiplication:

$$A_{\text{adj}} = A_{\text{transformer}} \odot A_{\text{fuzzy}}, \tag{16}$$

where $\odot$ represents the Hadamard product. This mechanism allows the model to focus on less noisy features, bridging the fuzzy system and the decoder for improved query-based object detection.

To prevent information leakage during training, the decoder employs an attention mask A, which enforces a separation between matching and denoising tasks. The mask is defined as follows:

$$A = [a_{ij}]_{W \times W}, \quad \text{where} \quad W = P \times M + N, \tag{17}$$

ensuring that

$$a_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ belong to different groups or tasks,} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Here, P is the number of denoising groups, M is the number of ground truth objects, and N is the number of matching queries. This design prevents interactions between noisy versions of the same object, reinforcing group separation.

The comprehensive attention matrix $A_{\text{comprehensive}}$ combines the adjusted attention weights with the attention mask as follows:

$$A_{\text{comprehensive}} = (A_{\text{transformer}} \odot A_{\text{fuzzy}}) \odot A, \tag{19}$$

where $A_{\text{transformer}}$ represents the attention weights from the Transformer model, A_{fuzzy} indicates the fuzzy adjustments, and A is the attention mask. This formulation effectively integrates the Transformer's attention, fuzzy logic-based refinements, and structural constraints, guiding the attention mechanism to enhance the model's robustness to noise while focusing on cleaner and more relevant data for accurate object detection.

3.3.2. Fuzzy System in FDTD

The FSA mechanism is the core component of the FDTD module and serves as an adapter within the Transformer Decoder to enhance the model's performance in noisy environments. This mechanism dynamically adjusts attention weights based on fuzzy logic, effectively mitigating the impact of noise on model predictions.

The calculation of attention weights in the FSA begins with aggregating outputs from fuzzy rules, each weighted by its activation strength. For a given target object within a denoising group, the FSA mechanism computes the activation strength of each rule based

on input membership values derived from the noise characteristics. The final attention weight is determined as a weighted sum of rule outputs:

$$W_{\text{Fuzzy-Attention}} = \sum_{i=1}^{n} w_i \cdot O_i^{\text{Rule}}, \tag{20}$$

where w_i represents the weight of the i-th rule, determined empirically or via data-driven approaches, and O_i^{Rule} is the output of the i-th rule. Each rule output is computed as the product of its activation strength, S_i^{Strength}, and its output weight, w_i:

$$O_i^{\text{Rule}} = S_i^{\text{Strength}} \times w_i. \tag{21}$$

The activation strength of a rule, S_i^{Strength}, quantifies the degree to which the rule is applicable to the current input conditions and is computed as the minimum membership value among all input features:

$$S_i^{\text{Strength}} = \min\Big(\mu_{\text{Shift}}(x, y), \mu_{\text{Scale}}(w, h), \mu_{\text{Flip}}(l)\Big). \tag{22}$$

This formulation ensures that a rule's contribution is proportional to its relevance, as indicated by the degree of satisfaction of its input conditions.

The fuzzy rules and corresponding membership functions play a pivotal role in defining the behavior of the FSA. Three primary rules are implemented: the Center Shifting Rule, the Box Scaling Rule, and the Label Flipping Rule. Each rule is equipped with membership functions that capture the "fuzziness" of the associated operations, facilitating a smooth transition between varying levels of distortion.

For the Center Shifting Rule, a bounding box center shift is classified as a "Small Shift" if the shift values Δx or Δy satisfy $|\Delta x| < \lambda_1 \times w/2$ and $|\Delta y| < \lambda_1 \times w/2$. The membership function $\mu_{\text{Shift}}(x, y)$ is defined as follows:

$$\mu_{\text{Shift}}(x, y) = \max\left(1 - \frac{|\Delta x|}{\lambda_1 \times w/2}, 1 - \frac{|\Delta y|}{\lambda_1 \times h/2}\right). \tag{23}$$

This function assigns higher membership values to smaller shifts, indicating minimal impact on the model's predictions. Similarly, the Box Scaling Rule governs the scaling of bounding box dimensions. Scaling operations are classified as "Minor Scaling" if the width and height lie within $[(1 - \lambda_2)w, (1 + \lambda_2)w]$ and $[(1 - \lambda_2)h, (1 + \lambda_2)h]$, respectively. The corresponding membership function is as follows:

$$\mu_{\text{Scale}}(w, h) = \max\left(1 - \frac{|w_{\text{scaled}} - w|}{\lambda_2 \times w}, 1 - \frac{|h_{\text{scaled}} - h|}{\lambda_2 \times h}\right), \tag{24}$$

where w_{scaled} and h_{scaled} denote the scaled dimensions. This formulation ensures higher membership values for minor deviations from the original dimensions, signifying less significant distortions. The Label Flipping Rule, controlled by the hyperparameter ξ, classifies label flipping as "Minor Flipping" for small values of ξ. Its membership function is defined as follows:

$$\mu_{\text{Flip}}(l) = 1 - \xi. \tag{25}$$

This simple function reflects the inverse relationship between label flipping probability and its disturbance impact.

The FSA mechanism employs a set of rules and membership functions to systematically adjust attention weights, thereby prioritizing inputs with lower levels of noise. This fuzzy inference approach enables the model to integrate refined attention weights in a

seamless manner, thereby enhancing the robustness and accuracy of object detection tasks, particularly in challenging conditions.

3.4. Loss

In the FDTD, two primary loss functions are employed: the reconstruction loss and the Hungarian loss. These loss functions are meticulously designed to guide the model in accurately recovering and matching object detection results under noisy and fuzzy conditions, thereby enhancing the model's robustness and precision.

The Reconstruction Loss consists of the L1 loss and the Generalized Intersection over Union (GIoU) loss. The L1 loss measures the absolute difference between predicted and ground truth bounding boxes, stabilizing predictions in noisy settings. The GIoU loss penalizes spatial discrepancies between predicted and ground truth boxes, encouraging tighter and more accurate localization.

$$\text{Loss}_{L1} = \sum_{i=1}^{N} |\hat{b}_i - b_i|, \tag{26}$$

where $\hat{b}_i$ represents the predicted bounding box, b_i is the ground truth bounding box, and N is the number of objects in the image.

$$\text{Loss}_{GIoU} = 1 - \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} + \frac{|C - (B_p \cup B_{gt})|}{|C|}, \tag{27}$$

where B_p and B_{gt} are the predicted and ground truth bounding boxes, and C is the smallest enclosing box containing both.

The Hungarian Loss optimizes the matching of predicted and ground truth boxes using the Hungarian algorithm, minimizing the matching cost. The label loss component of this, represented by Focal Loss, helps the model handle class imbalance by increasing the weight of hard-to-predict samples:

$$\text{Loss}_{\text{Label}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t), \tag{28}$$

where p_t is the predicted probability for class t, and α_t and γ are hyperparameters that control the balance between easy and hard samples.

The total loss combines these components, facilitating optimal matching between predicted and ground truth boxes while also refining the bounding box predictions.

3.5. Training Strategy

The training strategy for the FSDN-DETR model is divided into two main phases: the pretraining phase and the staged fine-tuning phase as illustrated in Figure 5. During the pretraining phase, the model's core components, including the ViT-based Transformer Encoder and Decoder, are initialized and trained on a large-scale, general-purpose dataset. This phase allows the model to learn fundamental visual patterns and feature representations without domain-specific adaptations, thereby establishing a robust foundation for visual processing. The objective is to equip the model with broad, transferable features that can be applied to a wide range of object detection tasks. Building on this foundation, the model progresses to the Staged Fine-tuning phase, during which domain-specific adaptations are introduced and implemented in two stages: the Initial Fine-tuning Stage and the Secondary Fine-tuning Stage.

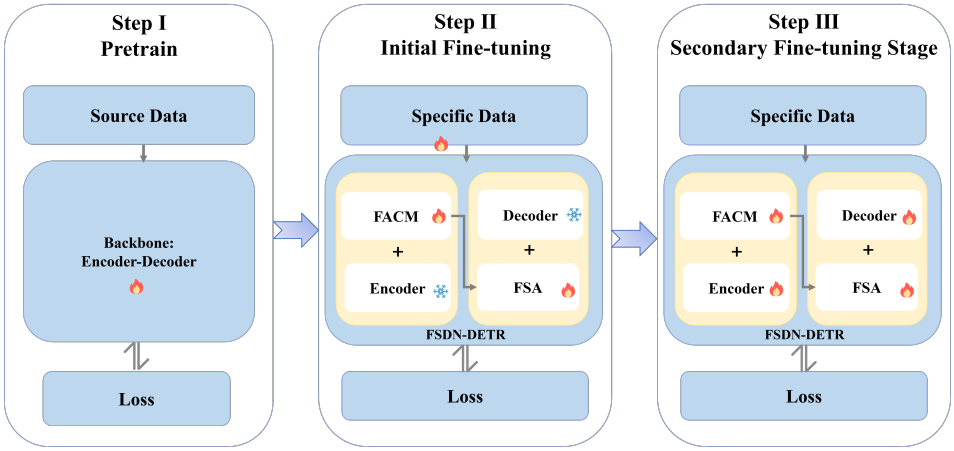


Figure 5. Training Strategy of FSDN-DETR Model. The strategy consists of two phases: pretraining to initialize the backbone with general features and staged fine-tuning to adapt the model to specific datasets through selective tuning of fuzzy logic and transformer components.

Initial Fine-tuning Stage: In this stage, the pre-trained Transformer Encoder is frozen, and the FACM in the FATE is fine-tuned to better manage uncertainties in complex visual data. The pre-trained Transformer Decoder is also integrated into the FSA module, and fine-tuned to enhance its denoising capabilities. This stage focuses on adapting the fuzzy logic modules without altering the pre-trained visual processing components.

Secondary Fine-tuning Stage: In the secondary stage, both the Encoder and Decoder are unfrozen and fine-tuned together as a unified system. This phase harmonizes the pre-trained components with the fuzzy logic adaptations from the initial fine-tuning, enabling the model to achieve a balanced optimization between general visual knowledge and task-specific fuzzy logic adjustments. The entire model is trained together to maximize performance in noisy and ambiguous cross-domain object detection tasks.

This staged approach ensures that the transfer of knowledge from the pre-trained backbone to the newly introduced fuzzy logic modules is gradual and efficient, resulting in improved robustness and accuracy in challenging environments.

4. Experiments

This section evaluates the FSDN-DETR model through a structured analysis, including its pre-training on COCO, fine-tuning on AI-TOD-V2, and ablation studies to examine key components. Performance comparisons and discussions highlight the model's robustness, generalization, and improvements over baselines.

4.1. Datasets

COCO [54]: The COCO dataset (Table 1) contains over 330,000 images with more than 2.5 million object instances, spanning 80 object categories. It provides detailed annotations, including object segmentation masks, key points, and captions, making it a comprehensive resource for object detection tasks. This dataset serves as a foundation for the pretraining of the FSDN-DETR model.

AI-TOD-V2 [55]: The AI-TOD-V2 dataset (Table 1) consists of 28,036 aerial images with a total of 752,745 labeled object instances. It is divided into three subsets: 11,214 images for training, 2804 images for validation, and 14,018 images for testing. The dataset features predominantly small objects, with 86% of instances smaller than 16 pixels and an average object size of 12.7 pixels. The number of objects per image ranges from 1 to 2667, with an average of approximately 24.64 objects per image and a standard deviation of 63.94, reflecting a significant variation in object density.

Table 1. Overview of the datasets used for training and evaluation, including image volume, number of classes, and data source information for COCO2017 and AI-TOD-V2 datasets.

Name	Image Volume	Source
COCO2017 [54]	121,408	Microsoft
AI-TOD-V2 [55]	28,036	Wuhan University

4.2. Experimental Design

The experimental protocol is designed to rigorously evaluate the performance of the FSDN-DETR model by first pre-training it on the COCO dataset using standard object detection techniques, including multi-scale training and data augmentation. These techniques help the model develop robust general feature representations, which are crucial for subsequent fine-tuning. After pre-training, the model fine-tunes on the AI-TOD-V2 dataset, specifically choosing to adapt the model's parameters for the detection of small and densely packed objects. This fine-tuning phase is essential for refining the model's ability to handle the unique challenges posed by small object detection in complex environments.

We conduct a series of baseline comparisons to benchmark its performance against well-established models. Specifically, we compare the FSDN-DETR with CNN-based detectors, such as FCIS [4], Faster R-CNN [21] and Mask R-CNN [56], as well as Transformer-based models, including DETR [9], Deformable DETR [5], DN-DETR [7] and DQ-DETR [12]. These comparisons are performed under consistent training conditions to ensure fairness and provide an objective assessment of the FSDN-DETR model's relative performance. By conducting these comparisons, we aim to demonstrate the advantages of our proposed model in terms of accuracy, robustness, and adaptability, especially in scenarios involving small densely packed objects.

4.3. Implementation Details

The training configuration for the FSDN-DETR model is meticulously optimized following established best practices to balance computational efficiency and model performance. The Adam optimizer is employed, equipped with a learning rate scheduler that initially sets the learning rate to 1×10^{-4} and gradually reduces it based on validation loss to enhance convergence stability. A batch size of 16 is selected to strike an optimal balance between computational demands and the stability of convergence during training. The loss function combines cross-entropy loss with a fuzzy membership loss and is specifically designed to capture the uncertainties in the predictions, thus enhancing the model's robustness in ambiguous scenarios. To maintain consistency and comparability in performance evaluations, the maximum number of detections per image is capped at 500, ensuring a reliable assessment of detection quality even in densely populated scenes. The model is implemented using the PyTorch framework [57], leveraging its flexibility and efficiency in deep learning tasks.

4.4. Evaluation Metrics

The performance of the proposed FSDN-DETR model is evaluated using a comprehensive set of metrics to ensure a robust assessment across various object scales.

AP: The primary metric is used to measure the detection accuracy of the model. AP is calculated as the mean value of precision scores from AP50 to AP95, with an Intersection over Union (IoU) interval of 0.05. This metric provides an overall evaluation of the model's precision across different overlap thresholds.

Scale-specific AP Metrics: To evaluate performance across different object sizes, we use the following scale-specific AP metrics:

- AP_{vt} **(Very Tiny):** The evaluation assesses the model's capacity to identify minute objects, with a particular focus on those measuring between 2 and 8 pixels. This metric represents the performance of the model in relation to the most challenging and intricate detection tasks within the dataset.
- AP_t **(Tiny):** Measures detection performance for tiny objects ranging from 8 to 16 pixels, offering insights into the model's capability to recognize slightly larger but still challenging targets.
- AP_s **(Small):** Assesses detection accuracy for small objects sized 16–32 pixels, commonly found in aerial imagery and complex scenes, highlighting the model's robustness in this size range.
- AP_m **(Medium):** Focuses on the performance for medium-sized objects between 32 and 64 pixels, capturing the model's effectiveness in handling moderately scaled objects.
- AP_l **(Large):** Measures the performance on large objects (>64 pixels), reflecting the model's capability to detect and localize prominently scaled objects with clear boundaries.

These metrics provide a comprehensive evaluation framework, highlighting the model's ability to maintain robust detection performance across a wide range of object sizes and complexities in the AI-TOD-V2 dataset.

4.5. Analysis of COCO Pre-Trained Models' Performance

We compare the performance of our proposed FSDN-DETR model with state-of-the-art DETR-based models on the COCO validation dataset. As shown in Table 2, FSDN-DETR achieves an overall AP of **53.64**, outperforming the best-performing DQ-DETR by **+3.42%**. Notably, FSDN-DETR excels in small object detection (AP_s), achieving a score of **35.42**, which is a significant improvement over all other models.

Table 2. Performance comparison of FSDN-DETR and baseline models on the COCO validation dataset, evaluated using AP metrics for small (AP_s), medium (AP_m), and large (AP_l) object sizes.

Basemodel	Method	Backbone	AP	AP_s	AP_m	AP_l
CNN-based	FCIS [4]+OHEM	ResNet-101-C5	32.21	10.14	34.30	50.38
	Mask R-CNN [56]	ResNet101-FPN	39.54	22.43	43.37	52.21
	Faster R-CNN [21]	ResNet101-FPN	42.02	26.58	45.41	54.75
DETR-like	DETR [9]	ResNet50	42.36	22.47	47.28	61.10
	Deformable DETR [5]	ResNet50	45.84	27.91	49.06	61.81
	DN-DETR [7]	ResNet50	46.07	26.68	50.03	63.38
	DQ-DETR [12]	ResNet50	50.22	31.85	53.19	64.70
Ours	FSDN-DETR	ResNet50	53.64	35.42	55.93	65.94

The enhanced performance in small object detection can be attributed to the integration of the Fuzzy Adapter Transformer Encoder (FATE) module. FATE introduces fuzzy logic into the encoder, allowing the model to focus more effectively on fine-grained details of small objects, which are typically challenging for traditional object detection models. By refining the attention mechanism, FATE improves the model's ability to capture subtle visual features that are critical for detecting smaller objects.

In addition to small object detection, FSDN-DETR shows consistent improvements across medium and large objects, with AP_m and AP_l scores of **55.93** and **65.94**, respectively. These gains are driven by the Fuzzy Denoising Transformer Decoder (FDTD), which enhances the model's ability to handle noise and occlusions—common challenges when detecting medium and large objects in complex scenes. By denoising the feature maps

during the decoding stage, FDTD helps improve localization and reduces false positives, ensuring higher accuracy across all object sizes.

Overall, the FATE and FDTD modules work synergistically, with FATE improving feature extraction for small objects and FDTD enhancing localization and robustness across multiple object sizes. This combination enables FSDN-DETR to achieve state-of-the-art performance on the COCO dataset, demonstrating the effectiveness of fuzzy logic in improving object detection, particularly for small and occluded objects.

4.6. Evaluation of Fine-Tuned Models on AI-TOD-V2

We evaluate the performance of our FSDN-DETR model after applying a Staged Fine-tuning strategy on the AI-TOD-V2 dataset, specifically designed to enhance the detection of small objects. The model is pre-trained on the COCO dataset, during which all parameters are unfrozen, allowing full optimization of both the backbone and Transformer components. For fine-tuning AI-TOD-V2, we employ a staged approach where initially only task-specific modules are trained. This strategy significantly reduces computational costs and improves convergence stability.

As shown in Table 3, FSDN-DETR achieves an overall AP of **31.58**, surpassing the best-performing DQ-DETR by **+1.85%**. Notably, FSDN-DETR excels in small object detection (AP_s), reaching **37.43**, a substantial improvement over all baseline models. Additionally, the model shows superior performance in detecting very small objects as indicated by the AP_{vt} score of **18.20**, demonstrating its effectiveness at handling objects that are even smaller than those typically addressed in the AP_s metric.

Table 3. Performance comparison of Fuzzy-DETR and baseline models on the AI-TOD-V2 test dataset, evaluated using AP metrics for very tiny (AP_{vt}), tiny (AP_t), small (AP_s), and medium (AP_m) object sizes.

Basemodel	Method	Backbone	AP	AP_{vt}	AP_t	AP_s	AP_m
CNN-based	RetinaNet [25]	ResNet50-FPN	8.92	2.49	8.36	13.06	19.89
	Mask R-CNN [56]	ResNet101-FPN	9.83	1.84	8.76	18.53	23.70
	Faster R-CNN [21]	ResNet101-FPN	13.65	0.04	9.53	25.71	37.84
DETR-like	Deformable DETR [5]	ResNet50	18.86	6.49	17.45	25.03	34.14
	DN-DETR [7]	ResNet50	23.38	9.86	22.60	29.74	38.81
	DINO-DETR [8]	ResNet50	25.88	13.02	25.09	32.02	39.65
	DQ-DETR [12]	ResNet50	29.73	15.09	29.38	36.38	44.64
Ours	Fuzzy-DETR	ResNet50	31.58	18.20	31.51	37.43	46.91

These enhancements can be attributed to both the Staged Fine-tuning strategy and the incorporation of the Fuzzy Adapter Transformer Encoder (FATE) and Fuzzy Denoising Transformer Decoder (FDTD) modules. The FATE module enables the model to more effectively capture fine-grained features, which are crucial for small object detection. Meanwhile, the FDTD module enhances localization precision by denoising feature maps, making it particularly effective for detecting smaller and occluded objects.

In addition to small and very small objects, FSDN-DETR demonstrates strong performance across medium objects, with AP_m scores of **46.91**, respectively. These gains further confirm the model's adaptability to various object sizes, driven by the combination of the fine-tuning strategy and fuzzy logic.

In comparison to DETR-like models, FSDN-DETR consistently outperforms across all object sizes, particularly in detecting small and very small objects, where it achieves the highest scores in both AP_s and AP_{vt}. This performance can be attributed to the effective integration of fuzzy logic into the attention mechanism, which enables the model to

more effectively refine feature extraction and enhance detection accuracy for small and occluded objects.

In summary, the results demonstrate that integrating fuzzy logic improves FSDN-DETR's ability to detect small and occluded objects, surpassing traditional DETR models. Its strong performance, particularly with very small objects, highlights the importance of refined feature extraction and attention mechanisms. These improvements emphasize the model's strong transferability to downstream tasks, making FSDN-DETR a promising solution for diverse real-world detection challenges.

4.7. Ablation Study

To further evaluate the adaptability and generalization capabilities of our FSDN-DETR model, we conduct an ablation study using a COCO pre-trained model and fine-tune it on progressively larger portions of the AI-TOD-V2 dataset. Specifically, we use 50%, 65%, and 80% of the training data to analyze the model's performance under varying data availability conditions, comparing it against two competitive DETR-based models, DINO-DETR and DQ-DETR.

As shown in Table 4, FSDN-DETR consistently outperforms DINO-DETR and DQ-DETR across all training data proportions, demonstrating its robust adaptability to varying data availability. When trained on 50% of the data, FSDN-DETR achieves an AP of **30.58**, significantly surpassing DQ-DETR (**15.21**) and DINO-DETR (**11.68**). This performance gap widens as the training data increases: with 65% of the data, FSDN-DETR reaches **33.92**, outpacing DQ-DETR (**23.78**) and DINO-DETR (**18.42**), and with 80% of the data, FSDN-DETR achieves **36.41**, outperforming DQ-DETR (**35.64**) and DINO-DETR (**26.53**). These improvements are particularly evident in AP_{vt} (very small object detection), where FSDN-DETR shows significant gains even with limited data. For example, with 50% of the data, FSDN-DETR achieves an AP_{vt} of **16.20**, surpassing DQ-DETR (**6.89**) and DINO-DETR (**3.86**). As the training data increases, FSDN-DETR continues to excel, reaching **21.65** with 80% of the data, significantly outpacing DQ-DETR (**16.49**) and DINO-DETR (**11.94**). These results underscore the superior performance of FSDN-DETR in detecting small and very small objects, showcasing its effectiveness even in data-limited settings.

Table 4. Ablation Study Results. Performance Comparison of DINO-DETR, DQ-DETR, and FSDN-DETR on AI-TOD-V2 Dataset Using Different Proportions of Training Data: 50%, 65%, and 80%.

Model	Training Data	AP	AP_{vt}	AP_t	AP_s	AP_m
DINO-DETR [8]	50%	11.68	3.86	4.93	8.60	11.53
DQ-DETR [12]		15.21	6.89	12.03	15.34	18.17
FSDN-DETR		30.58	16.20	28.51	33.43	39.81
DINO-DETR [8]	65%	18.42	5.71	9.31	11.32	14.08
DQ-DETR [12]		23.78	9.54	17.49	20.06	25.16
FSDN-DETR		33.92	19.74	33.01	39.16	47.07
DINO-DETR [8]	80%	26.53	11.94	21.36	27.02	32.57
DQ-DETR [12]		35.64	16.49	27.63	36.82	43.30
FSDN-DETR		36.41	21.65	34.23	41.88	48.42

The improvement across all object sizes, including AP_s (small objects) and AP_m (medium objects), further highlights the effectiveness of the fuzzy logic components integrated into the model, specifically the FACM and FSA modules within the FATE and FDTD components. The staged fine-tuning strategy, starting with fuzzy logic adaptation and followed by full model fine-tuning, allows the model to leverage both domain-specific

and general knowledge effectively, resulting in robust performance even under conditions with limited training data.

The ablation study shows that FSDN-DETR outperforms DINO-DETR and DQ-DETR across different training data sizes, demonstrating strong adaptability and transferability. Even with limited training data (50%), FSDN-DETR excels in detecting small and very small objects. The fuzzy logic components and staged fine-tuning strategy enable effective performance across varying data conditions, confirming FSDN-DETR's robustness in new datasets and data-limited scenarios.

5. Discussion and Future Work

5.1. Discussion

The experimental results demonstrate that the proposed FSDN-DETR model achieves notable improvements in small object detection and cross-domain transfer learning. However, several limitations and challenges were observed during the experiments which warrant further discussion.

Handling Dense and Small Objects: The proposed FSDN-DETR model demonstrates strong performance in detecting small and densely packed objects, as evidenced by the successful identification of small targets in various typical environments. However, challenges remain in detecting small objects under conditions such as shadows, low lighting, or occlusion. While the fuzzy logic components within the model significantly enhance its ability to handle slight positional shifts and scale variations, its performance may degrade when small objects are obscured by complex lighting conditions or shadows.

As shown in Figure 6, in scenarios where objects are partially shadowed or in low-light environments, the model's feature extraction capabilities may not fully capture the fine details of small objects, leading to detection errors or missed objects. To address these challenges, further optimization is needed. Future improvements could focus on refining the fuzzy attention mechanism to better handle illumination variations and occlusions. Additionally, incorporating more advanced image enhancement techniques or multi-modal data could improve the model's robustness under challenging lighting conditions, ultimately enhancing its performance in real-world, low-visibility environments.

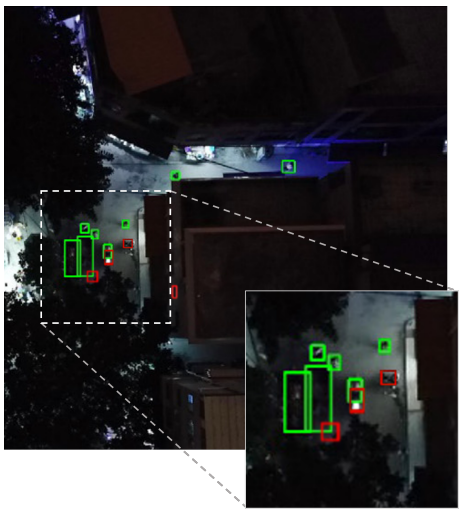


Figure 6. Sample of Error and Correct Detection in Low-Light Scenario. Red boxes indicate misidentified regions, while green boxes represent correctly detected areas.

Resource Constraints and Computational Overhead: The incorporation of fuzzy logic modules into the FSDN-DETR architecture, while beneficial for enhancing small object detection, also introduces increased computational overhead. This was particularly evident

during the pre-training phase on the COCO dataset, where the added complexity of the fuzzy logic modules required substantial computational resources for the deeper layers of the Transformer network. Limited GPU availability during the experiments further constrained our ability to perform comprehensive hyperparameter tuning and optimization, potentially limiting the overall performance of the model. Moreover, the increased model complexity resulted in longer training times and higher memory usage, which may restrict the applicability of FSDN-DETR in real-time or resource-constrained environments. Future work should focus on developing lightweight versions of the fuzzy logic modules to reduce computational overhead and improve the model's scalability across various deployment scenarios.

In summary, while FSDN-DETR demonstrates considerable improvements in small object detection and cross-domain learning, addressing these challenges is crucial to fully realize the model's potential and ensure its robust application in various real-world scenarios.

5.2. Future Work

To address the identified limitations and broaden the applicability of FSDN-DETR, future research will focus on developing lightweight fuzzy logic modules to improve computational efficiency and enhance suitability for real-time applications. Additionally, exploring hybrid learning paradigms that integrate unsupervised and semi-supervised techniques could further bolster robustness and generalization. Applying the model to other domains, such as medical imaging and remote sensing, would test its adaptability and broaden its impact. Finally, advancements in attention mechanisms, such as dynamic attention routing, may further refine feature aggregation and improve performance. These efforts aim to position FSDN-DETR as a versatile and efficient solution for diverse object detection challenges.

6. Conclusions

This paper introduces FSDN-DETR, a novel object detection framework that integrates fuzzy logic into the Transformer architecture, addressing critical challenges in small object detection and cross-domain transfer learning. The combination of the Fuzzy Adapter Transformer Encoder (FATE) and the Fuzzy Denoising Transformer Decoder (FDTD) results in a more robust model with enhanced noise tolerance and adaptability. In particular, the incorporation of fuzzy logic improves the model's ability to handle uncertainties and ambiguities in feature representations, thereby facilitating more accurate detection of occluded or small objects, particularly in noisy and complex environments. Moreover, a staged fine-tuning strategy allows for the integration of pre-trained knowledge with domain-specific adaptations, improving the efficiency of transfer learning on small, specialized datasets. Experimental results on COCO and AI-TOD-V2 demonstrate that the model achieves superior performance in detecting densely packed and small objects, outperforming state-of-the-art baselines. While the approach shows significant promise, it also presents challenges, such as increased computational costs and the complexity of optimizing fuzzy logic within the Transformer. These challenges will require further exploration in future work to enhance the efficiency and scalability of the model.

Author Contributions: Conceptualization, J.Z.; Methodology, X.Z.; Software, Y.Z.; Validation, D.Y.; Formal analysis, Y.Z.; Resources, Z.L.; Writing—original draft, Z.L. and J.Z.; Writing—review & editing, D.Y.; Supervision, X.Z. and W.D.; Project administration, M.W. and W.D. All authors have read and agreed to the published version of the manuscript.

Funding: The funding for this work was provided by the Shaanxi Provincial Housing and Urban-Rural Development Science and Technology Plan Project (2020-K09) and the Shaanxi Provincial Department of Education Collaborative Innovation Center Project (23JY038).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Li, Y.; Wang, H.; Dang, L.M.; Nguyen, T.N.; Han, D.; Lee, A.; Jang, I.; Moon, H. A deep learning-based hybrid framework for object detection and recognition in autonomous driving. *IEEE Access* **2020**, *8*, 194228–194239. [CrossRef]
2. Sandino, J.; Vanegas, F.; Maire, F.; Caccetta, P.; Sanderson, C.; Gonzalez, F. UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments. *Remote Sens.* **2020**, *12*, 3386. [CrossRef]
3. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3520–3529.
4. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2359–2367.
5. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
6. Yao, Z.; Ai, J.; Li, B.; Zhang, C. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv* **2021**, arXiv:2104.01318.
7. Li, F.; Zhang, H.; Liu, S.; Guo, J.; Ni, L.M.; Zhang, L. Dn-detr: Accelerate detr training by introducing query denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13619–13627.
8. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv* **2022**, arXiv:2203.03605.
9. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
10. Li, Y.J.; Dai, X.; Ma, C.Y.; Liu, Y.C.; Chen, K.; Wu, B.; He, Z.; Kitani, K.; Vajda, P. Cross-domain adaptive teacher for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7581–7590.
11. Gong, Y.; Luo, J.; Shao, H.; Li, Z. A transfer learning object detection model for defects detection in X-ray images of spacecraft composite structures. *Compos. Struct.* **2022**, *284*, 115136. [CrossRef]
12. Huang, Y.X.; Liu, H.I.; Shuai, H.H.; Cheng, W.H. Dq-detr: Detr with dynamic query for tiny object detection. *arXiv* **2024**, arXiv:2404.03507.
13. Tong, K.; Wu, Y. Deep learning-based detection from the perspective of small or tiny objects: A survey. *Image Vis. Comput.* **2022**, *123*, 104471. [CrossRef]
14. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
15. Yang, C.; Huang, Z.; Wang, N. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13668–13677.
16. Zeng, Y.; Zhang, P.; Zhang, J.; Lin, Z.; Lu, H. Towards high-resolution salient object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7234–7243.
17. Liu, Z.; Gao, G.; Sun, L.; Fang, Z. HRDNet: High-resolution detection network for small objects. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
18. Chen, W.; Luo, J.; Zhang, F.; Tian, Z. A review of object detection: Datasets, performance evaluation, architecture, applications and current trends. *Multimed. Tools Appl.* **2024**; *83*, 65603–65661. [CrossRef]
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
20. 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]
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]

22. 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.
23. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
24. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
25. Ross, T.Y.; Dollár, G. Focal loss for dense object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2980–2988.
26. Girshick, R. Fast r-cnn. *arXiv* **2015**, arXiv:1504.08083.
27. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; Volume 29.
28. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-head r-cnn: In defense of two-stage object detector. *arXiv* **2017**, arXiv:1711.07264.
29. Shi, Y.; Wang, N.; Guo, X. YOLOV: Making still image object detectors great at video object detection. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 2254–2262.
30. Tan, F.; Zhai, M.; Zhai, C. Foreign object detection in urban rail transit based on deep differentiation segmentation neural network. *Heliyon* **2024**, *10*, e37072. [CrossRef] [PubMed]
31. Li, Y.; Shang, J.; Yan, M.; Ding, B.; Zhong, J. Real-time early indoor fire detection and localization on embedded platforms with fully convolutional one-stage object detection. *Sustainability* **2023**, *15*, 1794. [CrossRef]
32. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
33. Farhadi, A.; Redmon, J. Yolov3: An incremental improvement. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1804, pp. 1–6.
34. Talpur, N.; Abdulkadir, S.J.; Alhussian, H.; Hasan, M.H.; Aziz, N.; Bamhdi, A. Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: A systematic survey. *Artif. Intell. Rev.* **2023**, *56*, 865–913. [CrossRef]
35. Chopade, H.A.; Narvekar, M. Hybrid auto text summarization using deep neural network and fuzzy logic system. In Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23–24 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 52–56.
36. Wang, X.; Zhang, X. Wireless network attack defense algorithm using deep neural network in internet of things environment. *Int. J. Wirel. Inf. Netw.* **2019**, *26*, 143–151. [CrossRef]
37. Aramuthakannan, S.; Ramya Devi, M.; Lokesh, S.; Manimegalai, R. Movie recommendation system via fuzzy decision making based dual deep neural networks. *J. Intell. Fuzzy Syst.* **2023**, *44*, 5481–5494. [CrossRef]
38. Jang, J.S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man, Cybern.* **1993**, *23*, 665–685. [CrossRef]
39. Ali, A.R.; Li, J.; Kanwal, S.; Yang, G.; Hussain, A.; Jane O'Shea, S. A novel fuzzy multilayer perceptron (F-MLP) for the detection of irregularity in skin lesion border using dermoscopic images. *Front. Med.* **2020**, *7*, 297. [CrossRef] [PubMed]
40. Liu, R.; Duan, S.; Xu, L.; Liu, L.; Li, J.; Zou, Y. A fuzzy transformer fusion network (FuzzyTransNet) for medical image segmentation: The case of rectal polyps and skin lesions. *Appl. Sci.* **2023**, *13*, 9121. [CrossRef]
41. Li, Q.; Wang, Y.; Zhang, Y.; Zuo, Z.; Chen, J.; Wang, W. Fuzzy-ViT: A Deep Neuro-Fuzzy System for Cross-Domain Transfer Learning from Large-scale General Data to Medical Image. *IEEE Trans. Fuzzy Syst.* **2024**. [CrossRef]
42. He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; Neubig, G. Towards a unified view of parameter-efficient transfer learning. *arXiv* **2021**, arXiv:2110.04366.
43. Mao, Y.; Mathias, L.; Hou, R.; Almahairi, A.; Ma, H.; Han, J.; Yih, W.t.; Khabsa, M. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv* **2021**, arXiv:2110.07577.
44. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Learning multiple visual domains with residual adapters. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–7 December 2017; Volume 30.
45. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Efficient parametrization of multi-domain deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8119–8127.
46. Karimi Mahabadi, R.; Henderson, J.; Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1022–1035.
47. Gu, Y.; Han, X.; Liu, Z.; Huang, M. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv* **2021**, arXiv:2109.04332.
48. Lester, B.; Al-Rfou, R.; Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv* **2021**, arXiv:2104.08691.
49. Li, X.L.; Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv* **2021**, arXiv:2101.00190.

50. Kim, K.; Laskin, M.; Mordatch, I.; Pathak, D. How to Adapt Your Large-Scale Vision-and-Language Model. 2021. Available online: https://openreview.net/forum?id=EhwEUb2ynIa (accessed on 15 March 2023).
51. Zhang, R.; Fang, R.; Zhang, W.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; Li, H. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv* **2021**, arXiv:2111.03930.
52. Chen, Z.; Duan, Y.; Wang, W.; He, J.; Lu, T.; Dai, J.; Qiao, Y. Vision transformer adapter for dense predictions. *arXiv* **2022**, arXiv:2205.08534.
53. Chen, T.; Zhu, L.; Ding, C.; Cao, R.; Wang, Y.; Li, Z.; Sun, L.; Mao, P.; Zang, Y. SAM Fails to Segment Anything?–SAM-Adapter: Adapting SAM in Underperformed Scenes: Camouflage, Shadow, Medical Image Segmentation, and More. *arXiv* **2023**, arXiv:2304.09148.
54. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
55. Wang, J.; Yang, W.; Guo, H.; Zhang, R.; Xia, G.S. Tiny object detection in aerial images. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3791–3798.
56. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
57. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 8–14 December 2019; Volume 32.

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.