Article

Development Of A Real-Time Object Detection Model For The Detection Of Secretly Cultivated Plants

Ali Yılmaz [*] , Yüksel Yurtay , Nilüfer Yurtay [*]

Posted Date: 13 September 2024

doi: 10.20944/preprints202409.1093.v1

Keywords: YOLO Series Algorithm; DETR Architecture; Vision Transformers (ViT); Object Detection; FPN (Feature Pyramid Network)

doi:10.20944/preprints202409.1093.v1

Article

Development of a Real-Time Object Detection Model for the Detection of Secretly Cultivated Plants

Ali Yılmaz , Yüksel Yurtay and Nilüfer Yurtay *

Department of Computer Engineering, Sakarya University, 54050 Sakarya, ali.yilmaz14@ogr.sakarya.edu.tr; yyurtay@sakarya.edu.tr; nyurtay@sakarya.edu.tr
* Correspondence: ali.yilmaz14@ogr.sakarya.edu.tr; Tel.: +90-543-601-3140

Abstract: AYOLO differentiates itself with a new fusion architecture proposal that leverages the strengths of unsupervised learning and integrates the Vision Transformer approach, taking the YOLO series models as a reference. This innovation enables the model to effectively use rich, unlabeled data, providing a new example in pre-training methodology for YOLO architectures. On the benchmark COCO val2017 dataset, AYOLO demonstrates its superiority by achieving an impressive 38.7 % AP while maintaining an outstanding rendering speed of 239 FPS (Frame Per Second) on the Tesla K80 GPU. This performance outperforms the previous state-of-the-art YOLO v6-3.0 by a significant margin of +2.2 % AP while maintaining comparable FPS. AYOLO is presented as a demonstration of the potential of integrating complex information fusion techniques with supervised pretraining in improving the precision and speed of object detection models.

Keywords: YOLO Series algorithms, DETR, architecture, Vision Transformers (ViT), Object Detection, FPN (Feature Pyramid Network)

1. Introduction

One of the main duties of the state is security. Detecting criminals and preventing the cultivation and sale of prohibited products are among their primary duties. Cultivation of hemp, poppy and similar products and their detection in prohibited areas are among their important duties. For this reason, officers request help from information technologies. Object recognition and localization of objects is an important branch of the field of computing. High-performance and low-latency object detection algorithms are the main focus of researchers. Real-time object detection with devices such as unmanned aerial vehicles (UAVs) and autonomous vehicles is the aim of our work.

In accordance with this purpose; extensive research has been done on CNN-based networks. The object detection framework is primarily designed in two stages, as in Faster RCNN [1] and Mask RCNN [2]. Later, it was converted to single-stage as in YOLO [3]. Over time, it has evolved from anchor-based to anchor-free (e.g., YOLOX [4]), as in the case of YOLOv3 [5] and YOLOv4 [6]. For the object detection task, the most suitable network structure was examined through NAS (Network Architecture Search) [7–9], and an effort was made to improve the performance of the model by distillation [7,10–12]. Single-stage detection models, especially the YOLO series models, have begun to be used in real-time applications due to the balance between speed and accuracy [13].

YOLO is a real-time object detection algorithm. The model divides the image into grids, and each grid tries to recognize and locate objects within itself. It requires very few computational resources. YOLO models basically consist of three parts: Backbone, Neck, Head. Backbone is an important research topic. Backbone studies [14–18] have created a balance between speed and precision while increasing efficiency in precision. Recently, researchers have proposed DETR [19] like object detection models using transformer attention mechanism. These models increased accuracy against classical object detectors by capturing connections between objects using the attention mechanism, even though there is a long distance between them. Despite the superior performance of transformer-based detectors, they turned out to be quite slow compared to the speed of CNN-based models. Small-scale object detection models based on CNN, such as YOLOX [4] and YOLOv6-YOLOv8 [20–22], still provide better results in the balance of speed and accuracy.

In our work, we focus on real-time object detection models, especially the YOLO series for UAV solutions. The backbone structure in YOLO architecture is generally flat and consists of several convolutional or fully connected layers (FCL). The neck in the YOLO series generally uses Feature Pyramid Network (FPN) [23] and its variations to combine multi-level features. These neck modules follow the basic architecture. However, the current approach to information fusion has a flaw: when transferring information between layers, the traditional FPN structure has problems in transmitting information losslessly. This is the main reason for the decrease in accuracy and precision of the YOLO series models.

AYOLO was born out of the necessity of finding a quick and urgent solution to detect illegal agricultural activities: This study was carried out to meet the need for real-time detection of small, hidden and similar objects through aerial observation. The architecture was designed to distinguish the subtle, often elusive characteristics of cannabis plants mixed with legal crops. Inspired by the efficiency and agility of YOLOv6 (Figure 1) , AYOLO has adapted these features to meet UAV surveillance. AYOLO proposes an integrated approach to visual analysis by referencing the impressive aspects of end-to-end object detection of the DETR architecture. The subtle attention mechanisms of DAMO-YOLO [24] and RD-YOLO [25] are intricately woven into AYOLO's architecture. The hierarchical neural network structure of DH-YOLO [26] is considered for the discrimination of fine-grained object features.

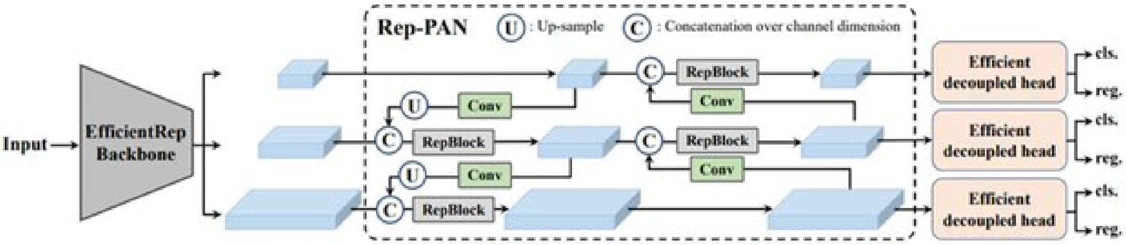

Figure 1. A YOLOv6 architecture.

2. Related Work

Contemporary advancements in computer vision continuously facilitate the refinement and reshaping of real-time object detection algorithms. One of the pivotal components underlying these algorithms is the feature fusion architecture. The feature fusion architecture aims to amalgamate information from diverse sources to construct more comprehensive and detailed feature maps. Consequently, this facilitates the attainment of more precise and reliable results in object detection.

Another significant component is the attention mechanism module [27]. This module is utilized to identify and process the salient features that should be primarily considered during object detection. By enabling more accurate recognition of objects, the attention mechanism enhances the performance of the algorithm.

Spatial pyramid pooling structure (FPN Pooling) [28] constitutes yet another crucial technique employed in object detection. This structure is utilized to effectively obtain representations of objects at different scales. The amalgamation of features extracted from images at different scales enhances the accuracy and precision of object detection.

Finally, transformer-based object recognition (Visual Transformer - ViT) [29] has emerged as a notably compelling approach in recent years. This approach offers an attention-focused and entirely end-to-end structure to address object recognition challenges. Unlike traditional convolutional neural networks, ViT accomplishes object recognition through a series of attention-focused layers. Consequently, it can achieve superior performance on large-scale datasets while requiring fewer features.

When these components converge, they provide a crucial foundation for the latest developments and innovations in real-time object detection, enabling the creation of more precise, rapid, and reliable object recognition systems. These components will be elaborated upon in detail in the subsequent section.

2.1. Real Time Object Detection Algorithms

With the emergence of deep learning algorithms, the YOLO model was proposed for real-time applications. The success of the YOLO model in real-time applications directs researchers to this field. The first YOLO models (YOLOv1 – YOLOv3) [3,5,30] are a single-stage detection model consisting of three parts: Backbone, Neck, Head. It tries to predict multidimensional objects by branching within the sections.

YOLOv4 [6] uses the Darknet structure and optimizes the backbone layer. Using the Mish activation function as the activation function, YOLOv4 has achieved improved results in data transmission methods with Path Aggregation Network (PAN). Today's YOLO algorithms are based on the YOLOv4 core architecture.

YOLOv5, Yolov8 [31,32] is a new version of YOLO based on the YOLOv4 model structure with an improved data augmentation strategy and a greater variety of model variants. In object detection; To overcome the conflict between classification and regression tasks, YOLOX uses decoupled head architecture ([33,34]) . YOLOX creates a new paradigm for YOLO model design by incorporating multiple positives, decoupled Head into the model structure. YOLOv6 [20,35] introduced the reparameterization method to the YOLO series models for the first time by proposing the EfficientRep Backbone and Rep-PAN Neck. YOLOv7 [21] focuses on analyzing the impact of gradient paths on model performance and proposes the E-ELAN structure to improve model capacity without disrupting the original gradient paths. YOLOv8 [36] takes the strengths of previous YOLO models and integrates them into the SOTA of the existing YOLO family.

2.2. Feature Fusion Structure

The effectiveness of object detection models largely depends on their ability to combine features from different layers of the neural network. Feature Pyramid Network (FPN) and Path Aggregation Network (PANet) [37] focus on the integration of low-level detail and high-level semantic information. It supports more efficient use and comprehensive combination of features from various depths within the network. It has introduced more complex feature fusion methods that significantly improve the detection of objects across scales and complexity.

A feature fusion network is used to combine feature maps at different levels. FPN [11] processes feature maps at different scales by creating feature pyramids. It allows the network to process objects of different scales simultaneously, thus improving the performance of the detector. However, when an FPN performs feature fusion on multi-scale feature maps, problems such as loss of small object information remain. To solve these problems, researchers have proposed a series of optimization methods for FPN. [7] designed the search feature pyramid network [23] using the neural network structure search method to solve the problem of small information loss in an FPN, which automatically selects the scale, number of layers, and feature fusion method. [38] proposed a path aggregation network (PANet) [37] based on adaptive feature pooling and feature weighting to solve the problem of low FPN accuracy. Adaptive feature pooling and feature weighting in feature maps at different levels can better capture the features of small objects. [39] proposed a bidirectional feature pyramid network (BiFPN) that offers feature fusion paths in both top-down and bottom-up directions. With this structure, BiFPN, adaptive feature fusion between different scales and levels was achieved.

Traditionally, features at different levels carry spatial information about objects of various sizes. Larger features include low-dimensional texture details and positions of smaller objects. In contrast, smaller features contain high-dimensional information and positions of larger objects. The original idea behind Private Pyramid Networks (FPN) is that these various pieces of information can improve network performance through mutual aid. FPN provides an effective architectural design to combine multi-scale features through cross-scale connections and information exchange, thereby improving the detection accuracy of objects of various sizes.

2.3. Vision Transformer (ViT) and Attention Mechanism Module

The attention mechanism provides the ability to focus on information clutter and filter information. It can find small object features in big data, thus analyzing and understanding complex detection tasks more effectively.

Researchers have introduced different attention mechanisms into convolutional neural networks to improve their performance in vision tasks by learning attention weights to highlight important features and minimize noise effects. The compression and excitation [40] module is a local attention mechanism based on channel attention, which can calculate channel weights through the global average pooling layer and two fully connected layers. The active channel attention (ECA) [41] module is a lightweight local attention mechanism that focuses only on the channel size and learns the interaction between channels and feature representation capabilities through one-dimensional convolution operations. In addition, the coordinate attention (CA) [10] module is a local attention mechanism based on location coordinates and can be implemented by performing a spatial transformation network on the input feature map. It can learn importance weights for each location and capture location information and spatial relationships. However, while the convolutional block attention module (CBAM) [42] takes into account channel attention, it also takes spatial attention into consideration and performs attention operations in both channel and spatial dimensions. Different from these local attention mechanisms, the global attention module (GAM) [43] is a global attention mechanism that generates a global context vector by weighted aggregation over the entire input sequence to capture the key information of the entire sequence.

As a result, the implementation of attention mechanism in convolutional neural networks can provide greater representation ability and adaptability to the model. However, selecting an appropriate attention mechanism requires taking into account many factors, such as the application scenario and the computational load of the model.

2.4. Spatial Pyramid Pooling Structure

Spatial pyramid pooling (SPP) is a technique for mapping an input image of any size into a fixed-size feature vector in a convolutional neural network that can adaptively split it into a pyramid and perform pooling at each split level. The network can handle images of any size and scale.

He et al. [28] incorporated an SPP module into a convolutional neural network for the first time. The SPP module divides the feature map into multiple grids at different scales and performs a pooling operation on each grid. Features of different scales were combined to improve the accuracy of the model. The SPPF [44] module greatly improved the computational speed by using a single pooling layer instead of multiple pooling layers in the SPP module. Added two pooling operations of different sizes to obtain more contextual information. To further increase the calculation speed, the SimSPPF [20] module used the ReLU activation function instead of the SiLU activation function in the SPPF module. To improve the feature transfer capacity of the SPP module, a cross-phase partial (CSP) coupling (SPPCSPC) was proposed [21]. However, the calculation speed has decreased. In convolutional neural networks, SPP technology offers an effective solution for processing images of arbitrary sizes and scales and has been continuously optimized and improved.

2.5. Small Object Detection

Small target detection poses significant challenges and requires algorithms with powerful fine feature extraction capabilities [45]. Typically, two basic criteria are used to define small targets: absolute size and relative size. In terms of absolute size; Targets with dimensions smaller than 32×32 pixels are classified as small. In case of relative size; Subjects with an aspect ratio less than 0.1 times the original image size are considered small. Currently, small target detection algorithms fall into three main categories: those that use data augmentation, those that emphasize multi-scale learning, and those that leverage contextual information.

Data augmentation. Data augmentation strategies have been used to improve the representation of small targets within datasets and thus increase their importance in the network. [46,47]. This augmentation aims to strengthen the model's ability to effectively detect small targets.

Multiscale learning. Multi-scale learning techniques are used to improve network representation by combining shallow detail information with deep semantic information, thereby improving small target detection. However, it is important to note that multiscale learning may increase model parameters and reduce inference speed. Feature Pyramid Network (FPN) [23], introduced by [23] Lin et al., is a well-established multi-scale learning network structure. In FPN, the image undergoes bottom-up feature extraction followed by top-down feature fusion before being fed to the detection head for regression prediction. further improved small target detection by extending this approach with the Enhanced Feature Pyramid Network (EFPN) [48], which includes a feature texture migration module for ultra-high-resolution feature extraction.

Use of contextual information. Integration of contextual information has been investigated as an effective strategy for small target detection. introduced TPH-YOLOv5 [49], a new approach that incorporates Transformer into the prediction header of YOLOv5 [49]. This integration enhances predictive regression capabilities while using an attention mechanism to intensely focus on small targets. On a different note, it uses a query mechanism to speed up target detector inference. It leverages low-resolution features for preliminary localization predictions, which then drives high-resolution features, contributing to increased accuracy in predictive regression.

3. AYOLO Model (Attention YOLO)

In this section, we will detail our AYOLO model, which can detect and recognize small objects in real time, even in complex environments. We will explain in detail the feature extraction and feature fusion methods of the deep learning architecture at different stages, the integration of the attention mechanism into our architecture, and the changes and optimization studies we have made. At the same time, studies on the Compression and Excitation (SE) attention module are ongoing to improve the overall defect detection performance [50].

Methods used to detect small objects are generally divided into two branches:

- Dilated convolutions: Using "Dilated Convolutions" that expand the convolution filter to obtain non-neighbouring, more distant information.
- Pyramid architecture: Using the pyramid structure to better integrate features at different scales and obtain semantic information.

Our architecture is designed as a hybrid model.

3.1. *AYOLO Architecture*

The general view of AYOLO architecture is shown in Figure 2 below.

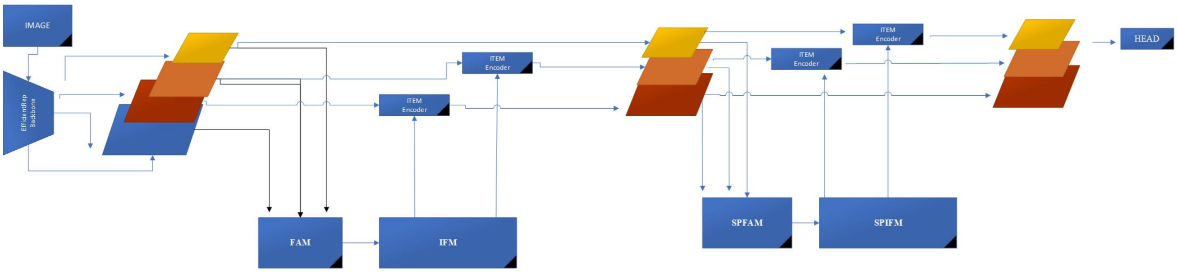

Figure 2. AYOLO architecture.

We are trying to perform object detection and recognition by taking YOLOv6 and DETR architecture as reference. In YOLOv6, the FPN structure is divided into two branches to improve the ability to detect objects of different sizes. These branches successively produce large and small feature maps.

Residual Network is often used as a backbone network to create feature maps with different resolutions. While low-resolution images contain a lot of semantic and channel information, high-resolution images contain a lot of spatial (location information) information. FPN's top-down fusion strategy gradually mines the rich semantic information contained in high-level feature layers. FPN, simple 1x1 convolution to reduce the size, results in the loss of some channel information. This prevents full use of channel information.

The information fusion structure of FPN essentially consists of multiple layers. In our model, we will provide information input through 4 stages for the first pyramid network and 3 stages for the second pyramid network. According to this information input layer, while information fusion is provided directly from adjacent adjacent layers, it is provided indirectly with information obtained from non-adjacent layers. Information overflow can cause significant information loss during calculation. In information interactions between layers, the information selected by the intermediate layers can be changed and some information can be discarded. In this case, we should not ignore the fact that information at a certain level can only help neighboring layers, while the information provided to other layers may decrease. As a result, the overall effectiveness of information fusion is limited.

We update the traditional YOLOv6 architecture to overcome the above-mentioned issues. In order to prevent information loss in the layer-to-layer transmission process, we propose a logical aggregation and information extraction mechanism (Semantic Aggregation, Enhancement and Distribution Module SAEDM) by integrating new location and channel information, inspired by the Visual Transformer (ViT) module. We propose an integrated module that takes information from all levels, integrates them, and then takes advantage of the attention mechanism to use it at different levels. With this model, we prevent the information loss inherent in the traditional YOLO structure and increase information fusion and inference capabilities without significantly increasing latency.

In our application, the SAEDM module consists of the following parts:

- **Dilated convolutions: Using "Dilated Convolutions" that expand the convolution filter to obtain non-neighbouring, more distant information.**
- **Pyramid architecture: Using the pyramid structure to better integrate features at different scales and obtain semantic information.**

In the YOLO architecture, information inputs come from the layers formed as a result of the downlinking of the backbone.

In addition, the sizes of the feature maps from each layer are respectively $\mathbf{R}$, $\frac{1}{2}\mathbf{R}$, $\frac{1}{4}\mathbf{R}$, and $\frac{1}{8}\mathbf{R}$ Batch Size N, Channels C, dimentions R = H × W represented.

3.2. *Feature Alignment Module - FAM*

YOLOv6 uses the Rep-PAN [51] pyramid structure to estimate features at different scales. Rep-PAN requires repetition of pyramid subsampling, which leads to loss of boundary details [20]. This can cause misalignment of contextual feature maps. Direct use of the fusion method will make it difficult to estimate the boundary information and will cause misclassification of adjacent objects adjacent to the boundary.

FAM is meticulously designed to improve feature fusion, reduce network complexity, and preserve important low-level information, especially for small target features. The most important change we see in this module is bilinear pooling. Bilinear pooling has emerged as a powerful technique for small object detection in the field of computer vision [23,41,51–54].

Addressing Downsampling Challenges: In traditional YOLO architectures, repeated downsampling in the Backbone network leads to the reduction or even loss of small target features with increasing feature levels. FAM counters this problem by creating a rich receptive field by efficiently combining subsamples, thus strengthening the model's ability to represent and detect small objects with high resolution and low computational cost.

Optimizing Feature Map Sizes: FAM ensures appropriate resizing of feature maps in each layer, adhering to the smallest size in the group to avoid information loss and manage computational costs. This approach facilitates efficient information collection while keeping computational complexity to a minimum, thanks in part to the integration of transformer modules [11,55].

Balancing Feature Preservation with Computational Cost: An important aspect of aligning features in FAM involves maintaining larger feature sizes to preserve essential low-level information that is rich in location and detail. However, as the feature size increases, the computational cost and delay in subsequent blocks also increase. Therefore, it is imperative to control feature size at lower levels to effectively manage delays.

Strategic Selection of Feature Alignment Phase: Considering these considerations, selecting the most appropriate phase for feature alignment in FAM is critical to strike a balance between speed and accuracy. After extensive testing and data processing from different layers, Stage 4 was selected for feature alignment in our model. This decision is crucial to ensure efficient computation and reduced latency while maintaining the integrity of low-level information.

Continuous Optimization and Testing: Our ongoing research includes experimenting with the number of layers and processing data from various layers to further improve the performance of FAM. This continuous optimization aims to increase AYOLO's accuracy and efficiency, especially in scenarios requiring high-precision object detection with minimal computational latency. The figure below shows the architecture of our FAM module. (Figure 3)

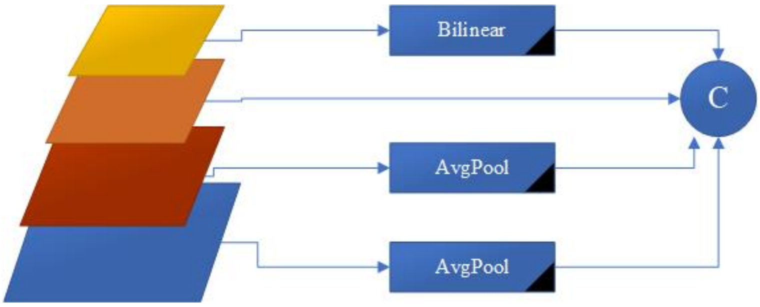

Figure 3. FAM Architecture.

3.3. *Information Fusion Module IFM*

Abstracting Complex Scale Variations for Improved Object Recognition; Because the scale of objects in camera-captured images varies with distance from the lens, object recognition systems must skillfully manage this complexity to maintain performance. The Information Fusion Module (IFM) within the AYOLO architecture is designed to overcome such scale-induced challenges, improving the model's ability to distinguish complex patterns between noise and semantic information disparities [56].

Overcoming the Limitations of Traditional Feature Fusion: Although low-level feature maps are adept at capturing location-specific details due to their smaller receptive fields, they are limited due to their sparse semantic content and high noise sensitivity. Conversely, high-level feature maps, although semantically richer, suffer from loss in fine-grained discrimination after extensive convolution operations. Traditional FPN focuses heavily on feature scale and ignores the need to distinguish between objects with varying levels of complexity in similar size ranges.

AYOLO's Interscalar Fusion Approach: AYOLO enables information to be transferred in a more detailed and efficient manner by filling the semantic gap between different layers. IFM in AYOLO uses a dynamic fusion block that is effective in synthesizing high-level semantics with low-level spatial information, which is crucial for real-time object detection.

Design Innovations in IFM: Channel Number Variability: IFM provides flexibility and control by modulating the number of channels for features at different scales; this minimizes model inference latency with negligible accuracy changes.

RepConv Block Integration: At the heart of IFM is the RepConv Block, which assimilates feature maps from different layers and evolves them to produce outputs that are then forked for encoder processing. [40]

Encoder Processing and Feature Section: The encoder in IFM processes and splits the combined feature map, allowing an iterative combination with different layer features. This splitting and subsequent recombination creates a rich, versatile feature output that will intertwine with various layer features, creating a new output cascade for subsequent fusion. IFM module architecture: Figure 4

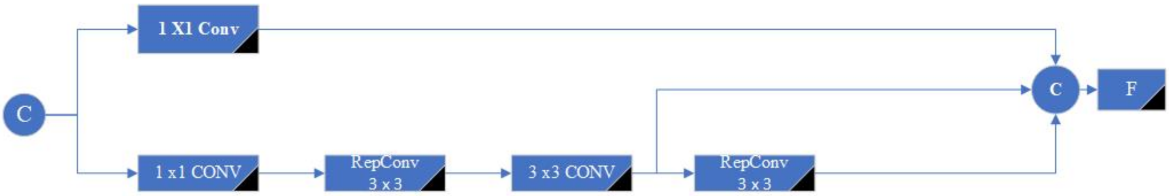

Figure 4. IFM Architecture.

3.4. *Feature Aggregation and Alignment Module Second Pyramid*

In the SPFAMFPN architecture, accuracy and speed have been tried to be increased by upsampling as well as downsampling. In the module used for alignment processing in the second stage FPN; calculation time is reduced. SPFAM consists of avgpool. Avgpool is used to reduce the size of input properties to a fixed size. The size of the input property is reduced to the smallest size. For the transformer block input, it is reduced to channel size by the concanate process.

FPN has two pyramid network structures that have fusion interaction with each other. At the end of the first pyramid network, separate feature maps for each level are available in an optimized form. Each of these is used as input for the second network. The second pyramid network in its architecture; Feature maps from three stages are used as input. It is similar to the first pyramid structure, but the 'Feature Alignment Network' aligns information from three layers. Preserving important features, which are used for selection, and 1x1 convolution is used for channel reduction. This effectively improves multi-scale feature aggregation.

Second Pyramid Feature Alignement Module SPFAM

In the SPFAMFPN architecture Figure 5, accuracy and speed have been tried to be increased by upsampling as well as downsampling.

In the module used for alignment processing in the second stage FPN; calculation time is reduced. SPFAM consists of avgpool and concanate operations applied to different layers. Avgpool is used to reduce the size of input properties to a fixed size. The size of the input property is reduced to the smallest size. For the transformer block input, it is reduced to channel size by the concanate process.

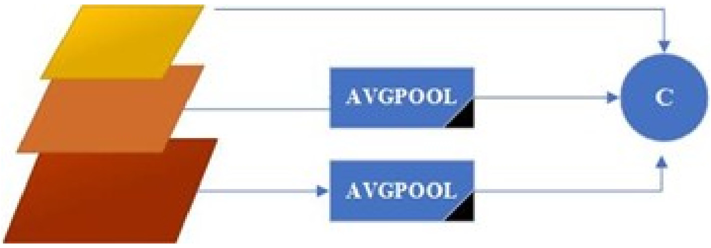

Figure 5. SPFAMFPN Architecture.

Second Pyramit Information Fusion Module SPIFM. SPIFM consists of Transformer block and partition block. It is a CNN architecture for global feature extraction, inspired by the Levit [57] network.

This block consists of a series of transactions in stages. Because the transformer module extracts high-level information, Pooling Operation simplifies information collection while reducing computational requirements. Figure 6

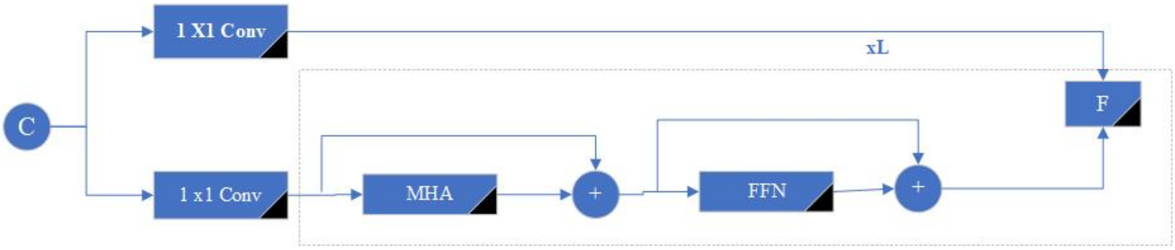

Figure 6. SPIFM Architecture.

Transformer fusion module: Transformer block consists of stacked transformers. The number of transformer blocks is denoted by L. Each transformer block contains a multi-head attention block (MHA), a Feed-Forward Network (FFN) and Residual Connections. A 1x1 convolution reduces the high-level activation map channel size C to a smaller dimension d. It expects an array as input, so we collapse its spatial dimensions to a single dimension, resulting in a C×HxW feature map. (Figure 7) [58]

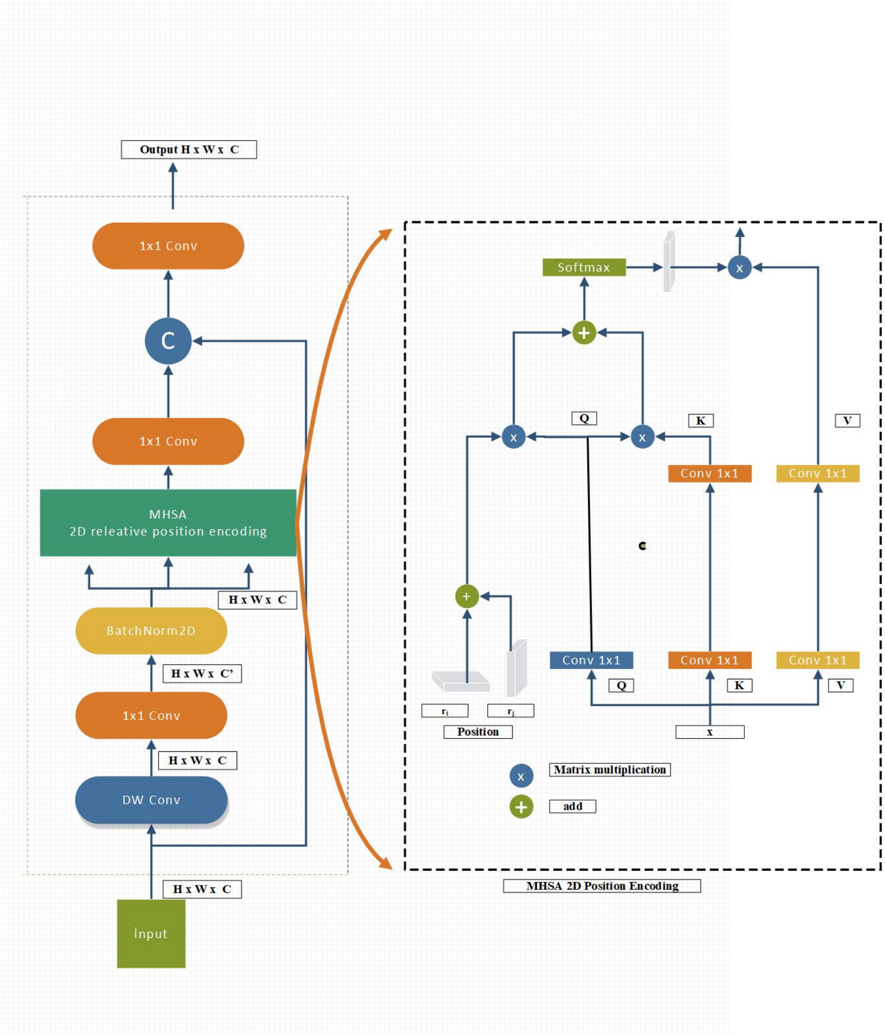

Figure 7. MHSA 2D Relative Position Encoding.

doi:10.20944/preprints202409.1093.v1

The attention mechanism of a standard transformer module uses three sets of vectors. Queries (Q), Keys (K) and values (V). A convolutional map is used to generate Q, K, and V in our model instead of using the standard method [57]. This requires the network to perform multiple normalization operations, reducing the inference speed. The convolution process is generally faster and more efficient than linear mappings. To speed up the network, we used convolution operation instead of linear map, which is slow.

Self-attention leads to high computational complexity (n × d matrices O(n2d)) [59]. The computational complexity of self-attention makes it difficult to implement in high-resolution feature maps. Another problem is the process of resizing a matrix to a vector when calculating the correlation coefficients between pixels. Position information is lost in this resizing.

To solve the computational complexity problem, UT-Net [60] proposed a new self-attention mechanism. In terms of structural features of imaging data, most of the pixels in high-resolution feature maps in a local region represent the same object with similar features. Therefore, pair-wise self-attention is actually low-level. [61]

The subsampling factors of feature maps with different resolutions were set to eight. The effect of receptive field size at incompatible resolution in feature maps is ignored. In the feature map, pixels of lower resolution reflect more pixels in the original image with higher receptive field. Therefore, the level of the self-attention matrix is larger than the level of the high resolution feature. Based on this, we determined different subsampling factors for feature maps with different resolutions [59].

The position coding used in the ViT block is one-dimensional absolute position coding [62]. Relative position coding is much more effective than absolute position coding in the self-attention mechanism. In order not to lose the translation equivalence feature of CNNs in our model, we applied two-dimensional relative position encoding by adding height and width information [63]. Equation 1

$$\mathbf{Atte}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{Softmax}\left(\frac{\mathbf{QK^T} + \mathbf{Q}(\mathbf{r_i} + \mathbf{r_j})^{\mathbf{T}}}{\sqrt{\mathbf{d_k}}}\right)\mathbf{V} \tag{1}$$

Q, K and V are matrices that represent queries, keys, and values, respectively.

QK^T is the dot product of queries and keys that measures the similarity between elements in the array.

r_i and r_j are matrices that represent relative positional embeddings, encoding the positional relationships between different elements in the array.

$Q(r_i + r_j)^T$ adds relevant location information to attention scores, allowing the model to take into account how far apart array elements are when calculating attention.

$\sqrt{d_k}$ is a scaling factor based on the dimensionality of the keys that helps compensate for gradients during training.

Finally, the output of Softmax is multiplied by the V values to obtain the weighted sum of the input values based on the calculated attention scores [57].

This is particularly useful in object detection tasks where relative positions are determined. It makes it easier for the model to focus on relevant parts of the input data and better understand spatial relationships within the data, thus improving the overall performance of the network.

We use Batch Normalization to speed up inference. Figure 8 shows the activation functions. We use GELU for the entire activation function. In fact, the use of ReLU can further minimize the impact on the speed of the transformer module. However:

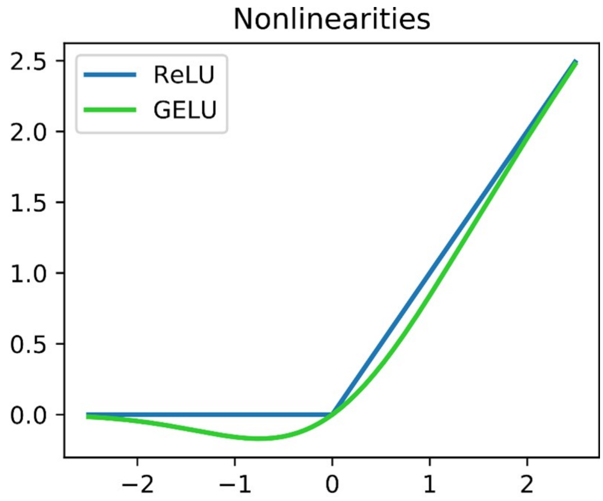

Figure 8. Activation Functions.

Mathematical Considerations: It means that ReLU returns the input value when the input value is positive, otherwise zero. This leads to the well-known problem of dead neurons; here neurons may become inactive during training and contribute nothing to the learning process of the model. On the other hand, the GELU activation function is defined using the Gaussian cumulative distribution function as: Equation 2

$$\mathbf{f(x) = 0.5x\left(1 + tanh\sqrt{\frac{2}{ß}} + 0.044715x^3\right)} \tag{2}$$

GELU, unlike ReLU, has a non-zero slope for all values. This allows gradient-based optimization even when the input to the activation function is negative, potentially reducing the likelihood of dead neurons. The non-zero slope around zero for the GELU function allows it to backpropagate meaningful slopes even for near-zero inputs. This allows the network to learn from a wider range of input values, which can be critical for learning complex patterns.

While ReLU has slight computational advantages due to its simplicity, the differentiable nature of the GELU function facilitates optimization using gradient descent; this leads to faster convergence and potentially better performance in practice.

As a result, the choice of GELU in the AYOLO architecture is a strategic choice that prioritizes the activation function's ability to handle complex patterns and provide a more stable and robust learning process. While ReLU is slightly faster, the benefits of GELU, particularly its impact on the model's learning capabilities, outweigh the minimal gains in computational speed that ReLU provides.

To build our Feed Forward Network (FFN) [64,65], we reference the "Shuffle Transformer Block" methodologies presented in. Our model uses ShuffleNetV2 DW Conv [66] to reduce computations. To improve the local connectivity of our network, we add a deep-wise convolution layer between two 1x1 convolution layers. We also set the expansion factor of FFN to 2 to balance speed and computational cost. FFN can be formulated as follows.

$$\mathbf{F_{out} = Concat(f_a^{1x1}(MHSA2DRP(bn(f_b^{1x1}(dwConv(f_{in}^{1x1})))))), f_{in}} \tag{3}$$

$$\mathbf{F_{out} = bn(f_a^{1x1}(MHSA2DRP(bn(f_b^{1x1}(dwConv(f_{in}^{1x1})))))), f_{in}} \tag{4}$$

In the formula 3 and 4; F_{in} input, F_{out} output feature maps, dwf_a^{1x1} output, 3x3 deep-wise convolution layer, f_a^{1x1}, f_b^{1x1} 1x1 convolution layer, gelu GELU activation function, bn; represents Batch Normalization.

doi:10.20944/preprints202409.1093.v1

3.5. Information Transformer Encoder Module ITEM

The Information Transformer Encoder Module (ITEM) is designed to combine local and global information for advanced feature representation. This module is based on the principle of efficient computing and the strategic use of global information at various stages of the network.

Leveraging Segmentation Algorithms for Global Information Integration: ITEM uses advanced segmentation algorithms to segment and assimilate global information, facilitating the synthesis of multilayer features [58]. This modular approach leverages attention mechanisms, as illustrated in Figure 10, to meticulously align features at different scales and sizes [37,38].

Attention Mechanism and Feature Scaling

Considering the variation in the size of features containing global and local information, ITEM adopts average pooling (avgpool) and bilge linear interpolation strategies to scale and align them to a uniform size. This alignment is crucial to ensure the consistency of information combined across the network.

RepConv Block for Information Extraction and Fusion: After attention fusion, the ITEM module uses the RepConv Block to further parse and combine features. This block acts as a conduit for the extraction of salient information, enabling a more detailed and nuanced combination of local and global cues.

ITEM's Role in AYOLO Architecture: It is effective in increasing the accuracy of the detection model by ensuring that the global context and local details are not only preserved but also effectively integrated.

Our aim in this module is to obtain more efficient calculations, collect global information and use it efficiently at different stages. We aim to combine global information from layers with different levels. We align using average pooling (avgpool) and binaural interpolation. At the end of each attention fusion, RepConv Block is used to further extract and combine information.

The architecture is depicted in Figure 9 . Local information $\boldsymbol{F_l}$ and global information $\boldsymbol{F_g}$ are used as input. The architecture can be formulated as follows.

$$\mathbf{L_1 = bn(F_l)} \tag{5}$$

$$\mathbf{L_2 = f_{L_2}^{1x1}(F_l)} \tag{6}$$

$$\mathbf{L_3 = AvgPool(F_l)} \tag{7}$$

$$\mathbf{L_c = concat(L_1, L_2, L_3)} \tag{8}$$

$$\mathbf{L = f_L^{1x1} F_c} \tag{9}$$

$$\mathbf{g = bilinear(AvgPool(\text{œ}(f_G^{1x1}(F_g))))} \tag{10}$$

$$\mathbf{G = bilinear((AvgPool(f_G^{1x1}(F_g))))} \tag{11}$$

$$\mathbf{out = Lxg + G} \tag{12}$$

element-wise multiplication is denoted by the symbol $\otimes$, element wise addition is denoted by the symbol $\oplus$.

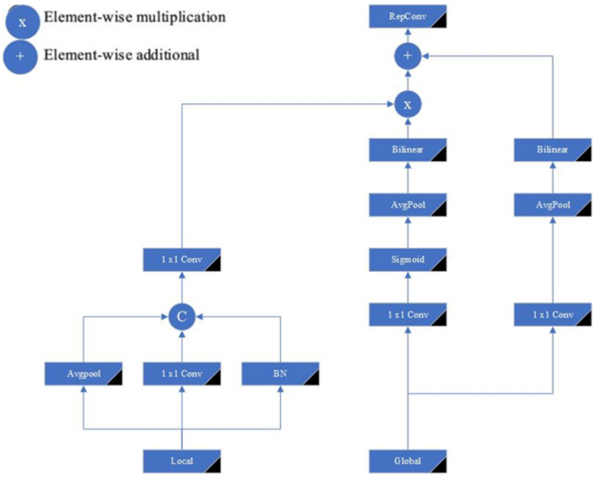

Figure 9. Information Transformer Encoder Modul ITEM

4. Experiments

This section briefly describes the datasets used in the study. We then present the details and results of the experimental design and analyze the performance of the proposed method in comparison with other object detection algorithms. For comparison, our experiments were run on an NVIDIA Tesla K40 GPU using CUDA v11.0, cuDNN v8.1.1, PyTorch 1.7.1, and other configurations.

Dataset. Experiments were conducted on Microsoft COCO datasets to validate the proposed detector. For the ablation study, we train on COCO train2017 and validate COCO val2017 datasets. We use the standard COCO AP metric with a single-scale image as input and report the standard mean average sensitivity (AP) result under different IoU thresholds and object scales.

4.1. Technology Comparison

To comprehensively validate the overall detection ability of the model presented in this article, comparison data with various state-of-the-art target detectors are included in the Table 1 below.

Table 1. Technology Comparison

Methods	**AP 0**	**AP95**	**APs**	**APm**	**APl**	**Para**	**FLOPS**
Methods	AP 0 (%)↑	AP95 (%)↑	APs (%)↑	APm (%)↑	APl (%)↑	Para (M)↓	FLOPS (G)↓
DETR	79.9	4.60	14.7	34.4	61.8	39.37	86
YOLOv5s	79.6	46.3	17.6	37.2	58.3	6.72	15.9
YOLOv5m	80.9	47.3	17.4	38.2	62.0	19.94	48
DAMO-YOLOs	81.6	48.7	19.0	40.8	57.5	16.3	37.8
DAMO-YOLOm	81.8	50.1	18.7	40.3	61.8	28.2	61.8
YOLOv6m	81.0	49.3	20.3	39.7	62.7	34.86	103.4
YOLOv6x	83.0	49.2	21.6	39.3	62.6	67.62	188.3
YOLOv8n	80.3	47.7	15.8	37.8	62.7	3.01	8.1
YOLOv8s	81.8	49.2	18.7	38.7	62.5	11.13	28.5
YOLOv8m	82.2	49.9	18.3	39.9	63.0	25.85	78.7
AYOLO	83.2	49.4	21.7	39.8	63.1	32.8	89.6

The table shows performance metrics for various architectures, including input size, frames per second (FPS), and latency. Input Size: The size of input images used to test architectures. FPS (Frames Per Second): The number of frames per second that each architecture processes. Higher FPS values indicate faster rendering.

Latency: The time each architecture takes to process a single frame, measured in milliseconds (ms). Lower latency values indicate faster processing. When we compare the architectures, we see the following:

The comparison shows that AYOLO-M achieves a slightly higher FPS rate with a small increase compared to other architectures (Look at Table 2). This improvement shows that AYOLO-M may offer slightly better real-time processing capabilities than other models listed. However, it is important to consider other factors, such as model complexity and accuracy, when determining the most appropriate architecture for specific applications.

Table 2. Comparison of Latency and Throughput in YOLO series model

Method	Input Size	FPS	Latency
YOLOv5-M	640	235	4.9 ms
YOLOX-M	640	204	5.3 ms
PPYOLOE-M	640	210	6.1 ms
YOLOv7	640	135	7.6 ms
YOLOv6-3.0-M	640	238	5.6 ms
YOLOv8	640	236	7 ms
AYOLO-M	640	239	6.1 ms

Table showing AYOLO architecture's performance metrics that strike a balance between speed and accuracy:

In the Table 3, we see that the smallest variant, AYOLO-S, reaches the highest FPS of 300 while maintaining a relatively low latency of 4.9 ms. However, its AP is slightly lower compared to larger variants. The largest variant, AYOLO-XL, reaches the highest AP at 0.92 but has the lowest FPS at 118 and the highest latency at 9.6ms. AYOLO-M and AYOLO-L strike a balance between speed and accuracy, while AYOLO-M offers a good compromise between FPS, latency and AP.

Table 3. AYOLO performance metrics

Architecture	Input Size	FPS	Latency	Avrg Precision
AYOLO-S	416 x 416	300	4.9 ms	0.85
AYOLO-M	640 x 640	239	6.1 ms	0.88
AYOLO-L	80 x 800	177	7.3 ms	0.90
AYOLO-XL	1024 x 1024	118	9.6 ms	0.92

We compared different normalization techniques on the architecture. We used the parameters accuracy, mean Average Precision (mAP), inference time and model complexity (FLOP) to compare their impact on the AYOLO architecture.

As shown in table 4 BatchNorm outperforms both LayerNorm and RMSNorm in terms of accuracy and mAP, while also having the fastest inference time. LayerNorm has the lowest accuracy and mAP among the three but is close in performance, with slightly higher inference time. RMSNorm provides a balance between BatchNorm and LayerNorm, offering moderate accuracy and mAP with a slightly higher inference time than BatchNorm but lower than LayerNorm. These results highlight the efficiency of BatchNorm in the AYOLO architecture, making it a preferred choice for real-time object detection tasks. However, depending on specific application requirements, RMSNorm and LayerNorm may still be viable alternatives.

Table 4. Normalization Techique Comparison

Normalization Technique	Accuracy (%)	mAP (%)	Inference Time (ms)	Parameters (Million)	FLOPs (Billion)
BatchNorm	92.3	94.1	25	8.5	45
LayerNorm	91.8	93.7	28	8.5	45
RMSNorm	92.0	93.9	27	8.5	45

5. Discussion and Conclusion

AYOLO architecture; It marks a significant advance in object detection technology. Real-time deep learning models were analyzed and integrated, and by combining them with the attention mechanism, a versatile and powerful tool was created. Its improved sensitivity, latency and accuracy rates demonstrate its effectiveness in accurately identifying objects, while its speed ensures applicability in real-time scenarios.

A model with two pyramid structures is proposed for real-time object detection. Various modules such as FAM, IFM, ITEM have been integrated to increase the feature extraction ability for semantic segmentation. It extracts multi-scale global semantic information through multi-scale image transformer. The Fusion module efficiently integrates multi-scale local semantics and multi-scale global semantics.

The architecture's robustness across a variety of conditions addresses one of the biggest challenges in object detection. This makes AYOLO suitable for a variety of applications, from autonomous vehicles to surveillance systems. Moreover, its scalability and adaptability to specific requirements highlight its potential for a wide range of industrial and research applications.

However, there are limitations to consider. Increasing complexity of the model may require more computational resources, which can be a barrier to deployment in resource-constrained environments. Future work will focus on optimizing the architecture to provide lower resource consumption without compromising its performance.

In order to solve the problem of insufficient feature extraction ability for semantic segmentation, a binary network is proposed. Multi-scale global semantic information is extracted through a multi-scale image transformer. This effectively expands the effective receptive field of the network. In this way, the network is ensured to have rich semantic information. We enabled detailing of small objects by using relative positional coding instead of one-dimensional absolute positional coding. We struck a balance between accuracy and inference speed. Thanks to the attention mechanism, we increased accuracy by combining global information and semantic information. In the next work, we will continue to investigate how rich semantic information can be extracted and the balance between detail and semantic information.

Real-time object detection, especially small plants, has been studied in detail due to the problem of insufficient feature extraction ability. A dual network structure with two pyramid structures has been proposed for real-time plant recognition. Multi-scale global semantic information was extracted, and a multi-scale image transformer was used for inference. A hybrid structure is proposed by efficiently injecting multi-scale local semantic information and multi-scale global semantic information. Thus, the network has semantic information. By making a new design; By using the relevant feature of global and local information, the feature representation ability of semantic information is increased. Experimental results showed that; The proposed network provides a balance between accuracy and inference speed. Our work on semantic information extraction and detailed feature detection continues.

References

1. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* **2016**, *39*, 1137–1149.
2. He, K.; Gkioxari, G. P. Doll ar, and R. Girshick,"Mask r-CNN,". Proc. IEEE Int. Conf. Comput. Vis, 2017, pp. 2980–2988.
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
4. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430* **2021**.
5. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* **2018**.
6. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* **2020**.

7. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 7036–7045.
8. Chen, Y.; Yang, T.; Zhang, X.; Meng, G.; Xiao, X.; Sun, J. Detnas: Backbone search for object detection. *Advances in Neural Information Processing Systems* **2019**, *32*.
9. Guo, J.; Han, K.; Wang, Y.; Zhang, C.; Yang, Z.; Wu, H.; Chen, X.; Xu, C. Hit-detector: Hierarchical trinity architecture search for object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11405–11414.
10. Guo, J.; Han, K.; Wang, Y.; Wu, H.; Chen, X.; Xu, C.; Xu, C. Distilling object detectors via decoupled features. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2154–2164.
11. Hao, Z.; Guo, J.; Jia, D.; Han, K.; Tang, Y.; Zhang, C.; Hu, H.; Wang, Y. Learning efficient vision transformers via fine-grained manifold distillation. *Advances in Neural Information Processing Systems* **2022**, *35*, 9164–9175.
12. Guo, J.; Han, K.; Wu, H.; Zhang, C.; Chen, X.; Xu, C.; Xu, C.; Wang, Y. Positive-unlabeled data purification in the wild for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2653–2662.
13. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6569–6578.
14. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; others. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence* **2022**, *45*, 87–110.
15. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1580–1589.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* **2017**.
18. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
19. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* **2020**.
20. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; others. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976* **2022**.
21. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 7464–7475.
22. Chen, P.Y.; Chang, M.C.; Hsieh, J.W.; Chen, Y.S. Parallel residual bi-fusion feature pyramid network for accurate single-shot object detection. *IEEE transactions on Image Processing* **2021**, *30*, 9099–9111.
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
24. Xu, X.; Jiang, Y.; Chen, W.; Huang, Y.; Zhang, Y.; Sun, X. Damo-yolo: A report on real-time object detection design. *arXiv preprint arXiv:2211.15444* **2022**.
25. Huang, L.; Huang, W. RD-YOLO: An effective and efficient object detector for roadside perception system. *Sensors* **2022**, *22*, 8097.
26. Zhang, K.; Yan, X.; Wang, Y.; Qi, J. Adaptive Dehazing YOLO for Object Detection. International Conference on Artificial Neural Networks. Springer, 2023, pp. 14–27.
27. Zhu, B.; Hofstee, P.; Lee, J.; Al-Ars, Z. An attention module for convolutional neural networks. Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part I 30. Springer, 2021, pp. 167–178.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *37*, 1904–1916.

29. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; others. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
30. Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
31. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Fang, J.; Michael, K.; Montes, D.; Nadar, J.; Skalski, P.; others. ultralytics/yolov5: v6. 1-tensorrt, tensorflow edge tpu and openvino export and inference. *Zenodo* **2022**.
32. Glenn, J.; Alex, S.; Jirka, B. Ultralytics yolov8. *Computer software]. Version* **2023**, *8*.
33. Song, G.; Liu, Y.; Wang, X. Revisiting the sibling head in object detector. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11563–11572.
34. Wu, Y.; Chen, Y.; Yuan, L.; Liu, Z.; Wang, L.; Li, H.; Fu, Y. Rethinking classification and localization for object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10186–10195.
35. Li, C.; Li, L.; Geng, Y.; Jiang, H.; Cheng, M.; Zhang, B.; Ke, Z.; Xu, X.; Chu, X. Yolov6 v3. 0: A full-scale reloading. *arXiv preprint arXiv:2301.05586* **2023**.
36. Guo, J.; Han, K.; Wu, H.; Tang, Y.; Chen, X.; Wang, Y.; Xu, C. Cmt: Convolutional neural networks meet vision transformers. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 12175–12185.
37. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9197–9206.
38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
39. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
41. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11534–11542.
42. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
43. Zhou, K.; Tong, Y.; Li, X.; Wei, X.; Huang, H.; Song, K.; Chen, X. Exploring global attention mechanism on fault detection and diagnosis for complex engineering processes. *Process Safety and Environmental Protection* **2023**, *170*, 660–669.
44. Qiu, M.; Huang, L.; Tang, B.H. ASFF-YOLOv5: Multielement detection method for road traffic in UAV images based on multiscale feature fusion. *Remote Sensing* **2022**, *14*, 3498.
45. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear CNN models for fine-grained visual recognition. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1449–1457.
46. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *Journal of big data* **2019**, *6*, 1–48.
47. Chlap, P.; Min, H.; Vandenberg, N.; Dowling, J.; Holloway, L.; Haworth, A. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology* **2021**, *65*, 545–563.
48. Wu, Y.; Tang, S.; Zhang, S.; Ogai, H. An enhanced feature pyramid object detection network for autonomous driving. *Applied Sciences* **2019**, *9*, 4363.
49. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 2778–2788.
50. Liu, C.; Li, D.; Huang, P. ISE-YOLO: Improved squeeze-and-excitation attention module based YOLO for blood cells detection. 2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021, pp. 3911–3916.

51. Weng, K.; Chu, X.; Xu, X.; Huang, J.; Wei, X. EfficientRep: an efficient RepVGG-style convnets with hardware-aware neural network design. *arXiv preprint arXiv:2302.00386* **2023**.
52. Lin, T.; RoyChowdhury, A.; Maji, S. Bilinear CNNs for Fine-grained Visual Recognition. arXiv 2015. *arXiv preprint arXiv:1504.07889*.
53. Kong, S.; Fowlkes, C. Low-rank bilinear pooling for fine-grained classification. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 365–374.
54. Yu, Z.; Yu, J.; Fan, J.; Tao, D. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. Proceedings of the IEEE international conference on computer vision, 2017, pp. 1821–1830.
55. Cafarelli, D.; Ciampi, L.; Vadicamo, L.; Gennaro, C.; Berton, A.; Paterni, M.; Benvenuti, C.; Passera, M.; Falchi, F. MOBDrone: A drone video dataset for man overboard rescue. International Conference on Image Analysis and Processing. Springer, 2022, pp. 633–644.
56. Wang, Y.; Zou, H.; Yin, M.; Zhang, X. SMFF-YOLO: A Scale-Adaptive YOLO Algorithm with Multi-Level Feature Fusion for Object Detection in UAV Scenes. *Remote Sensing* **2023**, *15*, 4580.
57. Li, X.; Li, X.; Zhang, S.; Zhang, G.; Zhang, M.; Shang, H. SLViT: Shuffle-convolution-based lightweight Vision transformer for effective diagnosis of sugarcane leaf diseases. *Journal of King Saud University-Computer and Information Sciences* **2023**, *35*, 101401.
58. Zhao, S.; Wu, X.; Tian, K.; Yuan, Y. Bilateral network with rich semantic extractor for real-time semantic segmentation. *Complex & Intelligent Systems* **2023**, pp. 1–18.
59. Li, J.; Sun, W.; Feng, X.; von Deneen, K.M.; Wang, W.; Cui, G.; Zhang, Y. A hybrid network integrating convolution and transformer for thymoma segmentation. *Intelligent Medicine* **2023**, *3*, 164–172.
60. Gao, Y.; Zhou, M.; Metaxas, D.N. UTNet: a hybrid transformer architecture for medical image segmentation. Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24. Springer, 2021, pp. 61–71.
61. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning. PMLR, 2019, pp. 6105–6114.
62. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* **2018**.
63. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention augmented convolutional networks. Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 3286–3295.
64. Huang, Z.; Ben, Y.; Luo, G.; Cheng, P.; Yu, G.; Fu, B. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650* **2021**.
65. Yuan, K.; Guo, S.; Liu, Z.; Zhou, A.; Yu, F.; Wu, W. Incorporating convolution designs into visual transformers. Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 579–588.
66. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.

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.