Towards Efficient Object Detection With Vision Transformers: Training and Inference Optimization Strategies

Prachi Natu[1], Shachi Natu[2]

[1] Associate Professor (Data Science Department, UMIT, SNDT University, Mumbai, India)

[2] Associate Professor (Department of Information Technology, TSEC, University of Mumbai, India)

*Corresponding Author

Abstract

In computer vision, object detection has its own significance having wide area real world applications from video surveillance to medical imaging field. After the proposal of Alexnet, convolutional neural network has become the backbone architecture of object detection and till today various popular YOLO architecture ruled the object detection use cases. But there are many challenges with respect to speed and accuracy of object detection using convolutional neural network. After the emergence of transformers in computer vision field, vision transformers have become main contributor in classification and object detection. These models performed better than convolutional neural network, but they pose significant training and inference challenges like heavy computation cost due to global attention, very high memory usage and increased latency. Also, huge amount of input data is needed to train such models. Looking at current trend of using AI on low energy devices deployment of these models for real time applications need reduced latency, less computational cost, low memory usage and extraction of both global and local features for more accurate object detection even in case of small object size. This paper exclusively focuses on how these challenges arise and how it can be resolved for real time deployment. It reviews optimization techniques used in vision transformers for object detection to tackle these challenges, majorly focusing on training time optimization and inference time improvements. By systematically analyzing these approaches we try to explore limitations in these models and possible future trends which may help to seek balance of accuracy with efficiency in object detection using vision transformers.

Keywords: Object Detection; Vision transformers; Latency; Self attention; Optimization

1. Introduction

In computer vision, task of object detection means to detect and localize the objects in an image. Object detection algorithms have now developed starting from traditional methods to CNN based methods and now it is highly influenced by vision transformers as detectors. Traditional object detection methods involve techniques like Histogram of Gradients (HoG) [1], Viola Jones detector [2], Deformable part-based model (DPM) etc. Viola Jone proposed face detection in real time without any constraints in 2001. This used simple sliding window method for detection. Further a new feature descriptor called histogram of gradients (HOG) was presented in 2005 by N. Dalal and B Triggs. It was scale invariant and used in many computer vision applications. DPM was proposed by Felzenszwalb [3] in 2008 as an extension of HOG detector. All these traditional object detectors were using hand crafted features for object detection.

Later, as deep neural networks emerged as automatic feature extractors and CNN became prominent for extracting image features, object detection evolved with high speed and many different models came up showing better performances in this field. Two types of object detectors were proposed using CNN based models by researchers: two stage detectors and single stage detectors. CNN based object detectors were at the heart of the object detection till transformers were introduced in computer vision field. Initially transformers were mainly used for natural language processing applications. But when they were used directly for image processing applications without any alterations, computer vision field has reached to a new level where CNNs are almost replaced by vision transformers with better performance.

The invention of transformer for natural language processing (NLP), started new era in deep neural networks. Similar to NLP domain, transformers were also introduced in computer vision domain which brought radical changes in the domain. Transformers in computer vision domain are called vision transformers and they played significant role in classification, segmentation and object detection applications.

This paper provides a detail review of CNN based methods which were popular and giving high performance before the use of vision transformers in this application domain. Further it reviews the detail development in object detection using vision transformers with different models so far. Then it discusses various training and inference time optimization strategies including how to use these models in energy constrained devices. In accordance to this, the paper is organized as follows: Section II discusses deep neural network based (CNN based) object detection techniques that include two stage and single stage detectors. Section III reviews the literature and discusses foundational architectures of object detectors using vision transformer and its variants obtained by modifications in transformer architecture along with their advantages and limitations. Section IV discusses

training time optimization strategies with their advantages and limitations. Section V describes inference time optimizations with their benefits and challenges. These strategies which makes deployment of these models practically possible for real world applications in latency sensitive and resource limiting environment such as mobile devices, embedded systems etc. In section VI summarizes datasets used in literature and their characteristics. Section VII focuses on limitations and challenges in existing methods followed by section VIII which comments on emerging trends and future directions. Finally, paper concludes with the remarks based on study and analysis of literature.

2. Object Detection Using CNN Models

CNN based models are mainly divided into two types: two stage detectors and single stage detectors. In two stage detectors, detection is done in two stages: first stage consists of proposing the candidate regions. Hence it is called as region proposal stage and second stage is classification stage.

A. Two stage detectors

Grishik et al. [4] proposed region based CNN called RCNN. It proposes category independent candidate regions where objects can be present by proposing bounding box around the object using selective search method [5]. Each proposal is passed through pretrained CNN to extract fixed length features. Image is converted to 227x227 size. From each region a feature vector of dimension 4096 was extracted. Five layered CNN with two fully connected layers was used for this purpose. Linear support vector machine classifier is used to detect if the object is present in each region and to recognize object categories. Authors had tested RCNN on PASCAL VOC dataset [6,7] and ILSVRC2013 [8] dataset and observed a significant improvement in mean average precision (mAP) over recently proposed methods at that time like OverFeat [9]. RCNN gave accurate object detection with different image sizes and complex background. But computationally it is very intensive as it involves region proposal, apply CNN to each proposal to extract features and then pass these features through classifier. Detection speed of RCNN was very slow at inference time as it was proposing over 2000 overlapped redundant bounding boxes from one image.

Improvement in RCNN called Fast RCNN was proposed in 2015 by same author Grishik [10] which gave better accuracy in object detection compared to R-CNN and solved some drawbacks of RCNN too. In Fast RCNN, image is first passed through CNN to generate a feature map. Then Region of Interest (RoI) pooling layer is used to obtain fixed size features for each proposal and it is passed through fully connected layer. Fast RCNN integrates classification and bounding box regression into a unified network with a single multi task loss function. Hence it provides end to end training. It is substantially faster than RCNN as it runs CNN once per image and generates shared feature map whereas RCNN, each proposal is processed individually through CNN resulting thousands of CNN forward passes per image. Further, RoI pooling layer eliminates the need to resize and warp image patches like in RCNN. In Fast RCNN, selective search is used for region proposal which is considerably slow. Classification and bounding box regression are done after region proposal, which does not optimize the performance for end-to-end functioning. Due to slow region proposal, Fast RCNN is not suitable for real time applications. Ren et al. [11] have proposed a solution to the region proposal bottleneck in Fast RCNN. Detection system provided by them is made up of two modules: first is deep fully convolutional network that proposes regions and another is Fast RCNN detector that uses proposed regions. Here region proposal network generates learnable region proposals directly from feature maps obtained after convolution. Further ROI pooling layer similar to fast RCNN is used for detection. This method was translation invariant and reduced the model size.

B. Single stage detectors

Lin et al. presented RetinaNet, described as a novel one-stage object detector. They stated that category imbalance was the reason for lower accuracy of single-stage detectors as compared to two-stage detectors. This problem of class imbalance in one-stage detectors involves difficulties with hard-to-classify examples [12]. The core contribution in retinanet was the Focal Loss, a modified cross entropy loss that dynamically scaled down the loss contribution of well-classified examples using a modulating factor $(1 - pt)^{\gamma}$. This focused the training on hard, misclassified examples, effectively preventing the easy negatives from overwhelming the training process. RetinaNet utilized a Feature Pyramid Network (FPN) backbone and employs anchors across multiple scales and aspect ratios. It attached separate, simple convolutional subnets for classification and bounding box regression to each FPN level. When trained with the proposed Focal Loss, RetinaNet had demonstrated the ability to match or exceed the accuracy of state-of-the-art two-stage detectors while maintaining the speed advantages of a one-stage system.

A first end to end single stage object detector was proposed by Redmon et al. [13] where instead of using sliding window, it sees entire image. It is called as ‘You Only Look Once’ (YOLO), a new method for object detection that treated the task as a single regression problem. Instead of using complex pipelines involving region proposals and classifiers, YOLO employed a single convolutional neural network that directly predicted bounding box coordinates and class probabilities from the full image in one evaluation. This unified architecture was extremely fast, capable of processing images at real-time speeds of 45 frames per second for the base model and up to 155 frames per second for Fast YOLO, significantly exceeding the speed and often the accuracy of previous real-time detection systems. A key advantage was that YOLO reasoned globally about the image during prediction, which helped reduce the number of false positive detections on background regions compared to methods that only examine local patches. Furthermore, YOLO learned generalizable representations, allowing it to perform effectively when applied to different domains, such as artwork. The system works by dividing the input image into an $S \times S$ grid, where each cell was responsible for detecting

objects whose center falls within it. Each cell predicted multiple bounding boxes, confidence scores (indicating the likelihood of an object and the box's accuracy), and conditional class probabilities. While innovative and fast, the initial YOLO version had limitations, particularly in precisely localizing small or grouped objects and generalizing to unusual aspect ratios.

Subsequent versions of YOLO introduced significant improvements. YOLOv2 [14] added batch normalization, high-resolution training, and, notably, the use of anchor boxes derived from k-means clustering to improve bounding box prediction. It also introduced finer-grained features via a passthrough layer and multi-scale training. Backbone architecture used by YOLOv2 is called as Darknet-19 which is inspired from network in network [15] as that of in YOLOv1. YOLOv3 [16,17]incorporated a larger backbone with residual connections, used logistic regression for objectness prediction, employed binary cross-entropy for multi-label classification, added Spatial Pyramid Pooling (SPP) [18], and introduced multi-scale predictions at three different grid sizes to improve small object detection.

From YOLOv4 [19]onwards, the architecture was often described in three parts: the backbone, neck and head. The backbone is tasked with extracting features from the input image. The neck serves as a vital link situated between the backbone and the detection head. Its function involves refining and processing the features derived from the backbone, often enhancing spatial contextual awareness and enabling efficient multi-scale feature fusion to better handle objects of varying sizes. Lastly, the detection head performs the final tasks of object classification and localization. Across the development of algorithms like the YOLO series, various techniques have been incorporated to boost detection performance. These include architectural enhancements and training strategies. For example, YOLOv4 and YOLOv5 optimized their architecture using components like CSP-Net and PAN. Specific training techniques, such as Mosaic data augmentation, were also utilized. Some methods are primarily focused on improving the training process, potentially increasing training costs. Other advancements involve modifications to the network structure that aim to improve accuracy, which can sometimes affect inference speed. The goal of these techniques is the continuous improvement of detection accuracy and efficiency [20].

A notable shift occurred with YOLOX, which moved back to an anchor-free architecture, simplified training, used multiple positives, and introduced a decoupled head for separate classification and regression tasks. This anchor-free trend has continued in later versions. YOLOv6 [21] also used a decoupled head and introduced an efficient backbone (EfficientRep) and neck (PAN with RepBlocks/CSPStackRep), along with advanced quantization techniques. YOLOv7 [22] focused on architectural efficiency with E-ELAN and model scaling strategies, incorporating ideas like planned re-parameterized convolutions and implicit knowledge from YOLOR [23]. DAMO-YOLO leveraged Neural Architecture Search (NAS) for an efficient design. YOLOv8, [24] developed by Ultralytics, also adopted an anchor-free decoupled head architecture and improved loss functions. The PP-YOLO series (PP-YOLO [25], PP-YOLOv2 [26], PP-YOLOE [27]) also evolved in parallel, incorporating various optimization tricks and eventually moving to an anchor-free design. YOLOv9 [28] was introduced to resolve the problem of information loss in deep neural networks during the feedforward process. This loss resulted into unreliable gradients during training. The paper proposed two key concepts: Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN). PGI utilized an auxiliary reversible branch to generate reliable gradients for the main network during training without adding inference cost. This helped the main branch learn important features more effectively. GELAN is a lightweight network architecture designed for efficiency in parameters, computation, and speed. It generalized existing architectures like ELAN. Combining PGI and GELAN, YOLOv9 achieved state-of-the-art performance on the MS COCO dataset, outperforming previous methods, particularly when trained from scratch, often with fewer parameters and computations. PGI specifically helps improve accuracy for both shallow and deep models.

In 2024, YOLOv10 [29] introduced object detection system which was end-to-end and real time. It improved upon previous YOLO architectures. Earlier YOLOs were using non-maximum suppression (NMS). Though it was used for post-processing, it was limiting end-to-end deployment and increasing inference latency. YOLOv10 addressed this by presenting consistent dual assignments for NMS-free training. This method employed both a one-to-many assignment branch for rich training supervision and a one-to-one head used solely for inference, eliminating the need for NMS without adding inference overhead. Furthermore, the paper detailed a holistic efficiency-accuracy driven model design strategy. This involves optimizing components like the classification head, down sampling layers, and block design, while incorporating features like large-kernel convolutions and partial self-attention for enhanced capability with minimal cost. Rigorous experimentation indicated that YOLOv10 achieved better performance and efficiency with fewer parameters and lower latency compared to prior leading detectors.

YOLOv11, proposed in 2024 [30], is built upon the foundation established by previous versions, including YOLOv8, YOLOv9, and YOLOv10. YOLOv11 introduced significant enhancements to architecture and training methodologies, aiming to push boundaries in accuracy, speed, and efficiency. Its architecture introduced C3k2 block, the retention of the SPPF block, and the addition of the C2PSA component. The C3k2 block replaces the C2f block from previous versions. It is described as a more computationally efficient implementation of the Cross Stage Partial (CSP) Bottleneck, using two smaller convolutions instead of one large one. The "k2" signifies a smaller kernel size, contributing to faster processing while aiming to maintain performance. C3k2 blocks are utilized in both the backbone and head sections of the network to efficiently process and refine multi-scale features. The SPPF block is retained from prior versions. The C2PSA block is a new addition placed after the SPPF block. It improved spatial attention in feature maps and model focused more effectively on important regions. It resulted in better detection accuracy for objects with varying sizes and positions. This specific attention mechanism distinguishes YOLOv11 from YOLOv8. The head also includes CBS layers for feature refinement, data normalization (Batch Normalization), and non-linearity (SiLU

activation). YOLOv11m variant achieved superior mAP while using 22% fewer parameters than YOLOv8m, showcasing improved efficiency. YOLOv11 variants consistently outperformed previous models (YOLOv5 through YOLOv10) in mAP with faster inference rates.

YOLOv12 is presented as the latest evolution in the YOLO series, following YOLOv11, and was introduced in April 2025 [31, 32]. Its core focus is the successful integration of attention mechanisms while preserving the real-time efficiency characteristic of the YOLO framework. This addresses the challenge where attention-based models, despite strong modelling capabilities, have traditionally lagged behind CNNs in speed. YOLOv12 aims to bridge this gap. YOLOv12 introduced following key architectural innovations:

- Area Attention (A2): A novel, efficient mechanism that partitions spatial regions to significantly reduce the computational complexity of self-attention while retaining a large receptive field. Unlike traditional methods, A2 achieves segmentation via a simple reshape operation. This reduces computational cost and parameter/memory usage. A2 enhances spatial attention, enabling the model to concentrate on key image regions, potentially leading to more accurate detection, especially for smaller or partially occluded objects.
- Residual Efficient Layer Aggregation Networks (R-ELAN): An enhancement over traditional ELAN, designed to stabilize training, particularly in larger-scale models. It ensured better gradient flow and optimization by incorporating block-level residual design with scaling and also used feature aggregation.

Beyond A2 and R-ELAN, YOLOv12 includes following refinements:

- Integration of Flash Attention: This optimizes GPU memory access, reducing memory overhead and enabling higher throughput and lower memory consumption, which is crucial for real-time performance with attention mechanisms.
- Removal of positional encoding to simplify computations.
- Computational balance of attention and feed forward networks is obtained by adjusting MLP ratio enhancing performance.
- Streamlined backbone with fewer stacked blocks in later stages for improved convergence and inference efficiency.
- Leveraging convolution operators for computational efficiency.

YOLOv12 is benchmarked on the MS COCO 2017 dataset and is shown to perform with good accuracy and competitive speed. However, challenges remain regarding memory constraints and inference latency on low-power edge hardware. The trade-off between speed and accuracy remains a core theme, with different model scales offered to suit diverse application needs and hardware constraints. The future of YOLO is envisioned to include incorporating the latest techniques, evolving benchmarks, expanding applications into new domains (like tracking and multi-modal tasks), and further adaptability to diverse hardware platforms.

Table 1 shows comparison of various CNN based object detection models. In this comparison we have considered above discussed two stage models, YOLOv1 which is very primary single stage detection model, YOLOv5, an intermediate YOLO version and YOLOv8 to YOLOv12 altogether as these latest versions are more prominently used for object detections than earlier YOLO versions.

These two stage and single stage detectors were successful due to their convolutional backbone and region proposal theme but after the emergence of transformers for natural language applications, researchers started exploring it for vision applications like segmentation, classification and object detection too. This shift caused a rise of new family that leverage self-attention mechanism to model global relationship between image regions more efficiently than CNN. It gave rise to new era of detectors called vision transformer-based object detectors. These detectors directly predict object locations instead of designing anchor boxes. This shifted the paradigm of object detection, balancing the tradeoff between accuracy and complexity.

3. Vision Transformer Based Object Detection Models

Variants of CNNs had dominated in the application fields of object detection due to their performance but in 2017, Vaswani et. al proposed attention mechanism [33] and since then rise of transformers shifted the paradigm. Originally transformers were invented for natural language processing applications. Transformers used attention mechanism at its core which was originally proposed by Bahdanau et. al. [34] to address the bottleneck of encoder decoder architecture. Further this attention mechanism was refined by Vaswani et. al. in 2017. In 2021, Dosovitskiy et. al. [35] have proposed the use of self attention in transformers for computer vision applications. Introduction of Vision Transformers (ViTs) has ushered in a new era, leveraging self-attention mechanisms to model global dependencies and achieve superior representational power. Need of using the handcrafted features and using hand crafted components like anchor boxes and non max suppression was eliminated due to transformers.

This section discusses the foundational vision transformer detector models and why training and inference cost becomes concern in these models.

In 2020, facebook proposed first vision transformer called Detection Transformer (DETR) [36] for object detection. Carion et. al. [34] stated that DETR views object detection as simple set prediction problem. It used three components in its architecture:

CNN backbone, encoder-decoder architecture and simple feed forward network. CNN learns 2D representation of image, giving global context information.

Table 1. Performance Comparison of CNN based Object Detection Models

Criteria	RCNN	Fast RCNN	Faster RCNN	RetinaNet	YOLOv1	YOLOv5	YOLOv8 to YOLOv12
Architecture	Two stage	Two stage	Two stage+RPN	Single stage	Single stage	Single stage	Single stage
Region Proposal	Selective Search	Selective Search	Region Proposal Network (RPN)	Anchors	Grid Cells	Grid + Anchors	Grid + Anchors+ Transforms
Backbone Network	Alexnet	VGG	ResNet + FPN	ResNet+FPN	Custom CNN	CSPNet	Custom
Speed (FPS)	2 FPS approx.	5 FPS	7-10 FPS	10-15 FPS	45 FPS	60-140 FPS	70-200+ FPS
Accuracy (mAP)	55 to 60%	65 to 70%	~ 75%	~ 78%	62 to 65%	80 to 85%	85 to 90%+
Loss Function	Softmax + L2	Softmax + L1	RPN + Smooth L1	Focal Loss + Smooth L1	MSE + BCE	CIoU + BCE	DIoU/CIoU + Class Loss
Training Complexity	Very high	High	Moderate	Moderate	Low	Low	Low
Real Time Suitability	No	No	To some extent	Yes	Yes	Yes	Yes
Small Object Detection	Poor	Fair	Good	Excellent	Weak	Good	Excellent
Strengths	High accuracy	Unified architecture	End-to-end, accurate	Handles class imbalance well	Real-time	Efficient, flexible	SOTA, real-time + high accuracy
Weaknesses	Very slow	Slow	Slower than YOLO versions	Moderate speed	Low speed	Tuning required	Complex to deploy in older systems

which is flattened and added with positional encoding. Then it is passed through encoder-decoder block. Encoder uses self attention mechanism to build global context. Decoder predicts bounding boxes and classes using fixed number of learned object queries. It is a simple end to end pipeline removing the need of region proposal, anchor generation and non max suppression as it used bipartite matching between predicted and ground truth bounding box which ensures one to one mapping. Since, it is fixed set prediction problem and for unused slots, predicts as no object. CNN backbone used at initial stage must be stronger like ResNet 101 [37] or may be higher. Though it has simplified the detection as end-to-end pipeline, it shows slow convergence as compared to CNN detectors. Detection of small objects is difficult due to global attention.

Deformable DETR [38] was introduced to overcome limitations of the original DETR model. While DETR successfully eliminated many traditional, hand-crafted components in object detection by using a Transformer encoder-decoder architecture, it faced significant issues. Specifically, DETR required a much longer training schedule to converge (e.g., 500 epochs on COCO compared to 10-20 times faster for Faster R-CNN) and exhibited limited performance on detecting small objects. These problems are primarily attributed to how the Transformer attention modules handle image feature maps. Standard Transformer attention computed attention weights for nearly all pixels in the feature maps, leading to high computational and memory complexity that grows quadratically with spatial size, making it inefficient for high-resolution maps needed for small objects. At initialization, attention weights are nearly uniform, requiring extensive training to learn to focus on meaningful locations. To address these challenges, Deformable DETR was proposed. Inspired by deformable convolution [39], which efficiently attends to sparse spatial locations, the deformable attention module focuses only on a small, fixed set of sampling points around a reference point for each query, regardless of the feature map's spatial size. This pre-filtering mechanism for prominent key elements mitigates the issues of slow convergence and limited feature resolution. The deformable attention mechanism calculates features by aggregating values from these sampled key points, weighted by learned attention weights. The sampling offsets and attention weights are predicted based on the query feature and reference point. Unlike standard attention which has quadratic complexity with feature map size, deformable attention applied in the encoder has a complexity that is linear with spatial size. When used as cross-attention in the decoder, its complexity is independent of the spatial size. The concept is extended to Multi-scale Deformable Attention, allowing aggregation of features from multiple input feature map levels. This module samples points from different scale levels, effectively exchanging information across scales without needing traditional feature pyramid networks (FPN) [42]. The reference point coordinates can be normalized for clarity across scales, and a scale-level embedding is added in the encoder to identify the feature level of query pixels.

In Deformable DETR, multi-scale deformable attention replaces the Transformer attention modules processing feature maps in both the encoder and the cross-attention modules of the decoder. The self-attention modules in the decoder, which operate on object queries rather than feature maps, remain unchanged. For object queries in the decoder, the reference point is predicted from the query embedding. The detection head predicts bounding box offsets relative to this reference point, which aids convergence. The researchers also explored additional improvements called Iterative Bounding Box Refinement [40], where each decoder layer

refines the box prediction from the previous layer, and a Two-Stage Deformable DETR variant. The two-stage approach uses an encoder-only model in the first stage to generate region proposals from pixels, which are then fed to the decoder for refinement in the second stage. Experiments on the COCO [41] benchmark show that Deformable DETR achieves better performance than DETR, notably on small objects, while requiring 10 times fewer training epochs. With iterative bounding box refinement and the two-stage mechanism, performance can be further improved. Deformable DETR also demonstrates faster runtime compared to DETR-DC5 and is competitive with Faster R-CNN. Using multi-scale features significantly increases the total number of tokens processed by the encoder (by about 20 times compared to DETR), making the encoder a new computational bottleneck in Deformable DETR, slowing down inference. Preliminary experiments showed that the decoder in a fully-trained Deformable DETR primarily references only about 45% of the total encoder tokens, and that updating only a subset of tokens preferred by the decoder minimally impacted performance. This suggests considerable redundancy among encoder tokens.

In 2021, Swin Transformer [42], a novel vision transformer architecture intended to function as a general-purpose backbone for various computer vision tasks was proposed by Liu et al. Authors identified that standard transformers have the difficulties like vast scale variations of visual entities and the significantly higher resolution of images compared to text, which lead to computational complexity issues. To address these challenges, the Swin Transformer introduced a hierarchical architecture that constructs feature maps by merging image patches in deeper layers, enabling compatibility with dense prediction methods such as FPN or U-Net [43]. Its core innovation is the Shifted Window-based Self-Attention, which limits self-attention computation to non-overlapping local windows. This crucial modification reduces the computational complexity to a linear relationship with respect to the image size, assuming a fixed window size. To facilitate interaction and connection across different windows, the window partitioning is shifted between consecutive layers. An efficient batch computation approach using cyclic shifting is employed for practical implementation of shifted windows. The model also incorporates relative position bias, which is shown to improve performance, especially for dense prediction, unlike absolute position embedding which can be detrimental. The paper evaluated different model variants (Swin-T, S, B, L) and demonstrates their effectiveness. Swin Transformer achieved state-of-the-art performance on major computer vision benchmarks, including ImageNet [44] classification, COCO object detection (showing significant gains), and ADE20K [45] semantic segmentation too. Its hierarchical design and linear complexity made it an efficient and capable general-purpose backbone for a wide range of vision tasks, suggesting its potential for unified vision and language modeling.

In 2021, Wenhai Wang et. al. proposed Pyramid Vision Transformer (PVT), which used convolution-free backbone network [46]. PVT incorporated a pyramid structure similar to Convolutional Neural Networks (CNNs). This allows PVT to generate multi-scale feature maps, which is essential for dense prediction methods like FPN or Mask R-CNN. The architecture is hierarchical, consisting of four stages where feature map resolution progressively shrinks from high (4-stride) to low (32-stride). To handle the computational burden of high-resolution feature maps, PVT introduces a Spatial-Reduction Attention (SRA) layer, which reduces the spatial dimension of keys and values before the attention operation, significantly cutting down computation and memory costs compared to standard Multi-Head Attention. Extensive experiments have shown that PVT serves as a versatile and effective backbone. It boosted performance across various downstream tasks. For instance, PVT+RetinaNet significantly outperformed ResNet+RetinaNet on COCO object detection, often with a comparable or fewer number of parameters. PVT also enabled the construction of pure Transformer pipelines for object detection by combining with DETR, showing improved performance over a ResNet-based DETR. This indicated PVT's potential as a strong alternative to CNN backbones for pixel-level predictions.

In 2022, Yanghao Li et. al. proposed a use of plain, non-hierarchical Vision Transformer (ViT) backbones [47] for object detection, challenging the traditional reliance on hierarchical backbone architectures common in modern detectors. The original ViT architecture maintains a single-scale feature map, which presents challenges when adapted for object detection tasks that require handling objects at multiple scales. This contrasts with hierarchical convolutional networks (ConvNets) [48] and Transformer variants like Swin or multiscale vision transformers [49], which naturally produce multi-scale features. Plain ViTs also face efficiency issues with the high-resolution images often used in detection due to the computational cost of global self-attention. The proposed approach, named ViTDet, aims to adapt plain ViT backbones for detection with minimal modifications applied only during the fine-tuning phase, rather than redesigning the pre-training architecture. Key adaptations in ViTDet include:

- Simple Feature Pyramid: Instead of the complex Feature Pyramid Network (FPN) [49] design that combines features from multiple hierarchical stages, ViTDet builds a multi-scale feature pyramid solely from the final, single-scale feature map of the plain ViT backbone. This is achieved using convolutions and deconvolutions to produce feature maps at different scales (e.g., 1/32, 1/16, 1/8, 1/4) based on the default 1/16 stride of ViT. Surprisingly, empirical results show that this simple pyramid is sufficient for achieving the benefits of a pyramid, and the top-down/lateral connections characteristic of FPN are not necessary for a plain ViT backbone.
- Backbone Adaptation for High-Resolution Input: To handle high-resolution images efficiently during fine-tuning, ViTDet uses non-overlapping window attention in most layers. To allow information propagation across these windows, which window attention inherently restricts, a small number of cross-window propagation blocks are strategically inserted (e.g., 4 blocks evenly spaced). These blocks can implement either global self-attention or convolutional layers. Ablation studies show that using even a few such blocks provides significant gains over using only window attention. Convolutional

propagation is noted as being highly practical. The research highlights the benefit of using plain ViT backbones pre-trained with Masked Autoencoders (MAE) [50]. MAE pre-training on ImageNet-1K (without labels) resulted in substantial performance gains (e.g., +4.6 APbox for ViT-L) compared to supervised pre-training on ImageNet-1K or ImageNet-21K [51]. The authors hypothesize that plain ViTs, having fewer inductive biases (like scale equivariance), may benefit more from the data-driven learning provided by self-supervised MAE training, especially at larger model capacities, which helps alleviate overfitting.

In 2021, Haoqi Fan et al. proposed Multiscale Vision Transformers (MViT) [52], an architecture that merges the concept of multiscale feature hierarchies with Transformer models. A key element of MViT is the introduction of Pooling Attention (MHPA). MHPA addresses this by incorporating a pooling operator that reduces the sequence length (resolution) of the key and value tensors before computing self-attention. Optionally, the query tensor can also be pooled to reduce the output sequence length. The MViT architecture is structured into multiple "scale stages". At the transition between stages, the channel dimension is expanded, and the spatiotemporal resolution is down-sampled. This channel expansion happens in the MLP layers, while resolution reduction is driven by pooling the query tensor in the first MHPA block of the subsequent stage. Key and value tensors are pooled in all MHPA blocks within a stage. The model also utilizes pooled skip connections to handle dimension mismatches during these transitions. Empirical results on various benchmarks had shown that MViT achieved significant performance gains, particularly on video recognition tasks like Kinetics [53], SSv2 [54], Charades [55], and AVA [56], often without requiring large-scale external pre-training data that some concurrent vision transformers rely upon.

In 2022, Yangaho et. al. introduced MViTv2 (Multiscale Vision Transformer version 2) [57], an extended version of the original MViT by creating a feature hierarchy, transitioning from high to low resolution across stages. MViTv2 incorporates two primary technical improvements to enhance performance:

- Decomposed relative positional embeddings: These introduce shift-invariance by encoding the relative distance between tokens along different axes (height, width, temporal) instead of absolute positions or a joint encoding. This significantly reduces complexity for high-resolution features and is empirically shown to be accurate and much faster than joint relative positional embeddings.
- Residual pooling connection: This connection adds the pooled query tensor to the attention block's output, improving information flow and facilitating the training of pooling attention blocks. Ablations indicate both the pooling operator for the query and this residual path are essential for significant performance gains.

For object detection, MViTv2's hierarchical structure naturally integrates with standard Feature Pyramid Networks (FPN), allowing the backbone to produce multiscale features suitable for FPN's top-down and lateral connections. The source also explores a Hybrid window attention (Hwin) scheme alongside pooling attention, demonstrating that their combination can yield competitive accuracy/compute tradeoffs. Empirical evaluations show that MViTv2 achieved 58.7 AP^{box} on COCO object detection (using Cascade Mask R-CNN with system-level enhancements), and strong performance on Kinetics and SSv2 video datasets. Comparisons indicate MViTv2 outperforms previous MViT versions and competes strongly with or surpasses other Transformer architectures like Swin, often with better computational efficiency or fewer parameters. Fundamentally, ViTDet promoted a methodology where task-agnostic backbone pre-training is decoupled from downstream task-specific designs, allowing the use of general-purpose ViT backbones developed through various advancements like self-supervised learning without requiring architecture redesign for detection.

Caron et. al. in 2021 [58] explored whether self-supervised learning (SSL) endows Vision Transformers (ViTs) with distinct properties compared to convolutional networks (convnets) or ViTs trained with supervision. The study was named as DINO, interpreted from self-distillation with no labels. A key finding is that ViT features trained with DINO explicitly capture information about the semantic segmentation of an image. Furthermore, they encode details about the scene layout and, notably, object boundaries. This crucial spatial information is directly accessible within the self-attention modules of the network's final layers. Researchers found that different attention heads within a DINO-trained ViT can automatically focus on and delineate distinct objects or parts. This capability extends even to objects that are small or partially hidden (occluded). It was contrasting to supervised ViTs, which show less effective attention to objects, particularly in cluttered scenes. The study indicates that employing smaller image patches within the ViT architecture is important for enhancing performance on these dense prediction challenges. DINO's self-supervised training imbues ViT features with fundamental, spatially-rich properties – explicit semantic awareness, object boundary detail, and scene layout understanding – that are directly relevant and highly beneficial for building effective object detection systems. Scaling traditional SSL methods to large, uncurated datasets has typically resulted in reduced feature quality due to a lack of data control and diversity.

DINOv2, a family of pretrained vision models addressed this challenge by employing a self-supervised approach trained on a large volume of curated data. A significant contribution was the development of an automatic pipeline for creating a dedicated, diverse dataset, named LVD-142M, from extensive uncurated sources. Inspired by NLP data processing, this pipeline utilized visual similarity instead of external metadata, involving embedding uncurated images, deduplicating them, and using self-supervised image retrieval to augment curated data sources. The DINOv2 pretraining method was based on existing discriminative SSL techniques like DINO and iBOT. Key technical modifications for scaling and stability included using separate MLP projection heads for the DINO and iBOT losses, implementing Sinkhorn-Knopp centering, incorporating the KoLeo

regularizer to encourage feature diversity, and adapting image resolution late in pretraining. Efficiency was enhanced through techniques such as Flash Attention, sequence packing, efficient stochastic depth, and Fully-Sharded Data Parallel (FSDP). DINOv2 utilizes Vision Transformer (ViT) architectures, including a large 1.1B parameter ViT-g model. Smaller models were effectively trained using knowledge distillation from the ViT-g teacher, yielding better performance than training from scratch. Qualitative results supported these findings for dense prediction tasks like object detection. In essence, DINOv2 demonstrated that scaling self-supervised pretraining with sufficient curated data and technical improvements can successfully learn transferable, general-purpose visual features competitive with state-of-the-art methods without requiring downstream finetuning.

In 2022, Roh et al. proposed Sparse DETR suggesting encoder token sparsification. The method selectively updated only a learned subset of encoder tokens, helping the model focus on detecting objects efficiently. Features for tokens not selected are passed through the encoder layers unchanged, still available to be referenced as keys by the selected tokens. This selection is achieved by introducing a scoring network that predicts the "saliency" of each encoder token. Two main criteria for determining token saliency and training the scoring network are explored:

- Objectness Score (OS): Uses a separate detection head on the backbone feature map to predict an objectness score for each token. The top-ρ% tokens with the highest scores are selected.
- Decoder cross-Attention Map (DAM): Aggregates the cross-attention weights from the decoder queries to the encoder output tokens, creating a map that reflects how much each encoder token is referenced by the decoder. A binarized version of this map (top-ρ% weights) serves as the pseudo ground truth for training the scoring network using a binary cross-entropy (BCE) loss. The DAM criterion explicitly aligns token selection with the decoder's focus. Compared to OS, DAM better captures object boundaries and internal areas rather than just high-frequency edges.

Sparse DETR incorporates two additional components to improve performance and stability:

- Encoder Auxiliary Loss: Auxiliary detection heads with Hungarian loss are added to the encoder layers. Unlike in other DETR variants, this loss is applied only to the selected, sparsified encoder tokens, keeping computational overhead low. This loss helps stabilize training, particularly for deeper encoders by providing intermediate gradients and alleviating vanishing gradient issues, ultimately improving detection performance.
- Top-k Decoder Queries: Instead of relying solely on learned object queries, the decoder takes the top-k encoder outputs (based on objectness score from an auxiliary head) as its queries, similar to Efficient DETR.

Experiments on the COCO 2017 benchmark demonstrated the effectiveness of Sparse DETR with backbone like swin transformer.

- It significantly reduced computational cost (FLOPs) and increased inference speed (FPS) compared to Deformable DETR. For instance, with a 10% keeping ratio on Swin-T, it outperforms Deformable DETR while reducing total computation by 38% and increasing FPS by 42%. The encoder computation itself can be reduced by approximately 82% at a 10% keeping ratio.
- The DAM selection criterion consistently outperforms the Objectness Score and random selection methods across different keeping ratios and backbones
- Sparse DETR shows robustness to dynamic sparsification during inference, allowing a single trained model to be used efficiently at various sparsity levels without significant performance degradation, unlike some prior methods.

These foundational models such as Detection Transformer (DETR) and its successors which are summarized in Table 2, have demonstrated that ViTs can match or even surpass CNN-based detectors in accuracy. However, the inherent computational demands, memory usage, and need for large-scale training data present notable challenges when deploying these models in real-world or resource-constrained environments. ViTs require extensive pre-training and large-scale datasets to achieve competitive performance. Self-attention mechanism in these transformers show quadratic complexity and hence leads to high memory and computational requirements. When multiple objects are present in image, inference can be slowed down and becomes resource intensive. So, major limitations are heavy computational cost due to global self-attention mechanism, slow inference speed and difficulty in capturing fine grained spatial features. Hence for real time applications as well as resource constrained applications these models may be less suitable. Further for complex situations like blurred images, occluded images, small objects, densely packed scenes more challenges are faced. It indicates a need for more robust scalable and efficient solution.

To address these challenges, in recent literature researchers have focused on improving these models by architectural innovations, variation in attention to optimize the performance in terms of reducing computational cost, memory usage and training time of model during learning as well as reducing latency and computational cost during deployment. This gives rise to two different optimizations strategies: at training time and at inference time.

4. Training Time Optimization Strategies

Training time optimization is used to speed up training of model, reduce memory usage or enable training on smaller datasets. It may contain modification in architectural design of a model or using variants of attention mechanism in such a way that it will reduce computational cost of a model. For training time optimization strategies, we have reviewed latest papers from year 2023 to

current year i.e. 2025 and selected the papers from reputed journals of Elsevier, Springer etc. and prestigious conferences like ICML, ICLR, European conference of Computer vision etc. We have rarely selected

Table 2. Comparison of Foundational Transformer Based Models for Object Detection Models

Model	Paper/Year	Key Contributions	Backbone Architecture	Accuracy	Training Time	Inference Complexity	Multiscale Feature use	Real Time Capability
DETR	*DETR*, ECCV 2020 (FAIR)	First end-to-end transformer-based object detector with bipartite matching	ResNet + Transformer Decoder	~42 AP (COCO)	Very Long (~500 epochs)	High	No	No
Deformable DETR	*ICLR 2021* (MSRA)	Deformable attention, multi-scale features, faster convergence	ResNet / Swin Backbone	~45 AP (COCO)	Fast (~50 epochs)	Moderate	yes	Moderate
Swin Transformer	*ICCV 2021* (MSRA)	Hierarchical ViT, shifted window attention for dense vision	Swin-T/B/L	83.5% top-1 (ImageNet); 49–53 AP (COCO)	Moderate (~100 epochs)	Medium	Yes	Moderate
PVT	*PVT: Pyramid Vision Transformer*, 2021	First pure ViT for dense prediction; pyramid structure like CNN	PVTv1/PVTv 2	~41–44 AP (COCO)	Fast	Low	Yes	Yes
MViT	*Multiscale Vision Transformers*, CVPR 2021 (Facebook AI)	Scales resolution/channel depth per stage, multiscale feature maps	MViT-T/S/B	82–84% top-1 (ImageNet); 45+ AP (COCO)	Moderate	Moderate	Yes	Moderate
MViTv2	*CVPR 2022*	Improves token interaction efficiency, training stability	MViTv2-T/B	85.3% top-1 (ImageNet); 52–56 AP (COCO)	Efficient (~100 epochs)	Medium-low	Yes	Yes
DINO	*DINO: DETR with Improved DeNoising*, ICLR 2023	Combines noise denoising + better query initialization for strong accuracy	Swin, ConvNeXt + Decoder	54.4 AP (COCO, Swin-B)	Moderate (~50–80 epochs)	High	Yes	No
DINOv2	*DINOv2: Self-supervised Representation Learning*, 2023	Foundation vision model; high-quality features without labels	ViT-G / ViT-S/B/L	88.5% top-1 (ViT-G); strong transfer	Long (SSL + fine-tuning)	High	Optional	No

any specific application of object detection, instead our focus was to select improvements in vision transformer to optimize training of the model. Our understanding of reviewed papers is presented below in this section.

Generally, we train any model at once but when object detector is trained in phases, it is called as incremental learning. Each phase in Incremental Object Detection (IOD), introduces annotations for new object categories while aiming to retain knowledge of previously learned categories. A major hurdle in this setting is catastrophic forgetting, where learning new information overrides earlier acquired knowledge. Traditional methods to combat catastrophic forgetting in machine learning, are Knowledge Distillation (KD) and Exemplar Replay (ER). They have been applied to IOD. However, Liu et. al [59] highlighted that these standard techniques do not work well when applied directly to state-of-the-art transformer-based object detectors like Deformable DETR and UP-DETR. Authors have identified two main issues causing this performance drop:

- **Unbalanced KD loss and contradictory supervision:** Transformer-based detectors test a large number of object hypotheses, most of which correspond to background. Standard KD compares the new model's output tokens to the old model's, leading to a loss dominated by redundant background information. Furthermore, since training images can contain both old and new category objects (only new ones are annotated in the current phase), the standard KD loss (preserving old predictions) and the regular training objective (learning new annotations) can provide contradictory evidence.
- **Mismatch in category distribution for ER:** While ER for image classification often samples an equal number of exemplars per category, this strategy is problematic in IOD datasets like COCO 2017, which have a naturally skewed object category distribution.

To address these challenges, the authors have proposed the ContinuaL DEtection TRansformer (CL-DETR). CL-DETR introduced two key improvements: Detector Knowledge Distillation (DKD) and an improved Exemplar Replay strategy with distribution-preserving calibration. Detector Knowledge Distillation (DKD) modified the standard KD loss. Instead of comparing raw outputs, DKD selected only the most confident foreground predictions from the old model and used them as "pseudo labels" for old categories. Redundant background predictions were ignored. These pseudo labels were then merged with the ground-truth labels for the new categories. The model was trained using the standard DETR loss applied to this merged set of labels, utilizing bipartite matching to ensure a one-to-one correspondence between predictions and labels and avoiding duplicate detections. This approach resolved conflicts between old and new information and avoided the dominance of background predictions. The improved Exemplar Replay strategy in CL-DETR addressed the distribution mismatch. Exemplars were selected to match the natural category distribution of the training set observed in that phase, rather than creating a balanced subset. The training process in each phase consists of two steps: a main training step using the DKD loss on the current data combined with the exemplar memory, followed by a smaller step fine-tuning the model on the new exemplar set to achieve better calibration.
Experiments on COCO 2017 showed state of the art results for CL-DETR in the IOD setting when applied to Deformable DETR and UP-DETR. Compared to directly applying KD and ER, CL-DETR significantly boosts the Average Precision (AP). Ablation studies confirm the effectiveness of both the DKD components (joint bipartite matching, pseudo label selection) and the distribution-preserving calibration in reducing forgetting and improving performance. To address the same challenge, Jichuan Zhang et. al. have proposed DyQ-DETR (Dynamic object Query-based DEtection TRansformer) [60]. Built upon the Transformer architecture, specifically Deformable DETR, DyQ-DETR is inspired by dynamic networks that expand model capacity incrementally. The core idea is to use dynamic object queries to incrementally expand the model's representation ability. At each incremental step for new classes, a new set of learnable object queries is added and aggregated with queries from previous steps. This creates segregated groups of queries, where each group is primarily responsible for detecting objects from the classes introduced at a specific step. This dynamic expansion and segregation alleviates the conflict between preserving old knowledge and learning new knowledge.
DyQ-DETR implements several mechanisms to make this dynamic query approach effective and efficient:

- **Disentangled self-attention:** Since object instances from different class sets rarely overlap, DyQ-DETR removes the self-attention interactions *between* query groups from different phases. This significantly reduces computational complexity, growing almost linearly, while maintaining performance. Each query group's ability to detect its assigned classes is preserved.
- **Risk-balanced partial calibration:** For IOD scenarios that use exemplar replay (retaining a small subset of old data), DyQ-DETR introduces a method to improve the selection and use of these exemplars.
    - **Risk-balanced selection** chooses exemplars based on a "risk score" derived from the partial loss of detecting new classes, selecting images with moderate scores that are informative and reliably annotated. This helps overcome limitations of using exemplars with incomplete labels.
    - **Partial calibration** specifically uses the incomplete real annotations in exemplars to train only the corresponding query group, avoiding reliance on potentially inaccurate pseudo labels and preventing bias towards certain classes.

Through these methods, DyQ-DETR was able to effectively address catastrophic forgetting and the limitations of fixed-capacity models in IOD. Experiments show it consistently outperforms state-of-the-art methods on the COCO dataset in both non-exemplar and exemplar-based settings, demonstrating improved stability and plasticity with limited parameter overhead.

Hang Chen et. al. observed that while heavy transformer heads perform well on complex images, lighter heads can achieve satisfactory results on simpler images with fewer, distinct objects. This led to the idea that using a single, heavy head for all images is inefficient. To overcome this challenge, authors have proposed HS-DETR, employing a dynamic head switching (DHS) [61] strategy. This involves incorporating multiple transformer heads with varying computational complexities into the DETR architecture. A lighter head is constructed by reducing the number of layers in the original heavy head. A lightweight module, typically a 3-layer MLP, is added to the backbone network. During inference, this module dynamically selects the most appropriate head for a given image based on its prediction. The goal is to assign lighter heads to easier images and heavier heads to harder ones, thereby improving the balance between accuracy and efficiency. The switching module is fine-tuned to operate within a specified computational budget, offering flexibility without requiring retraining for different efficiency needs. A potential challenge with using lighter heads is a decrease in detection accuracy. To mitigate this, the authors presented online head distillation (OHD). This technique is applied during the training process, transferring knowledge from the heavier head to the lighter heads. OHD includes both encoder distillation, which focuses on important encoder features weighted by attention maps, and decoder distillation, which aligns outputs and attention maps of matched object queries between the heads. Unlike some prior distillation methods, OHD works in an online manner and employs techniques like bipartite matching to align outputs for effective knowledge transfer. The multiple heads are trained jointly on a shared backbone, which experiments show provides a slight accuracy benefit even before applying OHD. OHD further enhances the performance of the lighter heads. Extensive experiments on the MS COCO dataset demonstrated a better accuracy–efficiency trade-off. The dynamic switching strategy is shown to effectively select heads based on image difficulty, and ablation studies confirm the effectiveness of OHD and other components.

Ke Li et. al. has proposed the Diagonal-shaped Window [62] attention mechanism. Vision Transformer architecture based on it is called DiagSWin Transformer. The DiagSWin attention mechanism tackles the computational and multi-scale challenges through several key innovations:

- **Hybrid Scale Attention:** In a single attention layer attentions in diagonal regions are modeled at hybrid scales. Tokens attend to their closest surroundings finely and tokens further away coarsely. This injects multi-scale receptive field sizes into tokens.
- **Multi-scale Downsampling:** Prior to attention, feature maps are downsampled to different sizes for different attention heads using multi-scale pooling (like convolution). This provides rich fine- and coarse-grain information in parallel.
- **Diagonal Aggregation:** Tokens are aggregated into diagonal-shaped windows. By attending to these diagonal regions, the mechanism can cover image regions with less cost compared to standard self-attention.
- **Alternating Windows:** Consecutive DiagSWin Transformer blocks alternate between computing self-attention in left and right diagonal-shaped windows. This effectively expands the attention region to achieve broader interaction.
- **Parallel Processing:** Multi-heads are split into parallel groups, applying different self-attention operations, which helps reduce extra computation.

The architecture is hierarchical, consisting of stages with overlapping embedding to build a feature pyramid suitable for dense prediction tasks. It also includes a Detail Feed-Forward (DFF) layer with a depth-wise convolution to incorporate local details and improve learning capability. Experimental results on ImageNet, COCO, and ADE20K demonstrate that DiagSWin Transformers consistently outperform state-of-the-art Vision Transformers and CNNs, achieving better accuracy-efficiency trade-offs. Ablation studies confirm the effectiveness of the DiagSWin attention and the DFF layer. To improve the performance and efficiency of Vision Transformers (ViTs), particularly when trained on smaller datasets. Zhang et. al. proposed combination of CNN and vision transformer [63]. A primary challenge with standard ViTs is their lack of inherent inductive bias compared to Convolutional Neural Networks (CNNs). ViTs process images by dividing them into patches and using self-attention to capture global relationships. However, this mechanism initially treats all tokens equally, overlooking the strong relationships between neighboring pixels or patches. Learning this inductive bias takes substantial data and training time, making ViTs less efficient and slower to converge than CNNs on limited datasets. Furthermore, while global attention is captured, ViTs may overlook fine-grained local details.

To address these challenges, authors have proposed incorporating a lightweight Depth-Wise Convolution (DWConv) module into the Vision Transformer architecture. This module is designed to act as a shortcut, bypassing entire Transformer blocks (which consist of attention and feed-forward layers). The core idea is to seamlessly integrate the convolutional network's strength in capturing local information with the Transformer's ability to learn global representations. The proposed method resolves the challenges in several ways:

- Introduces Inductive Bias: By adding the DWConv module, which inherently processes local pixel relationships using filters, the model gains the inductive bias that vanilla ViTs lack. This helps the model understand the structure of images more effectively from the outset.
- Enhances Local Detail Capture: The DWConv module specifically captures the fine-grained local details that might be missed by the Transformer blocks' focus on global interactions. The module reshapes the 1D patch tokens back into 2D

feature maps for convolution and then converts them back to 1D tokens before adding them to the Transformer block output, ensuring detail is integrated.

- Improves Training Efficiency: The added inductive bias and local detail capture lead to significantly faster convergence, especially when training from scratch on small datasets.
- Boosts Performance on Small Datasets: Substantial performance improvements are noted on small datasets like CIFAR-10, CIFAR-100, and Tiny-ImageNet. Remarkably, smaller ViT models with this mechanism can outperform larger original ViT models with considerably more parameters on these datasets.

The approach is designed to be a simple and can be easily integrated into most Transformer models with only a minimal increase in parameters and computational cost. Architecture variants are also proposed, such as having a single DWConv module bypass multiple transformer blocks to further reduce parameters/FLOPs in deeper models, or using multiple parallel DWConv modules with different kernel sizes to enhance local feature extraction. While highly effective when training from scratch, the benefits may be less pronounced with transfer learning or on very large datasets where the original ViT drawbacks are less severe. A novel approach to object detection is proposed by Zhang et. al. [64]. It addresses the challenge of effective extraction and fusion of multi-scale features. Traditional backbone networks lose spatial content details during down-sampling, and standard FPNs, while attempting to fuse features from different levels, suffer from insufficient feature capture at both local and global scales and inefficient information transmission. This makes it hard for models to distinguish foreground targets from similar backgrounds in challenging environments. To tackle these problems, authors have proposed the Adaptive Interactive Feature Extraction (AIFE) network. At its core is the Adaptive Multi-scale Feature Extraction Encoder (AMFEFormer), designed to enhance the feature processing within the model's neck structure, often incorporating an FPN.

The MCIEM addresses the local scale feature capture limitations. Situated at the highest input level of the FPN, it uses large kernel expansion convolution and attention mechanisms to dynamically extract multi-scale contextual information from different channels with a larger receptive field. This enhances semantic and spatial dependency capture, improving target area localization and pixel discrimination. It leverages channel splitting and grouping, along with depth-separable convolution, to improve efficiency and control parameters. The IETrans module is designed to improve global feature interaction, anti-jamming ability, and information transfer efficiency. It adopts a Transformer-based structure combined with CNN elements. By employing dual cross-attention, IETrans achieves dynamic interaction, fusion, and adaptive regulation of global and local features, establishing dependencies between distant pixels. This interactive extraction and adaptive adjustment of features from different scales helps overcome the rigid information flow of traditional FPNs and improves the model's ability to differentiate targets from backgrounds in complex scenes. IETrans, along with Feature Alignment Aggregation (FAA) and Feature Interaction Fusion (FIF) modules within AMFEFormer, enables efficient aggregation and transmission of features, compensating for the FPN's limitations in processing nonadjacent layers. Lightweight convolutional types and uniform channel design are used in IETrans to control computational costs.

Experimental results on the PASCAL VOC 2007 + 2012 and MS-COCO2017 datasets validate the effectiveness of the proposed method. AIFE-Net, built upon the TOOD baseline, significantly improves mean average precision (mAP). Ablation studies confirm the contribution of both MCIEM and IETrans modules to performance gains, demonstrating their ability to enhance feature extraction and interaction. While some state-of-the-art DETR-based methods achieve slightly higher accuracy, AIFE-Net demonstrates a strong balance between accuracy, computational complexity, and inference speed, making it more suitable for resource-constrained practical applications.

Evaluated on MS-COCO2017, AIFE-Net demonstrated excellent detection performance and stable efficiency with different backbone networks like ResNet50 and ResNet101, achieving APs of 44.2% and 46.9% respectively. When using Swin-T and Swin-S backbones, it achieved APs of 46.8% and 50.9%, comparable to state-of-the-art detectors. AIFE-Net showed significant advantages in computational complexity and resource consumption compared to methods like DINO-DETR, Co-Def-DETR, and Relation-DETR. It achieved higher inference speed (11.6 FPS) and significantly lower parameter count (35M) and FLOPs (196G) than these methods, demonstrating effective control of model complexity. Despite slightly lower AP compared to some leading methods, the study highlights AIFE-Net's lightweight design and efficiency benefits, while also noting limitations regarding feature richness and the incomplete nature of the transformer integration. Future work aims to optimize the encoder and explore more efficient structures to balance accuracy and efficiency.

Li et. al. have proposed DMCTDet [65], a novel composite transformer object detection network guided by a density map for urban scene object detection in UAV images. Detecting objects in UAV images is challenging due to small object size, variability, and diversity, especially for weak aggregated objects. DMCTDet addresses this by combining object density estimation with object detection. It is guided by density maps which provide prior information about object distribution. To accurately generate coarse-grained density maps, the paper introduces MSCSRNet, a new multiscale density estimation network that uses ResNet18 as a front-end for feature extraction and includes a multiscale feature fusion module (MSFFM). The core detection network features a composite backbone combining Swin Transformer and Vision Longformer to capture both large-range global context and smaller-range local details, mitigating the issue of small object feature loss. In the feature fusion stage, an adaptive multiscale feature pyramid enhancement module (AMFPEM) is used. This module dynamically senses object scale changes to learn connections between them and enhances feature response values for small objects. Experiments were conducted on the VisionDrone2019 dataset. Ablation studies verified the effectiveness of the proposed components. Comparative

experiments with methods like Cascade R-CNN, RetinaNet, CornerNet, ClusDet, and DMNet demonstrated that DMCTDet outperforms them in urban scene detection performance on UAV images. Though it was effective, the network's computational efficiency and the limited sample size of public UAV datasets for transformer-based networks are still the limitations.

Moorthy et. al. have presented a hybrid multi-attention (HyMAT) module and a Transformer-based framework called HyMATOD for improved object detection in videos [66]. The paper highlights the difficulty in independently computing correlations in conventional attention mechanisms, which can lead to noise and ambiguity. The proposed HyMAT module addresses the issue by consolidating agreement across correlation vectors to amplify relevant correlations and mitigate inaccurate ones, resulting in notable performance enhancements. HyMATOD utilizes the Transformer architecture and is designed for video object detection by efficiently employing encoded features and introducing embeddings of target backgrounds to better leverage temporal references. The framework starts processing from an initial frame with ground truth annotations to initialize the detector by extracting long and short-term information. Evaluation was conducted on the challenging ImageNet VID and UA-DETRAC datasets. On the UA-DETRAC dataset, HyMATOD significantly outperformed Faster R-CNN, demonstrating its practical applicability, particularly for objects with degraded appearance where strong temporal dependencies are beneficial. On the ImageNet VID dataset, HyMATOD achieved an impressive 86.7% mAP with a ResNet-101 backbone, surpassing state-of-the-art methods. The incorporation of both long-term and short-term references also enhanced the computational efficiency compared to methods used for comparison. However, a limitation of the approach is additional hyperparameters that require meticulous tuning for optimal performance.

Zhao et. al. addressed a significant challenge in applying Vision Transformers (ViTs) to object detection: their heavy reliance on traditional position embedding mechanisms. Position embedding, while crucial for encoding spatial relationships, requires extensive training data to be effective and can introduce redundant or conflicting information in diverse scenarios. The standard ViT process of flattening image patches also disrupts vertical positional relationships. This leads to limitations in flexibility, robustness, and generalization, particularly with limited datasets or in complex environments. To overcome these issues, the authors proposed a novel position-free embedding model called HV-SwinViT [67]. It replaces traditional position embedding by directly embedding both Horizontal and Vertical features using a modified self-attention mechanism. It has the module HVblock, designed to capture positional information without explicit position embeddings. Based on the Swin Transformer block, HVblock introduces a vertical flattening operation alongside the standard horizontal flattening. It transposes feature maps and uses fully connected layers on both the original and transposed versions, it learns relative positional relationships in both horizontal and vertical directions, even those disrupted by earlier flattening. The Q and K values generated from these processes (Q, K from horizontal; Q', K' from vertical) are then fused (specifically, added: Q+Q' and K+K') before being used in the self-attention mechanism. This HV-Self-Attention is the mechanism by which horizontal and vertical positional information is captured in a position-free manner.

HV-SwinViT integrates the HVblock into two hybrid modules: HV-Swin-B and Cf2-HV. These modules combine the HVblock with convolutional neural network (CNN) elements to enhance sensitivity to spatial data and capture both local and global features. HV-Swin-B, used in the backbone, splits input channels between the HVBlock and Depthwise Separable Convolution (DW-Conv) to reduce parameters and improve generalizability. Randomly adjusting channel indices during training further enhances robustness. Cf2-HV modifies the ELAN architecture, using channel splitting and 1x1 convolutions to efficiently fuse features from different depths, mitigating vanishing gradients and integrating features from various receptive fields. The HVblock within Cf2-HV also serves as an output head for object detection, leveraging the Transformer's ability to handle variable target information. The 1x1 convolution itself is redefined and utilized as a learnable activation function to capture nonlinear position features. It gave competitive results on benchmark datasets like MS-COCO2017 (53.1% AP with DNet-V8), VisDrone2019 (35.6% AP), and AI-TOD (32.1% AP). Outperforms specific SOTA models designed for small object detection on AI-TOD. While the most accurate model version (HV-SwinViT) had more parameters (60.4M) and a slightly slower inference speed (46 FPS) than some internal variants, it maintained real-time capability (above 30 FPS). Optimization via the SwinViT base significantly reduced parameters and computational cost compared to a traditional ViT baseline.

The HV-Self-Attention mechanism and the integration of multiple modules contribute to computational overhead. Certain design choices, like directly concatenating features without dimensionality adjustment, can lead to a significant increase in parameters and computational load. Although this model was designed to avoid redundant information from position embeddings, suboptimal configurations (e.g., Cat splicing for QK fusion) still led to parameter increase and potential learning biases.

To tackle the challenges in real time object detection in dynamic environments the researchers have introduced a novel framework called the Attention Transformer-YOLOv8 model in [68]. This approach integrates the strengths of the efficient YOLOv8 backbone with an attention mechanism and a Transformer-based detection head. The core idea is to enhance feature extraction and leverage global context and long-range dependencies while maintaining real-time efficiency. Authors used YOLOv8 backbone and extracted essential features from pre-processed data. An attention mechanism module refined these features by highlighting areas of importance for object detection. This mechanism helped in tracking objects across frames, even during occlusion or appearance changes, by understanding the links between sequential frames. The attention-refined features then passed to a Transformer-based detection head. The Transformer component incorporated mechanisms like Embedding Layers to transform image features into dense vectors, Positional Encoding to add spatial context, Multi-Head Self-Attention to capture complex relationships between features, and Feed-Forward Neural Networks for further data processing. The final Output

Layer used the processed features to predict bounding boxes for detected objects and assign class labels. The model also incorporated lightweight attention mechanisms, such as depth-wise separable convolutions, to optimize inference time. This hybrid design balances the lightweight efficiency of YOLOv8 with the contextual modeling capabilities of the Transformer. demonstrated superior performance compared to existing methods like Faster R-CNN, YOLOv3, YOLOv5n, and SSD. It achieves high precision (96.78%), recall (96.89%), and mean average precision (mAP) (89.67%). A key advantage is its real-time efficiency, with an inference time of only 5.2 ms per frame on a real-time dataset. The model effectively addressed challenges in complex and dynamic environments, handling crowded scenes, varying lighting conditions, occlusion, blur, and diverse object sizes with enhanced robustness. The integration of Transformer-based detection heads improved the accuracy of bounding box localization. The model's adaptability to changes in environmental factors like lighting and weather also contributed to its robust performance in real-world scenarios. Furthermore, the framework is designed to be scalable for large-scale deployments, such as citywide surveillance networks, due to its real-time processing and efficient resource utilization. The model also has potential for expansion into other real-time applications like autonomous driving and industrial automation. Although it achieves a low inference time on high-end hardware, this latency can still pose challenges on resource-limited edge devices, requiring further optimization techniques like model pruning or quantization. The model also showed some difficulty in distinguishing between certain object classes, with the confusion matrix indicating confusion between motorcycles and cars, and between people and the background, highlighting areas for potential improvement. Additionally, the performance may be less robust in extreme environmental conditions like severe lighting or weather, suggesting a need for future enhancements in this area.

Fusion Detection Transformer (F-DETR), which explicitly integrates multi-scale features into the decoder for the first time in an end-to-end DETR structure is proposed by Liu et. al. [69]. F-DETR is composed of several key parts: a backbone network (fine-tuned ResNet-50) for local feature extraction, a global encoder layer using self-attention on single-scale high-level features, a heterogeneous scale multi-branch fusion structure, query selection with added noise, and a stacked decoder layer. The core innovation lied in the heterogeneous scale multi-branch fusion structure, which takes features from different backbone stages ({L3, L4, L5}) and the global encoder output ({G5}) and fuses them using a diversified branch structure. This structure used fusion blocks with reparameter blocks to converge features of adjacent scales and diversify representations. It can be simplified during inference using reparameterization techniques. Multi-scale features, which contain local details, are fused with global context information from the encoder. The model also employs a query selection scheme with noise that selects fused features and uses noisy ground-truth boxes to initialize object queries, aiming to reduce optimization uncertainty and improve convergence stability. The decoder, built on deformable cross-attention, then refines these queries. It Introduced multi-scale features to the decoder, accelerating training convergence and reducing computational cost compared to encoder integration. It achieved significant multi-scale sequence length reduction compared to Deformable-DETR. Query denoising improved training stability and accelerated convergence. It showed significant improvements in small object detection compared to some models.

5. Inference Time Optimization Strategies

Real time systems like autonomous vehicles, robotics and surveillance systems require faster response time i.e. low latency. Also, due to compact technology, many applications are moving from cloud to edge devices like IoT devices, drones, mobile phones etc. These are energy constrained devices. When heavy models like vision transformers are deployed in energy constrained devices, balance needs to be maintained in model parameters, latency, energy consumption, accuracy and FLOPs to reduce energy consumption. Optimization at inference time reduces these parameters which leads to low energy consumption and hence cost savings at deployment. In this section we have reviewed recent research papers showing inference time optimization. Maximum papers we have selected are from year 2025 to study the latest research, few are from 2024 and 2023. Papers are selected from Scopus indexed journals like Elsevier, springer and reputed conferences like ICLR, ICML, ECCV or pre-prints of these conferences.

In 2023 Liu et. al. introduced EfficientViT (2023) [70], a new family of Vision Transformers (ViTs) specifically engineered for high-speed inference in real-time applications. The authors highlighted that while standard ViTs achieve impressive capabilities, their substantial model sizes and computational demands make them impractical for deployment in speed-critical scenarios. Crucially, they pointed that traditional metrics like reducing parameters or FLOPs don't always translate to actual inference speed or throughput on hardware. To address this, the authors performed a detailed analysis, identifying the underlying bottlenecks in ViT inference speed. They discovered that speed is often memory-bound, predominantly due to the memory-inefficient tensor reshaping and element-wise operations within the Multi-Head Self-Attention (MHSA) mechanism. Additionally, they found significant computation redundancy stemming from attention heads learning highly similar projections. Finally, suboptimal parameter allocation in existing lightweight models contributed to inefficiency. Based on these insights, EfficientViT is designed around three core principles:

A memory-efficient sandwich layout block positions a single memory-bound MHSA layer between multiple memory-efficient FFN layers. This arrangement reduces the time spent on memory-intensive MHSA operations while allowing effective channel communication via the FFNs. Depthwise convolutions (DWConv) are also integrated for local feature capture. Cascaded Group Attention (CGA) combats attention redundancy. Similar to grouped convolutions, CGA feeds different input feature splits to each head, explicitly decomposing attention computations. A cascading mechanism further enhances feature representations by adding the output of one head to the input of the next, increasing capacity without adding parameters. Parameter Reallocation optimizes

the distribution of parameters based on importance analysis. This involves expanding channel dimensions for critical components like V projections while reducing them for less critical ones such as Q, K, and FFN dimensions, boosting overall efficiency. The architecture is hierarchical, utilizing overlapping patch embedding and efficient subsampling layers. It employs hardware-friendly elements like BatchNorm (BN) and ReLU activation. EfficientViT-M4 gave competitive results on COCO object detection, outperforming other efficient models with fewer FLOPs. Ablation studies validate the effectiveness of the sandwich layout, CGA, parameter reallocation, and other design choices. But it was observed that model size was a slightly larger compared to some efficient CNNs.

Katare et. al. have proposed a model approximation approach called VI-ViT [71]. It is used for edge deployment. To tackle the issues of high energy consumption and latency in real-time processing caused by the self-attention mechanism and the accuracy losses associated with compression, VI-ViT utilizes variational inference and mixed-precision quantization. Variational inference is used to optimize parameter distribution and approximate inference to lower computational demands, both contributing to energy efficiency. Mixed precision involves using different numerical precisions (e.g., 16-bit, 32-bit) for different model components, which balances computational efficiency with model performance. Quantizing network weights or layers to lower precision after training can lead to gains in model inference metrics. For ViTs, mixed precision techniques have been explored to optimize deployment on edge devices by balancing power consumption and performance. Post-training quantization combined with mixed precision formats enabled efficient deployment and inference on resource-constrained devices. The proposed VI-ViT approach showed significant reductions in model parameters and FLOPs. For instance, experiments on autonomous driving datasets like nuScenes and Waymo has showed up to 37% reduction in parameters and 31% reduction in FLOPs while maintaining competitive accuracy. The paper also evaluated LiTeViT models, optimized versions using this approach, demonstrating competitive mIoU percentages with reductions in GFLOPs and latency, suggesting effective optimization for resource-constrained edge scenarios. Energy evaluations showed that LiTeViT models achieve high accuracy while conserving power, positioning them as viable solutions for energy-sensitive applications. The integrated approach of variational inference and mixed-precision quantization directly reduces computational resource requirements.

Wu et. al. introduced ELFATT (Efficient Linear Fast Attention) [72], a novel mechanism designed to improve the efficiency of Vision Transformers. Existing acceleration methods fall into two categories: memory-efficient methods that optimize memory I/O but still have quadratic computation complexity, and computation-efficient methods that achieve linear complexity but often sacrifice performance. ELFATT aimed to combine the advantages of both, achieving low memory I/O, linear computational complexity, and high performance. It incorporated two heads with almost linear complexity to capture both local and long-range dependencies. The sparse blockify attention head accelerated further using Flash Attention mechanisms to reduce memory I/O operations. Experiments have shown that ELFATT offered significant speedups over the vanilla softmax-based attention mechanism in high-resolution vision tasks without losing performance. On standard GPUs, ELFATT provided 4-7x speedups over vanilla attention without FlashAttention-2 and 2-3x speedups even with FlashAttention-2. For specific backbones like CSWin-T [73], ELFATT achieves the highest inference throughput compared to other attention mechanisms. Using CSWin-B ELFATT offers substantial speedups (e.g., 3.3x over vanilla attention with FlashAttention-2 at 384x384 resolution). Importantly, ELFATT also demonstrates efficiency on edge GPUs, offering 1.6x to 2.0x speedups compared to state-of-the-art attention mechanisms across various power modes (5W to 60W). Even compared to edge-optimized EfficientViT-B2, ELFATT is significantly faster in mixed precision on edge GPUs, particularly in high-power modes. This indicates that ELFATT effectively addresses the computational burden of attention, making ViTs more practical for speed-critical applications and resource-constrained edge devices.

In [74] authors Latibari et. al. has introduced a novel energy-aware, dynamically prunable ViT architecture called Incremental Resolution Enhancing Transformer (IRET). A core aspect of IRET is its dynamic pruning strategy. It leverages an attention-guided process: after the initial layers using a small embedding, a reliable attention matrix is formed. This matrix is then used to guide the sampling of additional information, but only for important tokens using a learnable 2D lifting scheme. Simultaneously, IRET drops the tokens receiving low attention scores. As the model "pays more attention" to important tokens, their focus and resolution increase incrementally. This process of attention-guided sampling and dropping unattended tokens allows IRET to significantly prune its computation tree on demand. The IRET architecture allows for a dynamic trade-off between complexity (and energy) and accuracy by controlling the threshold for dropping unattended tokens. A higher threshold drops more tokens, reducing computation but potentially decreasing accuracy. Furthermore, IRET is designed with early exiting capabilities, integrating classification heads to facilitate real-time predictions. This enables anytime prediction, where accuracy and complexity can be dynamically adjusted based on factors like battery life or reliability on edge devices. The dynamic nature of token dropping and the ability to enhance focus on attended tokens mean IRET can achieve a graceful degradation in accuracy when reducing complexity compared to prior art. The authors note that this approach modulates computational complexity via attention threshold adjustments, rather than changes in embedding size or architecture, making it suitable for applications balancing accuracy, energy efficiency, and latency.

Rohit Prasad has introduced [75] an ultra-low-power Coarse-Grained Reconfigurable Array (CGRA) architecture specifically designed to accelerate Transformer models for deployment on low-power edge devices. The proposed CGRA architecture focused on accelerating General Matrix Multiplication (GEMM) operations, which were identified as a core bottleneck in Transformer performance. The design integrated a 4x4 array of Processing Elements (PEs) for efficient parallel computation. To

optimize data handling and reduce memory bandwidth demands, the architecture included dedicated 4x2 Memory Operation Blocks (MOBs) for LOAD/STORE operations. This design enhanced data reuse by keeping data within the PE array as long as possible, reducing costly external memory accesses. Reduced memory bandwidth requirements are particularly valuable for edge devices where this resource is often limited.

Nixon et. al. have explored techniques for reducing the computational budget of deep networks, focusing on Early Exits (EEs) as a method within the group of Dynamic Inference (DI) techniques [76]. DI aims to modulate the complexity of a machine learning model based on constraints like compute budget or energy consumption, or to reduce complexity where it offers little benefit. The ultimate goal was to achieve a result similar to the final output of the full network using only a subset of its layers. While this generally leads to a reduction in accuracy, it comes with a worthwhile reduction in computational cost. The paper added Early Exits to a Vision Transformer (ViT) model. It proposed an exit criterion based on a closed set gallery, which can be computed ad-hoc using only the model's outputs. By adding these Early Exits to a ViT, combined with the defined exit criterion, the model can reduce its compute budget during inference. The core mechanism of EEs is that computation can stop at an earlier layer (or "exit") if a certain confidence or criterion is met, avoiding the computation of subsequent layers. The effectiveness of this approach was demonstrated through results, showing a 18.2% reduction in compute performance. This reduction is achieved at the cost of only a small loss in performance, specifically a 1% loss in TMr@FMr0.1% (a metric related to matching accuracy in face recognition). This clearly illustrates the compute benefit obtained by using Early Exits: significant computational savings are possible with only a minor impact on the model's task performance. The paper also investigated the relationship between EEs and model bias.

Liu et. al. have presented MatrixFlow [77], a novel co-designed system-accelerator architecture aimed at efficiently accelerating Transformers, which are central to recent AI advancements but computationally demanding.
MatrixFlow addressed these limitations through a co-designed approach involving both hardware and software. The system is based on a loosely coupled systolic array. A key innovation is a novel dataflow-based matrix multiplication technique co-optimized with a new software mapping approach. This technique is designed to reduce memory overhead. The system-level co-optimization involves designing a novel data structure and algorithm to optimize data sizes and arrangements, ensuring efficient storage and minimal data traffic between memory and computation units. This co-design significantly reduces instruction overhead, enhancing throughput and scalability. The dataflow-based approach, converting conventional matrix multiplication into processing page-sized blocks, frees the CPU from intensive instruction and data handling, ensuring continuous data feeding into the accelerator, maximizing throughput, and minimizing latency. Experiments benchmarking matrix multiplication and Transformer execution demonstrate substantial performance improvements. MatrixFlow achieves up to a 22x speedup compared to a many-core CPU system. Crucially, it outperforms closest state-of-the-art loosely-coupled and tightly-coupled accelerators by over 5x and 8x respectively. This highlights MatrixFlow's effectiveness in reducing the dominant data movement bottleneck and accelerating the critical GEMM operations in Transformers. In 2023, Zhu et. al have proposed a vision Transformer with Bi-Level Routing Attention (BRA) [78], a dynamic and query-aware sparse attention mechanism. BRA improves efficiency by dynamically identifying and attending to only the most semantically relevant key-value pairs for each query. This is achieved through a two-step process: first, a coarse region-level routing filters out irrelevant regions by building and pruning a region-level affinity graph. Second, fine-grained token-to-token attention is applied exclusively within the remaining "routed regions". The implementation is designed for efficiency on modern GPUs, utilizing GPU-friendly dense matrix multiplications after gathering relevant key-value tokens, which avoids inefficient sparse matrix operations. This dynamic sparsity allows BRA to achieve a much lower computational complexity of O((HW)^4/3) when using a proper region partition size, significantly reducing the computational cost compared to O((HW)^2) vanilla attention.

For object detection, BiFormer, built using BRA, is evaluated on the COCO 2017 dataset. BiFormer-S and BiFormer-B show competitive overall performance, but critically, they demonstrate a significant advantage in detecting small objects. This is attributed to BRA's sparse sampling technique which preserves fine-grained details important for small objects, unlike methods that rely on downsampling. By focusing computations on relevant areas and avoiding processing irrelevant ones, BRA reduces the overall computational load while maintaining or improving performance on tasks like small object detection, leading to more efficient inference. Although some overheads exist (GPU kernel launches, memory transactions) affecting throughput, the core mechanism offers a better computation-performance trade-off.

Zhang et. al. have introduced CAS-ViT (Convolutional Additive Self-attention Vision Transformers) [79], a new family of lightweight networks designed to achieve a better balance between efficiency and performance for mobile applications. The core idea is to recognize that token mixers derive global context from interactions in both spatial and channel domains. Based on this, they propose the Convolutional Additive Token Mixer (CATM), which uses underlying spatial and channel attention as novel interaction forms. A key innovation is that CATM eliminated computationally expensive operations like matrix multiplication and Softmax found in standard self-attention, replacing them with an easy-to-implement additive similarity function and Sigmoid-activated attention. They construct a hybrid architecture with CAS blocks that employ CATM. This architecture has designed to be easy-to-deploy. The effectiveness of CAS-ViT has been evaluated across various vision tasks, demonstrating strong performance with significantly reduced computational overhead. In object detection experiments using RetinaNet, CAS-ViT models achieve competitive accuracy (AP) while having fewer parameters and GFLOPs than comparable models. For instance, CAS-ViT-S achieves an AP of 38.6 with 25M parameters and 189 GFLOPs, outperforming ResNet-50 (36.3 AP, 44M

parameters, 260 GFLOPs) and competing well with PVT-T (36.7 AP, 33M parameters, 240 GFLOPs). The paper emphasizes that CAS-ViT models maintain a competitive edge in computational efficiency across all platforms, achieving high performance metrics with lower computational overheads, thus facilitating efficient inference and easy deployment on mobile devices.

A different approach called LaViT, is proposed by Zhang et. al. [80]. They have proposed a model that seeks to address these issues by rethinking the self-attention mechanism. Instead of relying solely on token reduction or pruning, which can lead to information loss, LaViT aims to diminish both computational complexity and attention saturation. Attention saturation, where the attention matrix shows limited variation across layers, can hinder the network's ability to learn and potentially reduce training stability. LaViT integrates decay weights of the retentive mechanism and designs a novel loss function to preserve the diagonality of attention matrices, contributing to resolving attention saturation. The model incorporates attention downsampling across stages, which helps in preserving crucial semantic information from earlier layers while enabling the transmission of global contextual information through alternative paths. This downsampling strategy, along with a "Less-Attention" mechanism, helps in reducing computational cost. The authors specifically highlight that LaViT's resource-efficient Less-Attention mechanism makes it a suitable choice for implementation on mobile platforms. The careful design of LaViT's self-attention mechanism is central to its ability to reduce the computations and the problem of attention saturation, leading to improved efficiency and performance.

Dong et. al. introduced SpeedDETR [81], a novel speed-aware transformer designed for end-to-end object detection that explicitly optimizes for real-world latency on multi-core processors. A primary contribution was a latency prediction model capable of accurately estimating network latency by considering network properties, hardware memory access patterns, and the degree of parallelism. This model was versatile and compatible with general GPU architectures. Guided by this prediction model and an effective visual modeling approach, they design a hardware-oriented architecture for SpeedDETR. Key architectural elements include an efficient embedding module, a hardware-oriented backbone for feature capturing, and a strong transformer encoder (TSP) to ensure performance without relying on training-inefficient decoders. The authors also incorporate and investigate fusion techniques for model implementation to improve efficiency. These include fusing Batch Normalization into linear layers, which achieved a 13%∼15% speedup with minimal accuracy loss. Merging multiple branches in reparameterized CNNs is another technique used to reduce data movement costs and improve computation balance across processing elements. SpeedDETR employs a speed-aware model slimming strategy guided by the latency prediction model to evolve the network architecture iteratively, reducing latency while minimizing accuracy drop. Experimental results on the MS COCO dataset demonstrate SpeedDETR's effectiveness, showing significantly faster inference speeds (1.09×∼3.6× speedup) with higher accuracy (1.5%∼9.2% AP) compared to other DETR-based methods on a Tesla V100 GPU. The approach also achieved acceptable inference speeds on resource-limited edge GPUs like NVIDIA JETSON TX2 and NANO, highlighting its practicality for deployment.

Wang et. al. proposed method that include a CNN-based auxiliary detection head and a transformer-based one-to-many dense supervision branch [82]. This suggests improvements to the RT-DETR [83] framework by incorporating hierarchical dense supervision branches to accelerate training. The CNN auxiliary branch integrates encoder output features and employs a one-to-many label assignment strategy during training to strengthen encoder supervision. The transformer-based branch similarly uses a parameter-shared one-to-many matching approach. These dense supervision branches are designed to be active only during the training phase; therefore, they do not introduce any additional latency during inference, allowing the improved model (RT-DETRv3) to maintain the same inference speed as the original RT-DETR. The study uses loss functions like VFL and DFL in the auxiliary branches to guide training. In addition to dense supervision, parameter-shared multi-group self-attention perturbation branches are added to the transformer decoder structure as training-only components. Extensive experiments using various backbones and training schedules validate the effectiveness of these methods. Compared to RT-DETR and RT-DETRv2, RT-DETRv3 achieves improved performance (AP) and significantly speeds up convergence. A key finding is that RT-DETRv3 requires substantially fewer training epochs (as little as 60% or less) to reach performance comparable to CNN-based real-time detectors, demonstrating significant training acceleration. Ablation studies confirm that adding the CNN-based and transformer-based one-to-many label assignment branches and the multi-group self-attention perturbation modules positively impact performance. The number of self-attention perturbation branches was also analyzed, with three branches providing optimal performance.

6. Datasets Used For Experimental Work

Throughout the literature which has been explored for this review paper, various object detection datasets were used from diverse application domains. These datasets have been summarized in Table 3.

Table 3. Datasets Used in Literature for Object Detection in Diverse Application Domains

Sr. NO.	Dataset	Domain/ Application Area	Description	Image Resolution	No of Classes	No of Images/ Videos
1	MS COCO 2017	General Object Detection	Rich object detection and segmentation dataset with complex scenes.	Variable (~640×480)	80	118K train, 5K val
2	PASCALVOC 2007	General Object Detection	Benchmark dataset for 20 object classes.	Variable (~500×375)	20	9,963 images
3	PASCALVOC 2012	General Object Detection	Extension of VOC 2007 with more annotations.	Variable	20	11,540 images
4	ImageNet VID	Video Object Detection	Video sequences with per-frame object annotations.	Variable	30	1M+ frames
5	Crowdhuman	Dense Pedestrian Detection	Dense pedestrian dataset with occlusion scenarios.	Variable (~640×480)	1	15K train, 4K val
6	nuScenes	Autonomous Driving	Multisensor dataset including images, LiDAR, radar, GPS, and IMU.	1600×900	23	1.4M images, 1000 scenes
7	Waymo	Autonomous Driving	High-resolution sensor suite data including LiDAR and camera.	1920×1280	4	12M images, 1,950 sequences
8	KITTI	Autonomous Driving	Annotated street scenes from car-mounted sensors.	1242×375	8	15K images
9	Cityperson	Pedestrian Detection	Built on Cityscapes with a focus on pedestrian annotations.	2048×1024	1	5K images
10	AI-TOD	Tiny Object Detection (Arial)	Tiny objects in aerial imagery, with extreme class imbalance	Variable	8	70K+ images
11	VisDrone2021-DET	Arial Object detection	Drone-captured images for small and dense object detection.	Variable (~960×540)	10	10K images

7. Limitations of Transformer Based Existing Models

DETR-based methods lack the ability to model strong local dependencies, leading to poor performance in capturing detailed information. When vision transformers are used for object detection, self attention plays very important role. Designing a self-attention mechanism itself is crucial task. It becomes more complex when image resolution is high, resulting in high computational cost. Large memory and high-performance processing are required in this case. Some local attention mechanisms reduce complexity and memory cost but are inefficient due to overlapped or restricted window size. Some multi-scale visual methods explored are unsuitable for use as a general-purpose backbone. Introducing multi-scale features into the decoder of DETR is a challenge.

In case of frequent overlapping and occlusions, performance of detector degrades. When these models are deployed on energy constrained devices, model compression through quantization may result in loss of accuracy. The work recognizes the challenges of environmental variability and the need for diverse training datasets. It is also observed that some lightweight strategies sacrificing richness of feature representations. It is observed that, for incremental object detection, knowledge distillation is used and sometimes teacher model gives negative prediction, i.e. it does not learn new object and detects background. Student model mimics the same from teacher model and shows no object i.e. background. It gets biased and thinking that either everything is background or old object class without learning any new object class. It creates excessive imbalance of negative predictions. In case of blurred and small object detection detecting weak aggregated objects in UAV images and small object features in large receptive field networks is a challenge. Some challenges remain in improving convergence speed and small object detection. The low quality of initial prediction boxes makes small objects difficult to learn. A portion of training iterations is spent addressing the information bias introduced during the decoding stage, resulting in slower convergence speeds.

The Computational complexity in some recent models poses challenges for real-time applications like surveillance and autonomous driving. In videos with rapid or complex motion, temporal references are particularly challenging to model. Video object detection introduces additional hyperparameters that must be meticulously adjusted for optimal performance, posing a limitation. Also, lack of standardized benchmarks for lightweight ViT models and limited studies on optimization in multimodal object detection are observed.

8. Emerging Trends And Future Directions

Literature shows that, the growing need of object detection in rear world scenarios not just require improved accuracy but it must also run on different types of hardware depending on application requirement. Hence newer research is focusing on approaches where model can run and adapt depending on how complex the input is or what kind of device it is running on. Two major developments are using dynamic vision transformers and hardware aware neural architecture search.

Dynamic ViTs (adaptive token pruning at runtime) [84]: In dynamic vision transformers Euclidean distance between token features with the graph theory algorithm is combined for accurate grouping of tokens. Dynamic vision transformers with adaptive token pruning at runtime offer substantial computational savings, adapt to input complexity, maintain or even improve accuracy and are well suited for deployment on resource constrained devices.

Hardware-aware NAS (neural architecture search) [85, 86]: Hardware neural architecture search enables automatic discovery of ViT architecture that are better suited for hardware specific platform and balances high detection accuracy with low latency, memory usage and energy consumption. NAS reduces the need for manual tuning and human intervention. By integrating hardware constraints into the search process, HW-NAS delivers ViT models that are both high-performing and deployable in real-world scenarios.

9. Conclusion

This review has explored recent advancements in Vision Transformer (ViT) architectures for object detection, focusing on training and inference time optimization strategies. Vision Transformers have demonstrated impressive capabilities in object detection tasks, often surpassing traditional CNN-based models. However, their computational demands and memory requirements pose challenges for real-time applications and deployment on resource-constrained devices.

Training time optimization strategies have focused on improving model efficiency and performance through architectural innovations, novel attention mechanisms, and enhanced learning techniques. Key developments include incremental learning approaches to combat catastrophic forgetting, dynamic head switching for balancing accuracy and efficiency, and adaptive feature extraction methods to enhance multi-scale feature processing.

Inference time optimization strategies have addressed the challenges of deploying ViTs in real-time and edge computing scenarios. Techniques such as efficient attention mechanisms, model compression, quantization, and hardware-aware optimizations have been proposed to reduce latency, energy consumption, and computational costs while maintaining competitive accuracy.

Despite these advancements, several challenges remain. These include the need for further improvements in modeling local dependencies, designing efficient self-attention mechanisms for high-resolution images, and addressing performance degradation in complex scenarios with frequent occlusions. Additionally, balancing model compression for edge deployment with maintaining accuracy remains an ongoing challenge. Future research directions may include developing more efficient and adaptable ViT architectures, exploring novel attention mechanisms that balance global and local feature capture, and investigating hardware-software co-design approaches for optimized deployment. There is also potential for further exploration of multi-modal fusion techniques and application-specific optimizations.

In conclusion, while Vision Transformers have shown great promise in object detection tasks, continued research is needed to address their limitations and fully realize their potential in real-world applications. The field is rapidly evolving, with a clear trend towards more efficient, scalable, and deployable models on edge devices that can meet the demands of diverse application scenarios.

References

[1] Dalal N, Triggs B (2005). Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1), 886–893.

[2] Viola P, Jones M (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of IEEE computer society conference on computer vision and pattern recognition, 511–518.

[3] P. Felzenszwalb, D. McAllester, and D. Ramanan (2008). A discriminatively trained, multiscale, deformable part model. In CVPR. IEEE, 1–8.

[4] R. Girshick, J. Donahue, T. Darrell and J. Malik (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition. 580-587.

[5] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders (2013). Selective search for object recognition. International journal of computer vision, 104(2), 154–171.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303–338.

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98–136.

[8] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fer gus, and Y. LeCun (2013). Overfeat: Integrated recogni tion, localization and detection using convolutional net works. arXiv preprint arXiv:1312.6229

[9] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun (2014). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. International Conference on Learning Representations.

[10] R. Girshick (2015). Fast R-CNN. IEEE International Conference on Computer Vision (ICCV), Santiago, 1440-1448.

[11] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28.

[12] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, 2980-2988.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 779–788.

[14] J. Redmon and A. Farhadi (2017). Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7263–7271.

[15] M. Lin, Q. Chen, and S. Yan (2013). Network in network. arXiv preprint arXiv:1312.4400.

[16] J. Redmon and A. Farhadi (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[17] B. Bhavya Sree, V. Yashwanth Bharadwaj, and N. Neelima (2021). An inter-comparative survey on state-of-the-art detectors—r-cnn, yolo, and ssd. Intelligent Manufacturing and Energy Sustainability: Proceedings of ICIMES, 475–483.

[18] K. He, X. Zhang, S. Ren, and J. Sun (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9),1904–1916.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 8759–8768.

[21] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie (2022). Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

[22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

[23] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao (2021). You only learn one representation: Unified network for multiple tasks. arXiv preprint arXiv:2105.04206.

[24] G. Jocher, A. Chaurasia, and J. Qiu (2023). YOLO by Ultralytics. https://github.com/ultralytics/ ultralytics, 2023. Accessed: February 30, 2023. (repeated as in ref no. 21)

[25] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding (2020). Pp-yolo: An effective and efficient implementation of object detector. arXiv preprint arXiv:2007.12099.

[26] X. Huang, X. Wang, W. Lv, X. Bai, X. Long, K. Deng, Q. Dang, S. Han, Q. Liu, X. Hu (2021). Pp-yolov2: A practical object detector. arXiv preprint arXiv:2104.10419.

[27] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du (2022). Pp-yoloe: An evolved version of yolo. arXiv preprint arXiv:2203.16250.

[28] Wang, Chien-Yao, I-Hau Yeh, and Hong-Yuan Mark Liao (2024). Yolov9: Learning what you want to learn using programmable gradient information. In European conference on computer vision, 1-21.

[29] Wang, Ao, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, and Jungong Han (2024). Yolov10: Real-time end-to-end object detection. Advances in Neural Information Processing Systems, 37, 107984-108011.

[30] Khanam, Rahima, and Muhammad Hussain (2024). Yolov11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725.

[31] Tian, Yunjie, Qixiang Ye, and David Doermann (2025). Yolov12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524.

[32] Khanam, Rahima, and Muhammad Hussain (2025). A Review of YOLOv12: Attention-Based Enhancements vs. Previous Versions. arXiv preprint arXiv:2504.11995.

[33]Vaswani Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). Attention is all you need. Advances in neural information processing systems, 30.

[34] D. Bahdanau, K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. International Conference on Learning Representations,1-15.

[36] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representation.

[36] Carion Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov and Sergey Zagoruyko (2020). End-to-End Object Detection with Transformers. ArXiv abs/2005.12872.

[37] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He (2017). Aggregated residual trans formations for deep neural networks. Computer Vision and Pattern Recognition.

[38] Zhu X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable DETR: Deformable Transformers for End-to-End Object Detection. In International Conference on Learning Representations.

[39] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. (2017). Deformable convolutional networks. In International Cobference on Computer Vision.

[40] Zachary Teed and Jia Deng (2020). Raft: Recurrent all-pairs field transforms for optical flow. In Europian Conference on Computer Vision.

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick (2014). Microsoft coco: Common objects in context. In Europian Conference on Computer Vision.

[43] Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision, 10012-10022.

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015). U net: Convolutional networks for biomedical image segmen tation.” International Conference on Medical image com puting and computer-assisted intervention, 234–241.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing sys tems, 1097–1105.

[45] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fi dler, Adela Barriuso, and Antonio Torralba (2018). Semantic understanding of scenes through the ade20k dataset. International Journal on Computer Vision.

[47] Wang, Wenhai, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF international conference on computer vision, 568-578.

[47]Li Yanghao, Hanzi Mao, Ross Girshick, and Kaiming He (2022). Exploring plain vision transformer backbones for object detection." In European conference on computer vision, 280-296.

[49] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel (1989). Backpropagation applied to handwritten zip code recognition. Neural computation.

[49] Tsung-Yi Lin, Piotr Doll´ ar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie (2017). Feature pyramid networks for object detection. In Computer Vision and Pattern Recognition.

[50] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll´ar, and Ross Girshick (2021). Masked autoencoders are scalable vision learners. arXiv:2111.06377.

[51] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). ImageNet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition.

[52] Fan, Haoqi, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. "Multiscale vision transformers (2021). In Proceedings of the IEEE/CVF international conference on computer vision, 6824-6835.

[53] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman (2018). A short note about Kinetics 600. arXiv:1808.01340.

[54] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michal ski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag (2017). The “Something Something" video database for learning and evaluating visual common sense. In International Conference on Computer Vision.

[55] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In Europian Conference on Computer Vision.

[56] Chunhui Gu, Chen Sun, David A. Ross, Carl Von drick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijaya narasimhan, George Toderici, Susanna Ricco, Rahul Suk thankar, Cordelia Schmid, and Jitendra Malik (2018). AVA: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of Computer Vision and Pattern Recognition.

[57] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jiten dra Malik, and Christoph Feichtenhofer (2021). MViTv2: Improved multiscale Vision Transformers for classification and detection. arXiv:2112.01526.

[58] Caron Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision, 9650-9660.

[59] Liu Yaoyao, Bernt Schiele, Andrea Vedaldi, and Christian Rupprecht (2023). Continual detection transformer for incremental object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 23799-23808.

[60] Zhang, Jichuan, Wei Li, Shuang Cheng, Yali Li, and Shengjin Wang (2025). Dynamic Object Queries for Transformer-based Incremental Object Detection. In ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing, 1-5.

[61] Hang Chen, Chufeng Tang, Xiaolin Hu (2024). DHS-DETR: Efficient DETRs with dynamic head switching. Computer Vision and Image Understanding, 248, 104-106.

[62] Ke Li, Di Wang, Gang Liu, Wenxuan Zhu, Haodi Zhong, Quan Wang (2024). DiagSWin: A multi-scale vision transformer with diagonal-shaped windows for object detection and segmentation. Neural Networks, 180(106653).

[63] Tianxiao Zhang, Wenju Xu, Bo Luo, Guanghui Wang (2025). Depth-Wise Convolutions in Vision Transformers for efficient training on small datasets. Neurocomputing, 617(128998).

[64] Zhenyi Zhang, Jianlong Zhang, Juan Wei, Lina Han, Tianping Li (2025). AIFE-Net: Transformer-based adaptive multi-scale feature extraction encoder for object detection. Neurocomputing, 640, (130366), https://doi.org/10.1016/j.neucom.2025.130366.

[65] Junjie Li, Si Guo, Shi Yi, Runhua He, Yong Jia (2025), DMCTDet: A density map-guided composite transformer network for object detection of UAV images. Signal Processing: Image Communication,136(117284). https://doi.org/10.1016/j.image.2025.117284.

[66] Sathishkumar Moorthy, Sachin Sakthi K.S., Sathiyamoorthi Arthanari, Jae Hoon Jeong, Young Hoon Joo (2025). Hybrid multi-attention transformer for robust video object detection. Engineering Applications of Artificial Intelligence, 139, Part B, 109606.

[67] XueZhuan Zhao, JiaWei Wang, LingLing Li, XiaoYan Shao, KeXin Zhang (2025). A unified solution for replacing position embedding in Vision Transformer for object detection. Engineering Applications of Artificial Intelligence, 152, 110679.

[68] Divya Nimma, Omaia Al-Omari, Rahul Pradhan, Zoirov Ulmas, R.V.V. Krishna, Ts. Yousef A.Baker El-Ebiary, Vuda Sreenivasa Rao (2025). Object detection in real-time video surveillance using attention based transformer-YOLOv8 model. Alexandria Engineering Journal, 118, 482-495.

[69] Fanglin Liu, Qinghe Zheng, Xinyu Tian, Feng Shu, Weiwei Jiang, Miaohui Wang, Abdussalam Elhanashi, Sergio Saponara (2025). Rethinking the multi-scale feature hierarchy in object detection transformer (DETR). Applied Soft Computing, 175, 113081.

[70] Liu, Xinyu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan (2023). Efficientvit: Memory efficient vision transformer with cascaded group attention. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 14420-14430.

[71] Katare, D., Leroux, S., Janssen, M., & Ding, A. Y. (2025). Approximating vision transformers for edge: variational inference and mixed-precision for multi-modal data. Computing, 107(3), 71.

[72] Wu, C., Che, M., Xu, R., Ran, Z., & Yan, H. (2025). ELFATT: Efficient linear fast attention for vision transformers. arXiv preprint arXiv:2501.06098.

[73] Dong, Xiaoyi, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo (2022). Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 12124-12134.

[74] B. S. Latibari, H. Homayoun and A. Sasan (2025). Optimizing Vision Transformers: Unveiling 'Focus and Forget' for Enhanced Computational Efficiency. IEEE Access, 13, 27908-27927.

[75] Rohit Prasad (2025). An ultra-low-power CGRA for accelerating Transformers at the edge. hal 04914400.

[76] Nixon, S., Ruiu, P., Cadoni, M., Lagorio, A., & Tistarelli, M. (2025). Assessing bias and computational efficiency in vision transformers using early exits. EURASIP Journal on Image and Video Processing, 2(1).

[77] Liu, Qunyou, Marina Zapater, and David Atienza (2025). MatrixFlow: System-Accelerator co-design for high-performance transformer applications. arXiv preprint arXiv:2503.05290.

[78] Zhu, Lei, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson WH Lau (2023). Biformer: Vision transformer with bi-level routing attention. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 10323-10333.

[79] Zhang, Tianfang, Lei Li, Yang Zhou, Wentao Liu, Chen Qian, Jenq-Neng Hwang, and Xiangyang Ji (2024). Cas-vit: Convolutional additive self-attention vision transformers for efficient mobile applications. arXiv preprint arXiv:2408.03703.

[80] S. Zhang, H. Liu, S. Lin and K. He (2024), You Only Need Less Attention at Each Stage in Vision Transformers. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6057-6066.

[81] Peiyan Dong, Zhenglun Kong, Xin Meng, Peng Zhang, Hao Tang, Yanzhi Wang, and Chih-Hsien Chou (2023). SpeedDETR: speed-aware transformers for end-to-end object detection. In Proceedings of the 40th International Conference on Machine Learning, 202(328), 8227–8243.

[82] Wang, Shuo, Chunlong Xia, Feng Lv, and Yifeng Shi (2025). RT-DETRv3: Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 1628-1636.

[83] Zhao, Yian, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen (2024). Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 16965-16974.

[84] Yu Yang, Xiaofang Hu and Yue Zhou (2024). Adaptive Token Pruning for Vision Transformer. 5th International Symposium on Computer Engineering and Intelligent Communications, 586-589.

[85] Hadjer Benmeziane, K. E. Maghraoui, Hamza Ouarnoughi, S. Niar, Martin Wistuba and Naigang Wang (2021). A Comprehensive Survey on Hardware-Aware Neural Architecture Search. ArXiv, abs/2101.09336.

[86] Zhengang Li, Alec Lu, Yanyue Xie, Zhenglun Kong, Mengshu Sun, Hao Tang, Zhong Jia Xue, Peiyan Dong, Caiwen Ding, Yanzhi Wang, Xue Lin and Zhenman Fang (2024). Quasar-ViT: Hardware-Oriented Quantization-Aware Architecture Search for Vision Transformers. Proceedings of the 38th ACM International Conference on Supercomputing.