Bridging the Gap: A Fusion of CNN and Transformer Models for Real-Time Object Detection

Yuanke Pan
College of Big Data and Internet
Shenzhen Technology University,
Shenzhen, 518118, China
dreamice@outlook.com

Chengmin Zhou
College of Big Data and Internet
Shenzhen Technology University,
Shenzhen, 518118, China
chuengminchou@gmail.com

Liyilei Su
College of Big Data and Internet
Shenzhen Technology University
Shenzhen, 518118, China
suliyilei@sztu.edu.cn

Haseeb Hassan*
College of Big Data and Internet
Shenzhen Technology University,
Shenzhen, 518118, China
haseeb@sztu.edu.cn

Bingding Huang*
College of Big Data and Internet,
Shenzhen Technology University
Shenzhen, China
huangbingding@sztu.edu.cn

2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC) | 979-8-3503-3366-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITAIC58329.2023.10408921

***Abstract*—Object detection is the main task in computer vision. Recently, object detection tasks have been performed through convolutional neural networks and the YOLO family, and gained substantial attention from the research community. Likewise, transformer-based models were introduced to improve the efficiency and accuracy of many detection models. However, real-time object detection still suffers from slow speed. To address this, a novel approach is proposed by fusing CNN with transformer-based detection. This fusion process positively impacts the accuracy and inference speed. In our experiments, we achieved a notable 51.8 mAP, which represents a 0.9% improvement. Note that it is performed by 36 million parameters, which is 5 million parameters fewer than transformer-based models. A 144 GFLOPS computation rate was measured for 800x1333 pixels (an excellent improvement over 640x640 pixels at 74 GFLOPS. All this is achieved through 40 epochs training schedule. In summary, the proposed model reduced millions of parameters and the computational cost of the model by restructuring the original detection-based architecture. Moreover, it significantly enhances inference speed compared to prior transformer-based models.**

***Index Terms*—End-to-end Object Detection; CNN (Convolutional Neural Network); Real-Time Object Detection; Transformer Models; Inference Speed**

I. Introduction

Within computer vision, object detection algorithms constitute a fundamental task primarily employed to identify the spatial coordinates of objects within an image and assign them to corresponding labels. In the recent past, an extensive amount of convolutional neural networks (CNNs) have been introduced to address this task.

Among the various CNN detection models, the Region Convolutional Neural Network (R-CNN) [1] stands out. It operates as a two-stage network, wherein regions identified through the selective search algorithm are initially selected. Subsequently, the AlexNet network [2] is employed for feature extraction, and a support vector machine is utilized for label assignment to these distinct regions. The requirement to process each region individually using a convolutional network results in a reduction in inference speed. In response to this limitation, the Fast R-CNN was developed. It processes the entire image using a convolutional neural network, followed by the projection of features to different regions using the Region of Interest (ROI) method. Building upon the foundation laid by Fast R-CNN [3], the authors proposed a Faster R-CNN R-CNN [4]. Notably, this model introduces the Region Proposal Network (RPN) as a replacement for the conventional candidate region generation method. The RPN takes charge of generating candidate regions, also called anchors, which are then utilized as the target regions for the earlier ROI method. These generated candidate regions play a pivotal role in determining category tags and anchor offset information. Despite the possibility of training Faster R-CNN as a holistic network, the inference process remains a two-stage classifier responsible for candidate region generation and subsequent classification based on these regions.

In recent times, pioneering advancements in one-stage target detection have been exemplified by the introduction of You Only Look Once (YOLO) [5] and the Single Shot MultiBox Detector (SSD) [6]. YOLO operates by partitioning the image into N cells. When the center point of a detected target aligns with the corresponding cell, that cell takes responsibility for identifying the associated target, eliminating the need for candidate region generation. Subsequent iterations of YOLO have incorporated predefined anchors and refined models to enhance both inference speed and accuracy. In contrast to YOLO, which maintains spatial density across the image, SSMD strategically shifts spatial density towards scaling considerations. This is achieved by introducing features at various scales, allowing for the location and classification of objects across multiple scales through predefined anchors and a final classifier.

Numerous convolutional neural network (CNN)-based models have been devised for object detection tasks. The inherent

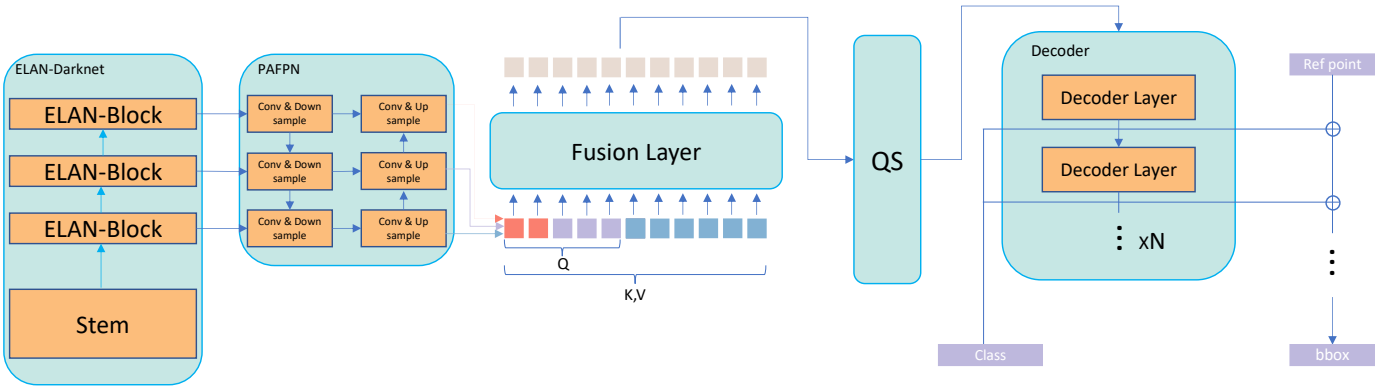


Fig. 1. **Proposed Model Architecture.** The initial stage involves the image's passage through the CNN's backbone, illustrated as "ELAN-Darknet." The outputs from the final three layers are then channeled into the PAFPN as essential features. Preceding their entry into the transformer segment, each layer's features undergo a flattening operation and are subsequently consolidated into a unified feature set open to processing by the transformer. Following a single layer of fusion processing, all components undergo a query selection step, which serves to optimize computational efficiency within the decoder section.

nature of their output necessitates specific post-processing procedures to yield results suitable for target detection. Non-maximum suppression (NMS) stands out as one of the most common techniques employed for this purpose. NMS is an algorithm designed to select the most plausible outcome from a range of results. Prior to model training, it is imperative to establish an Intersection over Union (IOU) threshold for NMS. This threshold is typically derived from the annotated features of diverse datasets. The presence of NMS, however, poses constraints on the advancement of target detection models.

Later on, with the successful integration of transformer architecture [7], [8] into computer vision, a notable advancement has been the introduction of an NMS-Free algorithm known as DEtection TRansformer (DETR) [9]. DETR incorporates the transformer architecture into the target detection model through a bipartite map matching algorithm, transforming what previously required Non-Maximum Suppression (NMS), a many-to-one process, into a one-to-one algorithm. This allows the model to provide the target's precise location and label directly. However, it's important to note that the stability of the bipartite graph matching algorithm presents challenges, leading to slower convergence of the model and a reduction in detection accuracy.

Recently, numerous variants of the DETR architecture have emerged, each designed to enhance both the accuracy and the convergence rate for the specific target detection task. For instance, DAB-DETR [10] incorporates a dynamic anchor frame mechanism that comprehensively interprets and leverages the outcomes generated at each decoder step, improving overall accuracy. Meanwhile, DN-DETR [11] takes a novel approach by introducing denoising training to address the instability inherent in the bipartite matching algorithm. This innovation enhances model convergence and contributes significantly to the augmentation of detection accuracy. In a similar vein, conditional-DETR [12] introduces a conditional spatial query to confine the decoder's search scope to image objects dynamically. This restriction expedites the convergence process, further optimizing the algorithm's efficiency and enhancing its overall performance.

In addition to enhancing accuracy and convergence speed, a vital aspect of this research centers on mitigating the issue of excessive complexity within the transformer model algorithm. Deformable-DETR [13], for instance, leverages the deformable attention component instead of the conventional attention component, resulting in a noteworthy reduction in algorithmic intricacy. Similarly, Sparse-DETR [14] mirrors PnP-DETR [15] by employing a spatial query component instead of a spatial query. Furthermore, DETR and PnP-DETR introduce the score net and PnP (poll and pool) sampler to enhance the sparsity of attention, respectively. Lastly, Lite-DETR [16] advocates for the utilization of only a subset of the features as queries, maintaining accuracy levels similar to those of the original model while significantly diminishing the computational cost of the encoder.

Despite the commendable progress achieved in enhancing the speed and robustness of DETR, it is noteworthy that CNN-based detectors, with YOLO as their prime representative, continue to dominate the field of real-time object detection. While CNN-based networks have found extensive application in target detection, the persistence of Non-Maximum Suppression (NMS) constrains their further advancement. In recent years, transformer-based models have undergone considerable refinement. Nonetheless, their relatively slower inference speeds and larger model parameters still impose restrictions on their utility in real-time target detection scenarios. Therefore, in this paper, our main objective is twofold: first, to facilitate the limitations of DETR concerning the number of parameters and inference speed, and second, to alleviate the constraints posed by NMS in the context of YOLO.

In recent years, many architectures have integrated convolutional neural networks and transformer models. For instance, BoTNet [17], replaces the conventional convolutional kernel with a multi-head self-attention module within the ResNet bottleneck. This adaptation has resulted in notable improvements in the model's accuracy when applied to visual tasks. Similarly, ConVit [18] introduces the Gated Positional Self-

Attention (GPSA) mechanism, enabling the model to prioritize the convolutional or attention tool dynamically. This strategic choice significantly enhances the model's performance, particularly in classification tasks. Furthermore, Conformer [19] incorporates a transformer architecture as an additional parallel branch to augment the network's overall performance. Additionally, a study of model performance concerning convolutional network and transformer occupancy ratios, as observed in CoAtNet [20], indicates that the optimal model capacity and task accuracy are achieved when these ratios are approximately equal.

The research discussed above reveals that investigations into CNN-based object detection models primarily center on reducing model parameters and enhancing inference speed. In contrast, research concerning transformer-based models predominantly concentrates on enhancing encoder sparsity and stabilizing bipartite graph matching algorithms. Moreover, the fusion of these two architectural paradigms offers promising avenues for augmenting overall network performance. Drawing from these insights, we introduce a novel approach that entirely replaces the transformer's backbone with a CNN-based architecture. The outcome is a model that strikes a more balanced equilibrium between inference speed and accuracy.

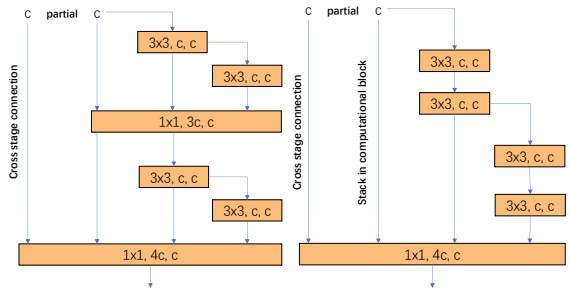
This paper introduces several contributions:

- Enhanced Network Adaptation: By substituting both the backbone and encoder components within the DINO framework, our work demonstrates the practicality of tuning the ratio between Convolutional Neural Networks (CNNs) and transformer networks. This adjustment is designed to enhance inference speed and reduce the total number of parameters.
- Addressing Transformer Detector Challenges: In response to challenges related to slow inference speed and excessive parameter count within the transformer-based detector, we replaced the DINO's backbone and encoder with a CNN architecture. This architectural modification resulted in a substantial 41.7% enhancement in inference speed and a notable 17% reduction in the number of parameters.
- Optimizing Label Loss: We introduced the varifocal loss function instead of the original focal loss for label loss computation to refine matching stability further. This change led to a significant +0.9 mAP improvement in the model's performance.

II. Method

A. Overview

The schematic representation of the model's main workflow is depicted in Figure 1. The process commences with the input image being propagated through a CNN architecture. Subsequently, the resultant feature vectors are directed to the transformer-based decoder. This operation yields a set of candidate bounding boxes. A filtering process based on bounding box confidence is executed to obtain the definitive bounding boxes and their respective class labels. It is worth noting that, in our approach, we have elected to utilize a CNN network as a substitution for the Backbone and Encoder components typically found in the DINO framework. As a result, the architecture encompasses elements of both CNN and transformer networks.



(a) CSPVoVNet [24] (b) ELAN

Fig. 2. Efficient layer aggregation networks.

B. CNN Part

Within the CNN segment of our architecture, we draw upon the Yolov7 design, incorporating CSPDarknet as our chosen backbone. This adaptation replaces the Cross Stage Partial (CSP) [21] mechanism with the Efficient Layer Aggregation (ELAN) [22] mechanism. The Path Aggregation Network with Feature Pyramid Network (PAFPN) [23] is at the model's helm, which also adopts the ELAN mechanism and synergizes it with the Repconv mechanism. PAFPN plays a pivotal role in transmitting the features acquired from the CNN component to the Decoder.

CSP is engineered with the primary objective of diversifying gradients, thus enhancing the model's parameter utilization. By doing so, it optimizes the training process. ELAN is introduced to address the challenge of gradient deterioration as models scale up. It accomplishes this by creating distinct gradient paths of varying lengths, mitigating the issue of convergence degradation during training. Repconv streamlines network efficiency during inference by consolidating multiple convolutional kernels into a single kernel. This optimization improves inference speed without compromising accuracy, as multiple convolutional kernels are still utilized during the training phase. The PAFPN is responsible for feature map fusion across various layers, leading to enhanced object detection accuracy. In the context of our model, the CNN network generates three distinct layers of features, which are subsequently passed to the transformer segment for further processing.

C. Transformer Part

We implement an encoder layer designed for this purpose to enhance the model's capacity to fuse features across varying levels effectively. Moreover, we introduce a learnable level embedding within this encoder segment to promote a more nuanced understanding of feature discriminations. Following the fusion of model layers, we employ the Query Selection mechanism under the DINO design principles to identify the

most probable queries. This strategic selection serves to mitigate computational demands within the subsequent decoder phase. Finally, within the final decoder segment, we apply a refinement approach to produce the ultimate bounding box outputs.

In the refinement process, we adopt a refinement operation from DAB-DETR. This involves taking the features generated by each decoder layer and directing them to the bounding box projection layer, which calculates the bounding box offsets. This iterative correction of bounding box coordinates consistently improves the final results, enhancing the accuracy of object localization.

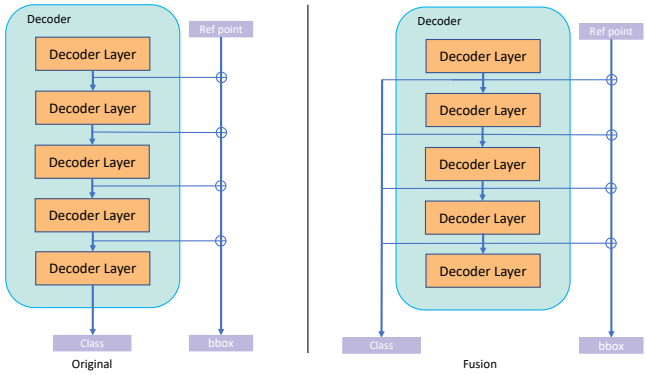


Fig. 3. **Late Fusion**. DINO uses only final features for labels (left), while our model utilizes multiple layers for better results (right).

D. Additional Enhancements

Varifocal Loss: In query selection, we incorporate a dense layer after the process to produce class-label embeddings for predicting encoder-generated feature outputs. Subsequently, we employ the maximum likelihood of these outputs to ascertain query scores for filtering. Note that this filtering mechanism presents a challenge, as the class labels are trained using focal loss, which may not introduce sufficient information about the bounding box when filtering the queries.The mathematical representation is provided as:

$$FL(p,y) = \begin{cases} -\alpha(1-p)^{\gamma}\log(p), & if y = 1 \\ -(1-\alpha)p^{\gamma}\log(1-p), & otherwise \end{cases} \quad (1)$$

To tackle this challenge, we replaced the local loss with varifocal loss [25] in the final loss computation stage. Varifocal loss uses Iou-aware Classification Scores (IACS) to calculate the loss when targeting positive classes. For positively classified instances, this loss is activated, while well-classified negative instances still employ the local loss, attenuated by the gamma parameter.

$$VFL(p,q) = \begin{cases} -q(q\log(p) + (1-q)\log(1-p)), & q > 0 \\ -\alpha p^{\gamma}\log(1-p), & q = 0 \end{cases} \quad (2)$$

Late Fusion: The previous approach used the last decoder layer to generate corresponding class embeddings. However, it led to concept drifting in the decoder, affecting data stability.

TABLE I
COMPARASION

Model	YOLOv7	DINO	Ours
Epochs	300	40	40
mAP	50.9	45.7	49.6
AP$_S$	34.7	23.8	32.6
AP$_M$	55.7	49.3	53.9
AP$_L$	66.3	63.2	65
GFLOPS	104	119	**74**
Parameters	36	41	**36**
Infer speed(FPS)	26	16	22

Therefore, a late fusion strategy is implemented to enhance the alignment between model labels post-decoding and query selection. This strategy leverages features from various decoder layers to boost the reliability of the final features. Thus, two fusion methods are explored: (1) average fusion, which computes the mean of features from different decoder layers, and (2) concatenate fusion, which stacks the features from different layers. Ultimately, an average fusion method is adopted due to its superior performance and prediction accuracy in our model.

III. Experiments and Results

A. Experiments

Dataset: Microsoft Common Objects COCO [26] dataset is utilized in our experimental setup. Model training was conducted on the train2017 set, and the val2017 set is considered for validation. Specific requirements of PAFPN were enforced, such as a condition where the input image size had to be divisible by 32. Consequently, we adjusted the image size in training to meet this criterion. We employed two distinct image sizes in the validation phase: a fixed 640x640 for computational and inference speed comparisons with YOLO and a maximum size of 800x1333 for benchmarking against other DETR variants. Our backbone also incorporated pretraining weights derived from a portion of YOLO-v7's training on the COCO dataset.

Implementation Details: For our model, we adopted training and configuration strategies from DINO [27]. In the backbone section, the learning rate was set to $1x10^{-5}$, while in other parts of the model, the learning rate was set to $1x10^{-4}$. Our training spanned 40 epochs, with the overall model weight decaying to $1x10^{-5}$ at the 30th epoch. The feature layers comprise four in total, with three being direct outputs from the CNN, while the fourth set of features is generated similarly to Deformable DETR, using a separate projection layer. Including the fourth layer resulted in only a marginal 0.2 mAP improvement, although it did somewhat enhance inference time efficiency. Regarding query management, we employed 900 queries, with 300 selected for entry into the decoder after query selection. Note that reducing the number of queries significantly enhanced reasoning speed but had a more pronounced impact on accuracy. When testing with only one-third of the queries, the final accuracy decreased by 0.8 mAP.

TABLE II
COMPARISON WITH DETR VARIANTS

Model	Epochs	mAP	AP_S	AP_M	AP_L	GFLOPS	Parameters	Infer speed(FPS)
Deformable DETR [13]	50	46.8	29.8	49.7	62.0	177	40M	
Lite-Deformable DETR H2L2-(2+1)x3 [16]	50	45.8	27.7	49.1	61.1	108	41M	
Lite-Deformable DETR H3L1-(6+1)x1 [16]	50	45.9	27.9	49.0	61.6	115	41M	
Lite-Deformable DETR H3L1-(3+1)x2 [16]	50	46.2	28.2	49.2	61.5	119	41M	
Lite-Deformable DETR H3L1-(2+1)x3 [16]	50	46.7	29.1	49.7	61.5	123	41M	
Sparse DETR-rho-0.1 [14]	50	45.3	28.4	48.3	60.1	111	41M	
Sparse DETR-rho-0.2 [14]	50	45.6	28.5	48.6	60.4	119	41M	
Sparse DETR-rho-0.3 [14]	50	46.0	29.1	49.1	60.6	127	41M	
Sparse DETR-rho-0.5 [14]	50	46.3	29.0	49.5	60.8	141	41M	
DINO-4Scale	40	50.6	33.8	53.8	65	234	42M	12
Stable-DINO-4scale [28]	24	51.5	35.2	54.7	66.5	-	-	
Ours	40	**51.8**	**38**	**55.7**	64.1	144	**36M**	**17**

B. Results

Table I provides a comparative analysis of our model against YOLO and DINO. As our model and DINO were not trained on 640 x 640 size images. Consequently, there is a reduction in accuracy compared to 800 x 1333 size images, where our model exhibits a decrease of 2.2 mAP, and DINO experiences a more pronounced 4.9 mAP drop. Upon further investigation, we observed that the decline in accuracy primarily stems from smaller objects. Our model registers a 2.6 mAP decrease in detecting small-scale objects, whereas DINO exhibits a substantial 10 mAP reduction.

Two key factors contribute to this distinction: (1) Our model's utilization of YOLO's pre-training weights endows the backbone with a degree of generalization for 640 x 640 imaging data. (2) Our model incorporates a more extensive CNN structure, enhancing its small object detection capabilities compared to DINO. This difference becomes even more noticeable with the reduction in image size. From a computational perspective, our model reduces computational requirements by 30 GFLOPS compared to YOLO. YOLO generates numerous detection bounding boxes in the head, incurring substantial computational expenses.

In contrast, our model dramatically reduces the number of queries through selective query processes, thereby decreasing computational demands in the decoder section. However, our algorithm lags behind YOLO in inference speed due in part to sub-optimal hardware support for the deformable attention operator. Addressing this discrepancy remains an area for future optimization.

Table II presents a comparative analysis of our model against the latest DETR variants. Specifically, we assess two computational optimization iterations, namely Lite deformable-DETR and sparse-DETR, both derived from deformable-DETR. These variants demonstrate a notable 38% enhancement in computational efficiency compared to deformable-DETR. However, this optimization comes at a slight cost in terms of accuracy, with a marginal 0.1-1 mAP reduction attributable to the omission of a portion of tokens in the token phase.

In contrast, our model leverages a combination of DINO and CNN to achieve a 38% reduction in computational complexity, while simultaneously improving accuracy by 1.2 mAP. Furthermore, we have managed to optimize the number of parameters by 14%.

IV. CONCLUSION

We proposed a novel object detection framework that integrates CNN and transformer networks. By amalgamating the gradient combination design inherent in CNN networks with the stable matching mechanism characteristic of transformers. The proposed model reduces computational time and decreases model parameters while enhancing accuracy. Additionally, our model inherits a key advantage from the DETR model—eliminating the Non-Maximum Suppression (NMS) component. This feature streamlines the model training process and facilitates ongoing model optimization, offering a dual benefit for model development.

V. USE OF AI TOOLS DECLARATION

We declare that we have utilized an AI-based proofreading and grammar-checking tool to review and enhance the quality of this research paper. While the tool has significantly assisted in identifying and rectifying grammatical errors and inconsistencies, the content, analysis, and conclusions presented herein remain the author's sole responsibility.

REFERENCES

[1] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:215827080

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84 – 90, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:195908774

[3] R. B. Girshick, "Fast r-cnn," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206770307

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:10328909

[5] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206594738

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:2141740

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225039882

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10 002, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232352874

[9] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *ArXiv*, vol. abs/2005.12872, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218889832

[10] S. Liu, F. Li, H. Zhang, X. B. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr," *ArXiv*, vol. abs/2201.12329, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246411225

[11] F. Li, H. Zhang, S. guang Liu, J. Guo, L. M. shuan Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 609–13 617, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247218212

[12] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3651–3660.

[13] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *ArXiv*, vol. abs/2010.04159, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222208633

[14] B.-S. Roh, J. Shin, W. Shin, and S. Kim, "Sparse detr: Efficient end-to-end object detection with learnable sparsity," *ArXiv*, vol. abs/2111.14330, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244714783

[15] T. Wang, L. Yuan, Y. Chen, J. Feng, and S. Yan, "Pnp-detr: Towards efficient visual analysis with transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4641–4650, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237513604

[16] F. Li, A. Zeng, S. Liu, H. Zhang, H. Li, L. Zhang, and L. M. shuan Ni, "Lite detr : An interleaved multi-scale encoder for efficient detr," *ArXiv*, vol. abs/2303.07335, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257495882

[17] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16 514–16 524, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231718848

[18] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: improving vision transformers with soft convolutional inductive biases," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2022, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232290742

[19] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye, "Conformer: Local features coupling global representations for visual recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 367–376.

[20] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *ArXiv*, vol. abs/2106.04803, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235376986

[21] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208310312

[22] C.-Y. Wang, H. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," *ArXiv*, vol. abs/2211.04800, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253420213

[23] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:3698141

[24] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 024–13 033, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:226964445

[25] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "Varifocalnet: An iou-aware dense object detector," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8510–8519, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221376816

[26] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14113767

[27] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J.-J. Zhu, L. M. shuan Ni, and H. yeung Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *ArXiv*, vol. abs/2203.03605, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247292561

[28] S. Liu, T. Ren, J.-Y. Chen, Z. Zeng, H. Zhang, F. Li, H. Li, J. Huang, H. Su, J.-J. Zhu, and L. Zhang, "Detection transformer with stable matching," *ArXiv*, vol. abs/2304.04742, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258049102