Object Detection using RT-DETR

Alan Devkota
University of Houston

I. INTRODUCTION

Object detection represents a specialized field within computer vision that intertwines recognition and spatial analysis. At its core, this technology processes digital imagery to accomplish two fundamental objectives: determining what objects exist within a scene and establishing their precise spatial coordinates.

The process unfolds as a seamless fusion of identification and positioning, where computational models examine each frame to recognize predefined objects while simultaneously calculating their exact locations. Rather than simply classifying image contents, object detection creates a comprehensive understanding of scene composition by generating bounding boxes that frame each identified item.

This technology stands apart through its capacity to handle multiple object instances in parallel, delivering real-time analysis that maps the spatial relationships between different elements in the visual field. The end result transforms raw visual data into structured information, expressing both the categorical nature of detected objects and their geometric placement within the scene. Through this dual functionality of recognition and localization, object detection serves as a cornerstone for modern computer vision systems, enabling applications ranging from autonomous navigation to security surveillance.

DETR is a state-of-the-art object detection model in computer vision. Unlike traditional object detection methods that rely on region proposal networks and anchor boxes[13,15,2], DETR uses a transformer architecture, which was originally developed for natural language processing tasks. In DETR, the image is divided into a fixed number of regions, and each region is treated as a separate "object query." The transformer processes these queries in parallel, attending to the entire image simultaneously, a method contrasting with learnable NMS methods and relation networks that explicitly model relations between different predictions with attention [7]. This approach eliminates the need for separate region proposals and feature extraction steps.

DETR's key innovation is its direct set-based prediction approach. Instead of predicting bounding boxes and class scores independently, DETR predicts all object queries and their corresponding positions and classes in a single forward pass. The model uses a bipartite matching mechanism to associate predicted queries with ground truth objects, allowing it to handle object detection as a set prediction problem[14]. This set-based paradigm enables DETR to achieve impressive results in terms of accuracy and efficiency. It has shown effectiveness in various object detection benchmarks and has become a notable model in the field of computer vision. DETR's design showcases the versatility of transformer architectures beyond natural language processing, illustrating their effectiveness in handling complex visual tasks.

A topic covered in this experimentation is using RT-DETR, an extension to Detection Transformers (DETR), as a means to conduct real-time Object Detection from video as well as webcam [18]. RT-DETR (Real-Time Detection Transformer) represents a significant advancement in object detection technology, combining the efficiency of real-time processing with the robust capabilities of transformer architecture. As a state-of-the-art detection framework, RT-DETR optimizes the balance between computational speed and detection accuracy.

The framework excels through its streamlined architecture, which processes image features through a transformer decoder to generate precise object predictions. Unlike traditional detection methods, RT-DETR achieves remarkable inference speeds while maintaining competitive accuracy metrics on standard benchmarks. What sets RT-DETR apart is its innovative approach to feature extraction and object localization, employing a refined attention mechanism that efficiently processes visual information. This design enables rapid object detection across diverse scenarios, making it particularly valuable for applications requiring real-time performance.

Despite the inherent complexity of transformer-based architecture, RT-DETR demonstrates that high-performance object detection can be achieved without sacrificing processing speed, establishing a new paradigm in efficient visual recognition systems.

Our focus of research is to develop a transformer model that elevates real-time object detection through its Enhanced Hybrid Encoder and Refined Query Selection System. The encoder processes multi-scale features efficiently, while the query selection minimizes uncertainty for improved detection accuracy. The framework offers flexible speed tuning without retraining and simplifies deployment by removing dual NMS thresholds. This innovative design establishes RT-DETR as a powerful alternative to YOLO-based methods, making it suitable for diverse real-time applications requiring both speed and precision [20].

II. RELATED WORK

The article produced by Facebook AI regarding Object Detection using Transformers[1] introduces a paradigm-shifting methodology for object detection by leveraging transformer architectures, similar to those used in sequence prediction [16]. Departing from the traditional two-stage approach, this model streamlines the entire process into a single end-to-end framework. The core innovation lies in the direct prediction of object detections without the need for explicit region proposals and extensive feature extraction, differing from methods relying on large sets of proposals [15,2], anchors [10], or window centers [17,16].

The transformer's self-attention mechanism proves instrumental in processing images divided into a set of learnable object queries[4]. Unlike traditional methods relying on predefined anchor boxes, the model adapts dynamically to different object scales and shapes. The capacity of the transformer to capture long-range dependencies is harnessed for understanding global context, enabling a more comprehensive perception of the entire scene.

A significant advancement is the incorporation of a bipartite matching algorithm during training[9]. This mechanism facilitates the direct association of predicted queries with ground truth objects, simplifying the training process and eliminating the need for post-processing steps. The end result is a model capable of simultaneously predicting object classes and bounding boxes in a single pass.

Evaluation on benchmark datasets like COCO [11] underscores the model's competitive performance compared to conventional two-stage detection approaches, such as Faster R-CNN [13]. The end-to-end design, coupled with the inherent strengths of transformers, suggests a promising direction for the evolution of object detection methods. Beyond achieving state-of-the-art results, the paper contributes to the broader exploration of transformer architectures in visual tasks, pushing the boundaries of what is achievable in object detection and paving the way for more efficient and effective computer vision systems.

The DETR architecture is characterized by its simplicity and is composed of three main components: a CNN backbone for feature extraction[5], an encoder-decoder transformer, and a straightforward feed-forward network (FFN) for the final detection prediction[6]. Unlike many contemporary detectors, DETR can be implemented in various deep learning frameworks with minimal lines of code[12]. The backbone starts with an initial image, extracting features through a conventional CNN to produce a lower-resolution activation map.

Positional encodings are added to account for the transformer's permutation invariance[4]. Then, after reducing the channel dimension of the activation map using a 1x1 convolution, and generating a new feature map[3], the transformer encoder is applied. This map is then treated as a sequence, and each encoder layer employs a multi-head self-attention module and a feed-forward network. The transformer decoder follows a standard architecture but decodes N objects in parallel at each layer, in contrast to the autoregressive model used by Vaswani et al[16]. The N object queries, serving as positional encodings, are transformed into output embeddings and independently decoded into box coordinates and class labels.

The final prediction is computed by a 3-layer perceptron with ReLU activation and a linear projection layer. This predicts the normalized box coordinates and class labels using a softmax function. To handle a fixed-size set of N bounding boxes, an additional special class label ∅ is introduced to represent cases where no object is detected within a slot[6].

Auxiliary decoding losses, including prediction FFNs and Hungarian loss, are employed during training to assist the model in outputting the correct number of objects for each class[8]. These losses are applied after each decoder layer, and shared layer normalization is used to normalize inputs to the prediction FFNs from different decoder layers.

Detection Transformers are introduced as a novel object detection system built on transformer architecture and bipartite matching loss for direct set prediction. In evaluations on the challenging COCO dataset, DETR achieves results comparable to an optimized Faster R-CNN baseline[13]. Its implementation is straightforward, and its flexible architecture easily extends to panoptic segmentation, delivering competitive performance[11]. Notably, DETR outperforms Faster R-CNN on large objects, attributed to its ability to process global information through self-attention[11].

While DETR presents promising results, it also introduces new challenges, particularly in training, optimization, and performance on small objects. The paper acknowledges that addressing these challenges might require further work, drawing parallels with the evolution of current detectors that needed several years of refinement to overcome similar issues. The authors anticipate future research efforts to successfully tackle these challenges and further enhance the capabilities of DETR[11].

Furthermore, in the extension of DETR, the RT-DETR model is designed with efficiency and real-time performance in mind while maintaining the advantages of DETR-like models. Its architecture consists of three main components: an efficient hybrid encoder, a lightweight IoU-aware DINO decoder, and task-aligned assignment and losses. The model uses a hybrid backbone that combines CNN and Vision Transformer (ViT) elements to extract multi-scale features efficiently [19].

The hybrid encoder is a key innovation that processes features at multiple scales (1/8, 1/16, and 1/32) through both spatial and channel dimensions. It employs efficient attention mechanisms and hybrid attention blocks that combine self-attention and cross-attention operations. This design allows the model to capture both local and global dependencies while maintaining computational efficiency. The encoder also includes a feature pyramid network (FPN) structure that helps in handling objects at different scales.

The decoder follows a DINO-style architecture but is optimized for real-time performance. It uses a set of learnable object queries to interact with the encoded features through cross-attention mechanisms. The decoder is made more efficient by reducing the number of decoder layers and incorporating IoU-aware prediction heads. The task-aligned assignment strategy helps in matching predictions with ground truth objects more effectively, while the loss functions are designed to balance classification and regression tasks. This architecture achieves a good trade-off between accuracy and speed, making it suitable for real-time applications.

III. EXPERIMENTAL METHODOLOGY

Our methodology leverages RT-DETR, in extension to DETR, to enhance the robustness and accuracy of object detection. The process is subdivided into three main stages, a backbone of CNN to extract the features from images, a transformer architecture that consists of an encoder and decoder to learn the contextual information between the embedded features, and two classifier units to predict class and bounding box for each object detected inside an image. Here, ResNet50 is used to extract features of images via ResNe50 to get learned feature representations. Our approach relies on transformer architectures, which are powerful tools that are typically used for language tasks. The versatility of these transformers allows us to extend behavior and comprehension of the larger-scale environment.

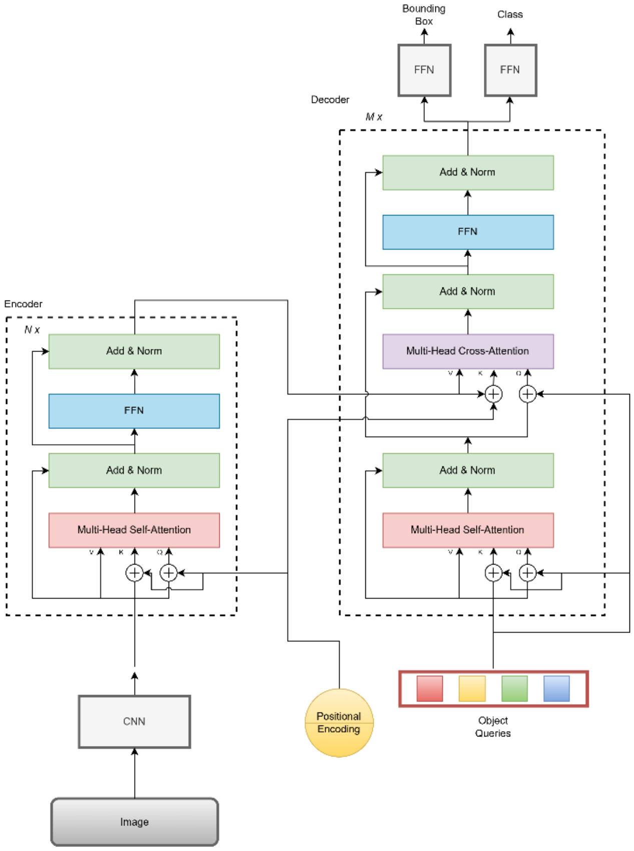

Figure 1: Overall Transformer Architecture

A. Features Extraction

The initial step involves tokenizing the images from the dataset by breaking down the visual information into manageable units that can be analyzed by the model. The feature representations are then extracted using a pre-trained Resnet50 model. The features extracted are then concatenated together and passed to the original DETR encoder, thus the embedding dimension would be double that of the original model. During the attention calculation, these features serve as the queries *(Q)*, keys *(K)*, and values *(V)* in the transformer's encoder. A multi-headed self-attention mechanism is applied, allowing the model to focus on relevant parts of the image.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

$$\text{where } Q = YW^Q, K = XW^K, V = XW^V$$

Equation 1: Attention Equation

Here, *W* is a trainable weight matrix, initialized as random in the beginning and used to form *Q*, *K,* and *V.* This attention mechanism on concatenated features allows the model to discern intricate details and relationships between different aspects of the image between two modalities, enhancing its ability to accurately detect objects from the provided dataset. Thus, the attention mechanism is crucial in enabling the model to weigh the importance of different visual elements that span across the image as it can extract missing features from one modality and also enhance the learned representation by combining the modalities.

Following this attention phase in the encoder, a fixed set of learnable encodings called object queries would attend with the keys and values output from the encoder via cross-attention. Therefore, object queries would detect corresponding objects in the output features from the encoder. A multi-layer perceptron (MLP) block is used to process these attention-guided features. The MLP block is a sophisticated component designed to process the features that have been refined through the cross-attention mechanism. The MLP block acts as a computational mechanism that applies non-linear transformations to the extracted features, further accentuating the similar and distinct patterns from the image dataset to provide precise object detection.

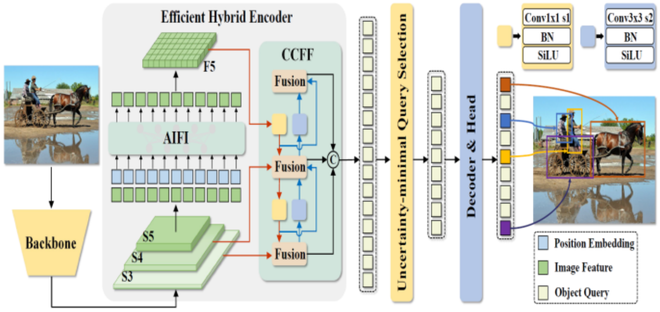

Figure 2: Overview of RT-DETR architecture (*from original paper*)

B. Set Prediction Loss

The amalgamated features after concatenation after the cross-attention between object queries and the key and values from the encoder output are then utilized for two concurrent objectives: classification and object localization. Classification is achieved through a predictive model, which employs the cross-entropy loss function to ascertain the probability distribution over predefined classes. For object localization, a bounding box regression model delineates the spatial coordinates of the object instances within the input data. The performance of the bounding box regression is quantified through a loss function suitable for continuous output values.

$$\mathcal{L}_{\text{class}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}),$$

Adopting the DETR model, instead of using the anchor boxes and processing the entire set of objects in an image, set-based paradigm was introduced for object detection. The key functions used for this process are Classification Loss, Bipartite Matching loss, No Object Loss and Bounding Box loss. The classification loss is computed as cross-entropy loss. Let *Pij* is the predicted probability of the j-th class and *yi* is the ground truth for the class label of the i-th object. The equation for classification loss and bipartite match is given as:

Equation 2: Classification Loss Equation

where *N* is the number of objects, C is the number of classes.

$$\mathcal{L}_{\text{no\_obj}} = \frac{1}{N}\sum_{i=1}^{N}(1 - \sum_{j=1}^{N_{\text{obj}}} M_{ij})\mathcal{L}_{\text{class}}(c_{ij}, \hat{c}_{ij}),$$

The Bipartite machine loss uses Hungarian algorithm to associate predicted objects with ground truth objects to find the optimal assignment matrix. If *M* is the assignment matrix, then the value of *Mij* is 1 when the i-th predicted object is matched with the j-th ground truth object, and 0 otherwise. The Bipartite loss is defined as:

$$\mathcal{L}_{\text{box}}(b_{ij}, \hat{b}_{ij}) = \sum_{k\in\{x,y,w,h\}} \text{SmoothL1}(b_{ij}^k - \hat{b}_{ij}^k),$$

Equation 3: Bipartite Loss Equation

where b_{ij} and $\hat{b_{ij}}$ are the predicted coordinates and the ground truth bounding box. λ_{box} is a smooth L1 for bounding box coordinates, and λ is a balancing power. The No Object Loss is defined as:

Equation 4: No Object Loss Equation

where c_{ij} and $\hat{c}_{ij}$ are the class probabilities for the i-th predicted objects and ground truth. The bounding box loss measures the difference between the predicted and ground truth bounding box is computed via smooth L1 loss.

Equation 5: Bounding Box Loss Equation

Thus, total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{class}} + \lambda_{\text{match}}\mathcal{L}_{\text{match}} + \lambda_{\text{no\_obj}}\mathcal{L}_{\text{no\_obj}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}},$$

Equation 6: Total Loss Equation

where λ_{match}, $\lambda_{\text{no\_obj}}$ and λ_{box} are weighting parameters whose values are typically determined through experiment. Also, the Intersection over Union (IoU) is more commonly used as an evaluation metric to assess how well the predicted bounding box aligns with the ground truth bounding boxes. It helps to evaluate the performance of our model.

Prior to processing the images, it was vital to understand the annotations associated with the provided dataset. Standard KAIST dataset provided annotations in a format different than what would be adaptable to our use cases regarding processing of images. The KAIST dataset natively used annotations provided in XML format, which would not be an ideal fit for our object detection and image processing architectures. Python code was developed to handle the conversion of the KAIST dataset from XML format to COCO format, which allowed for Dataset Sanitization. The main goal of Dataset Sanitization is to ensure that the data is accurate, reliable, and free from errors or inconsistencies prior to feeding into the model, which allows for meaningful analysis and training to take place.

C. Implementation of RT-DETR

Next, we implemented RT-DETR, whose architecture is shown in figure 2. The RT-DETR model processes visual information through a sophisticated three-stage pipeline. Starting with the backbone network's final three stages, the system extracts hierarchical features at multiple scales.

These features then pass through the efficient hybrid encoder utilizing AIFI (Attention-based Intra-scale Feature Interaction) for analyzing relationships within each scale and CCFF (CNN-based Cross-scale Feature Fusion) for merging information across scales via convolutional operations.

The architecture concludes with a precision-focused query selection process that identifies key feature points as object candidates, followed by iterative refinement in the decoder phase where specialized prediction heads enhance both classification and localization accuracy. For real time object detection using RT-DETR, pretrained model from ultralytics library was implemented and finetuned on our football detection dataset. Then, the object detection was further extended to real time object detection using the real time video captured from the webcam. Evaluations on the webcam video and football game recorded video provided desirable results.

IV. RESULTS

Figure 3 and Figure 4 shows the object detection results from DETR based object detection model. Figure 5 shows a snapshot of real time object detection from a saved video. Figure 6 shows a snapshot of real time object detection from webcam.

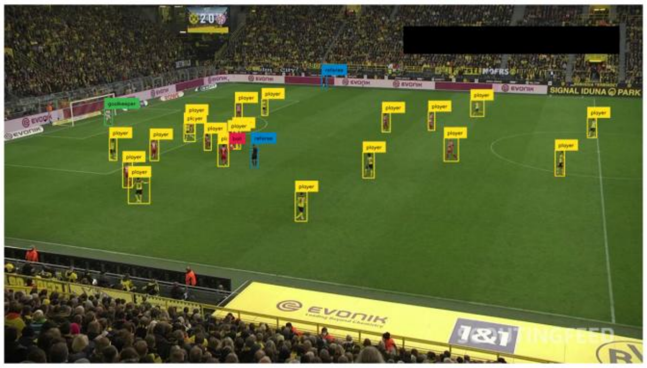

Figure 3: Detection using DETR model

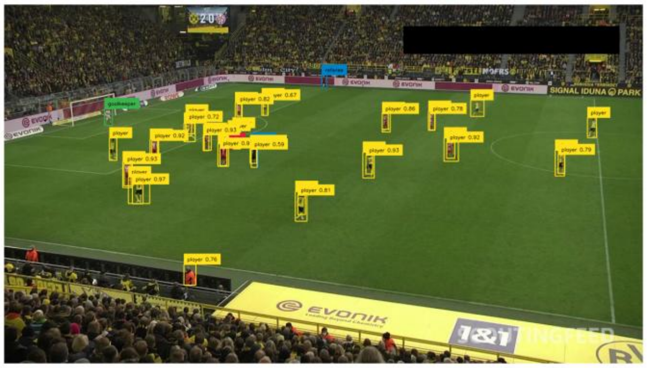
[redacted]

Figure 4: Detection without NMS (non-maximum suppression) using DETR model.

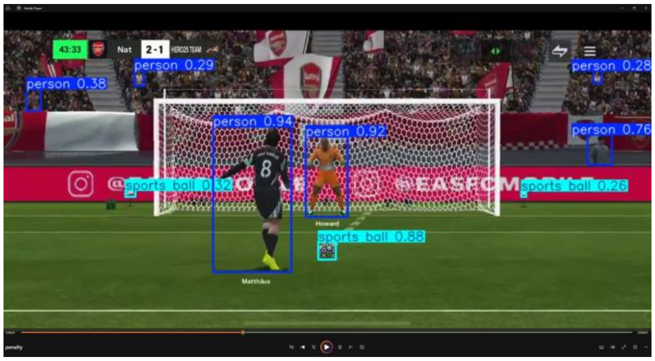

Figure 5: Detection using RT-DETR from a video

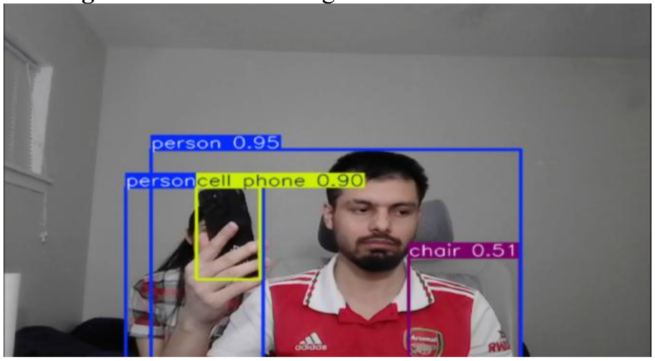

Figure 6: Detection using RT-DETR from a webcam video

V. CONCLUSION

The purpose of our research was to develop a transformer model that allows for the seamless integration of multiple modalities in order to improve the prediction of object detection models. Through our experimentation, we were able to leverage our modified DETR transformer model to capture and process image data from two localizations of the electromagnetic and light spectrum: RGB and IR. We fused these modalities features into the updated model which successfully performed object detection.

Through our results and findings, we were able to prove that our updated architecture performed slightly better than the standard DETR database in performing object detection. Our new model can help improve multiple sectors from healthcare, autonomous driving, to even military applications to support newer and more advanced object detection and processing techniques.

In our future work, we aim to expand the current framework from two modalities to three modalities, enhancing our capability to capture and integrate information from multiple sources comprehensively. Additionally, we propose a new architecture that incorporates two encoders. This development will enable us to process the data features of different modalities with greater precision. By introducing a second encoder, we can provide dedicated feature extraction and encoding for each modality, thereby improving the overall expressiveness and accuracy of the model. Such advancements will lay a solid foundation for addressing more complex issues such as cross-modal understanding and generation.

REFERENCES

[1]. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. arXiv preprint arXiv:2005.12872.

[2]. Cai, Z., Vasconcelos, N.: Cascade R-CNN: High quality object detection and instance segmentation. PAMI (2019)

[3]. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)

[4]. He, K., Girshick, R., Doll´ar, P.: Rethinking imagenet pre-training. In: ICCV (2019)

[5]. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

[6]. Hosang, J.H., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: CVPR (2017)

[7]. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: CVPR (2018)

[8]. Kirillov, A., Girshick, R., He, K., Doll´ar, P.: Panoptic feature pyramid networks. In: CVPR (2019)

[9]. Kuhn, H.W.: The hungarian method for the assignment problem (1955)

[10]. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Doll´ar, P.: Focal loss for dense object detection. In: ICCV (2017)

[11]. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)

[12]. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai,

J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)

[13]. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. PAMI (2015)

[14]. Salvador, A., Bellver, M., Baradad, M., Marqu´es, F., Torres, J., Gir´o, X.: Recurrent neural networks for semantic instance segmentation. arXiv:1712.00617 (2017)

[15]. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: ICCV (2019)

[16]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)

[17]. Zhou, X., Wang, D., Kr¨ahenb¨uhl, P.: Objects as points. arXiv:1904.07850 (2019)

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

[19]. Zhang, Hao, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. "Dino: Detr with improved denoising anchor boxes for end-to-end object detection." arXiv preprint arXiv:2203.03605 (2022).

[20]. Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." multimedia Tools and Applications 82, no. 6 (2023): 9243-9275.