Literature review on Transformers in object detection

Elkhan Ismayilzada[1], Minjae Lee[1] and Seungryul Baek[2]

[1] UNIST, Department of CSE, Ulsan, Republic of Korea, {elkhan, lmjbsj}@unist.ac.kr
[2] UNIST, AI Graduate School, Ulsan, Republic of Korea, srbaek@unist.ac.kr

Abstract

Object detection is an essential and challenging computer vision task that lately attracted a lot of attention. Convolutional neural networks (CNN) was the number one choice to use for this task due to its advantage of automatic feature extraction mechanism until the introduction of the vision transformers. Currently, transformer-based object detection models are considered as state-of-the-art due to their significant advantages over CNN-based ones and in this paper we will review recent works in this stream of works.

Keywords— *Object Detection, Transformers, Deep Learning*

I. Introduction

Object detection is a task in which we aim to find the objects and their bounding boxes in the images. Traditionally, convolutional neural networks (ConvNets) were extensively used due to its advantage of automatic feature extraction mechanism. However, using ConvNets alone is not sufficient for this task since it is not a basic set prediction task. Main difficulty in object detection is to get rid of near-identical bounding boxes and this requires to have handcrafted components in the model such as non-maximum suppression, anchor generation etc. Transformers can eliminate these post-processing steps as suggested first by [1] while performing better than the traditional methods. In this paper, we will explore several existing transformer-based object detection models starting with the foundations of transformers.

II. Transformers

A. Overview

Transformers were originally introduced by [9] as a new attention model that can be used for natural language processing tasks such as machine translation, speech recognition etc. The main advantage of this model over existing models such as recurrent neural networks (RNN) is the ability to learn from the entire input sequence. Additionally, although this model is computationally more expensive than RNN, it supports parallel computation unlike the autoregressive models.

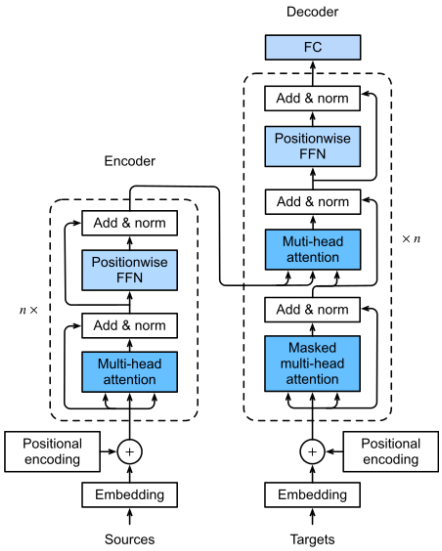

Fig. 1. Overall architecture of Transformers.

Transformers consist of four components: positional encoding, multi-head attention, add & norm and positionwise feedforward network as shown in Figure 1.

B. Positional Encoding

Since the transformers are permutation invariant we have to inject positional information to the input sequence and the authors of the paper suggest the following encoding function

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

where pos is the position, i is the dimension and d_{model} is the embedding size of the input.

C. Multi-head Attention

To understand multi-head self attention we need to first explain the self attention mechanism. Given a sequence

of words, self-attention predicts the correlation among the words. Given a sequence of n entries $(x_1, x_2, \ldots x_n)$ denoted as $X \in \mathbb{R}^{n \times d}$ where d is the embedding size, we try to learn three weight matrices i.e., W^q, W^k, W^v called as queries, keys, values respectively. First we transform the input sequence X using these weight matrices such that $Q = XW^q$, $K = XW^k$, $V = XW^v$ and apply softmax operation as follows

$$O = \text{softmax}\left(\frac{QK^T}{\sqrt{q}}\right)V$$

where q is the output dimension of W^q.

In decoder, since we predict the future word of the sequence, masked self-attention mechanism is used. The only difference with the standard self-attention is that we ignore the attention weights of future entities by element-wise multiply with an upper-triangular matrix

$$O = \text{softmax}\left(\frac{QK^T}{\sqrt{q}} \circ M\right)$$

where M is an upper-triangular mask matrix.

Multi-head attention is composed of multiple self-attention blocks in order to learn complex relationships between different words in the sequence.

D. Add & norm

After every layer except last fully connected layer (FC) the input and output of the layer is added and normalized.

E. Positionwise FFN

Positionwise FFN consists of two linear layers and ReLU activation function in between. It is used to further process the attention output potentially giving it a richer representation.

III. Transformers as Detectors

In this section we will go through several influential works on using transformers in object detection.

A. DETR

Detection Transformer (DETR) proposed by [1] is one of the earliest works in this area. Overall architecture is shown in Figure 2. They use CNN as a backbone network to extract features from the set of images and predict the set of bounding boxes using transformers. The main benefit of DETR is that there is no hand-designed postprocessing steps after the prediction and due to its simplistic architecture it can be used for other computer vision tasks such as segmentation. DETR significantly outperforms the CNN-based detectors in detecting large objects but struggles to detect small objects in the image. The another problem of this model is that it is computationally expensive. As we increase the size of feature map, the computational cost increases quadratically with $O(CW^2H^2)$ where W and H are width and height of the feature map respectively [12]. The model uses bipartite matching loss for matching the set of ground truth objects with the predicted ones and generalized IoU loss [8] for bounding box prediction.

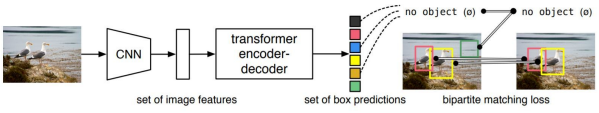

Fig. 2. Overall architecture of DETR.

B. D-DETR

Deformable DETR (D-DETR) proposed by [12] is one of the solutions to the problems related to the original DETR [1]. Motivated from deformable convolutions [4], they propose deformable version of self attention mechanism for processing the feature map. Unlike the standard self attention, regardless of the spatial size of the feature map, it attends to only sparse set of elements from it. As a result, deformable DETR converges 10 times faster than the original DETR model and even performs better in terms of AP (Average Precision) score.

C. A-DETR

Since transformers are permutation invariant, DETR model utilizes positional embedding as object queries to generate different kinds of bounding boxes. By doing so, each object query tries to focus on a specific region of the image. Problem is, however, that we cannot guess the region where the object query will focus on. Besides, it is possible that the region the object query focuses on might have multiple objects. Anchor DETR (A-DETR) [10] aims to solve these aforementioned problems. They use anchor points from CNN-based detectors as object queries so that each query learns to find objects near the anchor point. The model performs better than the original DETR [1] while requiring 10 times fewer training epochs.

D. Pix2Seq

Pix2Seq [3] treats the object detection task as language modeling task conditioned on previous predictions and image features. To do that, they convert the object descriptions i.e., class labels and bounding boxes to a sequence of discrete tokens and use a generic transformer-based encoder-decoder network to generate the desired sequence in an autoregressive manner. A major limitation of this approach, however, is it is significantly slower when there are a large number of objects in a single image because of autoregressive modeling.

E. YOLOS

All of the aforementioned DETR models have CNN as their backbone network for extracting the image features. You Only Look at One Sequence (YOLOS) [6], on the other hand, completely removes CNN from their pipeline and uses transformers instead. To understand YOLOS, we need to understand Vision Transformers proposed by [5]. It is the first work that classified the images using only transformers. They do so by dividing the images to small, fixed size patches flattened as vectors and then feed them to the transformers to generate class tokens. It significantly outperformed the CNN-based approaches for image classification task. YOLOS is directly built upon this model but is trained for object detection task rather than image classification so similar to the original DETR [1] they use object queries to learn and bipartite matching as a loss function.

F. DFFT

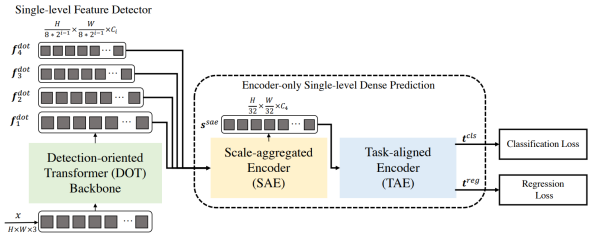

Fig. 3. Overall architecture of DFFT.

DFFT (Decoder-Free Fully Transformer-based) [2] is yet another recent fully transformer-based object detection model that aims to decrease computation cost by a large margin. Unlike the previous transformer-based models, as the name implies, DFFT has no decoder and utilizes two novel encoders, namely scale-aggregated encoder and task-aligned encoder to retain the accuracy of single-level feature map prediction as shown in Figure 3. The proposed framework outperforms DETR [1] by a small margin in terms of AP score with about 28% reduction in computation and converges in more than 10 times fewer training epochs. However, it has low accuracy in detecting small objects in the image.

G. DESTR

Object detection consists of two tasks i.e., image classification and bounding box regression and aforementioned DETR models treat these tasks as one since a single estimation of cross-attention in decoder is computed for both of them. Detection Split Transformer (DESTR) [7] changes the decoder layer to include two cross-attention mechanisms which independently trained for classification and box regression so that they can optimally focus on relevant visual cues for the respective tasks as shown in Figure 4. The proposed model outperforms DETR and its successors. However, it suffers from high computational cost due to the introduction of another cross-attention layer in decoder.

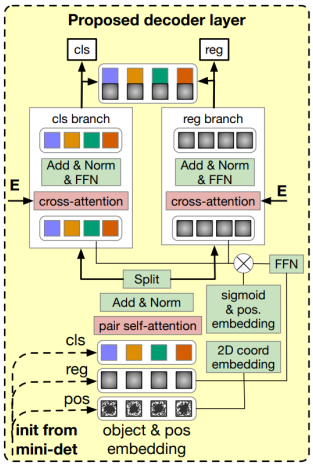

Fig. 4. Proposed decoder of DESTR.

H. MTTrans

Transformer-based object detectors require large amount of labeled data which may not be available and if there is shift in domain or variation of data distribution the performance of these models degrades significantly. MTTrans (Transformer based on the mean teacher framework) [11] is an approach to solve this problem. It is an end-to-end cross-domain detection Transformer framework that uses mean teacher architecture to transfer knowledge between the domains and produces pseudo labels for unlabeled data. To generate high quality pseudo labels they make use of cross-scale self-attention mechanism used in Deformable DETR [12]. The model outperforms the existing methods by a large margin in all scenarios they have shown in the paper. However, it suffers from detecting blurred objects in the image.

IV. Conclusion

In this paper, we have reviewed several recent transformer-based approaches for object detection. First, we explained the basics of transformers and then introduced the earliest approach i.e., DETR [1] that detects the objects using transformers. Then we moved to the recent approaches that tackle the issue of slow convergence and high computational cost in DETR [1] such as D-DETR [12] A-DETR [10], DFFT [2], DESTR [7]. We also introduced a recent approach that deals with unlabeled data for object detection using transformers [11] and the method that treats object detection as a language modeling task [3]. We have summarized the highlights and the limitations of each method in Table 1.

Method	Highlights	Limitations
DETR [1]	**a)** First model that uses transformers for object detection, **b)** Significantly outperforms CNN-based models in detecting large objects.	**a)** Slow convergence, **b)** Quadratical increase in computational cost as the size of feature map increases, **c)** Struggle in detecting small objects.
D-DETR [12]	**a)** Better results in detecting small objects, **b)** Faster convergence than DETR [1].	**a)** Has two-stage object detector design.
A-DETR [10]	**a)** Uses anchor points as prior for learned object queries, **b)** Better performance than D-DETR [12].	**a)** No consideration to the objects of different scales.
Pix2Seq [3]	**a)** Language modeling approach for object detection.	**a)** Slow inference due to autoregressive modeling.
YOLOS [6]	**a)** Fully transformer-based pipeline.	**a)** Poor performance compared to DETR models.
DFFT [2]	**a)** Decoder-free fully-transformer based model, **b)** 28% reduction in computation and significantly faster convergence than DETR. [1]	**a)** Poor performance in detecting small objects compared to DETR. [1]
DESTR [7]	**a)** Independent training for each subtask in Object detection, **b)** Outperforms DETR models in all measures.	**a)** Less-efficient due to the addition of another cross-attention layer.
MTTrans [11]	**a)** Support for unlabeled data in object detection using mean teacher framework, **b)** SOTA results in every scenario.	**a)** Does not work well with blurred objects.

Table 1. Summary of the highlights and limitations of the methods discussed in this paper.

REFERENCES

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ECCV*, 2020.

[2] Peixian Chen, Mengdan Zhang, Yunhang Shen, Kekai Sheng, Yuting Gao, Xing Sun, Ke Li, and Chunhua Shen. Efficient decoder-free object detection with transformers. *ECCV*, 2022.

[3] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *ArXiv:XXXX.XXXXX*, 2021.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *ICCV*, 2017.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv:XXXX.XXXXX*, 2020.

[6] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *NeurIPS*, 2021.

[7] Liqiang He and Sinisa Todorovic. Destr: Object detection with split transformer. *CVPR*, 2022.

[8] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. *CVPR*, 2019.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[10] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *AAAI*, 2022.

[11] Jinze Yu, Jiaming Liu, Xiaobao Wei, Haoyi Zhou, Yohei Nakata, Denis Gudovskiy, Tomoyuki Okuno, Jianxin Li, Kurt Keutzer, and Shanghang Zhang. Mttrans: Cross-domain object detection with mean teacher transformer. *ECCV*, 2022.

[12] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ArXiv:XXXX.XXXXX*, 2020.

SUMMARY OF THIS PAPER

A. *Problem Setup*

Recently, researchers are using transformers for object detection due to its high potential and in this paper we reviewed the most influential transformer-based object detection architectures.