End-End Object Detection by using Transformers

Kheem P. Dharmani *NUCES, University,* Islamabad

Abstract-- **This paper presents a method for object detection that addresses the problem of direct set prediction. It eliminates the need for multiple components and simplifies the process by avoiding the need for manual intervention.The main components of a framework known as DEtection TRansformer are the transformer and the matching mechanism. This paper shows that the relationship between the two components is strong enough to explain the relationship between the two datasets,The DETR framework achieves a fast R-CNN baseline and is capable of providing uniform panoptic segmentation. It is also well-designed to improve its performan**ce.

***Keywords*— DETR Transformers, R-CNN**

I. INTRODUCTION

The leading sequence transduction models are based on the neural networks that are complicated recurrent or convolutional include an encoder and a decoder. The finest models additionally link the encoder and decoder. via a mechanism of attention [1], depending solely on attention to draw global dependencies between input and output, Convolutional neural networks are the foundation of all of them. block, concurrent computation of hidden representations for all input and output positions. A stack of similar layers makes up the encoder, All sub-layers in the model were designed to support these residual connections., the embedding layers, and so on, provide dimensioned outputs, where the query(Q), keys(K), values(V), and output are all vectors, consists of two linear transformations with a ReLU activation in between

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

Very long sequences, self-attention could be restricted to considering only a neighborhood of size in the input sequence centered around the respective output position.

On the standard, used previously trained data. WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs. Empirical evaluation of gated recurrent neural networks on sequence modelling enables neural machine translation by learning to align and translate together. [1].

Jifeng Dai DETR was recently proposed as a method for reducing the need for multiple hand-crafted components Despite this, it does well in object detection [3].

Because of Transformer attention modules' limitations in processing image feature maps, [2] It has a slow convergence rate and a low feature spatial resolution.

Deformable was designed to address these concerns, DETR, has attention modules that only pay attention to a small number of critical sample locations near a reference [2].

With 10 fewer training epochs, Deformable DETR can achieve greater performance than DETR (particularlyon small objects). Extensive testing on the COCO benchmark show that this method is effective [2].

Carion introduced DETR to remove the need for such handcrafted components, and created the first fully end-to-end object detector with exceptional performance.

DETR uses fundamental architecture that combines convolutional neural networks (CNNs) with Transformer encoder-decoders (Vaswani et al., 2017). They use Transformers' versatile and powerful relation modelling capability to replace hand-crafted rules when properly designed training signals are used [2]. DETR has its own drawbacks, despite its innovative design and strong performance:It takes substantially longer training epochs than existing object detectors to converge.

DETR, for example, takes 500 epochs to converge on the COCO (Lin et al., 2014) benchmark, which is 10 to 20 times slower than Faster R-CNN (Ren et al., 2015). When it comes to detecting small items, DETR has a bad track record High-resolution feature maps are used to recognize small objects.by modern object detectors, which use multi-scale features. Meanwhile, DETR meets unacceptable complications as a result of high-resolution feature maps. Transformer component deficiencies in processing image feature maps are primarily to blame for the mentioned difficulties.The attention modules assign approximately consistent attention weights to all pixels in the feature maps during initialization [2].

Long periods of practice are required to master attention weights needed to focus on sparse meaningful sites. In the Transformer encoder, on the other hand, the attention weights computation is a quadratic computation with respect to pixel numbers. As a result, processing high-resolution feature maps has a significant computational and memory complexity. Deformable convolution (Dai et al., 2017) is a powerful and

efficient method for attending to sparse spatial locations in the picture domain. As illustrated in Figure 1, we use (multi-scale) deformable attention modules to replace the Transformer attention modules that process feature maps in Deformable DETR.

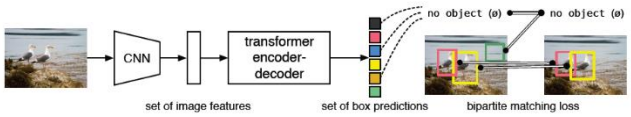

Figure 1 [6] (Carion et al., n.d.)

II. Comparative Analysis

A. Builds On Previous Research

Both the input and output of the encoder are multi-scale feature maps of the same resolution. We extract multi-scale feature maps from ResNet stages C3 through C5 output feature maps (processed by a 1 1 convolution) in encoder, where Cl is 2l lower in resolution than the source image [4]. Other hyper-parameter settings and training strategies are similar to DETR, with the exception that for bounding box classification, Focal Loss with a loss weight of 2 is employed, and object queries number is raised from 100 to 300. For a fair comparison, we also provide the performance of DETR-DC5 with these improvements, indicated as DETR [4].

B. Differ From Previous Research

The lowest resolution feature map xL is produced via a 33 percent stride 2 convolution on the final C5 stage. FPN (Lin et al, 2017a) is not needed because our proposed multi-scale deformable attention may transfer information among multi-scale feature maps. The instances of Deformable DETR are already divided in the encoder, identical to DETR. Instead of focusing on the extreme points as observed in DETR, our decoder model focuses on the full foreground instance.

C. Additional Improvements

Deformable DETR allows us to use many versions of end-to-end object detectors due to its fast convergence and computational and memory efficiency. Inspired by two-stage object detectors [5, we examine a variant of Deformable DETR for developing region proposals as the first step.
The resulting region recommendations will be sent to the decoder as object queries for further refinement, culminating in a two-stage Deformable DETR. If object inquiries are directly set as pixels, the decoder's self-attention modules, whose complexity grows quadratically with the number of queries, will incur excessive computational and memory expenses. To circumvent this problem, we remove the decoder and replace it with an encoder-only Deformable DETR for region proposal creation. An object query is paired with each pixel, which predicts a bounding box. Before feeding the region proposals to the second step, NMS is used.

D. Transformers Architecture

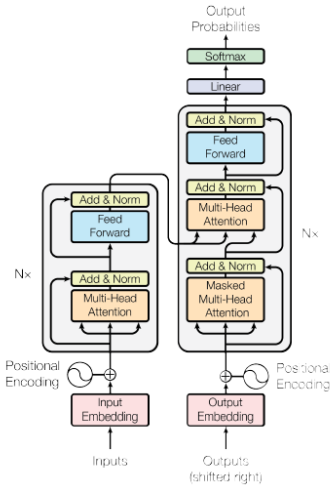

Figure 2 [7] (Vaswani et al., 2017)

Most competitive neuronal sequence transduction models [5, 10] have an encoder-decoder structure. A sequence of symbol representations (x1 to xn) is converted into a sequence of continuous representations (z = zn) via the encoder (z1 to zn). Given z, the decoder produces a symbol output sequence (y1 to ym) one element at a time. At each step [10], the model is auto-regressive, using previously created symbols as supplementary input to generate the next.
The Transformer uses layered self-attention and point-wise, totally linked layers for both the encoder and decoder, as seen in the left and right portions of Figure 2, respectively.

E. Encoder & Decoder Stack

We utilise a residual connection [2] around each of the two sub-layers, followed by layer normalisation [2]. The output of each sub-layer is LayerNorm (x + Sublayer(x)), where Sublayer(x) is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, give outputs of dimension d model = 512.*Attention*

An attention function maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The weights assigned to each value are defined by the query's compatibility function with the related key [9], and the result is a weighted sum of the values. We use the attention function to generate dv-dimensional output values by running it in parallel on each of these projected versions of queries, keys, and values.

F. Multi Head Attention

We found that linearly projecting the queries, keys, and values h times to dk, dk, and dv dimensions with independent, learned linear projections was more useful than running a single attention function with model-dimensional keys, values, and queries [8].

III. DATASET

The COCO dataset (Common Objects in Context) is a large-scale object detection dataset supplied by Microsoft that was

utilised with already trained data of 80 objects (resource is specified in abstract). It recognises things in two dimensions and three dimensions by localising their properties and characterising their relationships. The purpose of this dataset is to improve picture recognition [11]. This dataset has 80 classes, which include objects that can be simply labelled for individual occurrences (person, car, chair, etc.). It comprises 330K photos, more than 200K of which are labelled, and 1.5 million object instances that are divided into 80 object categories.

IV. Results

In DETR implementation, PyTorch framework of machine learning is used to detect the bounding boxes of COCO dataset images. In DETR architecture there are two main components; a convolution backbone (ResNet-50) and transformers (Pytorch nn). By constructing the model with 80 COCO output classes + 1 ⃠ (no object) class and load the pretrained weights, the weights are saved in half precision to save bandwidth without hurting model accuracy. The pre-trained DETR model on the 80 COCO classes, with class indices ranging from 1 to 90 defines the mapping from class indices to names. DETR uses standard ImageNet normalization, and output boxes in relative image coordinates in [***x_center, y_tcenter, w, h***] format, where [***x_center, y_center***] is the predicted center of the bounding box, and ***w***, ***h*** its width and height. Because the coordinates are relative to the image dimension and lies between [0, 1], so convert predictions to absolute image coordinates and [***x_0, y_0, x_1, y_1***] format for visualization purposes [12].

Figure 3 (Object Detection)

In Figure three the bounding boxes are created for different classes to accurate labels with score on top left of each box. DETR detects objects precisely to overlap another object class, as so can't classify under inaccurate labels, as shown in both figures 3 and 4.

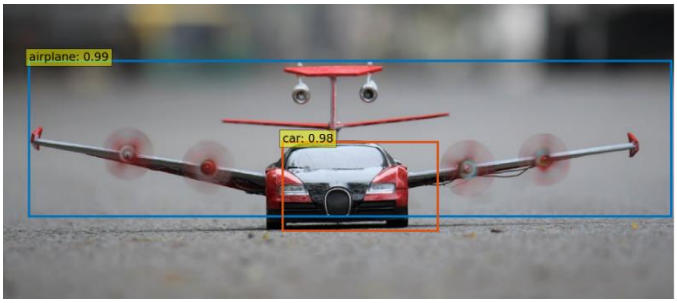

Figure 4 (Object Detection)

V. References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

[2] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. http://arxiv.org/abs/2010.04159

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and YichenWei. Deformable convolutional networks. In ICCV, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[5] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020.

[6] Figure 1 taken from Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (n.d.). *End-to-End Object Detection with Transformers*.

[7] Figure 2 taken from Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In Proc. of NAACL, 2016.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122v2, 2017.

[10] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509, 2019.

[11] https://viso.ai/computer-vision/coco-dataset/#:~:text=The%20MS%20COCO%20dataset%20is,for%20various%20computer%20vision%20projects.

[12] Code, Facebook AI https://github.com/facebookresearch/detr