End-to-End Object Detection with Transformers [1]

Minji Kim (minji@snu.ac.kr; 2020-28702), Dept. of Electrical and Computer Engineering, Seoul National University

1. Introduction

Previous modern object detectors such as Faster R-CNN [3] require hand-crafted components such as pre-defined anchor generation and non-maximum suppression (NMS) post-processing. Such pipelines are not fully end-to-end and require manual hyper-parameter adjustment for each dataset. To address this issue, this paper suggests a new detection method, called DEtection Transformer (DETR), which sees object detection as a direct set prediction problem. DETR predicts all objects at once based on end-to-end training with a set loss function which helps bipartite matching between prediction and ground-truth objects.

2. DETR

2.1. Architecture

DETR consists of three main component: a CNN backbone to extract a feature representation, an encoder-decoder transformer, and a lightweight feed forward network (FFN) that generates the final detection prediction. The overall pipeline is illustrated in Fig. 1. DETR first extracts a 2D feature representation from a standard CNN backbone. The flattened feature is combined with a positional encoding and passes through a transformer encoder. Along with the encoded features, a transformer decoder takes as input a small fixed number of learned positional embeddings, called *object queries*, resulting the decoded features where the positional and semantic information are attended. Finally, each output embeddings of the decoder passes through a shared feed forward network (FFN) that predicts class, bounding box, with a “no object” class.

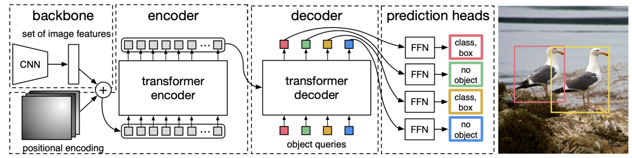

Figure 1. The overall pipeline of DETR. DETR consists of CNN backbone, encoder-decoder for transformer, and multiple prediction heads with shared feed-forward networks.

2.2. Object Detection Set Prediction Loss

Given y the ground truth set of objects and the set of N predictions $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$, we find a bipartite matching by searching for a permutation of N elements $\sigma \in C_N$ with the lowest cost:

$$\hat{\sigma} = \arg\min_{\sigma \in C_N} \sum_{i}^{N} \mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)}), \tag{1}$$

where $\mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)})$ is a pair-wise matching cost between ground truth y_i and a prediction with index $\sigma(i)$. Here, each element i of the ground truth set is $y_i = (c_i, b_i)$ where c_i is the target class label (including no-object) and $b_i \in [0,1]^4$ that represents a relative coordinate and size of a bounding box. This matching procedure plays same role with the previously used proposal matching or anchors, but the main difference is that in this scenario the matching should be one-to-one without any duplication. Finally, the Hungarian loss for all matched pairs can be obtained as follows:

$$\mathcal{L}(y, \hat{y}) = \sum_{i=1}^{N} [-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{c_i \neq \emptyset} \mathcal{L}_{box}(b_i, \hat{b}_{\hat{\sigma}}(i))], \tag{2}$$

where $\hat{\sigma}$ is the optimal assignment found in the previous step. The bounding box loss $\mathcal{L}_{box}$ is a linear combination of the l_1 loss and the generalized IoU loss [4].

3. Discussion

DETR is the first detection framework to integrate Transformers into the detection pipeline. It eliminates the burdensome hand-craft components of previous detection frameworks. In experiments on COCO [2] dataset, it achieves competitive performance with previous modern object detectors such as Faster R-CNN. However, the convergence of DETR is 10x 20x slower than Faster R-CNN, and its feature resolution is limited, showing limited performance compared to the Faster R-CNN with FPN setting.

References

[1] N. Carion, et al. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[2] T.-Y. Lin, et al. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[3] S. Ren, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1

[4] H. Rezatofighi, et al. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 1