Supplementary Material
DESTR: Object Detection with Split Transformer

First Author
Institution1
Institution1 address
firstauthor@i1.org

Second Author
Institution2
First line of institution2 address
secondauthor@i2.org

This supplementary material presents additional experimental results. For the evaluation settings and implementation details, please refer to the paper. In the following, we evaluate: 1) Number of training epochs for DESTR, 2) Object detection produced directly by the mini-detector, 3) Comparison to C-DETR with similar #Params, 4) Object detection from different decoder layers (a.k.a. stages). Finally, we illustrate some additional qualitative results for the proposed classification and regression cross-attention.

1. Number of Training Epochs

As mentioned in the paper, C-DETR [2] significantly reduces the training epochs in comparison to DETR [1], because C-DETR conditions the positional embedding of each query with the corresponding decoder output embedding of the previous stage. However, their content embedding is inferred from scratch. Our DESTR uses a mini-detector to initialize the content and positional embeddings of the decoder, which further reduces the training epochs relative to that of C-DETR, as shown in Tab. 1.

Tab. 1 compares the performance of C-DETR-R50 and DESTR-R50, with ResNet-50 as the backbone network, on COCO 2017 val, for different numbers of training epochs. The learning rate is decreased by 0.1 at epoch 20, 40, 60, and 80, respectively. For the results of C-DETR trained with 25 epochs in Tab. 1, we follow C-DETR's training script provided on the authors' GitHub.

From Tab. 1, DESTR converges faster than C-DETR. At epoch 25, DESTR-R50 outperforms C-DETR by 2.8 in AP, and at epoch 50, DESTR-R50 outperforms C-DETR-R50 with 108 epochs of training.

Model	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
C-DETR-R50 [2]*	25	37.3	58.2	39.5	17.7	40.6	55.7
C-DETR-R50 [2]	50	40.9	61.8	43.3	20.8	44.6	59.2
C-DETR-R50 [2]	75	42.1	62.9	44.8	21.6	45.4	60.2
C-DETR-R50 [2]	108	43.0	64.0	45.7	22.7	46.7	61.5
DESTR-R50	25	40.1	61.5	42.4	20.9	43.2	58.5
DESTR-R50	50	43.6	64.7	46.5	23.6	47.5	62.1
DESTR-R50	75	**43.8**	**64.8**	**46.7**	**23.8**	**47.6**	**62.3**

Table 1. The performance of DESTR and C-DETR with ResNet-50 as the backbone network for the increasing number of training epochs. "*" denotes we train C-DETR with the training script provided at the authors' GitHub.

2. Evaluation of the Mini-detector and Ablation of Decoder Layer Output

Tab. 2 shows object detection results produced directly by the the mini-detector. Also, Tab. 2 tests ablations of the decoder stages for DESTR-R50 trained with 50 epochs. As can be seen, the mini-detector does not give good results (e.g., relative to FCOS [3]), and hence is used only for initializing the object queries. Also, as the number of decoder stages grows, the performance steadily improves.

3. Comparison to C-DETR with Similar #Params

The table 3 shows how our performance changes for different numbers of parameters (#Params). Our paper reports results for a DESTR architecture that simply adds our contributions to C-DETR, for fair comparison. This kind of extension, of course, increases #Params relative to C-DETR, as reported in Tab. 1 in paper. But our contributions also allow other design choices. As shown in the table below, when we reduce the dimension of feature embedding in Transformer from 256 to 160, and reduce the number of heads, the resulting DESTR architecture has a similar #Params as C-DETR, and still outperforms C-DETR. Hence, our performance improvement fundamentally stems from our contributions – not from increasing #Params.

4. Additional Qualitative Results

Fig. 1 shows cross-attention maps on some example images from COCO-val, estimated by: (a) DETR trained with 500 training epochs, and (b) C-DETR trained with 50

output layers	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
mini-det	36.2	58.7	38.3	20.2	41.2	47.1
layer 1	38.0	59.9	40.5	20.7	42.1	52.3
layer 2	40.6	61.9	43.1	21.5	44.4	56.8
layer 3	42.3	63.4	45.0	22.8	46.2	59.5
layer 4	43.1	64.3	46.1	23.3	47.1	60.9
layer 5	43.4	64.5	46.3	23.5	47.3	61.7
layer 6	**43.6**	**64.7**	**46.5**	**23.6**	**47.5**	**62.1**

Table 2. Object detection results on COCO-val prduced by the mini-detector and different decoder layers (a.k.a. stages).

model	hidden dim	nhead	Gflops	#params (M)	AP
C-DETR-R50	256	8	90.0	44	40.9
C-DETR-R50	160	8	81.8	34	38.9
C-DETR-R50	160	5	81.6	34	39.5
DESTR-R50	256	8	104.4	69	43.6
DESTR-R50	160	8	87.1	46	41.9
DESTR-R50	160	5	86.9	46	**42.4**

Table 3. Comparison of our DESTR and C-DETR on COCO-val when the dimension of embedding is reduced from 256 to 160, and the number of Transformer heads is reduced from 8 to 5. The backbone is R50, and the results are obtained after 50 training epochs.

epochs. (c) DESTR's classification cross-attention, and (d) DESTR's regression cross-attention. DESTR is trained with 50 epochs. As can be seen, DESTR's classification cross-attention focuses on discriminative object parts, whereas DESTR's regression cross-attention highlights horizontal and vertical edges in the image.

Fig. 2 illustrates five out of eight attention multi-heads of C-DETR, and the classification and regression cross-attention of DESTR, for the heads which focus on the object's four boundaries and center, for a few example images from COCO-val. The attention produced by the other three heads are duplicates of the illustrated five. As can be seen, the attention heads of C-DETR and DESTR that focus on the object's four boundaries are similar. But for the head that represents the object's center, the DESTR's classification and regression cross-attention are much broader than C-DETR's attention. In addition, DESTR's center head of classification cross-attention gets wider response from object's center appearance. Moreover, DESTR's center head of regression cross-attention is even larger, including some regions that interact with other objects or background. We think this is due to the intrinsic differences between the classification and regression tasks. This justifies our design to separate these two tasks into two independent branches in the decoder.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

[2] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. *arXiv preprint arXiv:2108.06152*, 2021. 1

[3] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 1

Figure 1. Cross-attention maps estimated by: (a) DETR trained with 500 training epochs, and (b) C-DETR trained with 50 epochs. (c) DESTR's classification cross-attention focuses on discriminative object parts. (d) DESTR's regression cross-attention highlights horizontal and vertical edges in the image. DESTR is trained with 50 epochs. R50 is used as the backbone for all of the three models. For better visualization, the figure shows square-root values of cross-attention. Warmer colors indicate higher cross-attention values.

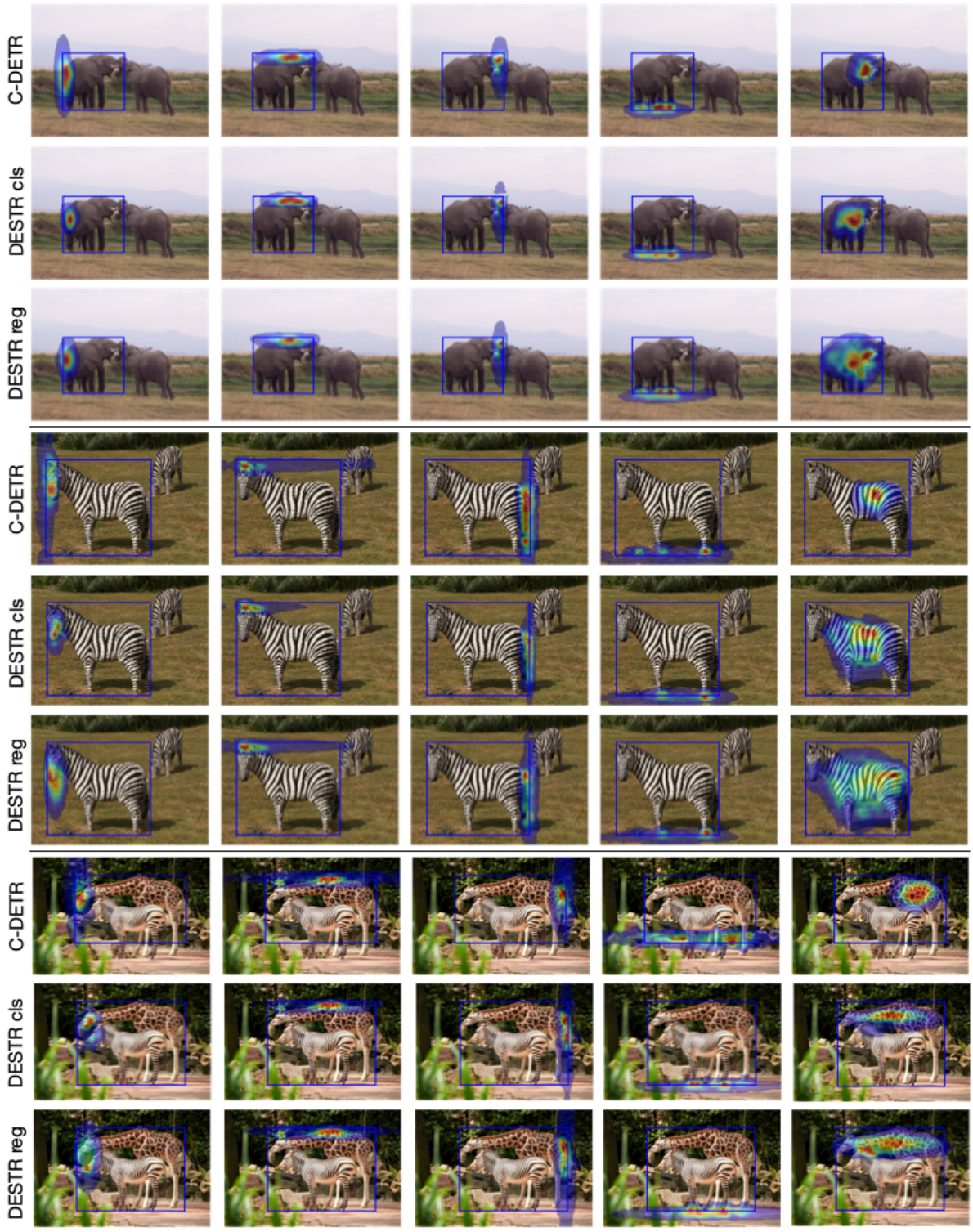

Figure 2. 5 out of 8 attention multi-heads. This figure visualizes the cross-attention map for the heads which focus on the object's four boundaries and center. The attentions of the other three heads are duplicates of the 5 illustrated attentions. For clarity, we show square-root values of attention.