كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

Selected Topics in Computer Science 4
CS496 2022-2023

Dr.Wessam Al Behaidy

PROJECT

Object detection in images

End-to-End Object Detection with Transformers

TA.Omar Tarek.

- Team NO. 06

Team Members

ID	Name
- 201900547	- غيداء الطاهر احمد محمد
- 201900573	- كاترين حبيب جورج ناداب
- 201900935	- هدير سامح داود
- 201900124	- اسامه محمد عبدالوهاب السيد
- 201900534	- عمر نبيل صابر ندا
- 201900772	- محمود طه احمد على

MAY_2023
Computer Science and Artificial Intelligencee
HELWAN UNIVERSITY

DETR

DEtection TRansformer

Implemented Paper

Citation:@article{DBLP:journals/corr/abs-2005-12872,

author = {Nicolas Carion and Francisco Massa and Gabriel Synnaeve and Nicolas Usunier and Alexander Kirillov and Sergey Zagoruyko},

title = {End-to-End Object Detection with Transformers},

journal = {CoRR},

volume = {abs/2005.12872},

year = {2020},

url = {https://arxiv.org/abs/2005.12872v3},

archivePrefix = {arXiv},

eprint ={ 2005.12872},

timestamp = {Thu, 28 May 2020 17:38:09 +0200},

biburl = {https://dblp.org/rec/journals/corr/abs-2005-12872.bib},

bibsource = {dblp computer science bibliography, https://dblp.org} }

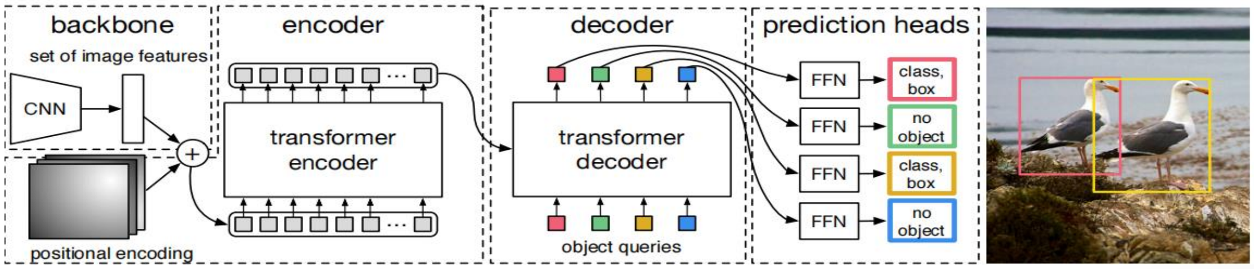
Dataset Used:They evaluated DETR on the well-known COCO 2017 object detection dataset, and other panoptic segmentation datasets, containing 118k training images and 5k validation images.

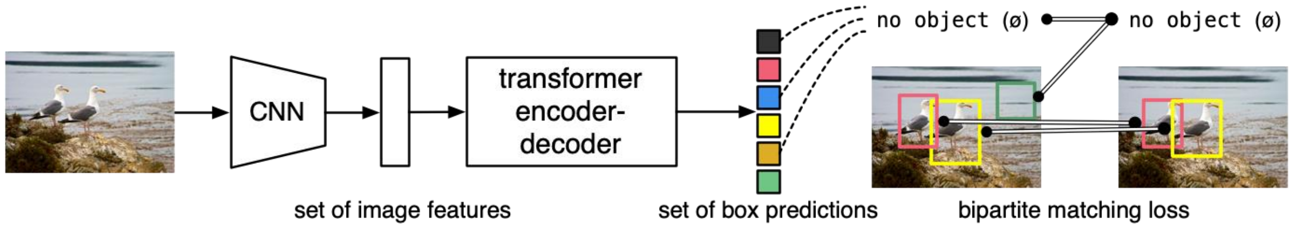
Each image is annotated with bounding boxes and panoptic segmentation. There are 7 instances per image on average, up to 63 instances in a single image in training set, ranging from small to large on the same images.

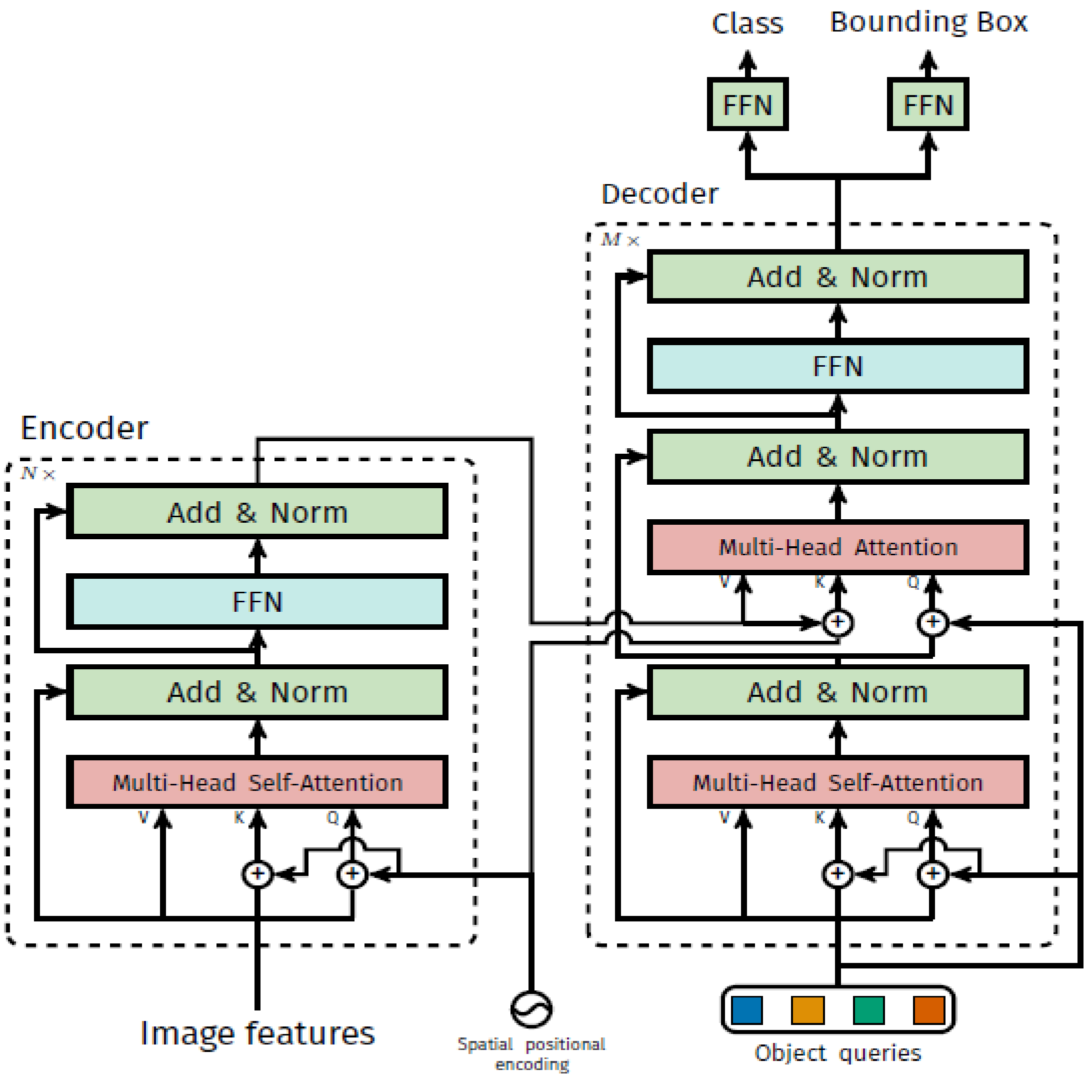
Implemented Algorithms:

The overall DETR architecture contains three main components:

a CNN backbone to extract a compact feature representation, an encoder-decoder transformer, and a simple feed forward network (FFN) that makes the final detection prediction.

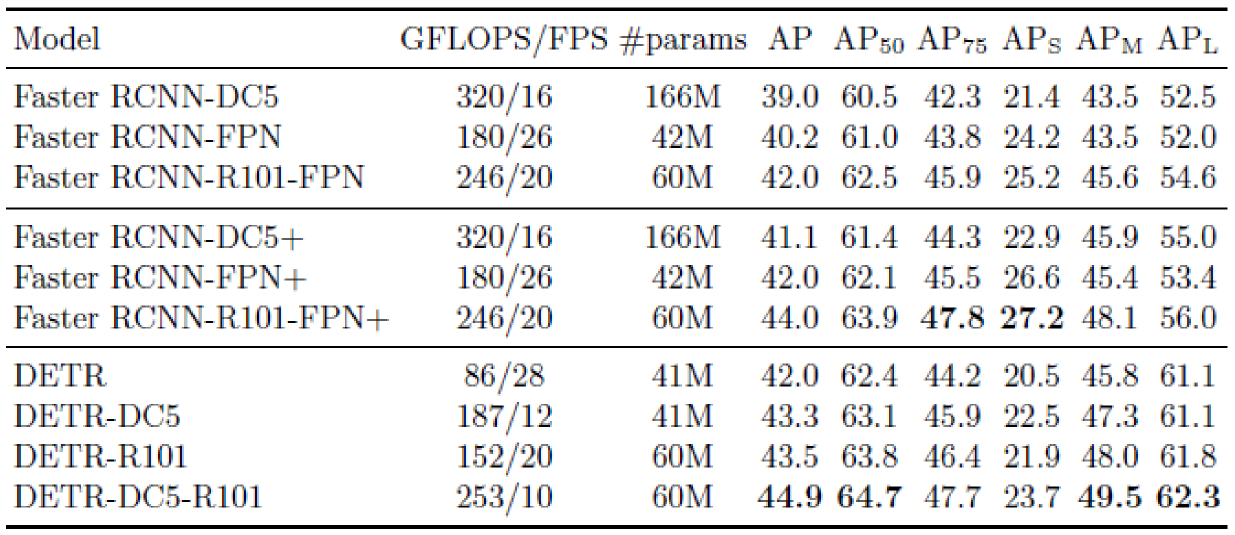






DETR uses a conventional CNN backbone (pretrained ResNet-50 or ResNet-101) to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a transformer encoder. A transformer decoder then takes as input a small, fixed number of learned positional embeddings, which are called *object queries*, and additionally attends to the encoder output. They then pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a “no object” class. DETR is trained end-to-end with a set loss function which performs **bi-partite matching** between predicted and ground-truth objects. Prediction with no match should yield a "no object" (Φ) class prediction.

Model Results: They compared the results with Faster R-CNN and reported validation AP at the last training epoch, for cut outs they reported the median over validation results from the last 10 epochs.



Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Comparison with Faster R-CNN with a ResNet-50 and ResNet-101 backbones on the COCO validation set.

DETR can be competitive with Faster R-CNN with the same number of parameters, achieving an AP (average precision) of **42.0** on COCO 2017 validation subset.

Project Documentation

Selected Dataset:

Name_ Car Object Detection

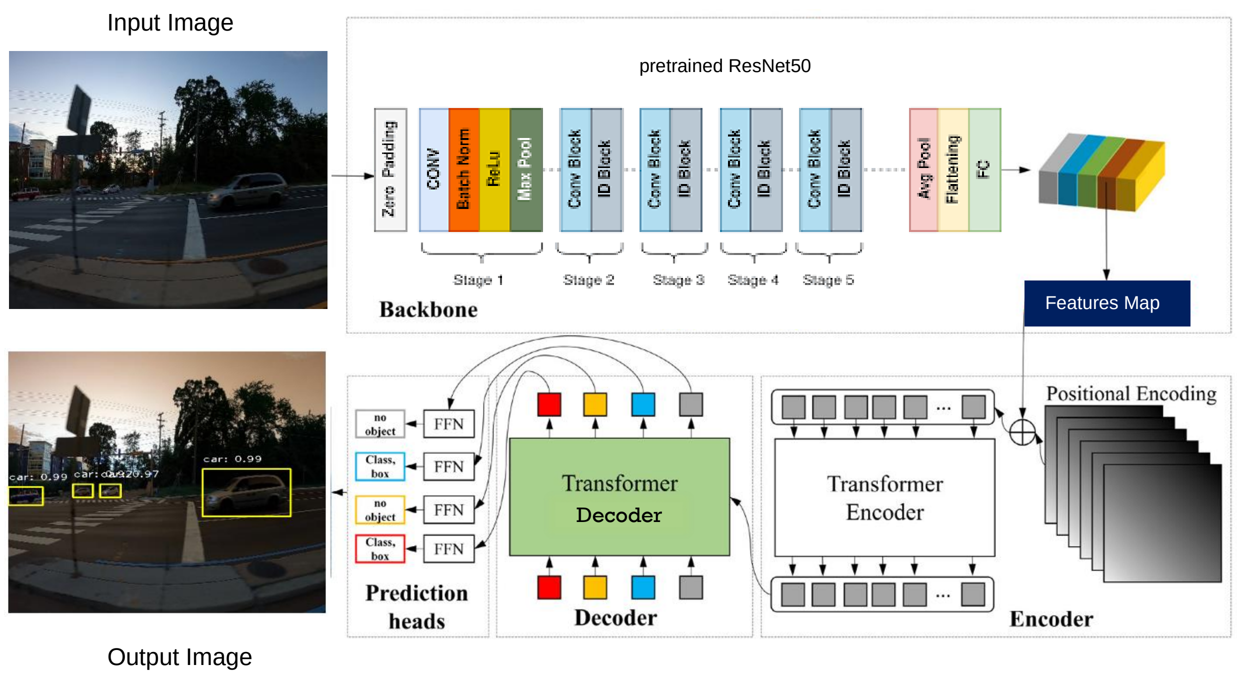
Link_ https://www.kaggle.com/datasets/sshikamaru/car-object-detection

Samples_ 1001 training-images & 175 testing-images.

Dimension of images_ 676 x 380

Class_ 1 class with 'car' label.

Implementation Details:

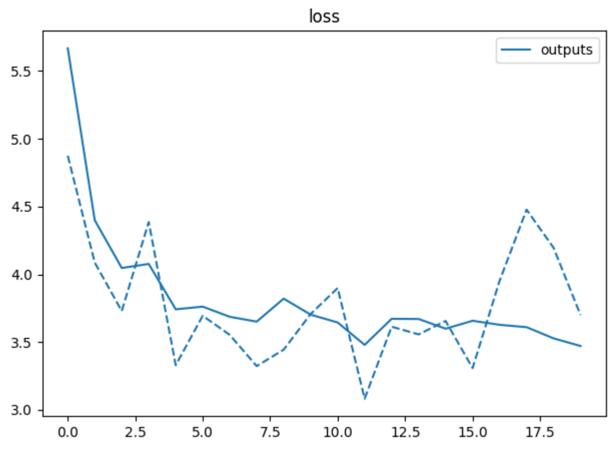


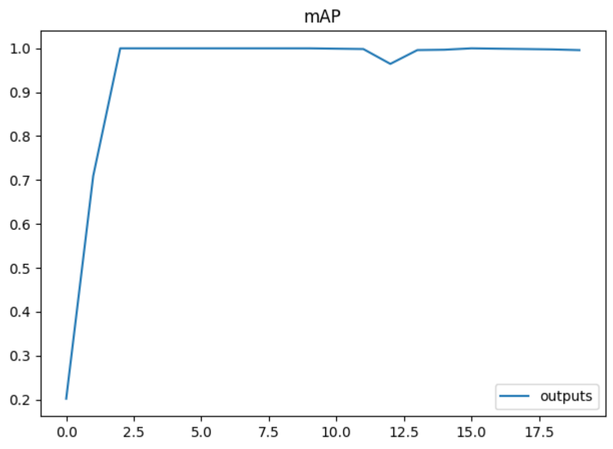
We used the DETR model architecture with backbone ResNet50 _pretrained on COCO Dataset_ to train the model on a car object detection dataset that we imported from Kaggle.
The annotations of the training images were recorded in a csv file format, to train the model we had to convert them to Jason file format. We divided the training images into 95% to 5% train-validation subsets respectively. Then we trained the model on a Colab notebook with PyTorch using 1TeslaGPU accelerator.

Hyperparameters used: Train-test split ratio, 85% to 15% respectively.
Learning_rate, 0.0001.
Number of iterations (epochs), 20.
Batch size, 6.
Number of classes, 1.
Number of workers (dataloaders), 1.
Backbone, ResNet50 with COCO pretrained weights.
And a list of other parameters which were with default values in the DETR model training.

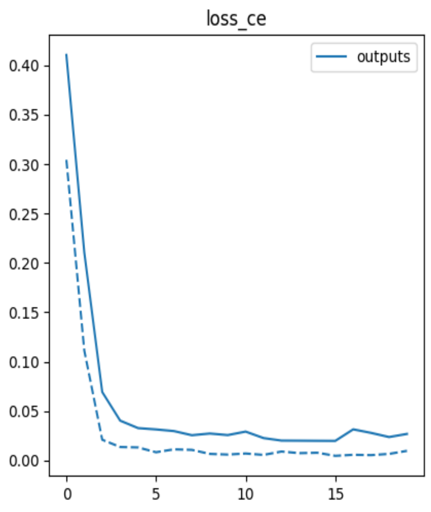
Result & evaluation Details:

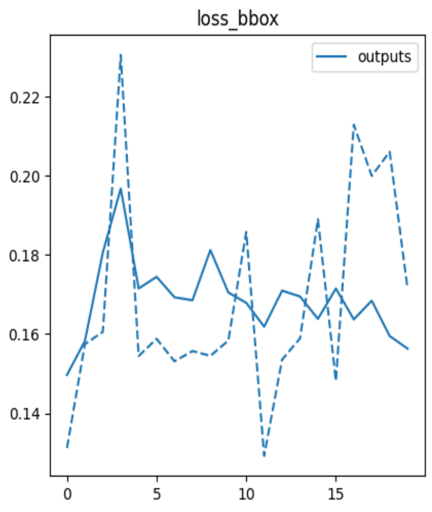
Evaluation: The mean average precision (mAP) metric is used to show our model's accuracy, we also showed the validation loss (loss) metric to assess the performance of the model on the validation set.

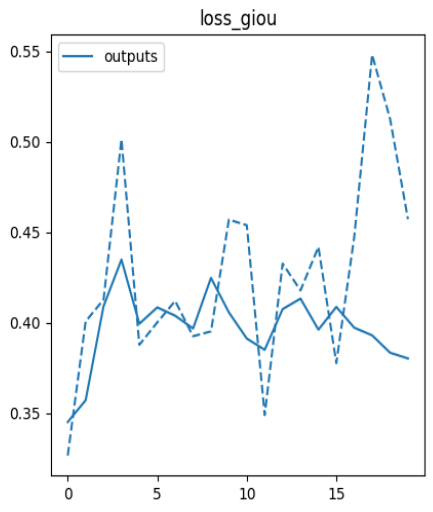


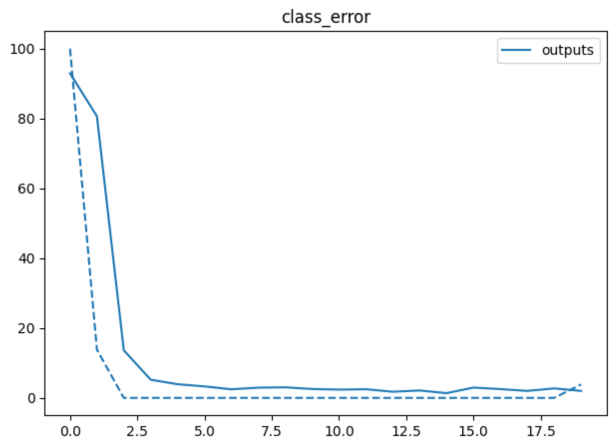


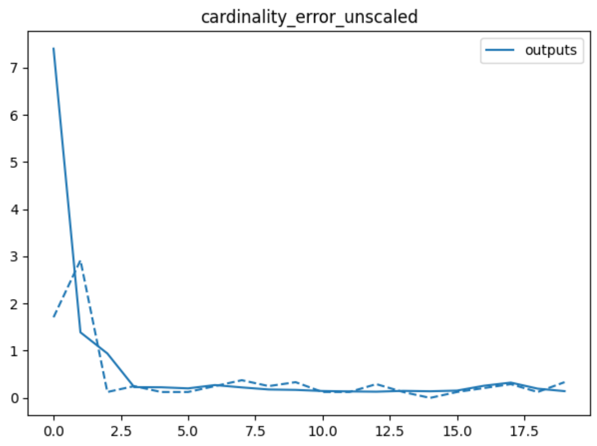
We added other metrics for more details such as Cross-entropy loss (loss_ce) metric to measure how good the model is at predicting the correct classes at the location of the predicted bounding boxes, Bounding box loss (loss_bbox) metric a loss to measure how "tight" the predicted bounding boxes are to the ground truth object and, Generalized intersection over union (loss_giou) metric to measure the loss for bounding box regression.











Results on testing data: