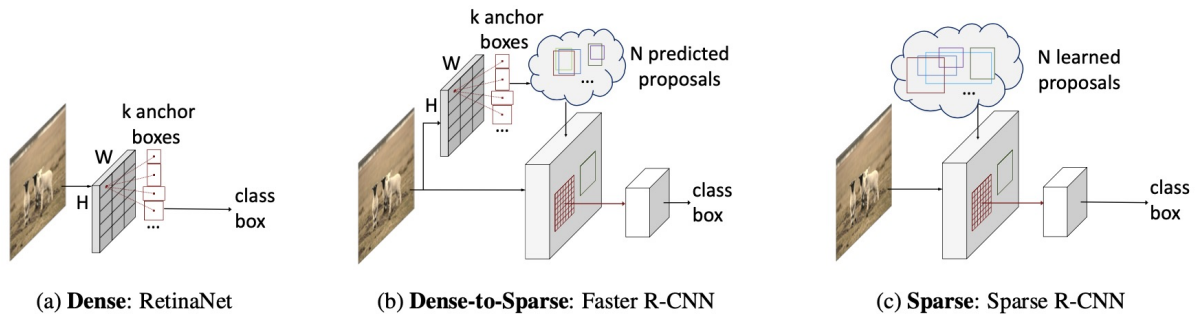

Sparse R-CNN: End-to-End Object Detection with Learnable Proposals

License MIT



Paper (CVPR 2021)

Sparse R-CNN: End-to-End Object Detection with Learnable Proposals

Updates

- (02/03/2021) Higher performance is reported by using stronger backbone model PVT.
- (23/02/2021) Higher performance is reported by using stronger pretrain model DetCo.
- (02/12/2020) Models and logs(R101_100pro_3x and R101_300pro_3x) are available.
- (26/11/2020) Models and logs(R50_100pro_3x and R50_300pro_3x) are available.
- (26/11/2020) Higher performance for Sparse R-CNN is reported by setting the dropout rate as 0.0.

Models

Method	inf_time	train_time	box AP	download
R50_100pro_3x	23 FPS	19h	42.8	model log
R50_300pro_3x	22 FPS	24h	45.0	model log
R101_100pro_3x	19 FPS	25h	44.1	model log
R101_300pro_3x	18 FPS	29h	46.4	model log

If download link is invalid, models and logs are also available in Github Release and Baidu Drive by code wt9n.

Notes

- We observe about 0.3 AP noise.
- The training time is on 8 GPUs with batchsize 16. The inference time is on single GPU. All GPUs are NVIDIA V100.
- We use the models pre-trained on imagenet using torchvision. And we provide torchvision's ResNet-101.pkl model. More details can be found in the conversion script.

Method	inf_time	train_time	box AP	codebase
R50_300pro_3x	22 FPS	24h	45.0	detectron2
R50_300pro_3x.detc	22 FPS	28h	46.5	detectron2
PVTSmall_300pro_3x	13 FPS	50h	45.7	mmdetection
PVTv2-b2_300pro_3x	11 FPS	76h	50.1	mmdetection

Installation

The codebases are built on top of Detectron2 and DETR.

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.5 and torchvision that matches the PyTorch installation. You can install them together at pytorch.org to make sure of this
- OpenCV is optional and needed by demo and visualization

Steps

1. Install and build libs

```
1 git clone https://github.com/PeizeSun/SparseR-CNN.git
2 cd SparseR-CNN
3 python setup.py build develop
```

2. Link coco dataset path to SparseR-CNN/datasets/coco

```
1 mkdir -p datasets/coco
2 ln -s /path_to_coco_dataset/annotations datasets/coco/annotations
3 ln -s /path_to_coco_dataset/train2017 datasets/coco/train2017
```

```
4 ln -s /path_to_coco_dataset/val2017 datasets/coco/val2017
```

3. Train SparseR-CNN

```
1 python projects/SparseRCNN/train_net.py --num-gpus 8 \  
2   --config-file projects/SparseRCNN/configs/sparsercnn.res50.100pro.3  
   x.yaml
```

4. Evaluate SparseR-CNN

```
1 python projects/SparseRCNN/train_net.py --num-gpus 8 \  
2   --config-file projects/SparseRCNN/configs/sparsercnn.res50.100pro.3  
   x.yaml \  
3   --eval-only MODEL.WEIGHTS path/to/model.pth
```

5. Visualize SparseR-CNN

```
1 python demo/demo.py\  
2   --config-file projects/SparseRCNN/configs/sparsercnn.res50.100pro.3  
   x.yaml \  
3   --input path/to/images --output path/to/save_images --confidence-  
   threshold 0.4 \  
4   --opts MODEL.WEIGHTS path/to/model.pth
```

Third-party resources

- mmdetection implementation: sparse_rcnn. Thank Shilong Zhang!
- cvpod implementation:sparse_rcnn. Thank Benjin Zhu!
- paddledetection implementation:sparse_rcnn. Thank FL77N!

License

SparseR-CNN is released under MIT License.

Citing

If you use SparseR-CNN in your research or wish to refer to the baseline results published here, please use the following BibTeX entries:

```
1  
2 @article{peize2020sparse,  
3   title = {{SparseR-CNN}: End-to-End Object Detection with Learnable  
   Proposals},
```

```
4   author = {Peize Sun and Rufeng Zhang and Yi Jiang and Tao Kong and  
        Chenfeng Xu and Wei Zhan and Masayoshi Tomizuka and Lei Li and  
        Zehuan Yuan and Changhu Wang and Ping Luo},  
5   journal = {arXiv preprint arXiv:2011.12450},  
6   year    = {2020}  
7 }
```