

DETECTRON 2

MLUGS

Tin Votan, 17.11.2020

Overview

Detectron2

- FAIR => Object detection
- Ursprünglich aus Mask R-CNN-Benchmark; Nachfolger von Detectron
 - ➡ Mask R-CNN-Benchmark ist veraltet
- PyTorch 1.0
- Enthält weitere Features: Panoptic Segmentation, Densepose, Cascade R-CNN, Rotated Bounding Boxes, PointRend, DeepLab, etc.
- Training ist schneller geworden -> unterstützt Single oder Multiple GPU Server

Production Use-Cases

Facebook AI Research

We are now using Detectron2 to rapidly design and train the next-generation pose detection models that power Smart Camera, the AI camera system in Facebook's Portal video-calling devices.

Verbesserungen

Detectron2

PyTorch

- Detectron war in Caffe2 implementiert; Detectron2 ist vollständig in PyTorch umgesetzt worden

Neue Models / Features

- alle aus Detectron vorhanden¹ + weitere Models² + zusätzliche Features³

Neue Tasks

- alle aus Detectron vorhanden⁴ + Unterstützung von weiteren Tasks⁵

¹ Faster R-CNN, Mask R-CNN, RetinaNet, and DensePose

² Cascade R-CNN, Panoptic FPN, and TensorMask

³ Synchronous Batch Norm, LVIS

⁴ Object detection with boxes and instance segmentation masks, as well as human pose prediction

⁵ Semantic segmentation and panoptic segmentation, a task that combines both semantic and instance segmentation

Verbesserungen

Detectron2

Implementation Quality

- Verbesserung von Detectron2 im Gegensatz zu Detectron

Geschwindigkeit und Skalierung

- Training Pipeline komplett auf GPU + Unterstützung von Distributed Training auf mehreren GPU-Servern

Detectron2go

- Einfaches Deployment von komplexeren neuen Models in Production
- Weitere Features⁶

6 standard training workflows with in-house data sets, network quantization, and model conversion to optimized formats for cloud and mobile deployment

Quellen

- <https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/>
- <https://github.com/facebookresearch/detectron2>
- <https://github.com/facebookresearch/maskrcnn-benchmark/>
- <https://towardsdatascience.com/object-detection-in-6-steps-using-detectron2-705b92575578>
- <https://detectron2.readthedocs.io/index.html>

Live Section

Detectron2

- Zusatz: SageMaker, RISE, Jupyter, OpenCV, PyYAML, Cython
- Python 3.7
- Rest siehe Pipfile
- MacOS Big Sur (Version 11.0.1) -> Probleme mit C-Compiler (Xcode)

```
≡ Pipfile
1  [[source]]
2  name = "pypi"
3  url = "https://pypi.org/simple"
4  verify_ssl = true
5
6  [dev-packages]
7
8  [packages]
9  numpy = "*"
10 pandas = "*"
11 matplotlib = "*"
12 jupyter = "*"
13 torch = "==1.5.0"
14 torchvision = "==0.5.0"
15 opencv-python-headless = "*"
16 cython = "*"
17 pyyaml = "*"
18 rise = "*"
19 ninja = "*"
20
21 [requires]
22 python_version = "3.7"
23
```

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

Build Detectron2 from Source

gcc & g++ ≥ 5 are required. [ninja](#) is recommended for faster build. After having them, run:

```
python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'
# (add --user if you don't have permission)

# Or, to install it from a local clone:
git clone https://github.com/facebookresearch/detectron2.git
python -m pip install -e detectron2

# Or if you are on macOS
CC=clang CXX=clang++ python -m pip install .....
```