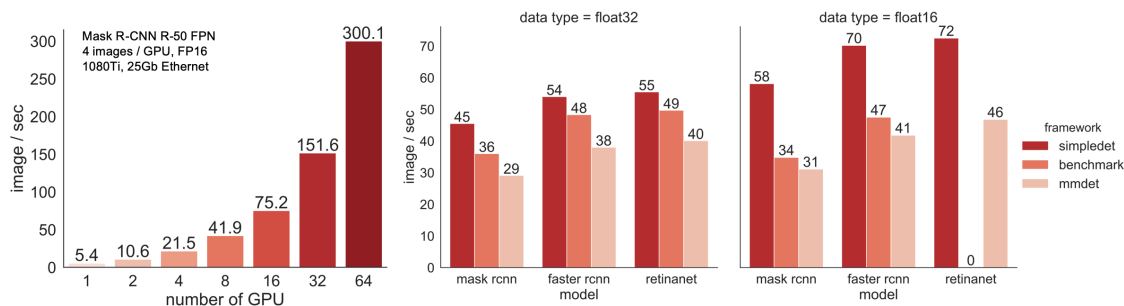

SimpleDet - A Simple and Versatile Framework for Object Detection and Instance Recognition

Major Features



- FP16 training for memory saving and up to **2.5X** acceleration - Highly scalable distributed training available **out of box** - Full coverage of state-of-the-art models including FasterRCNN, MaskRCNN, CascadeRCNN, RetinaNet, DCNv1/v2, **TridentNet**, **NASFPN**, **EfficientNet**, and **Knowledge Distillation**
- Extensive feature set including **large batch BN**, **loss synchronization**, **automatic BN fusion**, soft NMS, multi-scale train/test - Modular design for coding-free exploration of new experiment settings - Extensive documentations including annotated config, Fintuning Guide

Recent Updates

- Add RPN test (2019.05.28)
- Add NASFPN (2019.06.04)
- Add new ResNetV1b baselines from GluonCV (2019.06.07)
- Add Cascade R-CNN with FPN backbone (2019.06.11)
- Speed up FPN up to 70% (2019.06.16)
- Update NASFPN to include larger models (2019.07.01)
- Automatic BN fusion for fixed BN training, saving up to 50% GPU memory (2019.07.04)
- Speed up MaskRCNN by 80% (2019.07.23)
- Update MaskRCNN baselines (2019.07.25)
- Add EfficientNet and DCN (2019.08.06)
- Add python wheel for easy local installation (2019.08.20)
- Add FitNet based Knowledge Distill (2019.08.27)
- Add SE and train from scratch (2019.08.30)
- Add a lot of docs (2019.09.03)
- Add support for INT8 training(contributed by Xiaotao Chen & Jingqiu Zhou) (2019.10.24)
- Add support for FCOS(contributed by Zhen Wei) (2019.11)

-
- Add support for Mask Scoring RCNN(contributed by Zehui Chen) (2019.12)
 - Add support for RepPoints(contributed by Bo Ke) (2020.02)
 - Add support for FreeAnchor (2020.03)
 - Add support for Feature Pyramid Grids & PAFPN (2020.06)
 - Add support for CrowdHuman Dataset (2020.06)
 - Add support for Double Pred (2020.06)
 - Add support for SEPC(contributed by Qiaofei Li) (2020.07)

Setup

All-in-one Script We provide a setup script for install simpledet and preppare the coco dataset. If you use this script, you can skip to the Quick Start.

Install We provide a conda installation here for Debian/Ubuntu system. To use a pre-built docker or singularity images, please refer to INSTALL.md for more information.

```
1 # install dependency
2 sudo apt update && sudo apt install -y git wget make python3-dev
   libglib2.0-0 libsm6 libxext6 libxrender-dev unzip
3
4 # create conda env
5 conda create -n simpledet python=3.7
6 conda activate simpledet
7
8 # fetch CUDA environment
9 conda install cudatoolkit=10.1
10
11 # install python dependency
12 pip install 'matplotlib<3.1' opencv-python pytz
13
14 # download and intall pre-built wheel for CUDA 10.1
15 pip install https://1dv.aflat.top/mxnet_cu101-1.6.0b20191214-py2.py3-
   none-manylinux1_x86_64.whl
16
17 # install pycocotools
18 pip install 'git+https://github.com/RogerChern/cocoapi.git#subdirectory
   =PythonAPI'
19
20 # install mxnext, a wrapper around MXNet symbolic API
21 pip install 'git+https://github.com/RogerChern/mxnext#egg=mxnext'
22
23 # get simpledet
24 git clone https://github.com/tusimple/simpledet
25 cd simpledet
26 make
```

```
27
28 # test simpledet installation
29 mkdir -p experiments/faster_r50v1_fpn_1x
30 python detection_infer_speed.py --config config/faster_r50v1_fpn_1x.py
    --shape 800 1333
```

If the last line execute successfully, the average running speed of Faster R-CNN R-50 FPN will be reported. And you have successfully setup SimpleDet. Now you can head up to the next section to prepare your dataset.

Preparing Data We provide a step by step preparation for the COCO dataset below.

```
1 cd simpledet
2
3 # make data dir
4 mkdir -p data/coco/images data/src
5
6 # skip this if you have the zip files
7 wget -c http://images.cocodataset.org/zips/train2017.zip -O data/src/
    train2017.zip
8 wget -c http://images.cocodataset.org/zips/val2017.zip -O data/src/
    val2017.zip
9 wget -c http://images.cocodataset.org/zips/test2017.zip -O data/src/
    test2017.zip
10 wget -c http://images.cocodataset.org/annotations/
    annotations_trainval2017.zip -O data/src/annotations_trainval2017.
    zip
11 wget -c http://images.cocodataset.org/annotations/image_info_test2017.
    zip -O data/src/image_info_test2017.zip
12
13 unzip data/src/train2017.zip -d data/coco/images
14 unzip data/src/val2017.zip -d data/coco/images
15 unzip data/src/test2017.zip -d data/coco/images
16 unzip data/src/annotations_trainval2017.zip -d data/coco
17 unzip data/src/image_info_test2017.zip -d data/coco
18
19 python utils/create_coco_roidb.py --dataset coco --dataset-split
    train2017
20 python utils/create_coco_roidb.py --dataset coco --dataset-split
    val2017
21 python utils/create_coco_roidb.py --dataset coco --dataset-split test-
    dev2017
```

For other datasets or your own data, please check DATASET.md for more details.

Quick Start

```
1 # train
2 python detection_train.py --config config/faster_r50v1_fpn_1x.py
```

```
3
4 # test
5 python detection_test.py --config config/faster_r50v1_fpn_1x.py
```

Finetune Please check FINTUNE.md

Model Zoo Please refer to MODEL_ZOO.md for available models

Distributed Training

Please refer to DISTRIBUTED.md

Project Organization

Code Structure

```
1 detection_train.py
2 detection_test.py
3 config/
4     detection_config.py
5 core/
6     detection_input.py
7     detection_metric.py
8     detection_module.py
9 models/
10     FPN/
11     tridentnet/
12     maskrcnn/
13     cascade_rcnn/
14     retinanet/
15 mxnext/
16 symbol/
17     builder.py
```

Config Everything is configurable from the config file, all the changes should be **out of source**.

Experiments One experiment is a directory in **experiments** folder with the same name as the config file. > E.g. r50_fixbn_1x.py is the name of a config file

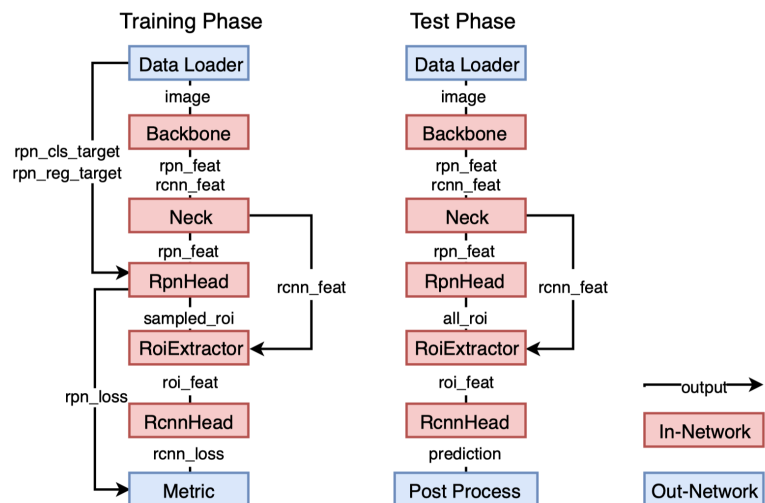
```
1 config/
2     r50_fixbn_1x.py
3 experiments/
4     r50_fixbn_1x/
```

```

5     checkpoint.params
6     log.txt
7     coco_minival2014_result.json

```

Models The `models` directory contains SOTA models implemented in SimpleDet.



How is Faster R-CNN built

Simpledet supports many popular detection methods and here we take **Faster R-CNN** as a typical example to show how a detector is built.

- *Preprocessing.* The preprocessing methods of the detector is implemented through `DetectionAugmentation`.
 - Image/bbox-related preprocessing, such as `Norm2DImage` and `Resize2DImageBbox`.
 - Anchor generator `AnchorTarget2D`, which generates anchors and corresponding anchor targets for training RPN.
- *Network Structure.* The training and testing symbols of Faster-RCNN detector is defined in `FasterRcnn`. The key components are listed as follow:
 - *Backbone.* `Backbone` provides interfaces to build backbone networks, e.g. ResNet and ResNext.
 - *Neck.* `Neck` provides interfaces to build complementary feature extraction layers for backbone networks, e.g. `FPNNeck` builds Top-down pathway for Feature Pyramid Network.
 - *RPN head.* `RpnHead` aims to build classification and regression layers to generate proposal outputs for RPN. Meanwhile, it also provides interplace to generate sampled proposals for the subsequent R-CNN.

-
- *Roi Extractor*. `RoiExtractor` extracts features for each roi (proposal) based on the R-CNN features generated by `Backbone` and `Neck`.
 - *Bounding Box Head*. `BboxHead` builds the R-CNN layers for proposal refinement.

How to build a custom detector The flexibility of **simpledet** framework makes it easy to build different detectors. We take **TridentNet** as an example to demonstrate how to build a custom detector simply based on the Faster R-CNN framework.

- *Preprocessing*. The additional processing methods could be provided accordingly by inheriting from `DetectionAugmentation`.
 - In TridentNet, a new `TridentAnchorTarget2D` is implemented to generate anchors for multiple branches and filter anchors for scale-aware training scheme.
- *Network Structure*. The new network structure could be constructed easily for a custom detector by modifying some required components as needed and
 - For TridentNet, we build trident blocks in the `Backbone` according to the descriptions in the paper. We also provide a `TridentRpnHead` to generate filtered proposals in RPN to implement the scale-aware scheme. Other components are shared the same with original Faster-RCNN.

Contributors

Yuntao Chen, Chenxia Han, Yanghao Li, Zehao Huang, Naiyan Wang, Xiaotao Chen, Jingqiu Zhou, Zhen Wei, Zehui Chen, Zhaoxiang Zhang, Bo Ke

License and Citation

This project is release under the Apache 2.0 license for non-commercial usage. For commercial usage, please contact us for another license.

If you find our project helpful, please consider cite our tech report.

```
1 @article{JMLR:v20:19-205,  
2   author  = {Yuntao Chen and Chenxia Han and Yanghao Li and Zehao Huang  
3             and Yi Jiang and Naiyan Wang and Zhaoxiang Zhang},  
4   title   = {SimpleDet: A Simple and Versatile Distributed Framework  
5             for Object Detection and Instance Recognition},  
6   journal = {Journal of Machine Learning Research},  
7   year    = {2019},  
8   volume  = {20},
```

```
7   number = {156},  
8   pages  = {1-8},  
9   url    = {http://jmlr.org/papers/v20/19-205.html}  
10 }
```