Journal of Machine Learning Research 20 (2019) 1-8 Submitted 3/19; Revised 9/19; Published 10/19

SimpleDet: A Simple and Versatile Distributed Framework for Object Detection and Instance Recognition

Yuntao Chen CHENYUNTAO2016@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China

Chenxia Han CHENXIAHAN18@GMAIL.COM

Yanghao Li LYTTONHAO@GMAIL.COM

Zehao Huang ZEHAOHUANG18@GMAIL.COM
TuSimple, Beijing, China

Yi Jiang JIANGYI0425@GMAIL.COM

Naiyan Wang WINSTY@GMAIL.COM
TuSimple, Beijing, China

Zhaoxiang Zhang ZHAOXIANG.ZHANG@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China

Editor: Balazs Kegl

Abstract

Object detection and instance recognition play a central role in many AI applications like autonomous driving, video surveillance and medical image analysis. However, training object detection models on large scale datasets remains computationally expensive and time consuming. This paper presents an efficient and open source object detection framework called *SimpleDet* which enables the training of state-of-the-art detection models on consumer grade hardware at large scale. SimpleDet covers a wide range of models including both high-performance and high-speed ones. SimpleDet is well-optimized for both low precision training and distributed training and achieves 70% higher throughput for the Mask R-CNN detector compared with existing frameworks. Codes, examples and documents of SimpleDet can be found at `https://github.com/tusimple/simpledet`.

Keywords: Object Detection, Instance Recognition, Distributed Training, Mixed Precision Training

1. Introduction

During recent years, challenging datasets and sophisticated detection models have emerged with an ever-growing demand in high performance detection framework. The public available large-scale datasets grow exponentially in size. Open Images (Kuznetsova et al., 2018) contains 1700× images compared with PASCAL VOC (Everingham et al., 2010). In order to take full advantage of large-scale datasets, the state-of-the-art CNNs also scale up in companion with the datasets. For example, EfficientNet (Tan and Le, 2019) requires 50× FLOPs compared with AlexNet (Krizhevsky et al., 2012).

These two factors bring the training time of a detector from tens of GPU hours up to tens of thousands of GPU hours, which calls for a distributed detection framework that scales out. Built on top of MXNet, SimpleDet is a high performance open source detection framework which provides an optimized training pipeline. Compared with existing frameworks, SimpleDet could achieve up to 70% higher mixed precision distributed training throughput and nearly linear scalability for the training of Mask R-CNN. Besides its high efficiency, SimpleDet also takes user experience as priority. SimpleDet features a modular design of detector and a configuration system in pure python, which could ease the use for users and also provides great extendibility. To further ease the adoption of SimpleDet, we provide pre-built Python wheels, Singularity images, and Docker containers for installation. The full codes, examples and documents of SimpleDet can be found at `https://github.com/tusimple/simpledet`.

2. Benchmark Platforms

We evaluate all frameworks on two platforms P1 and P2. P1 reflects the traditional GPGPU accelerators and P2 reflects the specialized accelerator (Tensor Core) built for Deep Learning. Detailed configuration could be found in `doc/BENCHMARK.md`

P1:

- 2X E5-2682 v4 + 8X 1080Ti + 25GbE
- CUDA 9.0 + cuDNN 7.5.0

P2:

- 2X Platinum 8163 + 8X 2080Ti + 25GbE
- CUDA 10.0 + cuDNN 7.5.0

3. Features of SimpleDet

In this section, we will introduce the model coverage, dataset support, scaling efficiency, memory saving techniques and other features for SimpleDet.

3.1. Models and Methods

SimpleDet covers both high performance (Dai et al., 2017; Zhu et al., 2019; Cai and Vasconcelos, 2018; Li et al., 2019; Ghiasi et al., 2019) and high speed detectors (Tan and Le, 2019; Redmon and Farhadi, 2018; Hinton et al., 2014), in addition to standard benchmark detectors (Ren et al., 2015; He et al., 2017; Lin et al., 2017). Furthermore, we provide an integrated knowledge distillation utility which could combine merits the high performance and high speed detectors for real-work applications.[1]

Standard Methods:

- Faster R-CNN
- Mask R-CNN
- RetinaNet

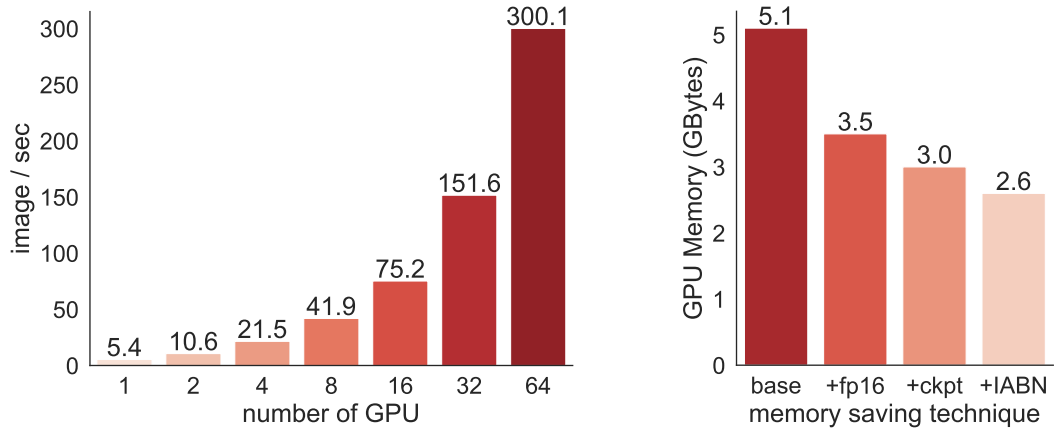
Performance-Oriented Methods:

- DCN v1/v2
- Cascade R-CNN
- TridentNet
- NASFPN

Speed-Oriented Methods:

- EfficientNet
- YOLOv3 (WIP)
- FCOS
- Knowledge Distill

1. Up to git commit 03b808, Sep 2019, we will keep this project up to date.



(a) (b)

Figure 1: (a) Scaling efficiency of SimpleDet. Throughput results are measured with a R50-FPN Mask R-CNN on the P1 platform. (b) GPU memory usages ablation with different memory saving techniques. Training memory usages with mixed precision training(+fp16), memory checkpoint(+ckpt), and in-place activation BN(+IABN) are reported. Results are obtained with a R50-C4 Faster R-CNN.

Besides a good coverage of models, SimpleDet also provides extensive pre-processing and post-processing routines in detection with their best practices, including various data augmentation techniques, multi-scale training and testing, soft NMS (Bodla et al., 2017) and weighted NMS, etc. All these features are provided based on the unified and versatile interfaces in SimpleDet, which allows the users to easily customize and extend these features in training.

3.2. Supported Datasets

SimpleDet provides generic parsing utilities for both COCO-style and VOC-style annotations. These tools allow users to utilize a wide range of existing annotations (Wang et al., 2019). A light-weight JSON format of annotation is also supported, which enables users to train on their own data. More details can be found in `doc/DATASET.md`. Currently supported datasets includes

1. Cityscapes
2. COCO
3. DeepLesion
4. DOTA
5. Kitchen
6. KITTI
7. VOC
8. WiderFace

3.3. Distributed Training

In order to take full advantage of ever-growing data, we need to utilize data parallelism to scale out to multiple machines. The efficiency of parameter communication lies in the core of scalable distributed training. Thanks to the underlying MXNet, SimpleDet supports

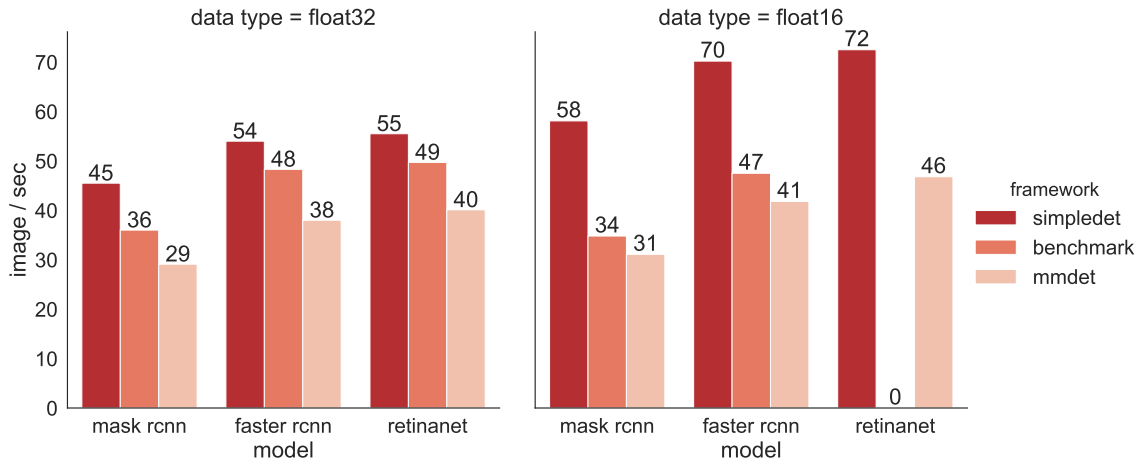


Figure 2: Compare with other detection frameworks on various detectors with different input data types. Here benchmark denotes maskrcnn-benchmark. Throughput results are obtained on the P2 platform with the R50-FPN backbone. RetinaNet with FP16 are not supported on maskrcnn-benchmark@24c8c9, so the result is 0.

both parameter server and all-reduce algorithms for model parameter update. Sophisticated communication algorithms like gradient aggregation and compression are also available out-of-the-box. Along with the mixed precision training technique which will be introduced in the next section, SimpleDet could give nearly linear scaling efficiency for a 8-node cluster as shown in Figure 1a. SimpleDet achieves 20% higher performance on the training of Mask R-CNN than MMDet with much lower end harwards (1080Ti vs V100, and 25Gb Ethernet vs 100Gb Infiniband).

3.4. Mixed Precision Training

Modern specialized hardwares like NVIDIA Tensor Core provide 10 times throughput for computation in half precision (FP16) over traditional GPGPUs. Besides speed up the training, low precision training also reduces the memory footprint and in turn saving the communication bandwidth. The main obstacle for the adoption in mixed precision training is the convergence issue and accuracy drop due to the limited range of representation. SimpleDet implements the scale loss proposed by Micikevicius et al. (2018) to mitigate these problems. In practice, we find that the mixed precision training yields identical training curves and detection mAP as the full precision one.

3.5. INT8 Training

Low-bit neural networks have been gaining more popularity in practice. The reduced storage usage, memory footprint and latency makes low-bit networks especially favorable. The traditional post-training quantization techniques work well in general scenarios but provide little flexibility for the corner cases. *SimpleDet* provides an integrated pseduo-int8 training pipeline, which enables the quantization of a wider range of models and provides more

flexibility for the users. In practice, we find that our minmax-based quantization method gives similar detection mAP as the full precision one.

3.6. Beyond Fixed Batch Normalization

Due to the limit of GPU memory, modern detectors are trained in a $1 \sim 2$ images per GPU setting. But BN is usually implemented in a per-GPU manner, which forces researchers to freeze BN statistics and parameters during training as a workaround. As indicated by Peng et al. (2018), fixed BN detectors trained with linear learning rate scaling scheme fail to converge when the total batch size increases beyond a threshold. This harms the scalability of a detection framework. In order to mitigate this limitation, SimpleDet integrates Cross-GPU Batch Normalization(CGBN) and Group Normalization. These normalization methods are available to users as a one-line configuration. In practice, we find that scaling a detector to a mini-batch size of 256 with CGBN or GN leads to stable convergence.

3.7. Memory Saving Techniques

A major limiting factor for the design of new detectors is the amount of memory available for a single GPU. Since the main training paradigm of CNN detector is data parallelism, designs are bounded by the amount of memory that a single GPU could provide. To mitigate this problem, SimpleDet combines mixed precision training, in-place activation batch normalization (Rota Bulò et al., 2018), computation graph merge, and layer-wise memory checkpointing (Chen et al., 2016) together to minimize the demand of GPU memory. Combining all these techniques, SimpleDet could save up to 50% memory as in Figure 1b with a marginal increase in computation cost compared with the vanilla setting.

4. Framework Design

Figure 3 gives a brief overview of the API design of SimpleDet. SimpleDet explicitly separates the implementation of the training and the test phases to reduce the code complexity. For example, `RpnHead` gives all proposals for test while only sampled proposals for training. Components are well-decoupled and only needs to process a small sets of input and output tensors. More details could be found in `doc/FRAMEWORK_OVERVIEW.md`.

5. Comparison with Other Detection Frameworks

We compare four other detection frameworks with SimpleDet in terms of training speed, supported models and advanced features in Table 1.

1. `detectron`[2] is the first general framework for object detection. But its training speed is a major problem as it uses Python operators in the core part of the framework extensively.

2. `mmdetection`(Chen et al., 2019) is a well-designed framework written in PyTorch which supports a wide range of detection models. Again, the training speed is a major problem of `mmdetection` as demonstrated in Figure 2.

2. `https://github.com/facebookresearch/Detectron`

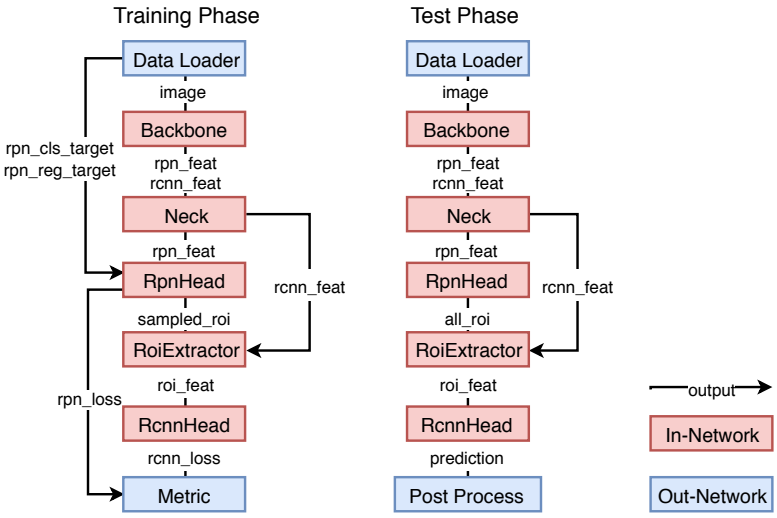


Figure 3: An overview of a generalized R-CNN detector in SimpleDet. The colored rectangles denotes components for a detector and the arrowed lines denote input and output tensors.

3. `tensorpack`[3] supports some advanced training features like Cross-GPU Batch Normalization and distributed training, but it lacks supports of some new models.

4. `maskrcnn-benchmark`[4] is a well optimized framework with amazing training speed. But it supports the least models of all frameworks. The apex-based FP16 support is also quite limited.

	`detectron`	`tensorpack`	`mmdetection`	`maskrcnn-benchmark`	`simpledet`
Commit ID	6efa99	cda5fd	d71184	24c8c9	2d8144
R50-FPN FRCNN Speed	29 images/s	29 images/s	38 images/s	48 images/s	54 images/s
FasterRCNN	✓	✓	✓	✓	✓
MaskRCNN	✓	✓	✓	✓	✓
RetinaNet	✓	✗	✓	✓	✓
Other Models	few	few	lots	few	some
Cross-GPU BN	✗	✓	✓	✗	✓
Mixed Precision Training	✗	✗	✓	✓	✓
Distributed Training	✗	✓	✓	✗	✓
Memory Checkpointing	✗	✗	✓	✗	✓

Table 1: Comparison of the training speed and supported features for `detectron`, `tensorpack`, `mmdetection`, `maskrcnn-benchmark` and `simpledet`.

3. https://github.com/tensorpack/tensorpack/tree/master/examples/FasterRCNN
4. https://github.com/facebookresearch/maskrcnn-benchmark

6. Conclusion

In this work, we present the SimpleDet framework for object detection and instance segmentation. SimpleDet features optimized mixed precision training and nearly linear scaling distributed training over 25Gb Ethernet, which achieves 70% higher throughput for Mask R-CNN compared with existing frameworks. It also integrates various memory-saving techniques to enable the training of large detectors. Besides, SimpleDet covers a wide range of detection models and datasets. We hope that this framework could help users design and benchmark new detection systems more efficiently.

References

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *International Conference on Computer Vision*, 2017.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv:1604.06174*, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *International Conference on Computer Vision*, 2017.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *International Conference on Computer Vision*, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems Workshop*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *International Conference on Computer Vision*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, 2017.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. MegDet: A large mini-batch object detector. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of DNNs. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets v2: More deformable, better results. In *Conference on Computer Vision and Pattern Recognition*, 2019.