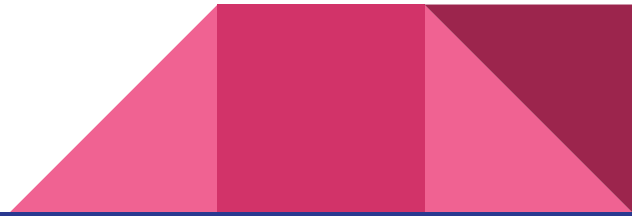
Designing Object Detectors with MMDetection

Alan L. Yuille

Bloomberg Distinguished Professor
Cognitive Science and Computer Science
Johns Hopkins University

Outline

- Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution." CVPR 2021.
- Hao Ding, Siyuan Qiao, Alan Yuille, Wei Shen. Deeply Shape-guided Cascade for Instance Segmentation. CVPR 2021.

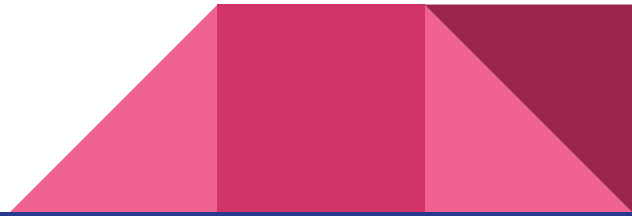


DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution

Siyuan Qiao, Liang-Chieh Chen, Alan Yuille

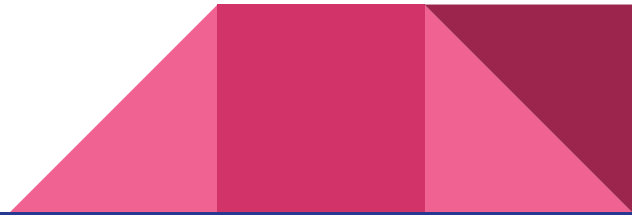
Motivation

- It is conjectured that visual perception selectively enhances and suppresses neuron activation by passing high-level information through feedback connections.



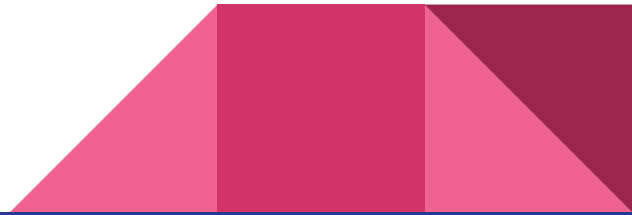
Motivation

- Human visual perception selectively enhances and suppresses neuron activation by passing high-level information through feedback connections.
- In computer vision, the mechanism of looking and thinking twice:
  - Two-stage object detectors.
  - Cascade R-CNN: a multi-stage detector.

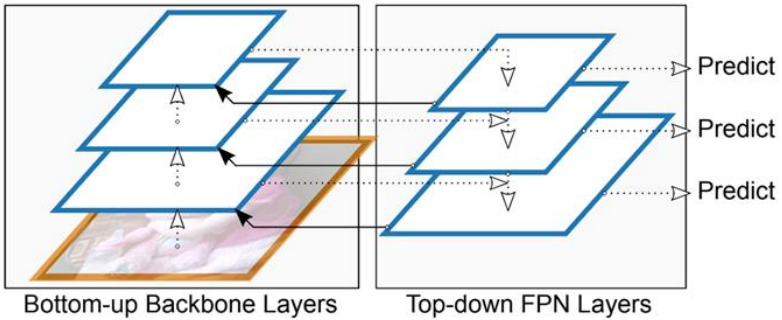


Motivation

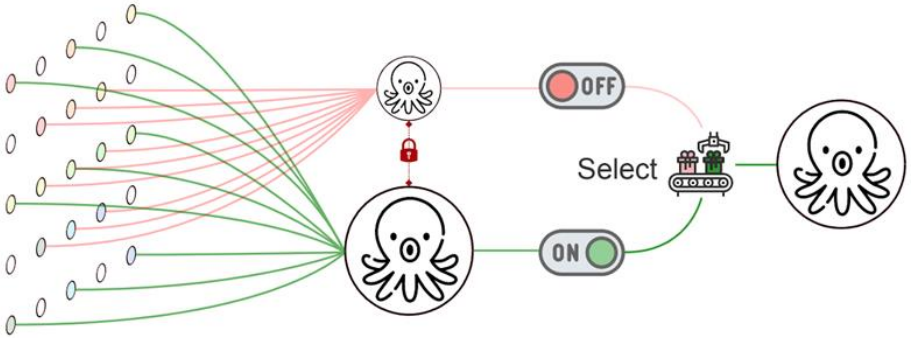
- Human visual perception selectively enhances and suppresses neuron activation by passing high-level information through feedback connections.
- In computer vision, the mechanism of looking and thinking twice:
  - Two-stage object detectors.
  - Cascade R-CNN: a multi-stage detector.
- Explore it in the backbone design?



Two techniques

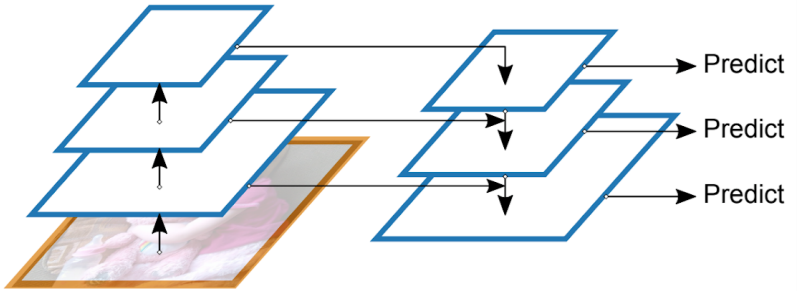


(a) Macro Design: Recursive Feature Pyramid.



(b) Micro Design: Switchable Atrous Convolution.

Recursive Feature Pyramid (RFP)

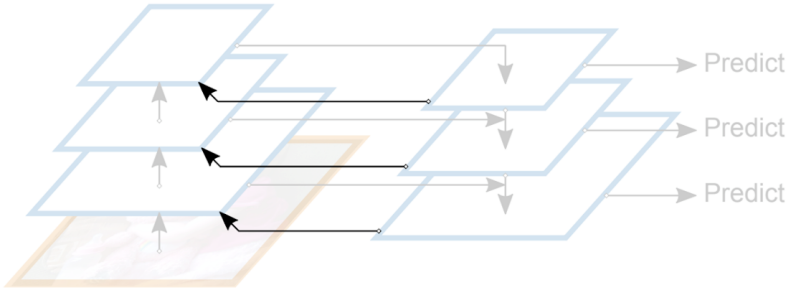


(a) FPN.

Let Bi denote the i-th stage of the bottom-up backbone, and Fi denote the i-th top-down FPN operation. FPN is then defined by

$$\mathbf{f}_i = \mathbf{F}_i\left(\mathbf{f}_{i+1}, \mathbf{x}_i\right), \quad \mathbf{x}_i = \mathbf{B}_i\left(\mathbf{x}_{i-1}\right),$$

Recursive Feature Pyramid (RFP)

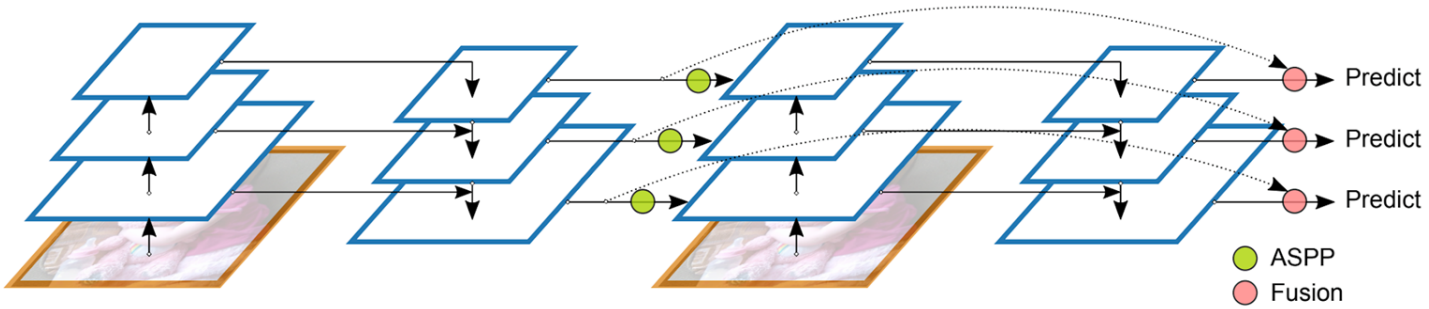


(b) RFP adds feedback connections to FPN.

Let Ri denote the feature transformations before connecting them back to the bottom-up backbone.

$$\mathbf{f}_i = \mathbf{F}_i\left(\mathbf{f}_{i+1}, \mathbf{x}_i\right), \quad \mathbf{x}_i = \mathbf{B}_i\left(\mathbf{x}_{i-1}, \mathbf{R}_i(\mathbf{f}_i)\right)$$

Recursive Feature Pyramid (RFP)

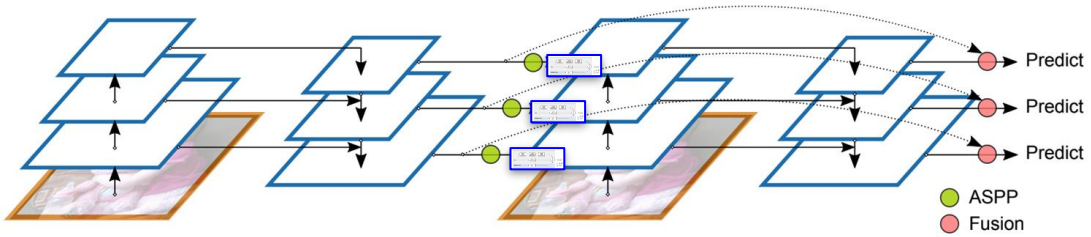


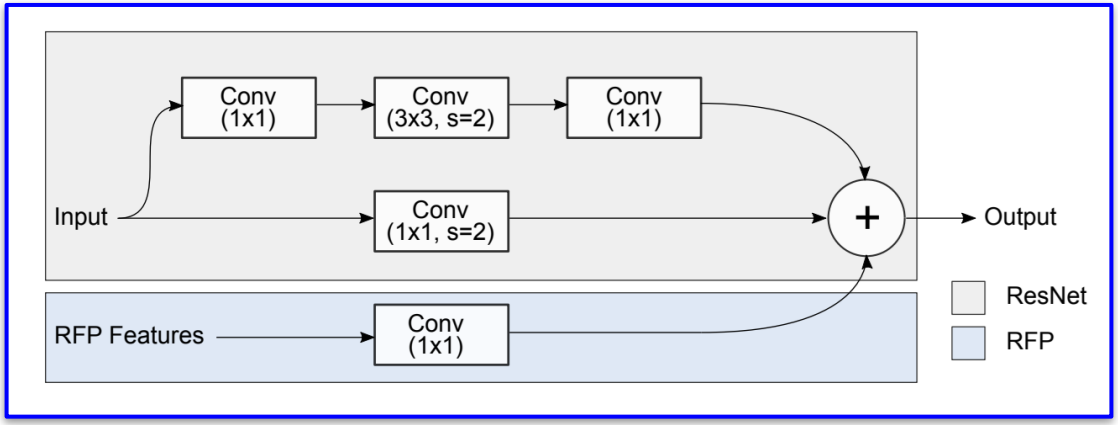
(c) Unrolling RFP to an example 2-step sequential implementation.

We unroll it to a sequential network, where superscript t denotes operations and features at the step t.

$$\mathbf{f}_i^t = \mathbf{F}_i^t\left(\mathbf{f}_{i+1}^t, \mathbf{x}_i^t\right), \ \mathbf{x}_i^t = \mathbf{B}_i^t\left(\mathbf{x}_{i-1}^t, \mathbf{R}_i^t\left(\mathbf{f}_i^{t-1}\right)\right)$$

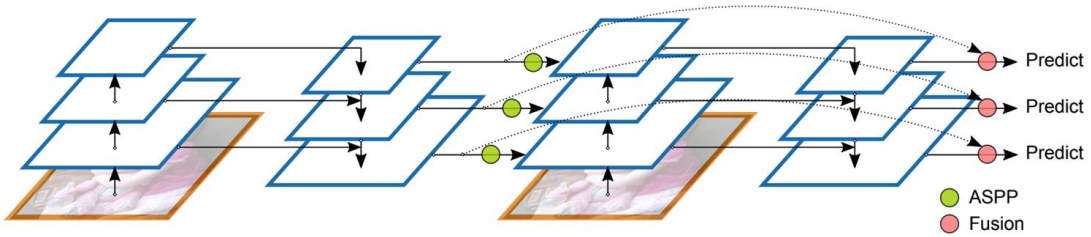
Recursive Feature Pyramid (RFP)

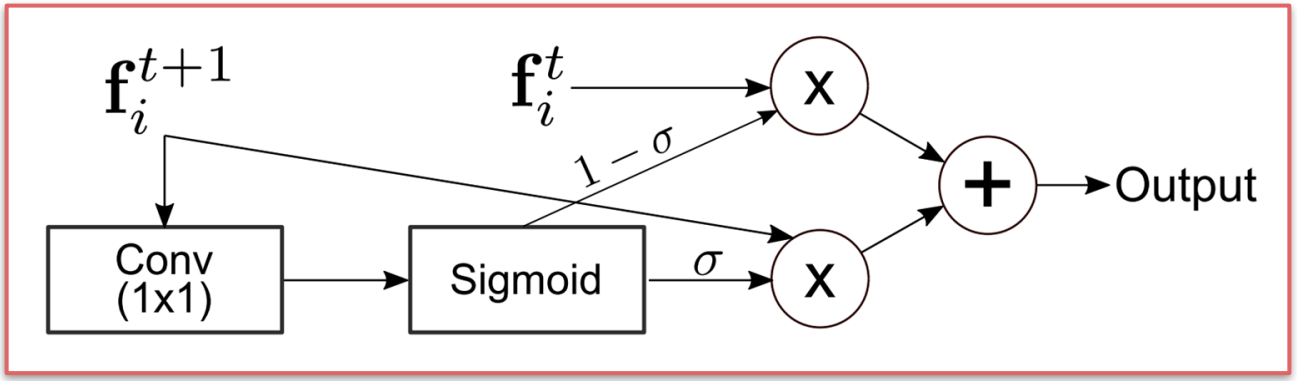




RFP adds transformed features to the first block of each stage of ResNet.

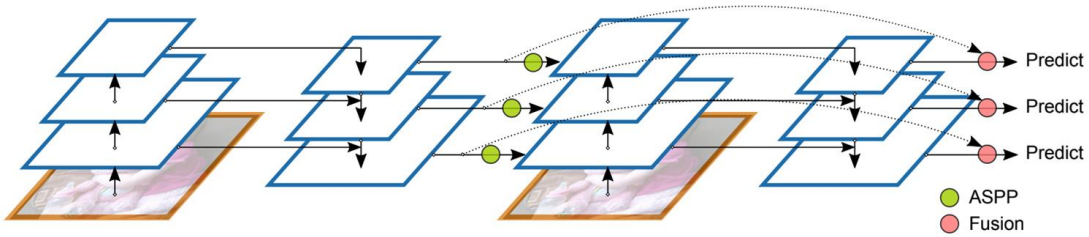
Recursive Feature Pyramid (RFP)

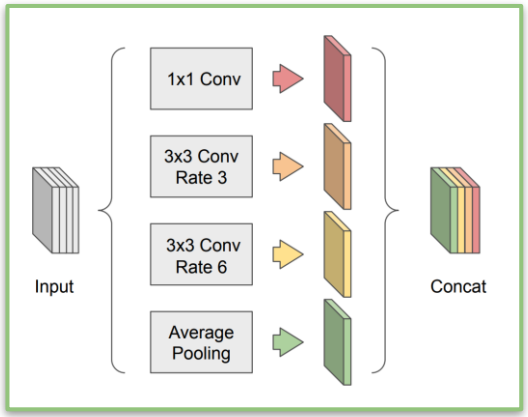




The Fusion module used in RFP.

Recursive Feature Pyramid (RFP)

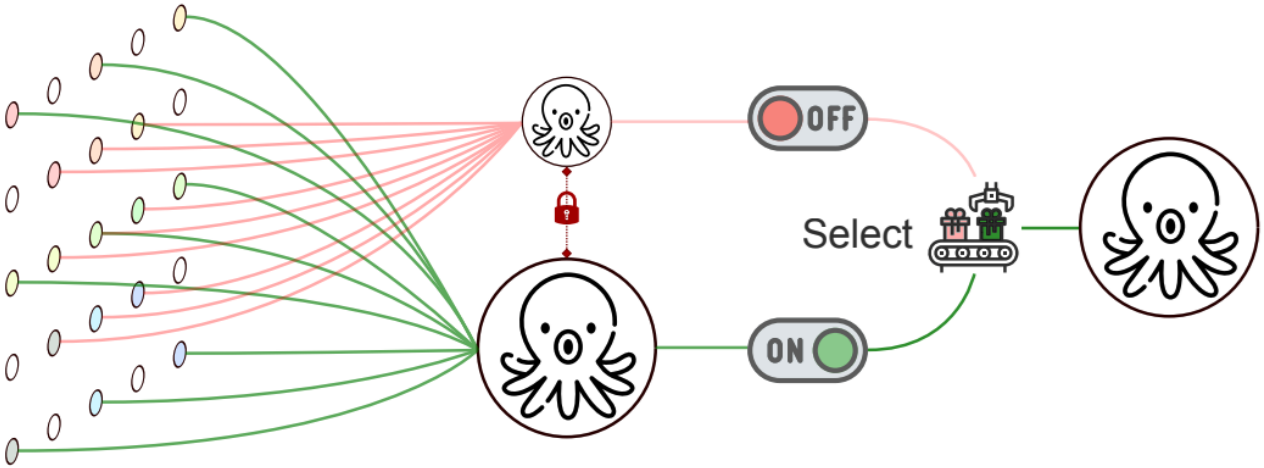




The ASPP module used in RFP.

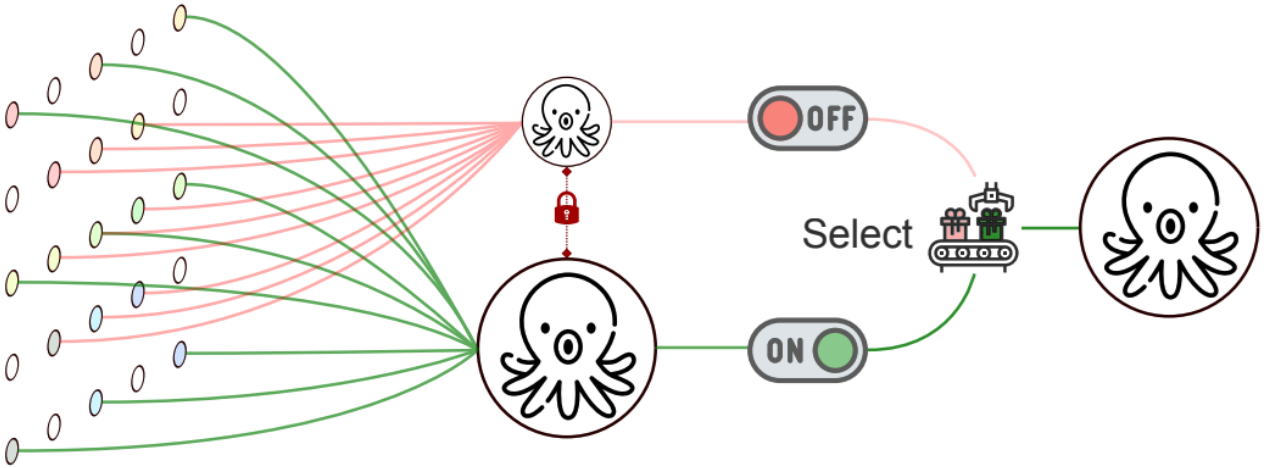
Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE TPAMI.

Switchable Atrous Convolution (SAC)



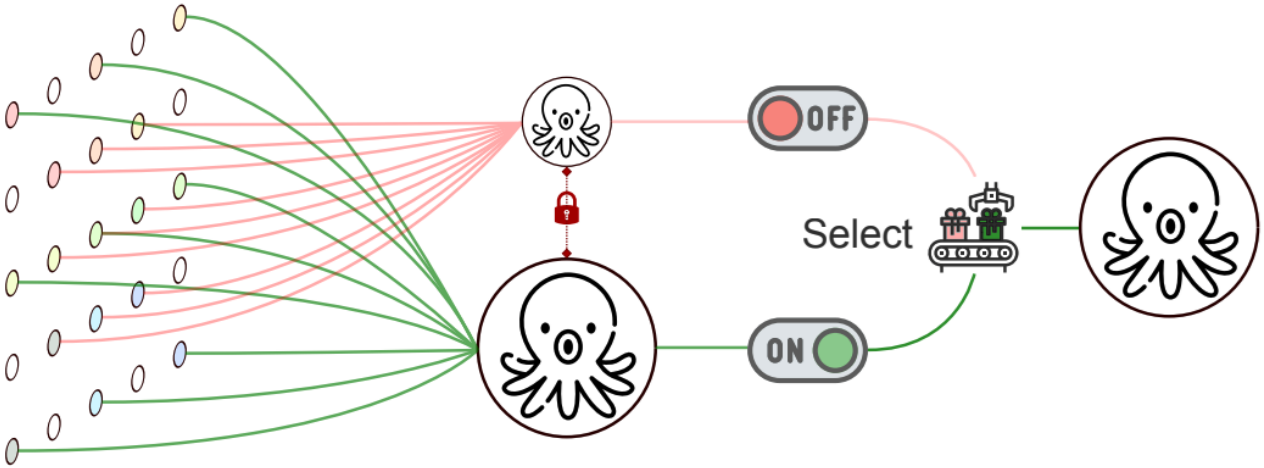
The same object at different scales could be roughly detected by the same set of convolutional weights using different atrous rates.

Switchable Atrous Convolution (SAC)



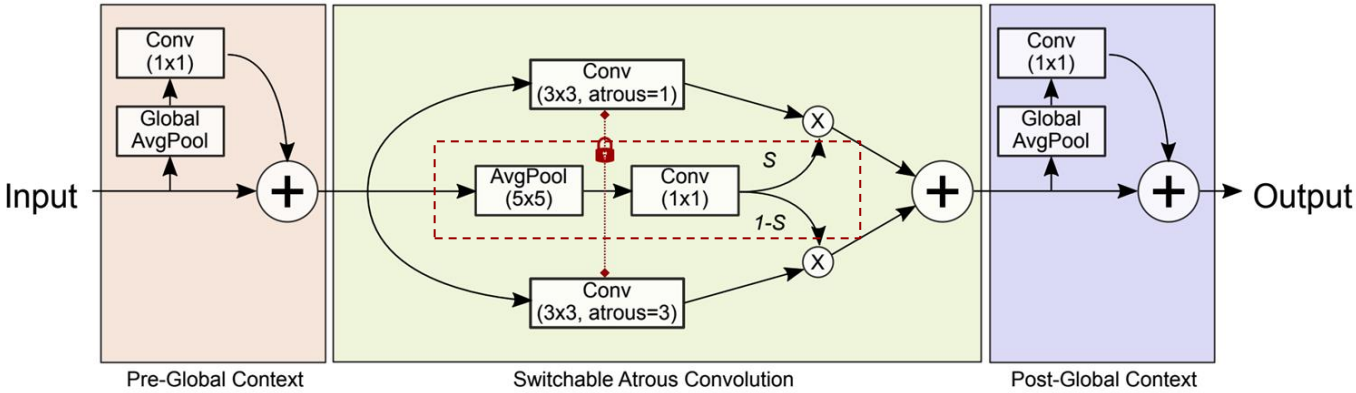
$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, r)$$

Switchable Atrous Convolution (SAC)



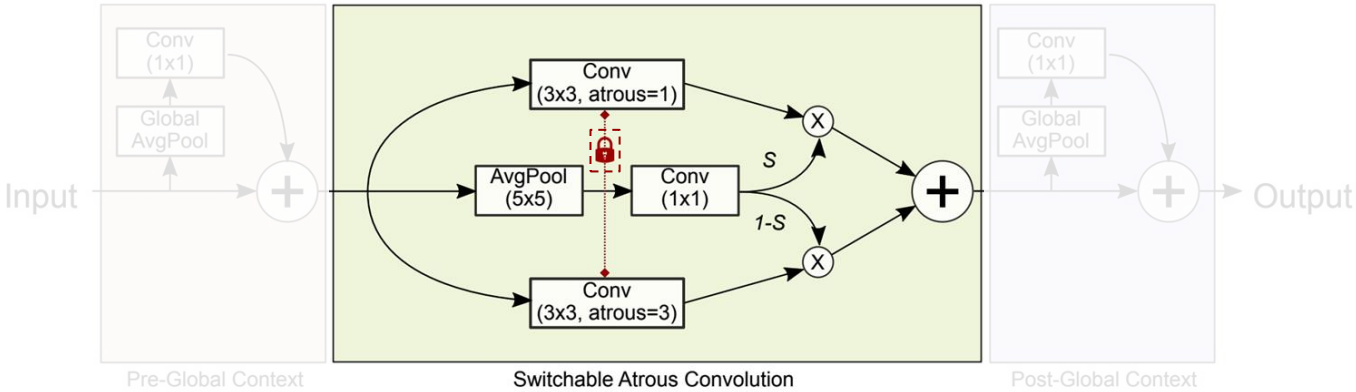
$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow[\text{to SAC}]{\text{Convert}} \mathbf{S}(\mathbf{x}) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1)$$
$$+(1 - \mathbf{S}(\mathbf{x})) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w} + \Delta\mathbf{w}, r)$$

Switchable Atrous Convolution (SAC)



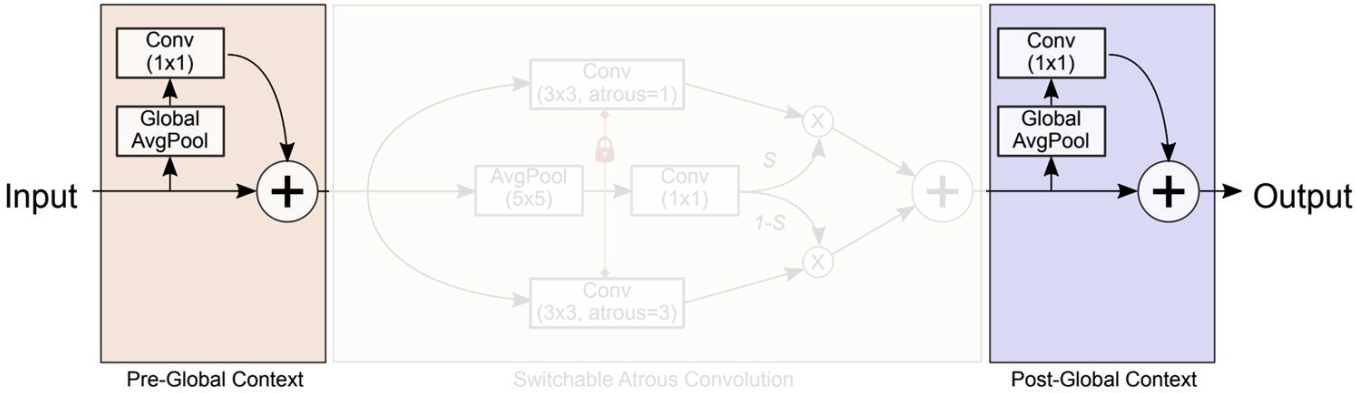
$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow[\text{to SAC}]{\text{Convert}} \mathbf{S}(\mathbf{x}) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1)$$
$$+ \big(1 - \mathbf{S}(\mathbf{x})\big) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w} + \Delta\mathbf{w}, r)$$

Switchable Atrous Convolution (SAC)



$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow[\text{to SAC}]{\text{Convert}} \mathbf{S}(\mathbf{x}) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1)$$
$$+\big(1 - \mathbf{S}(\mathbf{x})\big) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w} + \Delta\mathbf{w}, r)$$

Switchable Atrous Convolution (SAC)



$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow[\text{to SAC}]{\text{Convert}} \mathbf{S}(\mathbf{x}) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) + (1 - \mathbf{S}(\mathbf{x})) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w} + \Delta\mathbf{w}, r)$$

Experiments

HTC	RFP	SAC	Box AP	Box AP_{50}	Box AP_{75}	Box AP_S	Box AP_M	Box AP_L	Mask AP	Mask AP_{50}	Mask AP_{75}	Mask AP_S	Mask AP_M	Mask AP_L	Runtime FPS
✓			42.0	60.8	45.5	23.7	45.5	56.4	37.1	58.2	39.9	19.1	40.2	51.9	4.3
✓	✓		46.2	65.1	50.2	27.9	50.3	60.3	40.4	62.5	43.5	22.3	43.8	54.9	4.1
✓		✓	46.3	65.8	50.2	27.8	50.6	62.4	40.4	63.1	43.4	22.7	44.2	56.4	4.2
✓	✓	✓	49.0	67.7	53.0	30.1	52.6	64.9	42.1	64.8	45.5	23.9	45.6	57.8	3.9

Detection results on COCO val2017
with ResNet-50 as backbone.

Chen, Kai, et al. 2019. Hybrid task cascade for instance segmentation. In CVPR. 2019.

Experiments

HTC	RFP	SAC	Box AP	Box AP_{50}	Box AP_{75}	Box AP_S	Box AP_M	Box AP_L	Mask AP	Mask AP_{50}	Mask AP_{75}	Mask AP_S	Mask AP_M	Mask AP_L	Runtime FPS
✓			42.0	60.8	45.5	23.7	45.5	56.4	37.1	58.2	39.9	19.1	40.2	51.9	4.3
✓	✓		46.2	65.1	50.2	27.9	50.3	60.3	40.4	62.5	43.5	22.3	43.8	54.9	4.1
✓		✓	46.3	65.8	50.2	27.8	50.6	62.4	40.4	63.1	43.4	22.7	44.2	56.4	4.2
✓	✓	✓	49.0	67.7	53.0	30.1	52.6	64.9	42.1	64.8	45.5	23.9	45.6	57.8	3.9

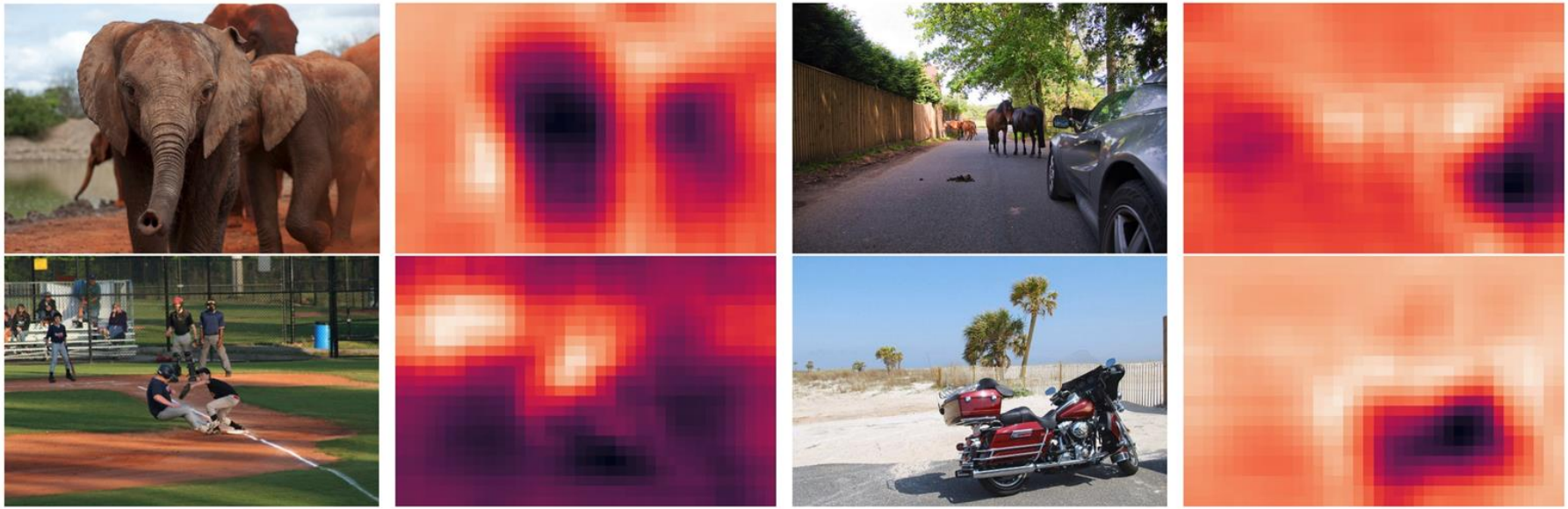
Detection results on COCO val2017
with ResNet-50 as backbone.

Experiments

HTC	RFP	SAC	Box AP	Box AP_{50}	Box AP_{75}	Box AP_S	Box AP_M	Box AP_L	Mask AP	Mask AP_{50}	Mask AP_{75}	Mask AP_S	Mask AP_M	Mask AP_L	Runtime FPS
✓			42.0	60.8	45.5	23.7	45.5	56.4	37.1	58.2	39.9	19.1	40.2	51.9	4.3
✓	✓		46.2	65.1	50.2	27.9	50.3	60.3	40.4	62.5	43.5	22.3	43.8	54.9	4.1
✓		✓	46.3	65.8	50.2	27.8	50.6	62.4	40.4	63.1	43.4	22.7	44.2	56.4	4.2
✓	✓	✓	49.0	67.7	53.0	30.1	52.6	64.9	42.1	64.8	45.5	23.9	45.6	57.8	3.9

Detection results on COCO val2017
with ResNet-50 as backbone.

Visualization



Visualizing the outputs of the learned switch functions in Switchable Atrous Convolution. Darker intensity means that the switch function for that region gathers more outputs from the larger atrous rate.

Experiments

Method	Backbone	TTA	AP_{bbox}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SpineNet [22]	SpineNet-190		52.1	71.8	56.5	35.4	55.0	63.6
EfficientDet-D7 [70]	EfficientNet-B6		52.2	71.4	56.3	-	-	-
EfficientDet-D7x (Model Zoo on GitHub)	-	-	55.1	74.3	59.9	37.2	57.9	68.0
CBNet [55]	ResNeXt-152	✓	53.3	71.9	58.5	35.5	55.8	66.7
DetectoRS	ResNet-50		51.3	70.1	55.8	31.7	54.6	64.8
DetectoRS	ResNet-50	✓	53.0	72.2	57.8	35.9	55.6	64.6
DetectoRS	ResNeXt-101-32x4d		53.3	71.6	58.5	33.9	56.5	66.9
DetectoRS	ResNeXt-101-32x4d	✓	54.7	73.5	60.1	37.4	57.3	66.4
DetectoRS	ResNeXt-101-64x4d	✓	55.7	74.2	61.1	37.7	58.4	68.1

SOTA comparison on COCO test-dev for object detection.

Experiments

Method	Backbone	TTA	AP_{mask}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
HTC [8]	ResNet-50		38.5	60.1	41.7	20.4	40.6	51.2
HTC	ResNeXt-101-32x4d		40.7	63.2	44.1	22.0	43.3	54.2
HTC	ResNeXt-101-64x4d		41.3	63.9	44.8	22.7	44.0	54.7
HTC + DCN [19] + multi-scale training	ResNeXt-101-64x4d		44.2	67.8	48.1	25.3	47.2	58.7
DetectoRS	ResNet-50		44.4	67.7	48.3	25.6	47.5	58.3
DetectoRS	ResNet-50	✓	45.8	69.8	50.1	29.2	48.3	58.2
DetectoRS	ResNeXt-101-32x4d		45.8	69.2	50.1	27.4	48.7	59.6
DetectoRS	ResNeXt-101-32x4d	✓	47.1	71.1	51.6	30.3	49.5	59.6
DetectoRS	ResNeXt-101-64x4d	✓	48.5	72.0	53.3	31.6	50.9	61.5

SOTA Instance segmentation on COCO test-dev.

Experiments

Method	TTA	PQ	PQ^{Th}	PQ^{St}
DeeperLab [81]		34.3	37.5	29.6
SSAP [24]	✓	36.9	40.1	32.0
Panoptic-DeepLab [18]	✓	41.4	45.1	35.9
Axial-DeepLab-L [72]	✓	44.2	49.2	36.8
TASCNet [41]		40.7	47.0	31.0
Panoptic-FPN [36]		40.9	48.3	29.7
AdaptIS [67]	✓	42.8	53.2	36.7
AUNet [44]		46.5	55.8	32.5
UPSNet [78]	✓	46.6	53.2	36.7
Li *et al.* [42]		47.2	53.5	37.7
SpatialFlow [16]	✓	47.3	53.5	37.9
SOGNet [82]	✓	47.8	-	-
DetectoRS	✓	50.0	58.5	37.2

SOTA Panoptic segmentation on COCO test-dev.

DetectoRS

Introduction

We provide the config files for DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution.

```
@article{qiao2020detectors,
  title={DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution},
  author={Qiao, Siyuan and Chen, Liang-Chieh and Yuille, Alan},
  journal={arXiv preprint arXiv:2006.02334},
  year={2020}
}
```

hhaAndroid [Refactor]: Unified parameter initialization (#4750) ... ✕ Latest commit 670ecc2 on Apr 28 History

4 contributors

134 lines (120 sloc) | 4.89 KB | Raw | Blame

```python
import torch
import torch.nn as nn
import torch.nn.functional as F
from mmcv.cnn import constant_init, xavier_init
from mmcv.runner import BaseModule, ModuleList

from ..builder import NECKS, build_backbone
from .fpn import FPN


class ASPP(BaseModule):
    """ASPP (Atrous Spatial Pyramid Pooling)

    This is an implementation of the ASPP module used in DetectoRS
    (https://arxiv.org/pdf/2006.02334.pdf)

    Args:
        in_channels (int): Number of input channels.
        out_channels (int): Number of channels produced by this module
        dilations (tuple[int]): Dilations of the four branches.
            Default: (1, 3, 6, 1)
        init_cfg (dict or list[dict], optional): Initialization config dict.
    """

    def __init__(self,
                 in_channels,
```

yhcao6 fix saconv (#489) ...

Latest commit 7b18b97 on Aug 15, 2020 History

1 contributor

132 lines (124 sloc) | 4.91 KB | Raw | Blame

```python
import torch
import torch.nn as nn
import torch.nn.functional as F

from mmcv.cnn import CONV_LAYERS, ConvAWS2d, constant_init
from mmcv.ops.deform_conv import deform_conv2d
from mmcv.utils import TORCH_VERSION


@CONV_LAYERS.register_module(name='SAC')
class SAConv2d(ConvAWS2d):
    """SAC (Switchable Atrous Convolution)

    This is an implementation of SAC in DetectoRS
    (https://arxiv.org/pdf/2006.02334.pdf).

    Args:
        in_channels (int): Number of channels in the input image
        out_channels (int): Number of channels produced by the convolution
        kernel_size (int or tuple): Size of the convolving kernel
        stride (int or tuple, optional): Stride of the convolution. Default: 1
        padding (int or tuple, optional): Zero-padding added to both sides of
            the input. Default: 0
        padding_mode (string, optional): ``'zeros'``, ``'reflect'``,
            ``'replicate'`` or ``'circular'``. Default: ``'zeros'``
        dilation (int or tuple, optional): Spacing between kernel elements.
```

Deeply Shape-guided Cascade for Instance Segmentation

Hao Ding, Siyuan Qiao, Alan Yuille, Wei Shen

Introduction

- **Key to success**: Fully leverage the relationship between bounding box detection and mask segmentation across multiple stages.
  - Others' direction: Mask Segmentation benefit from better box detection.
  - Our direction: Box detection can utilize better mask segmentation.

Introduction

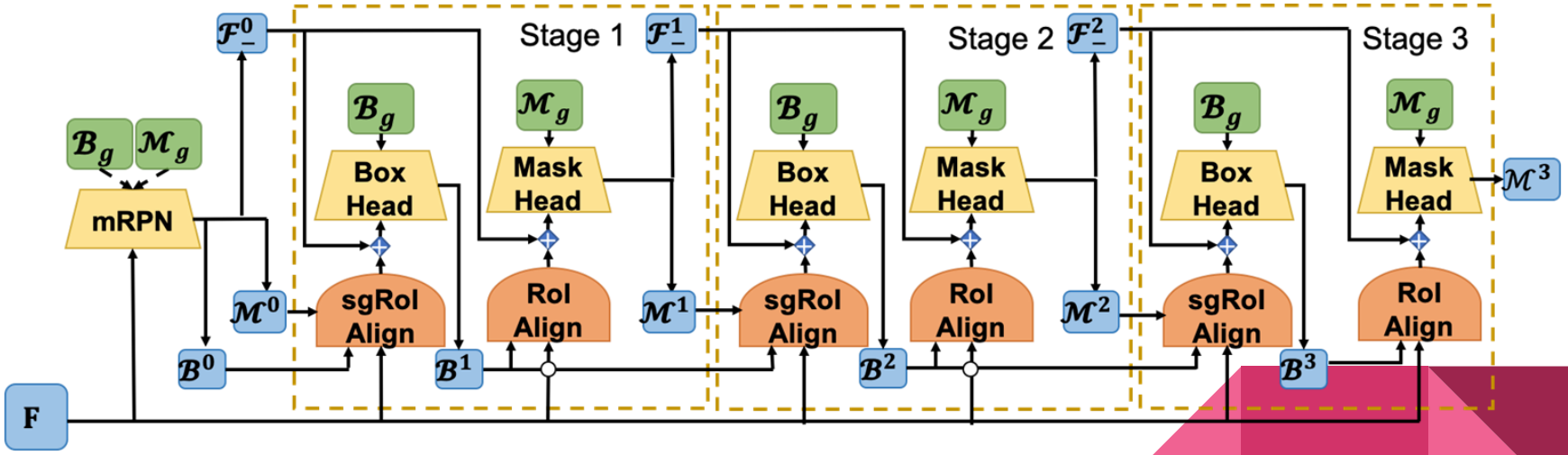
- **Deeply Shape-guided Cascade (DSC):** To imposes the shape guidance extracted from mask prediction, We propose DSC for instance segmentation with there key components:
  - Initial shape guidance: mask-supervised Region Proposal Network
  - Explicit shape guidance: mask-guided region feature extractor
  - Implicit shape guidance: feature fusion & alignment operation

Introduction

- **Promising Results:** We outperforms the state-of-the-art instance segmentation cascade, HTC, by a non-negligible margin on COCO instance segmentation benchmark.
  - **2.1 box AP and 1.5 mask AP** on COCO 2017 val set
  - **1.9 box AP and 1.4 mask AP** on COCO 2017 test-dev set
  - More significant improvements on **huddled instances subset**

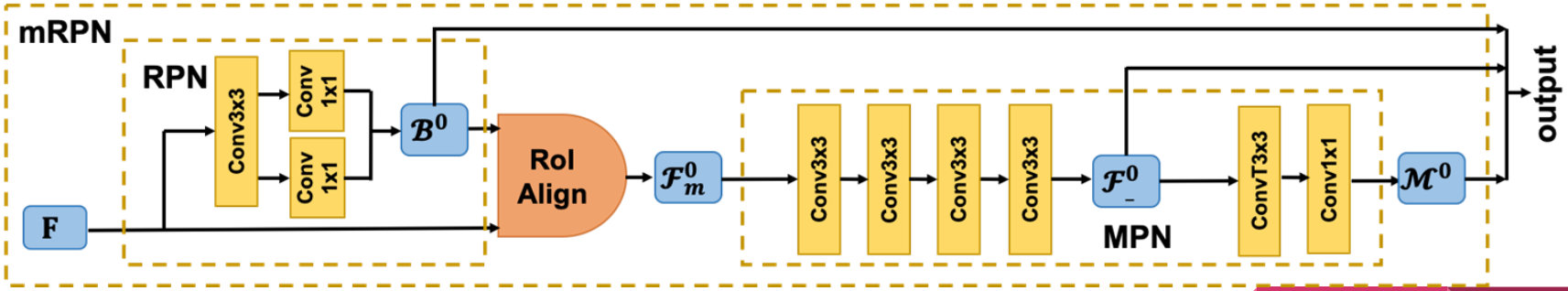
Methodology

DSC - Overall Framework



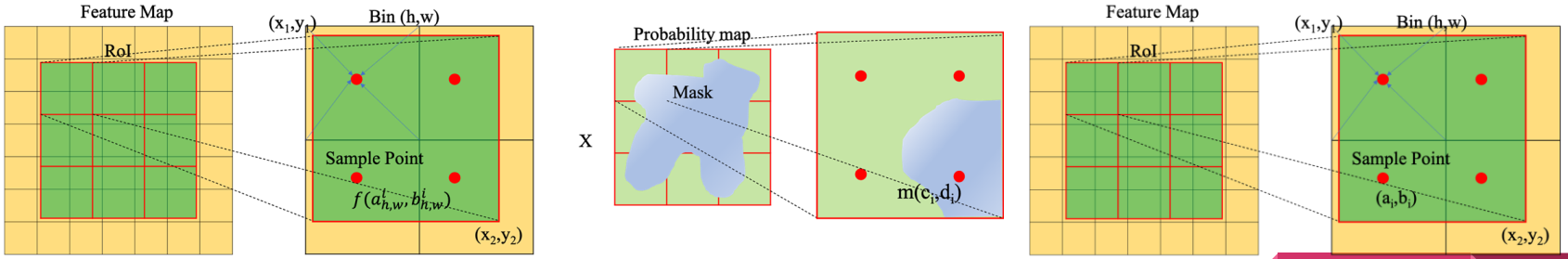
Methodology

DSC - Initial shape guidance : mask-supervised Region Proposal Network (mPRN) with the ability to generate class-agnostic masks.



Methodology

DSC - Explicit shape guidance : mask-guided region-of-interest (RoI) feature extractor, which employs mask segmentation to focus feature extraction within a region aligned well with the instance shape rather than a rectangular RoI.

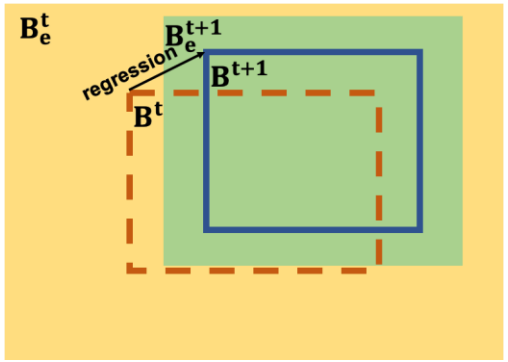


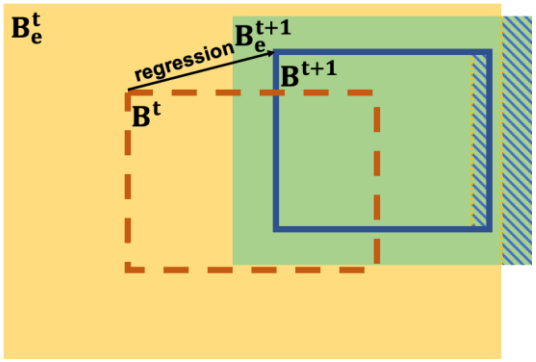
SgRoIAlign: $f_{B,M}(h,w) = \frac{\sum_{i=1}^{N} f(a^i_{h,w}, b^i_{h,w}) \times (m(c^i_{h,w}, d^i_{h,w}) + 1)}{N}$

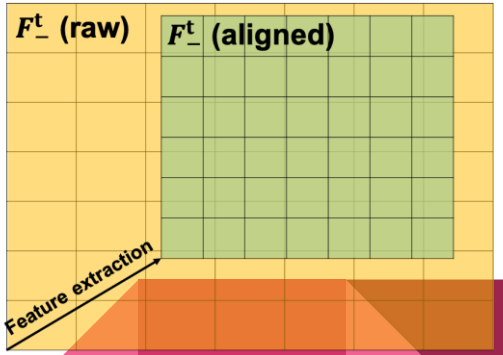
RoIAlign: $f_{B,M}(h,w) = \frac{\sum_{i=1}^{N} f(a^i_{h,w}, b^i_{h,w})}{N}$

Methodology

DSC - Implicit guidance : feature fusion operation feeding intermediate mask features to the box head. To deal with **feature misalignment,** we introduce the **adaptive feature alignment strategy.**







Methodology

DSC - Cascade Pipeline: The Cascade pipeline is shown as the equations:

$$\mathbf{F}_m^t = \mathbb{f}(\mathbf{B}_e^t, \mathbf{F}) \quad (\mathbf{M}^t, \mathbf{F}_-^t) = \mathbb{m}^t(\mathbf{F}_m^t \oplus \mathbf{w}_m^t \mathbb{a}(\mathbf{F}_-^{t-1}, \mathbf{B}_e^{t-1}, \mathbf{B}_e^t))$$

$$\mathbf{F}_b^{t+1} = \mathbb{f}_s(\mathbf{B}_e^t, \mathbf{F}, \mathbf{M}^t) \qquad \mathbf{B}^{t+1} = \mathbb{b}^t(\mathbf{B}^t, \mathbf{F}_b^{t+1} \oplus \mathbf{w}_b^{t+1} \mathbf{F}_-^t)$$

Experiment Results

Benchmarking Results:

- Comparison with HTC on COCO val

Method	Backbone	AP_b	AP_b^{50}	AP_b^{75}	AP_m	AP_m^{50}	AP_m^{75}
HTC	R-50 FPN	43.3	62.2	47.1	38.3	59.3	41.4
DSC	R-50 FPN	45.8(+2.5)	63.4 (+1.2)	49.8(+2.7)	40.2 (+1.9)	61.0(+1.7)	43.5(+2.1)
HTC	R-101 FPN	44.8	63.3	48.8	39.6	61.0	42.8
DSC	R-101 FPN	46.6(+1.8)	64.5 (+1.2)	50.8(+2.0)	40.7 (+1.1)	62.0 (+1.0)	44.1(+1.3)
HTC	X-101-32x4d FPN	46.1	65.3	50.1	40.5	62.5	43.7
DSC	X-101-32x4d FPN	48.0(+1.9)	65.9(+0.6)	52.2(+2.1)	42.0(+1.5)	63.7(+1.2)	45.6(+1.9)

Experiment Results

Benchmarking Results:

- Comparison with HTC on COCO val

Method	Backbone	AP_b	AP_b^{50}	AP_b^{75}	AP_m	AP_m^{50}	AP_m^{75}
HTC	R-50 FPN	43.3	62.2	47.1	38.3	59.3	41.4
DSC	R-50 FPN	45.8(+2.5)	63.4 (+1.2)	49.8(+2.7)	40.2 (+1.9)	61.0(+1.7)	43.5(+2.1)
HTC	R-101 FPN	44.8	63.3	48.8	39.6	61.0	42.8
DSC	R-101 FPN	46.6(+1.8)	64.5 (+1.2)	50.8(+2.0)	40.7 (+1.1)	62.0 (+1.0)	44.1(+1.3)
HTC	X-101-32x4d FPN	46.1	65.3	50.1	40.5	62.5	43.7
DSC	X-101-32x4d FPN	48.0(+1.9)	65.9(+0.6)	52.2(+2.1)	42.0(+1.5)	63.7(+1.2)	45.6(+1.9)

Experiment Results

Ablation Study:

- Inference time vs Precision

Methods	AP_b	AP_m	Inference Time
HTC	42.3	37.4	238ms
DSC	44.8(+2.5)	39.5(+2.1)	434ms(+196ms)
F-DSC	44.5(+2.2)	39.4 (+2.0)	256ms(+18ms)

Experiment Results

Ablation Study:

- Contribution of Components

Method	AP_b	AP_m
F-DSC	44.5	39.5
F-DSC - ExSG	44.2(-0.3)	39.1(-0.4)
F-DSC - ImSG	43.4(-1.1)	38.8(-0.7)
F-DSC - Plus1	44.3(-0.2)	39.3(-0.2)
F-DSC - AFA	44.0(-0.5)	39.0(-0.5)
F-DSC - AFA + ReComp	44.9(+0.4)	39.9(+0.4)

Experiment Results

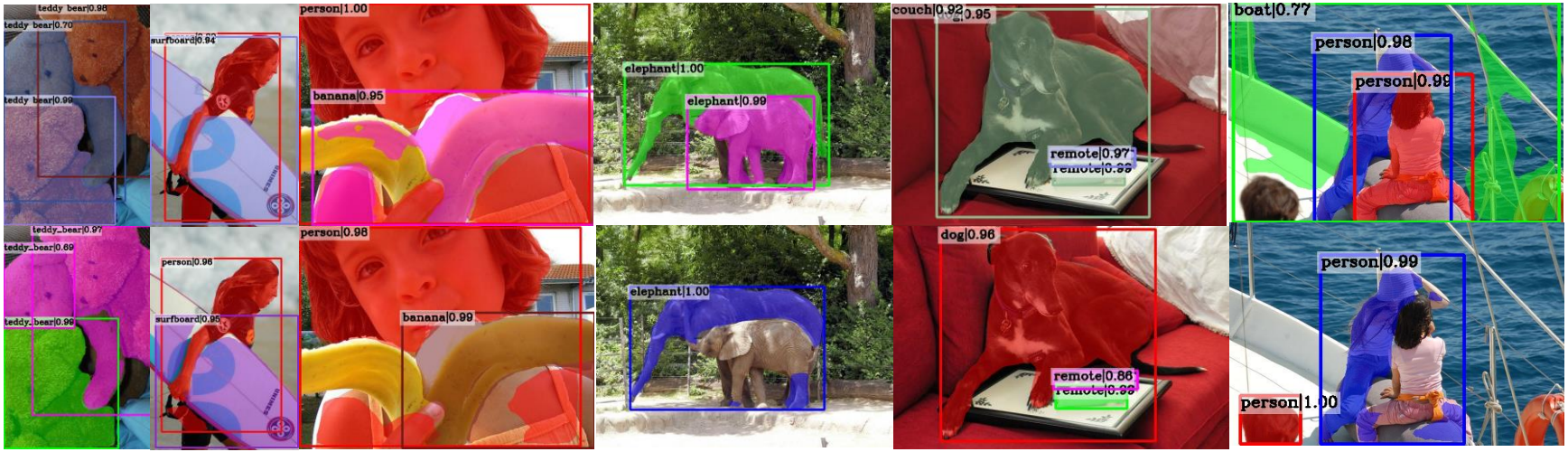
Ablation Study:

- Results on huddle Instances

T_O \ T_P	0.0	0.1	0.2	0.3	0.4
-1.0	2.4 / 2.1	-	-	-	-
0.0	2.5 / 2.0	2.5 / 2.0	2.6 / 2.2	2.7 / 2.4	2.5 / 2.4
0.1	2.5 / 2.0	2.4 / 2.1	2.6 / 2.3	2.8 / 2.5	2.8 / 2.8
0.2	2.5 / 2.1	2.5 / 2.2	2.7 / 2.6	2.9 / 2.9	3.0 / 3.3
0.3	2.4 / 2.0	2.6 / 2.4	3.0 / 2.9	3.7 / 3.6	3.6 / 3.9
0.4	2.5 / 2.1	2.9 / 2.6	3.2 / 3.2	3.8 / 3.9	**4.2 / 4.7**

Experiment Results

Qualitative comparison between DSC (top) and HTC (bottom) on COCO val.



MMDetection

- Our work is based on MMDetection which is an open-source object detection toolbox based on PyTorch. It is a part of the OpenMMLab project.
- We added all our methods in an object-oriented way in MMDetection by creating new classes.
- The following pages are about more detailed implementation on MMDetection

Code is available here:
https://github.com/hding2455/DSC

MMDetection – new files

- configs/dsc/*
- mmdet/core/bbox/sampler/dsc_pseudo_sampler.py
- mmdet/core/post_processing/dsc_bbox_nms.py
- mmdet/models/roi_heads/bbox_heads/dsc_bbox_head.py
- mmdet/models/roi_heads/mask_heads/dsc_mask_head.py
- mmdet/models/roi_heads/roi_extractors/relative_roi_extractor.py
- mmdet/models/roi_heads/roi_extractors/sg_single_level_roi_extractor.py
- mmdet/models/detectors/dsc.py
- mmdet/models/roi_heads/dsc_roi_head.py

Code is available here:
https://github.com/hding2455/DSC

MMDetection – file description

configs/dsc/*

- Configs files for DSC of different backbones and training settings.
- This kind of files use dict in python to store design choice (configs of models, training settings, dataset, ets.)
- In our work the main difference is the model design.

```
roi_head=dict(
    type='DSCRoIHead',
    num_stages=3,
    stage_loss_weights=[dict(loss_mpn=1, loss_bbox=1, loss_cls=1),
                        dict(loss_mpn=1, loss_bbox=0.5, loss_cls=1),
                        dict(loss_mpn=1, loss_bbox=0.5, loss_cls=1),
                        dict(loss_mask=1)],
    relative_roi_extractor=dict(
        type='RelativeRoIExtractor',
        roi_layer=dict(type='RoIAlign', out_size=14, sample_num=0),
        out_channels=256,
        featmap_strides=[1.0]),
mpn=[
    dict(
        type='DSCMaskHead',
        with_conv_res=False,
        num_convs=4,
        in_channels=256,
        conv_out_channels=256,
        class_agnostic=True,
        loss_mask=dict(
            type='CrossEntropyLoss', use_mask=True, loss_weight=1.0)),
```

Some example codes

Code is available here:
https://github.com/hding2455/DSC

MMDetection – file description

mmdet/core/bbox/sampler/dsc_pseudo_sampler.py

- Sampler that retain the number and the order of the RoIs for every stage.

mmdet/core/post_processing/dsc_bbox_nms.py

- NMS operation which returns the selecting indices of the RoIs.

```python
@BBOX_SAMPLERS.register_module()
class DSCPseudoSampler(BaseSampler):
    """A pseudo sampler that does not do sampling actually."""
def sample(self, assign_result, bboxes, gt_bboxes, gt_labels,**kwargs):
    """Directly returns the positive and negative indices  of samples
        sampling_result = SamplingResult(pos_inds, neg_inds, bboxes, gt_bboxes,
                                         assign_result, gt_flags)
        return sampling_result
```

```python
def dsc_multiclass_nms(multi_bboxes,
                       multi_scores,
                       score_thr,
                       nms_cfg,
                       max_num=-1,
                       score_factors=None):
```

...

```python
    if max_num > 0:
        dets = dets[:max_num]
        keep = keep[:max_num]

    return dets, labels[keep], rois_inds[keep]
```

Some example codes

Code is available here:
https://github.com/hding2455/DSC

MMDetection – file description

mmdet/models/roi_heads/bbox_heads/dsc_bbox_head.py

- box head for DSC

mmdet/models/roi_heads/mask_heads/dsc_mask_head.py

- mask head for DSC

```python
@HEADS.register_module()
class DSCBBoxHead(ConvFCBBoxHead):
    def forward(self, x, res_feat=None):
        if res_feat is not None and self.with_conv_res:
            res_feat = self.conv_res(res_feat)
            res_feat = nn.functional.adaptive_avg_pool2d(res_feat, x.shape[-2:])
            x = x + res_feat
    def _get_unsampled_target_single(self, bboxes, pos_inds, pos_gt_bboxes,
                                     pos_gt_labels, cfg):
    def get_unsampled_targets(self, sampling_results, gt_bboxes,
                              gt_labels, rcnn_train_cfg, concat=True):
    @force_fp32(apply_to=('bbox_pred', ))
    def refine_bboxes(self, rois, enlarged_rois, next_enlarge_ratio, labels,
                      bbox_preds, pos_is_gts, img_metas):
    @force_fp32(apply_to=('bbox_pred', ))
    def regress_by_class(self, rois, enlarged_rois, next_enlarge_ratio,
                         label, bbox_pred, img_meta):
```

```python
@HEADS.register_module()
class DSCMaskHead(FCNMaskHead):
    def forward(self, x, res_feat=None, return_logits=True, return_feat=True):
        if res_feat is not None:
            assert self.with_conv_res
            res_feat = self.conv_res(res_feat)
            res_feat = nn.functional.adaptive_avg_pool2d(res_feat, x.shape[-2:])
            x = x + res_feat
                                    ...
        return outs if len(outs) > 1 else outs[0]
```

Some example codes

Code is available here:
https://github.com/hding2455/DSC

MMDetection – file description

mmdet/models/roi_heads/roi_extractors/relative_roi_extractor.py

- The relative RoI feature extractor using in adaptive feature alignment.

mmdet/models/roi_heads/roi_extractors/sg_single_level_roi_extractor.py

- The shape-guided RoIAlign feature extractor

```python
@ROI_EXTRACTORS.register_module()
class RelativeRoIExtractor(BaseRoIExtractor):
    def compute_relative_rois(self, rois, base_rois, feature_shape):
    def forward(self, feats, rois, base_rois):
        """Forward function"""
        out_size = self.roi_layers[0].out_size
        feature_shape = feats.shape[-2:]
        relative_rois = self.compute_relative_rois(rois,base_rois,feature_shape)
        if len(rois) == 0:
            return feats.new_zeros(
                rois.size(0), self.out_channels, *out_size)
        return self.roi_layers[0](feats, relative_rois)
```

```python
@ROI_EXTRACTORS.register_module()
class SqSingleRoIExtractor(BaseRoIExtractor):
    def map_roi_levels(self, rois, num_levels):
    @force_fp32(apply_to=('feats', ), out_fp16=True)
    def forward(self, feats, rois, roi_scale_factor=None, masks=None):
```

```python
if masks is not None:
    resized_masks = nn.functional.adaptive_avg_pool2d(masks, roi_feats.shape[-2:])
```

```python
            if masks is not None:
                resized_masks_t = resized_masks[inds]
                roi_feats_t = roi_feats_t * (resized_masks_t + 1.0)
            roi_feats[inds] = roi_feats_t
```

Some example codes

Code is available here:
https://github.com/hding2455/DSC

MMDetection – file description

mmdet/models/detectors/dsc.py

- File that describe the high-level architecture of DSC

mmdet/models/roi_heads/dsc_roi_head.py

- File that detailed implement the cascade looping for the DSC RoI heads

```
@DETECTORS.register_module()
class DSC(TwoStageDetector):
```

...

```
    @property
    def with_semantic(self):
        """bool: whether the detector has a semantic head"""
        return self.roi_head.with_semantic
```

```
@HEADS.register_module()
class DSCRoIHead(BaseRoIHead, BBoxTestMixin, MaskTestMixin):
    def init_bbox_head(self, bbox_roi_extractor, bbox_head):
    def init_mpn(self, mpn_roi_extractor, mpn):
    def init_mask_head(self, mask_roi_extractor, mask_head):
    def init_assigner_sampler(self):
    def enlarge_rois(self, rois, img_shape, ratio=2):
```

...

```
def _bbox_forward(self, stage, x, rois,  semantic_feat,  res_feat, mpn_pred,  img_shape, enlarge=False):
def _bbox_forward_train(self, stage, x, sampling_results, gt_bboxes,
                        gt_labels, rcnn_train_cfg, semantic_feat, res_feat, mpn_pred, img_metas):
def _mask_forward(self, stage, x, rois, former_enlarged_rois, semantic_feat, res_feat, img_shape, enlarge=False):
def _mask_forward_train(self, stage, x, sampling_results, gt_masks, rcnn_train_cfg,
                        semantic_feat, res_feat, former_enlarged_rois, img_metas):
def _mpn_forward(self, stage, x, rois, former_enlarged_rois, semantic_feat, res_feat, img_shape,
                 enlarge=True, return_logits=True):
def _mpn_forward_train(self, stage, x, sampling_results, gt_masks, rcnn_train_cfg,
                       semantic_feat, res_feat, former_enlarged_rois, img_metas):
def forward_train(self, x, img_metas, proposal_list, gt_bboxes, gt_labels,
                  gt_bboxes_ignore=None, gt_masks=None, gt_semantic_seg=None):
def simple_test(self, x, proposal_list, img_metas, rescale=False):
```

Some example codes

Code is available here:
https://github.com/hding2455/DSC

Thank you!
Q&A