Hybrid Task Cascade for Instance Segmentation

Kai Chen¹, Jiangmiao Pang²,³, Jiaqi Wang¹, Yu Xiong¹, Xiaoxiao Li¹, Shuyang Sun⁴, Wansen Feng²
Ziwei Liu¹, Jianping Shi², Wanli Ouyang⁴, Chen Change Loy¹,⁵, Dahua Lin¹

¹The Chinese University of Hong Kong  ²SenseTime Research  ³Zhejiang University
⁴The University of Sydney  ⁵Nanyang Technological University of Singapore

Team: MMDet
Comparison of our approach with 2017 winning entries on COCO test-dev.
Overview

1. We developed a **hybrid cascading and branching** pipeline for detection and segmentation.
1. We developed a **hybrid cascading and branching** pipeline for detection and segmentation.

2. We proposed a **feature guided anchoring** scheme to improve the average recall (AR) of RPN by 10 points. (submitted to AAAI 2019)
1. We developed a **hybrid cascading and branching** pipeline for detection and segmentation.

2. We proposed a **feature guided anchoring** scheme to improve the average recall (AR) of RPN by 10 points. (submitted to AAAI 2019)

3. We designed a new backbone **FishNet**. (accepted to NIPS 2018)
1. We developed a **hybrid cascading and branching** pipeline for detection and segmentation.

2. We proposed a **feature guided anchoring** scheme to improve the average recall (AR) of RPN by 10 points. (submitted to AAAI 2019)

3. We designed a new backbone **FishNet**. (accepted to NIPS 2018)
A hybrid architecture with interleaved task branching and cascade.
Baseline: Cascade R-CNN
Baseline: Cascade R-CNN

Problem: designed for detection, not segmentation
Baseline: Cascade R-CNN + Mask R-CNN
**Baseline:** Cascade R-CNN + Mask R-CNN

**Problem:** mismatch of training and testing pipeline
Problem: mismatch of training and testing pipeline
**Task cascade**: ordinal bbox prediction and mask prediction
Task cascade: ordinal bbox prediction and mask prediction

Problem: no connection between mask branches of different stages
Interleaved execution: box cascade & mask cascade
Interleaved execution: box cascade & mask cascade

Problem: contextual information is not much explored
Hybrid branching: additional semantic segmentation branch
Overview

1. We developed a hybrid cascading and branching pipeline for detection and segmentation.

2. We proposed a feature guided anchoring scheme to improve the average recall (AR) of RPN by 10 points. (submitted to AAAI 2019)

3. We designed a new backbone FishNet. (accepted to NIPS 2018)
Guided Anchoring

• From sliding window to sparse, non-uniform distribution
• From predefined shapes to learnable, arbitrary shapes
• Refine features based on anchor shapes
Guided Anchoring

RPN w/ FPN
Guided Anchoring

feature pyramid

GA-RPN w/ FPN
Guided Anchoring
Guided Anchoring

feature pyramid

Guided anchoring

anchors

prediction

predicted anchor probabilities

predicted anchor aspect ratios

wide

tall

predicted anchors
Guided Anchoring

Guided anchoring

feature pyramid

predicted anchor probabilities

predicted anchor aspect ratios

predicted anchors

Guided anchoring

predicted anchors

prediction

Guided anchoring

predicted anchors

prediction

Guided anchoring

predicted anchors

prediction

Guided anchoring

predicted anchors

prediction
Guided Anchoring

Guided anchoring

Anchor generation

Feature adaption

predicted anchor probabilities

predicted anchor aspect ratios

predicted anchors

wide

tall
Guided Anchoring

RPN

GA-RPN
Guided Anchoring

![Graph showing the performance of different models with varying runtime on TITAN X](image)

- GA-RPN (SENet-154)
- RPN (SENet-154)
- RPN (ResNeXt-101)
- RPN (ResNet-152)
- GA-RPN (ResNet-50)
- RPN (ResNet-50)
1. We developed a **hybrid cascading and branching** pipeline for detection and segmentation.

2. We proposed a **feature guided anchoring** scheme to improve the average recall (AR) of RPN by 10 points. (submitted to AAAI 2019)

3. We designed a new backbone **FishNet**. (accepted to NIPS 2018)
Motivation

• The basic principles for designing CNN for region and pixel level tasks are diverging from the principles for image classification.

• Unify the advantages of networks designed for region and pixel level tasks in obtaining deep features with high-resolution.
FishNet

Motivation

• Traditional consecutive down-sampling will prevent the very shallow layers to be directly connected till the end, which may exacerbate the vanishing gradient problem.

• Features from varying depths could be used for refining each other.

FishNet

Top-1 Classification Error on ImageNet
MS COCO \textit{val-2017} detection and instance segmentation results.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Instance Segmentation $\text{AP}^s/\text{AP}^S/\text{AP}^M/\text{AP}^L$</th>
<th>Object Detection $\text{AP}^d/\text{AP}^D/\text{AP}^D/\text{AP}^L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [3]</td>
<td>34.5/15.6/37.1/52.1</td>
<td>38.6/22.2/41.5/50.8</td>
</tr>
<tr>
<td>ResNet-50$^\dagger$</td>
<td>34.7/18.5/37.4/47.7</td>
<td>38.7/22.3/42.0/51.2</td>
</tr>
<tr>
<td>ResNeXt-50 (32x4d)$^\dagger$</td>
<td>35.7/19.1/38.5/48.5</td>
<td>40.0/23.1/43.0/52.8</td>
</tr>
<tr>
<td>FishNet-188</td>
<td>\textbf{37.0}/19.8/40.2/50.3</td>
<td>\textbf{41.5}/24.1/44.9/55.0</td>
</tr>
<tr>
<td>vs. ResNet-50$^\dagger$</td>
<td>+2.3/+1.3/+2.8/+2.6</td>
<td>+2.8/+1.8/+2.9/+3.8</td>
</tr>
<tr>
<td>vs. ResNeXt-50$^\dagger$</td>
<td>\textbf{+1.3}/+0.7/+1.7/+1.8</td>
<td>\textbf{+1.5}/+1.0/+1.9/+2.2</td>
</tr>
</tbody>
</table>
Experiments

Training/Testing details

1. Training scales
   • short edge: random sampled from 400 ~ 1400
   • long edge: 1600
2. Test scales
   • (600, 900), (800, 1200), (1000, 1500), (1200, 1800), (1400, 2100)
3. Pipeline
   • Joint training
   • Finetune with GA-RPN proposals
   • Test with GA-RPN proposals
4. Resources
   • 32 Tesla V100 GPUs (16GB) for 3 days
Experiments

Backbones

- SENet-154
- ResNeXt101 (64×4d)
- ResNeXt101 (32×8d)
- DPN-107
- FishNet

\[ \sim 0.8 \text{ points higher} \]

comparable
Experiments

Other tricks

• w/ SoftNMS
• w/o OHEM
• w/o classwise balance sampling
• w/o voting for bbox or mask
Experiments

mask AP on test-dev

36.7

baseline
R-50 Cascade
with mask
Experiments

mask AP on test-dev

36.7

37.3 (+0.6)

baseline
R-50 Cascade
with mask
interleaved cascade
Experiments

mask AP on test-dev

- Baseline
- R-50 Cascade
- Interleaved cascade
- Semantic branch

Scores:
- Baseline: 36.7
- Interleaved cascade: 37.3 (+0.6)
- Semantic branch: 38.1 (+0.8)
Experiments

mask AP on test-dev

36.7 (+0.6) 37.3 (+0.8) 38.1 (+1.4)
deformable conv

baseline
R-50 Cascade
with mask
Experiments

mask AP on test-dev

- baseline R-50 Cascade with mask: 36.7
- interleaved cascade: 37.3 (+0.6)
- semantic branch: 38.1 (+0.8)
- deformable conv: 39.5 (+1.4)
- synchronize BN: 40.7 (+1.2)
Experiments

mask AP on test-dev

42.5 (+1.8)

multi-scale training

36.7

baseline R-50 Cascade with mask

interleaved cascade

semantic branch

deformable conv

synchronize BN

37.3 (+0.6)

38.1 (+0.8)

39.5 (+1.4)

40.7 (+1.2)
Experiments

mask AP on test-dev
Experiments

mask AP on test-dev

- 45.3 (+1.0)
- 44.3 (+1.8)
- 42.5 (+1.8)
- 40.7 (+1.2)
- 39.5 (+1.4)
- 38.1 (+0.8)
- 37.3 (+0.6)
- 36.7

- baseline
- R-50 Cascade with mask
- deformable conv
- semantic branch
- synchronize BN
- multi-scale training
- better backbone
- GARPN finetune
Experiments

mask AP on test-dev

47.4 (+2.1) multi-scale & flip testing

44.3 (+1.0)

42.5 (+1.8)

40.7 (+1.2)

39.5 (+1.4)

38.1 (+0.8)

37.3 (+0.6)

36.7

baseline
R-50 Cascade
with mask

synchronize BN

defeormable conv

semantic branch

multi-scale training

better backbone

GARN finetune

multi-scale & flip testing
Experiments
Visualization
Visualization
Visualization
Visualization
Visualization
Experience

1. What can bring large gains?

Fundamental improvements of pipelines and structures
- Mask R-CNN
- FPN
- Cascade R-CNN
- (Synchronized) BN
- Deformable ConvNet
- ...
2. What may not?

Improvements of specific modules
• Precise RoI Pooling
• DetNet
• GCN
• Fitness NMS

Extra marginal components
• ASPP
• Spatial attention
• Additional R-CNN/PSPNet
Experience

2. What may not?

• Increasing model complexity can eat most of the gains
• Combination of ideas is not trivial
• May not be universal or robust
• Time is limited or wrong implementation
Experience

3. The annotation quality may limit the performance.

ground truth  segmentation results
Experience

3. The annotation quality may limit the performance.
Experience

4. Engineering tricks matter.

Reproducing detection pipelines is not very easy.

- Some component works well in one DL framework, but it takes us long time to reimplement and debug it with another framework.
- It takes only 2 hours to implement an algorithm, but it may take 1 week to reproduce the performance reported in the paper.
- ...

Experience

4. Engineering tricks matter.

There are traps everywhere.

• A wrong implementation of flip testing even decreases the mAP, the cause proves to be the rounding operation of bbox coordinates.
• A single pixel shift can lead to 1 point drop.
4. Engineering tricks matter.

reproduce existing methods
(20 days)

performance tuning
(30 days)

explore new ideas
(30 days)

We could do better if we already have a good codebase.
One more thing
**Codebase**

- **Comprehensive**
  - ✓ RPN
  - ✓ Mask R-CNN
  - ✓ Cascade R-CNN
  - ✓ More … …
  - ✓ Fast/Faster R-CNN
  - ✓ FPN
  - ✓ RetinaNet

- **High performance**
  - ✓ Better performance
  - ✓ Optimized memory consumption
  - ✓ Faster speed

- **Handy to develop**
  - ✓ Written with PyTorch
  - ✓ Modular design

GitHub: mmdet
Thank you!