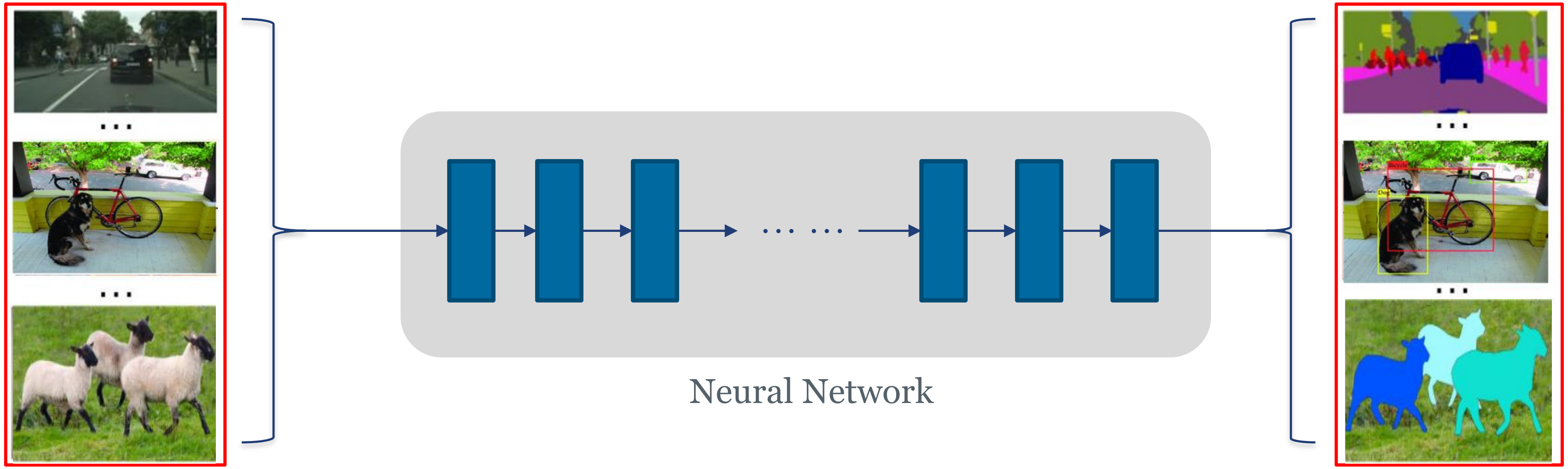
# FaPN: Feature-aligned Pyramid Network for Dense Image Prediction

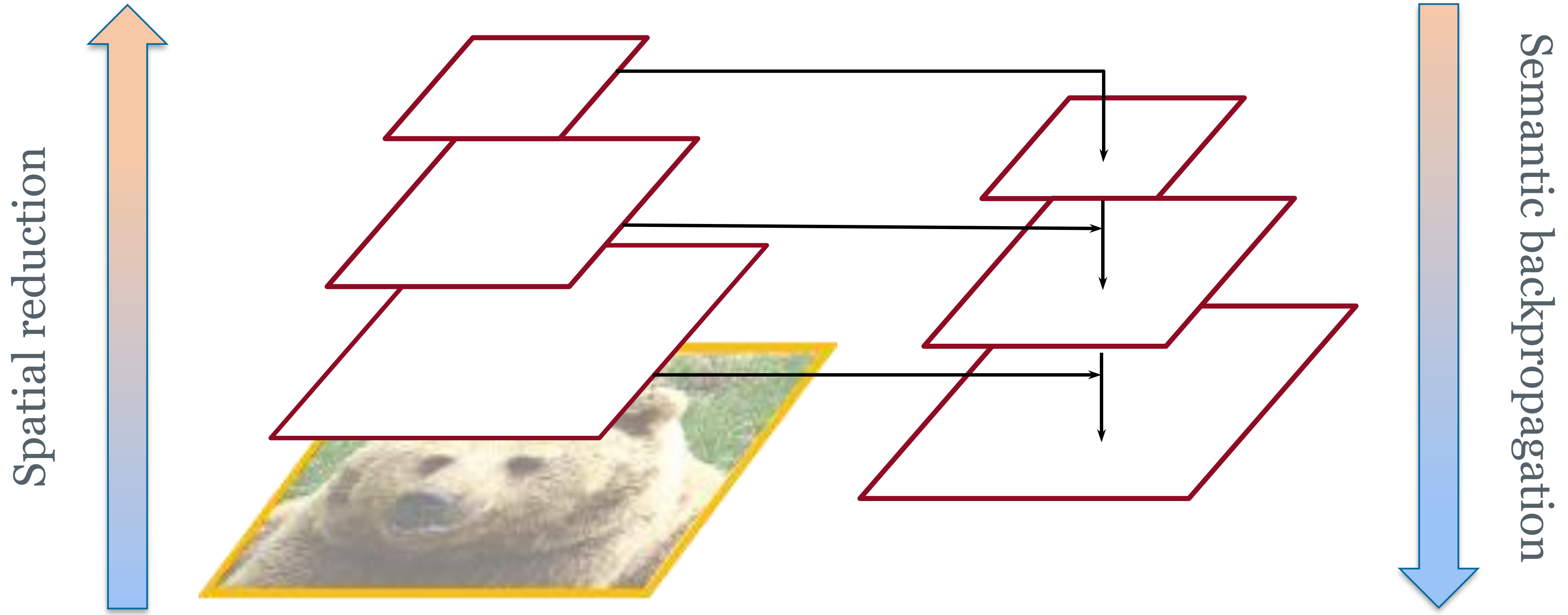
Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He
Department of Computer Science and Engineering,
Southern University of Science and Technology

# Dense image prediction




Dense image prediction is a pixel-level classification task that includes semantic segmentation, object detection, instance segmentation, et.al.

# Feature pyramid network for dense image prediction

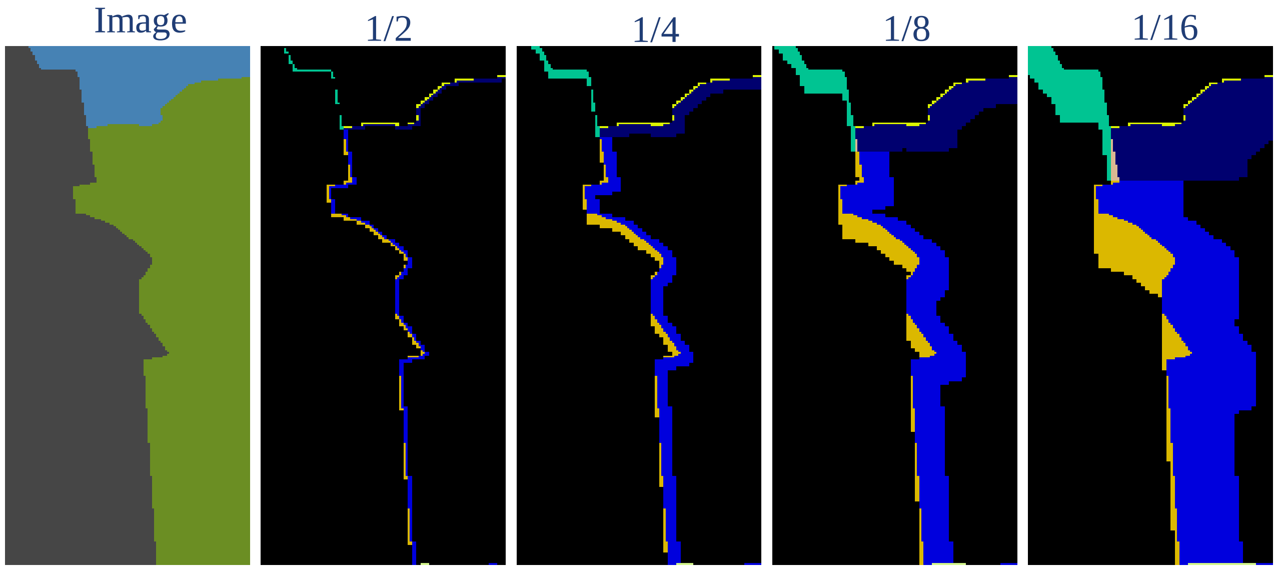



- Spatial reduction with downsampling will make the features on the top have larger perceptive fileds as stronger semantics for better classification.
- Semantic backpropagation with upsampling aims to distribute the semantics back to their corresponding locations at each scale to achieve rich semantics at all levels.

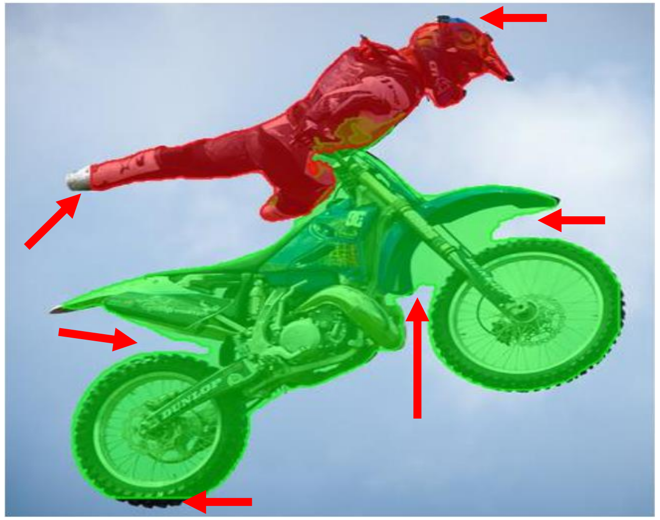
# Motivation

- Step-by-step downsampling makes the features lose location details progressively and dramatically.
- When without any accurate location reference , non-learnable upsampling operations will misplace the semantic feature into the upscaled map, i.e.,  misaligned context.
- The locality of convolution and upsampling makes the scope of misaligned context is local in which the object boundaries will suffer from severe misclassification due to the ambiguous context.

**Two examples to illustrate the misaligned boundaries:**

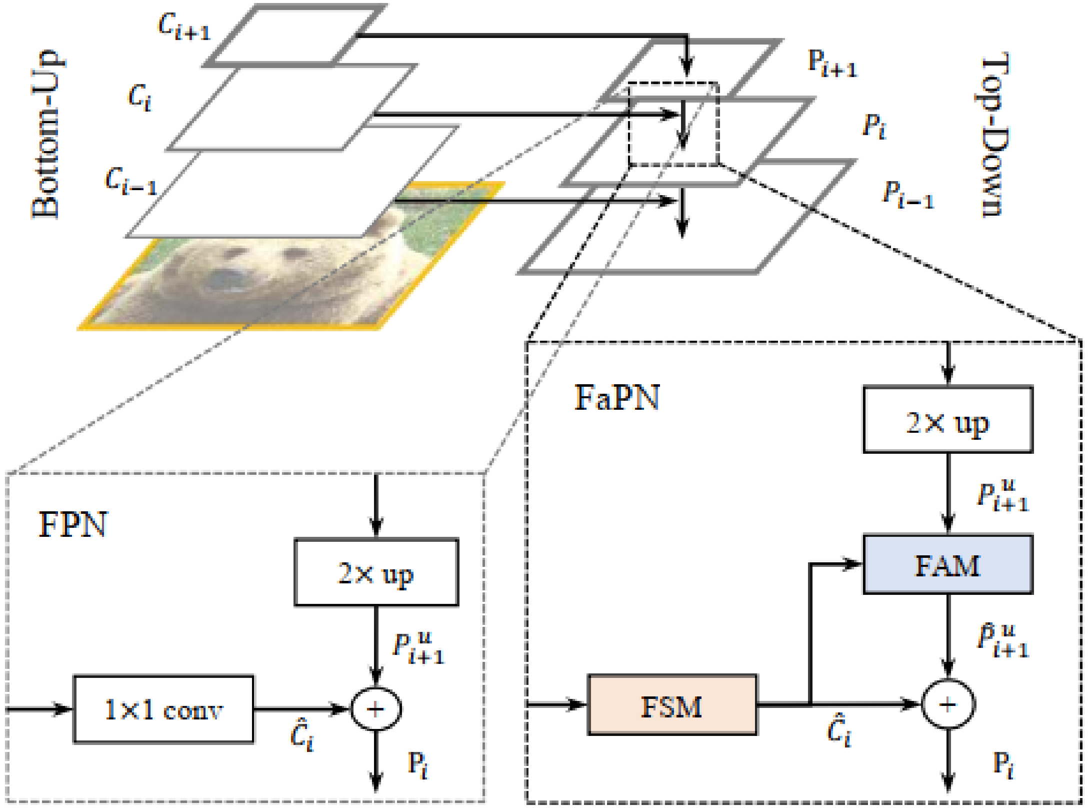



Differences between the image and image after rescaled. The difference exists over object boundaries and the area of difference is increasing as the downsamping rate.



An example result from FPN.

# FaPN: Feature-aligned pyramid network for dense image prediction



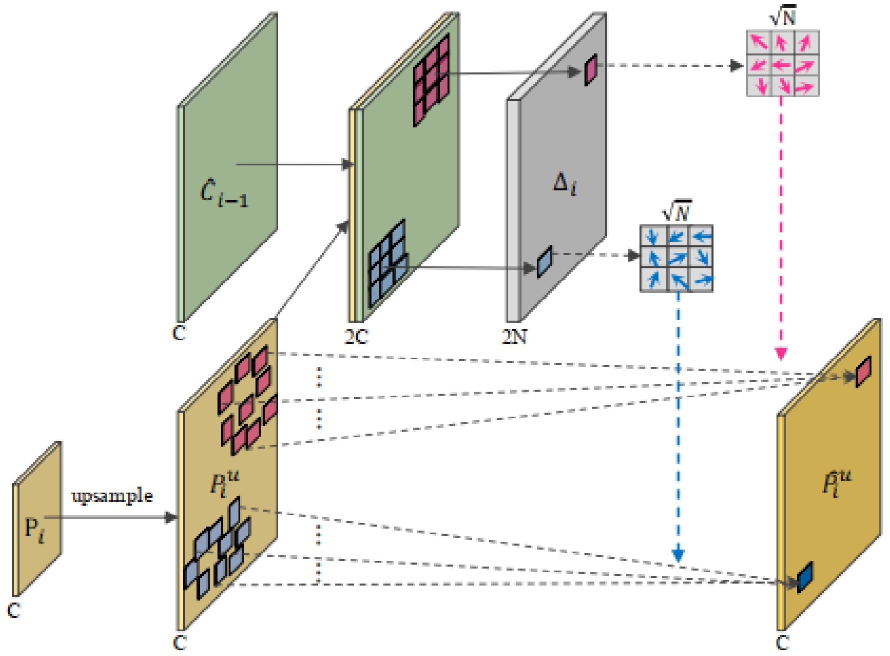

- Compared to FPN, our FaPN has two additional modules, i.e., FAM and FSM.
- Our FaPN is flexible and can be placed in any FPN-based method by simple replacement.

# Feature alignment and selection modules

**Feature alignment module (FAM)**:

$$\hat{\mathbf{P}}_i^u = f_a\big(\mathbf{P}_i^u, \boldsymbol{\Delta}_i\big),$$
$$\boldsymbol{\Delta}_i = f_o\big([\hat{\mathbf{C}}_{i-1}, \mathbf{P}_i^u]\big)$$

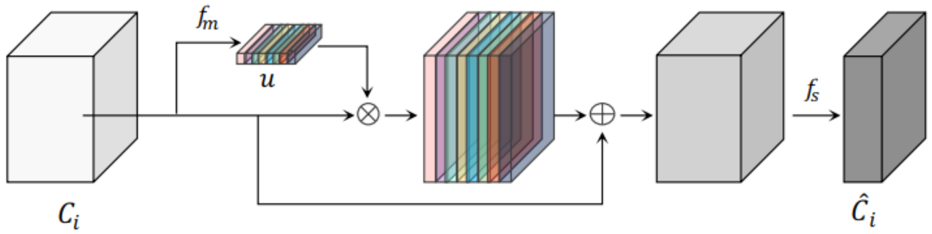
- ➢ Learning the offsets from the differences between the detailed and upsampled features.
- ➢ Aligning upsampled features with the learned offsets.




**Feature selection module (FSM)**:

$$\hat{\mathbf{C}}_i = f_s(\mathbf{C}_i + \mathbf{u} * \mathbf{C}_i),$$
$$\mathbf{u} = f_m(\mathbf{z}),$$

- ➢ Modeling the importance of each feature map in the detailed features.
- ➢ Emphasizing the detailed rich features by multiplying the importance values before channel reduction.



# Ablation Study

| method | backbone | #Params (M) | mIoU (%) |
|---|---|---|---|
| FPN | R50 | 28.6 (+4.5) | 77.4 (+2.6) |
| FPN + extra 3×3 conv. | R50 | 33.4 (-0.3) | 77.5 (+2.5) |
| FPN | R101 | 47.6 (-14.5) | 78.9 (+1.1) |
| FPN + FAM | R50 | 31.7 (+1.4) | 79.7 (+0.3) |
| FPN + FAM + SE | R50 | 33.1 (+0.0) | 78.8 (+1.2) |
| FPN + FAM + FSM (FaPN) | R50 | 33.1 (+0.0) | **80.0** (+0.0) |
| FPN + deconv + FSM | R50 | 32.7 (+0.4) | 76.7 (+3.3) |
| FPN + FAM† + FSM | R50 | 32.7 (+0.4) | 79.3 (+0.7) |

- # 2~3: Additional learnable parameters in FPN would not boost the performance greatly as our FaPN.
- # 4~5: FAM is compatible with FSM, while the SE module adversely affects the performance.
- # 7: Replacing the non-learnable upsampling with a learnable one could not improve the performance, i.e., addressing feature misalignment
- # 8: Location reference matters during alignment.

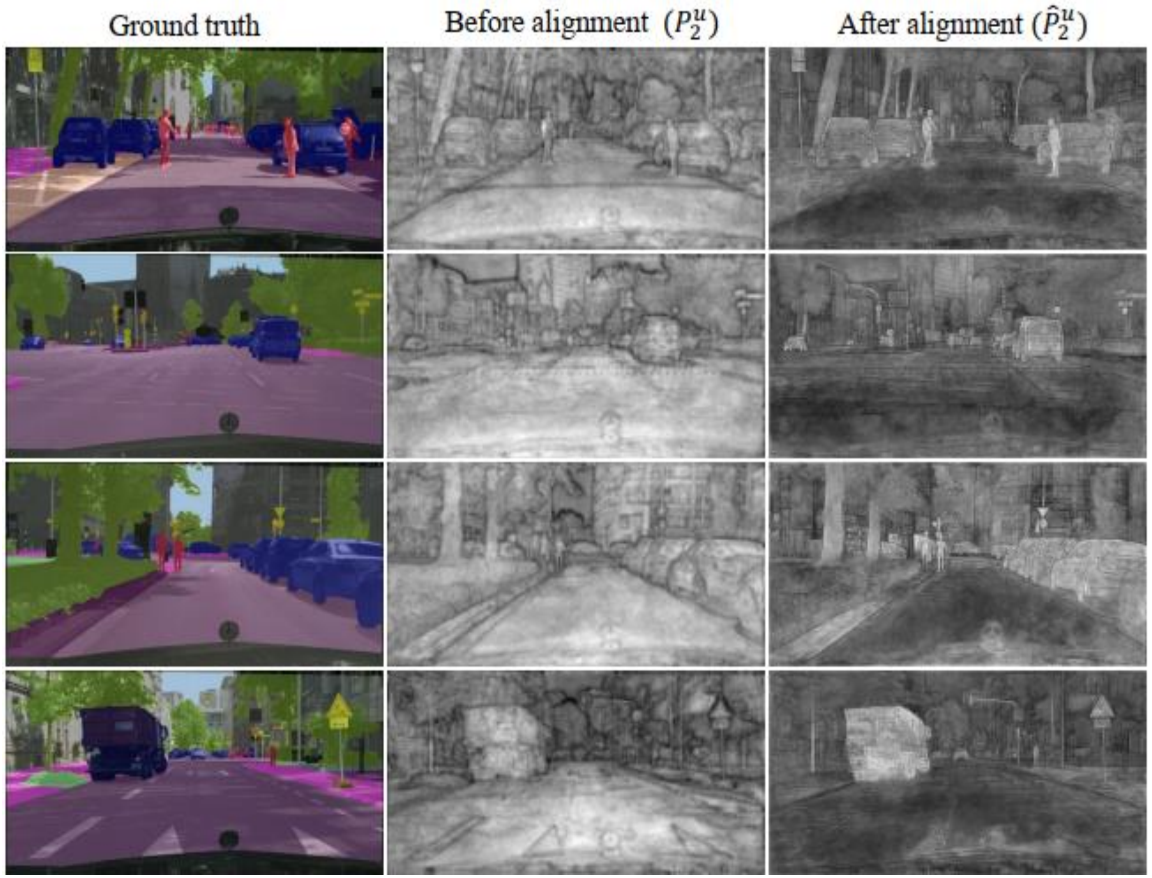
# Boundary prediction analysis

| method | backbone | 3px | 5px | 8px | 12px | mean |
|---|---|---|---|---|---|---|
| FPN | PointRend [21] R50 | 46.9 | 53.6 | 59.3 | 63.8 | 55.9 |
| FaPN | | 49.2 | 56.2 | 62.0 | 66.4 | 58.5 |
| *improvement* | | (+2.3) | (+2.6) | (+2.7) | (+2.6) | (+2.6) |
| FPN | PointRend [21] R101 | 47.8 | 54.6 | 60.5 | 64.9 | 57.0 |
| FaPN | | 50.1 | 57.1 | 62.9 | 67.2 | 59.3 |
| *improvement* | | (+2.3) | (+2.5) | (+2.4) | (+2.3) | (+2.3) |

Segmentation performance around boundaries

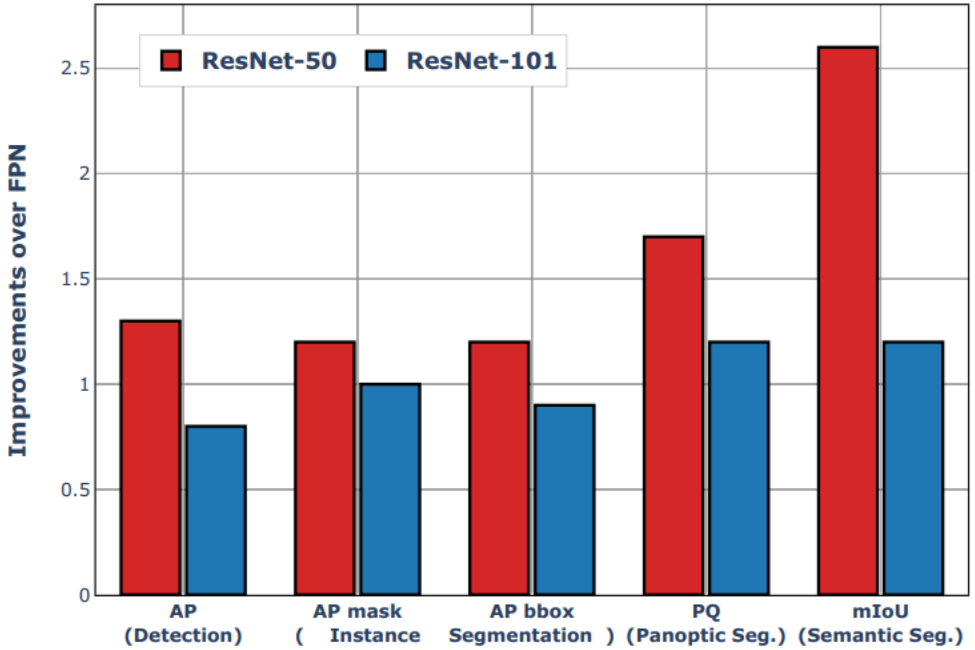
Both the quantitative evaluation and qualitative observation are consistent:

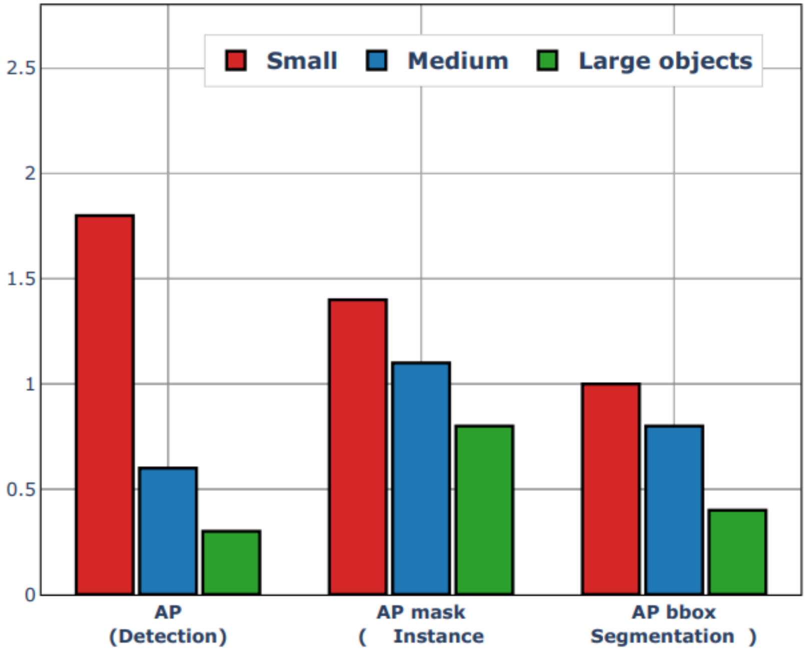
- Compared with FPN, FaPN achieves higher mIoU over the boundary segmentation.
- Raw upsampled features are noisy and fluctuating, while the aligned features are smooth and containing more precise object boundaries.




Visualization of the input to and the output from FAM

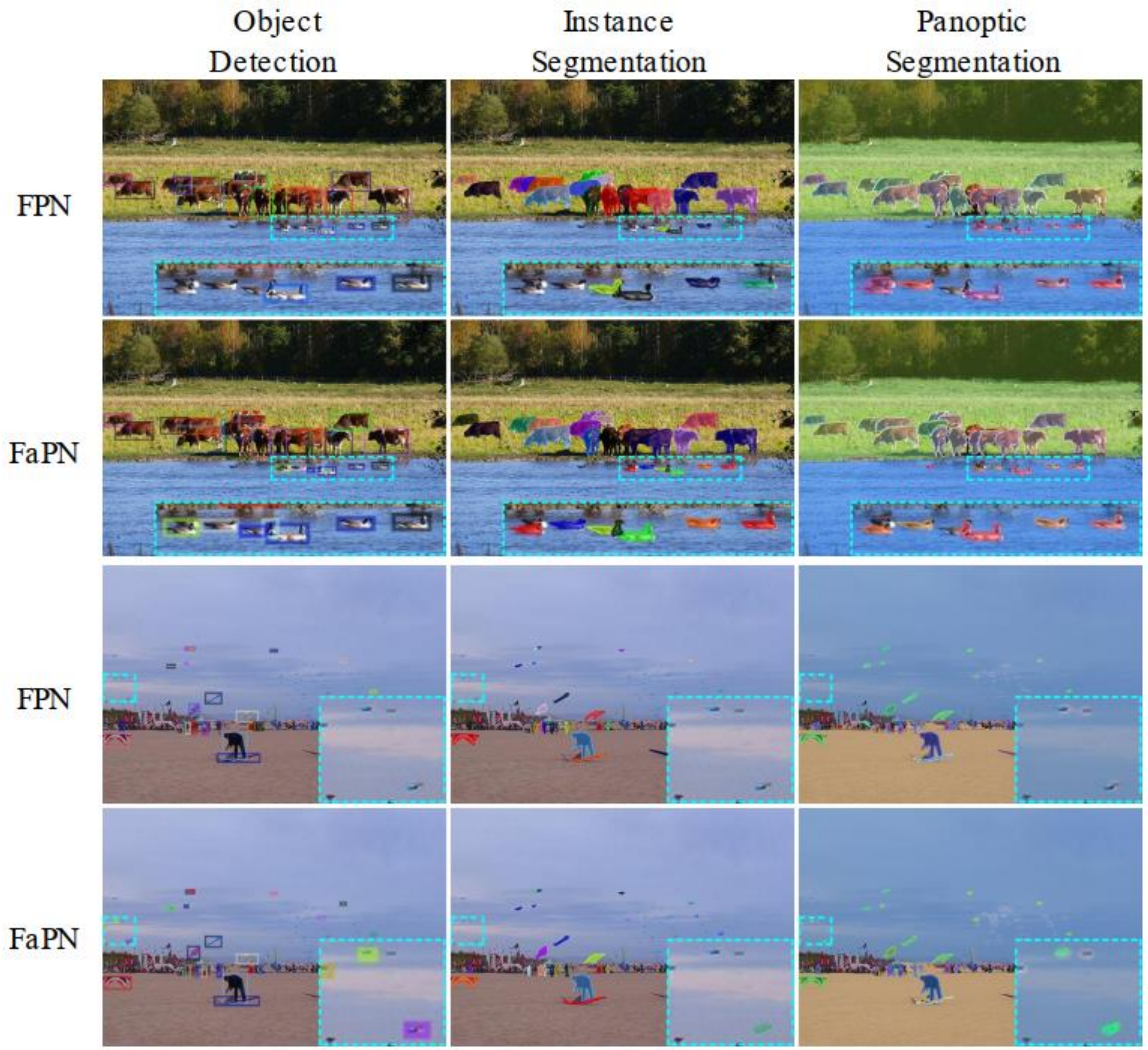
# Main results



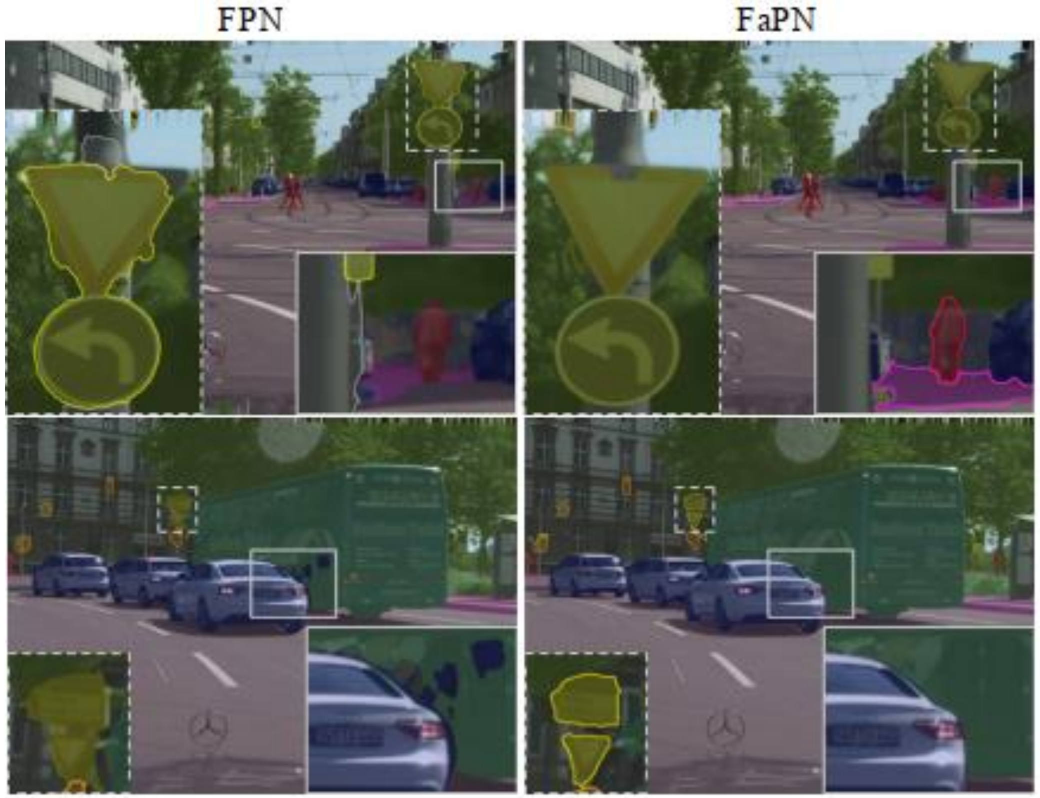




- ➢ Our FaPN can be applied in four dense image prediction tasks.
- ➢ A simple replacement of FPN with FaPN in five representative methods yields an overall improvement of 1.2 - 2.6 points in AP / mIoU.
- ➢ Our FaPN mainly improves the performance of small objects.

# Example prediction visualizations




- Compared to FPN, FaPN significantly improves the performance of small objects.
- FaPN also has finer segmentation on object boundaries.



# Border explorations

(a) ADE20K `val`

| method | backbone | crop size | mIoU (s.s.) | mIoU (m.s.) |
|---|---|---|---|---|
| OCRNet [49] | R101 | 520 × 520 | - | 45.3 |
| AlignSeg [17] | R101 | 512 × 512 | - | 46.0 |
| SETR [51] | ViT-L† | 512 × 512 | - | 50.3 |
| Swin-UperNet [27] | Swin-L† | 640 × 640 | - | 53.5 |
| MaskFormer [8] | Swin-L† | 640 × 640 | 54.1 | 55.6 |
| MaskFormer + FaPN | Swin-L† | 640 × 640 | **55.2** | **56.7** |

(b) COCO-Stuff-10K `test`

| method | backbone | crop size | mIoU (s.s.) | mIoU (m.s.) |
|---|---|---|---|---|
| OCRNet [49] | | 520 × 520 | - | 39.5 |
| MaskFormer [8] | R101 | 640 × 640 | 38.1 | 39.8 |
| MaskFormer + FaPN | | 640 × 640 | **39.6** | **40.6** |

- Our FaPN also advances the Transformer-based methods.
- When augmented with MaskFormer, our FaPN achieves the 2nd best result over ADE20k-150.

(a) Cityscapes

| method | backbone | crop size | FPS | mIoU (*val*) | mIoU (*test*) |
|---|---|---|---|---|---|
| ESPNet [36] | † | 512 × 1024 | 113 | - | 60.3 |
| ESPNetV2 [37] | † | 512 × 1024 | - | 66.4 | 66.2 |
| FaPN | R18 | 512 × 1024 | **142** | **69.2** | **68.8** |
| BiSeNet [49] | R18 | 768 × 1536 | 65.6 | 74.8 | 74.7 |
| FaPN | R18 | 768 × 1536 | **78.1** | **75.6** | **75.0** |
| SwiftNet [39] | R18 | 1024 × 2048 | **39.9** | 75.4 | 75.5 |
| ICNet [51] | R50 | 1024 × 2048 | 30.3 | - | 69.5 |
| FaPN | R34 | 1024 × 2048 | 30.2 | **78.5** | **78.1** |

- Our FaPN is also efficient and applied to real-time segmentation methods.
- A simple replacement of FPN with our FaPN achieves competitive results against existing dedicated methods.

(b) COCO-Stuff-10K

| method | backbone | crop size | FPS | mIoU (*val*) |
|---|---|---|---|---|
| BiSeNet [49] | R18 | | - | 28.1 |
| BiSeNetV2 [48] | † | | 42.5 | 28.7 |
| ICNet [51] | R50 | 640 × 640 | 35.7 | 29.1 |
| FaPN | R18 | | **154** | 28.4 |
| FaPN | R34 | | 110 | **30.3** |

# Thanks!

The code is available at:
https://github.com/ShihuaHuang95/FaPN-full