Deformable Convolutional Networks

Jifeng Dai^
With Haozhi Qi*^, Yuwen Xiong*^, Yi Li*^, Guodong Zhang*^, Han Hu, Yichen Wei
Visual Computing Group
Microsoft Research Asia
(* interns at MSRA, ^ equal contribution)

Highlights

- **Enabling effective modeling of spatial transformation** in ConvNets
- **No additional supervision** for learning spatial transformation
- **Significant accuracy improvements** on sophisticated vision tasks

Code is available at https://github.com/msracver/Deformable-ConvNets

Modeling Spatial Transformations

- A long standing problem in computer vision

Deformation:

Scale:

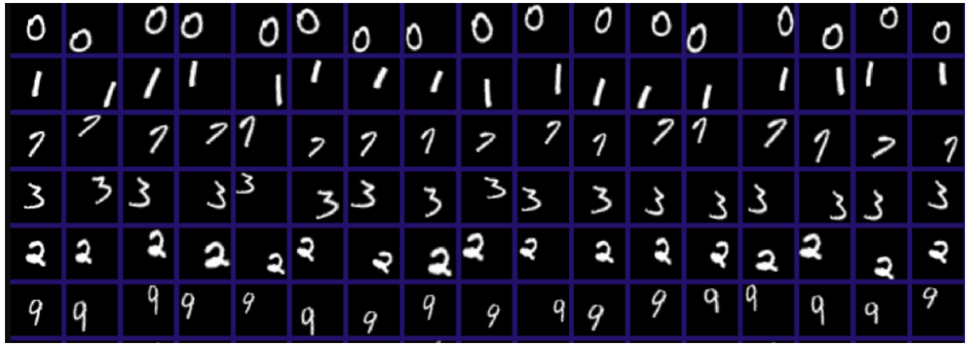
Viewpoint variation:

Intra-class variation:

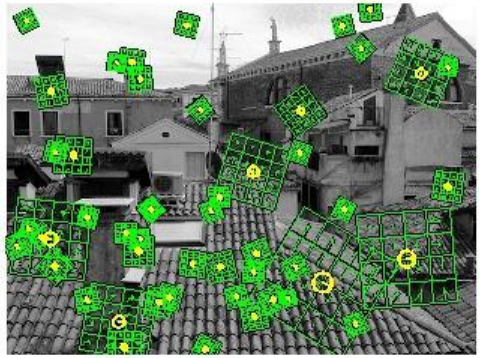
(Some examples are taken from Li Fei-fei's course CS223B, 2009-2010)

Traditional Approaches

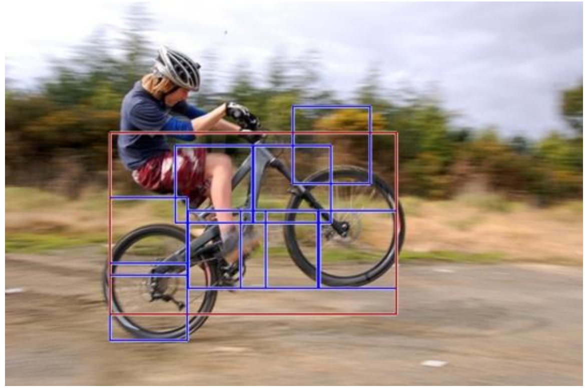
- 1) To build training datasets with sufficient desired variations



- 2) To use transformation-invariant features and algorithms



Scale Invariant Feature Transform (SIFT)

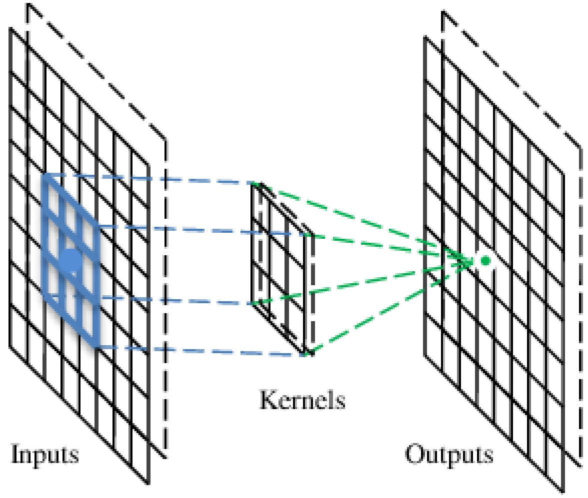


Deformable Part-based Model (DPM)

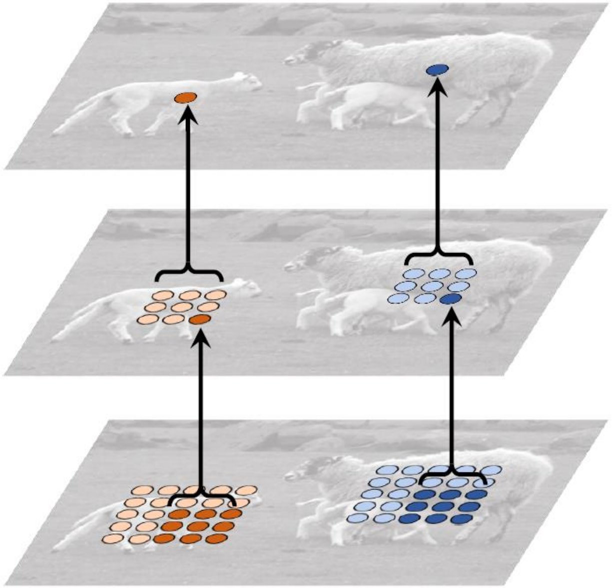
- Drawbacks: geometric transformations are assumed fixed and known, hand-crafted design of invariant features and algorithms

Spatial transformations in CNNs

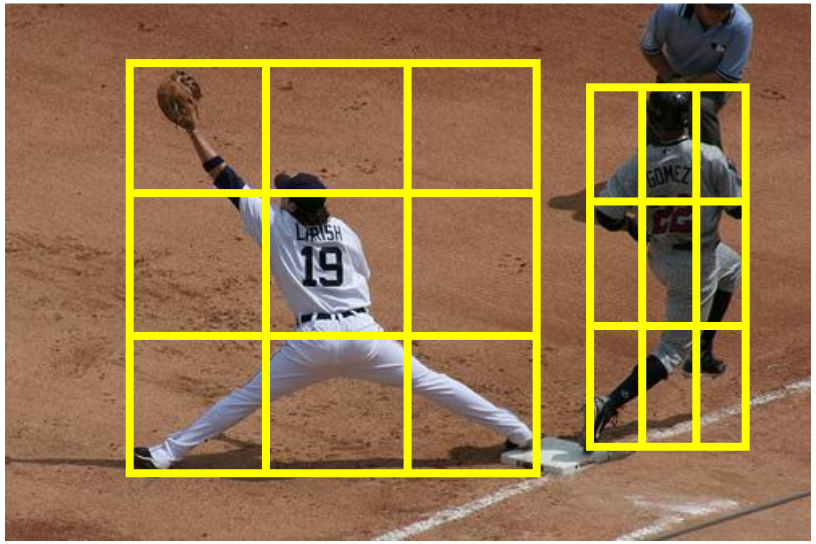
- Regular CNNs are inherently limited to model large unknown transformations
    - The limitation originates from the fixed geometric structures of CNN modules



regular convolution



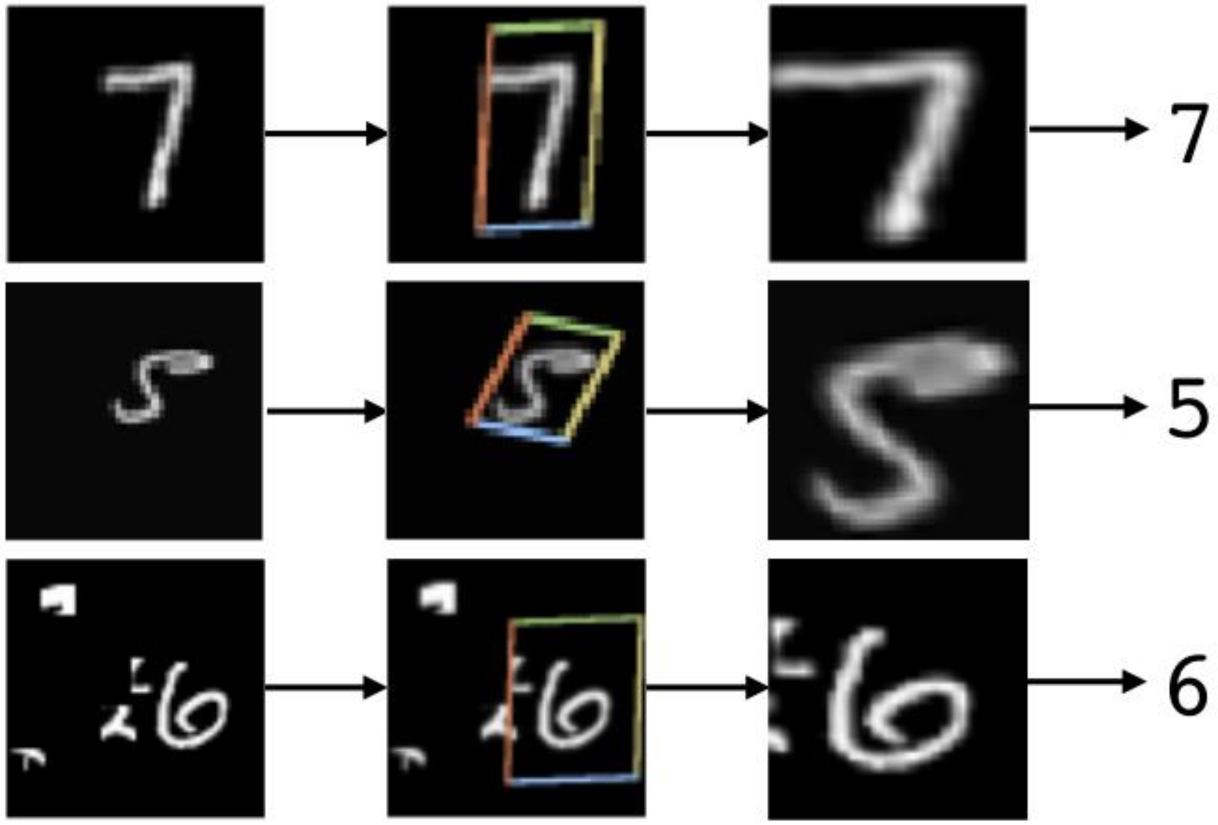
2 layers of regular convolution



regular RoI Pooling

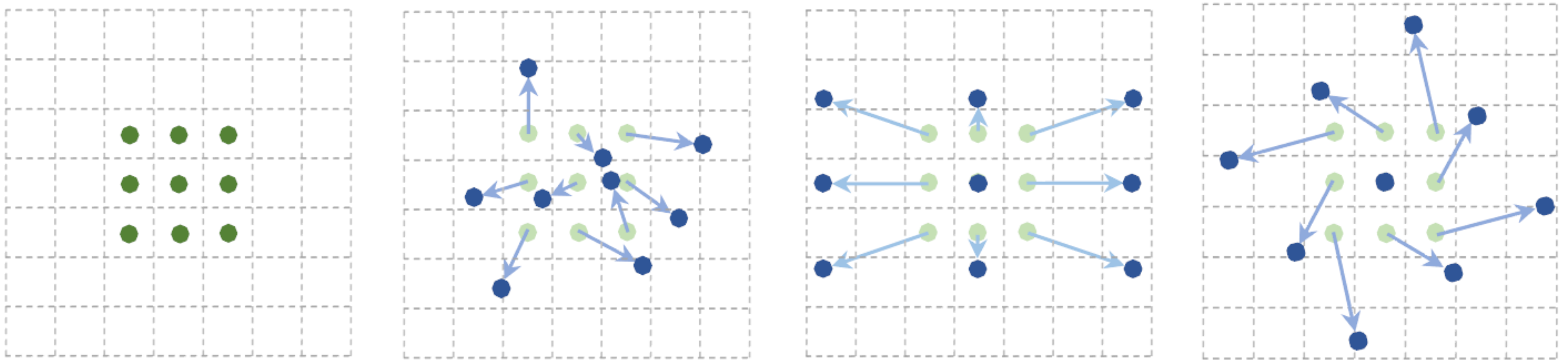
Spatial Transformer Networks

- Learning a global, parametric transformation on feature maps
  - Prefixed transformation family, infeasible for complex vision tasks



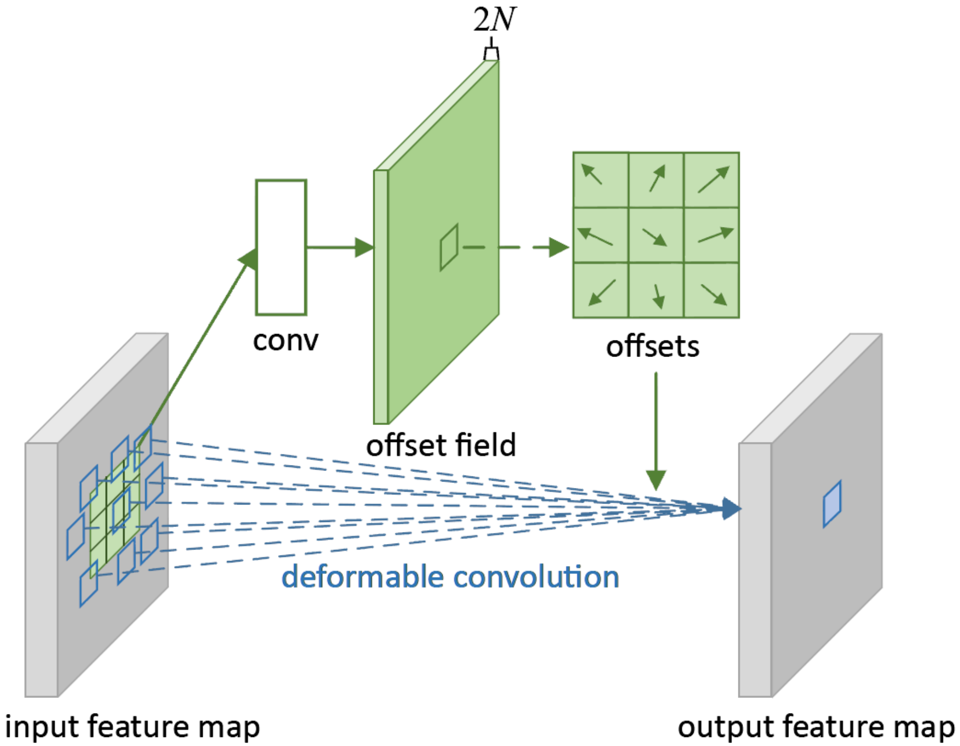
Deformable Convolution

- Local, dense, non-parametric transformation
  - Learning to deform the sampling locations in the convolution/RoI Pooling modules



regular   deformed   scale & aspect ratio   rotation

Deformable Convolution



Regular convolution

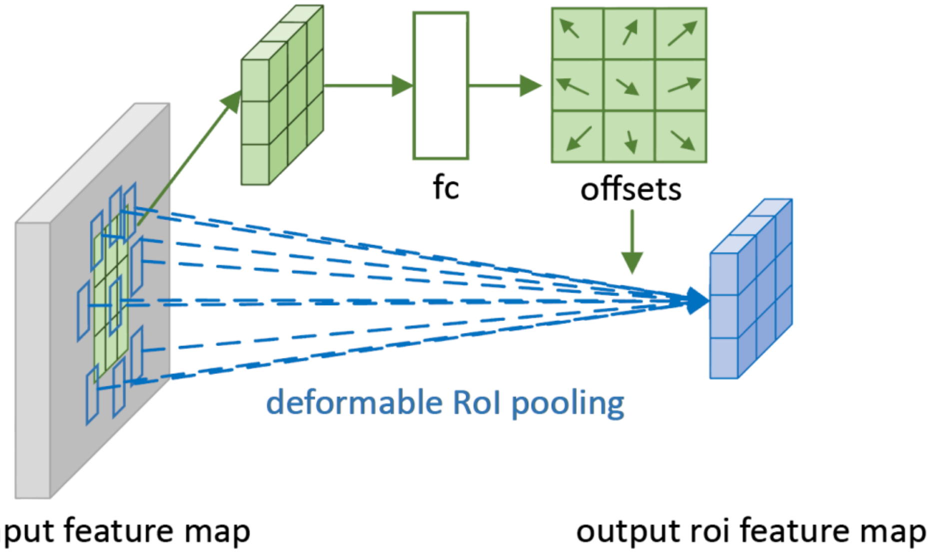
$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n)$$

Deformable convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)$$

where $\Delta\mathbf{p}_n$ is generated by a sibling branch of regular convolution

Deformable RoI Pooling



deformable RoI Pooling

Regular RoI pooling

$$\mathbf{y}(i,j) = \sum_{\mathbf{p} \in bin(i,j)} \mathbf{x}(\mathbf{p}_0 + \mathbf{p}) / n_{ij}$$

Deformable RoI pooling

$$\mathbf{y}(i,j) = \sum_{\mathbf{p} \in bin(i,j)} \mathbf{x}(\mathbf{p}_0 + \mathbf{p} + \Delta\mathbf{p}_{ij}) / n_{ij}$$

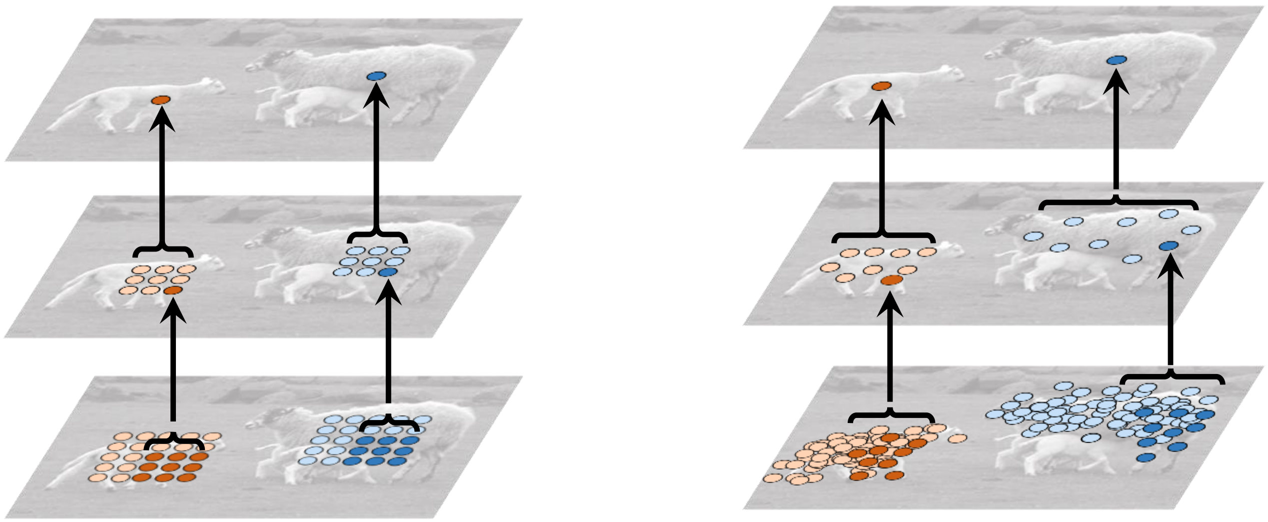
where $\Delta\mathbf{p}_{ij}$ is generated by a sibling fc branch

Deformable ConvNets

- Same input & output as the plain versions
  - Regular convolution -> deformable convolution
  - Regular RoI pooling -> deformable RoI pooling

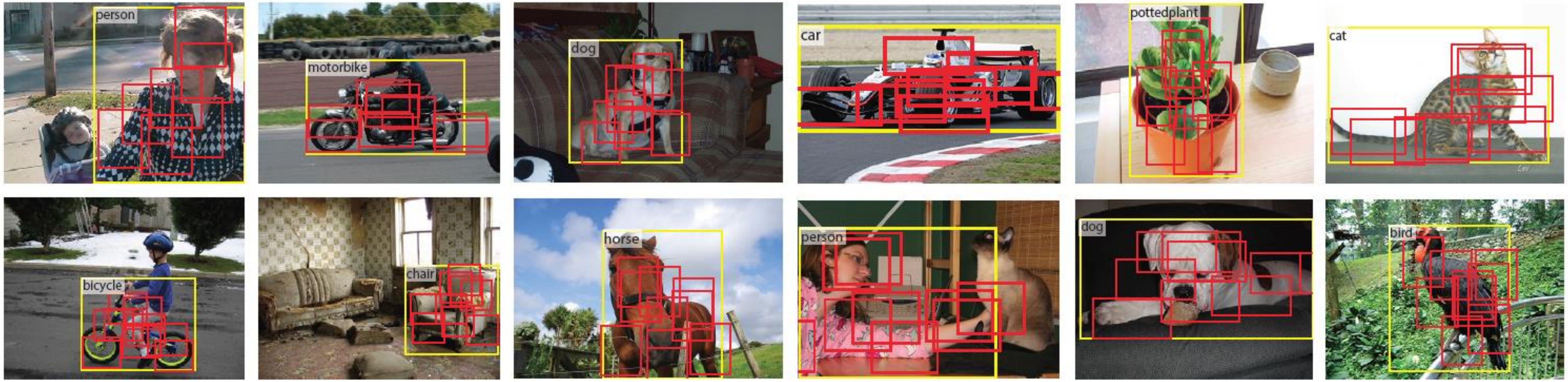
- End-to-end trainable without additional supervision

Sampling Locations of Deformable Convolution



(a) standard convolution (b) deformable convolution

Part Offsets in Deformable RoI Pooling



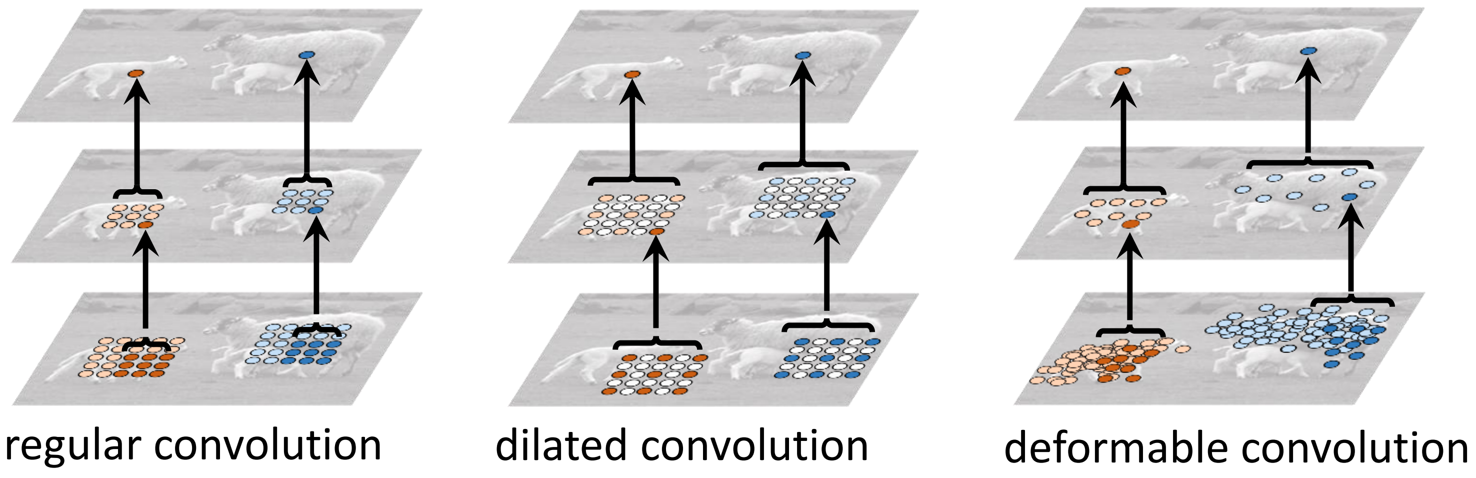
Ablation Experiments on VOC & Cityscapes

- Number of deformable convolutional layers (using ResNet-101)

# deformable layers	DeepLab		Class-aware RPN		Faster R-CNN (2fc)		R-FCN	
	mIoU@V (%)	mIoU @C (%)	mAP@0.5 (%)	mAP@0.7 (%)	mAP@0.5 (%)	mAP@0.7 (%)	mAP@0.5 (%)	mAP@0.7 (%)
None (0, baseline)	69.7	70.4	68.0	44.9	78.1	62.1	80.0	61.8
Res5c (1)	73.9	73.5	73.5	54.4	78.6	63.8	80.6	63.0
Res5b, c (2)	74.8	74.4	74.3	56.3	78.5	63.3	81.0	63.8
Res5a, b, c (3) (default)	**75.2**	**75.2**	74.5	57.2	78.6	63.3	81.4	64.7
Res5 & res4b22, b21, b20 (6)	74.8	75.1	**74.6**	**57.7**	**78.7**	**64.0**	**81.5**	**65.4**

Deformable ConvNets v.s. dilated convolution

Deformable modules	DeepLab mIoU@V/@C	Class-aware RPN mAP@0.5/@0.7	Faster R-CNN mAP@0.5/@0.7	R-FCN mAP@0.5/@0.7
Dilated convolution (2, 2, 2) (default)	69.7 / 70.4	68.0 / 44.9	78.1 / 62.1	80.0 / 61.8
Dilated convolution (4, 4, 4)	73.1 / 71.9	72.8 / 53.1	78.6 / 63.1	80.5 / 63.0
Dilated convolution (6, 6, 6)	73.6 / 72.7	73.6 / 55.2	78.5 / 62.3	80.2 / 63.5
Dilated convolution (8, 8, 8)	73.2 / 72.4	73.2 / 55.1	77.8 / 61.8	80.3 / 63.2
Deformable convolution	**75.3 / 75.2**	**74.5 / 57.2**	78.6 / 63.3	81.4 / 64.7
Deformale RoI pooling	N.A	N.A	78.3 / 66.6	81.2 / 65.0
Deformale convolution & RoI pooling	N.A	N.A	**79.3 / 66.9**	**82.6 / 68.5**



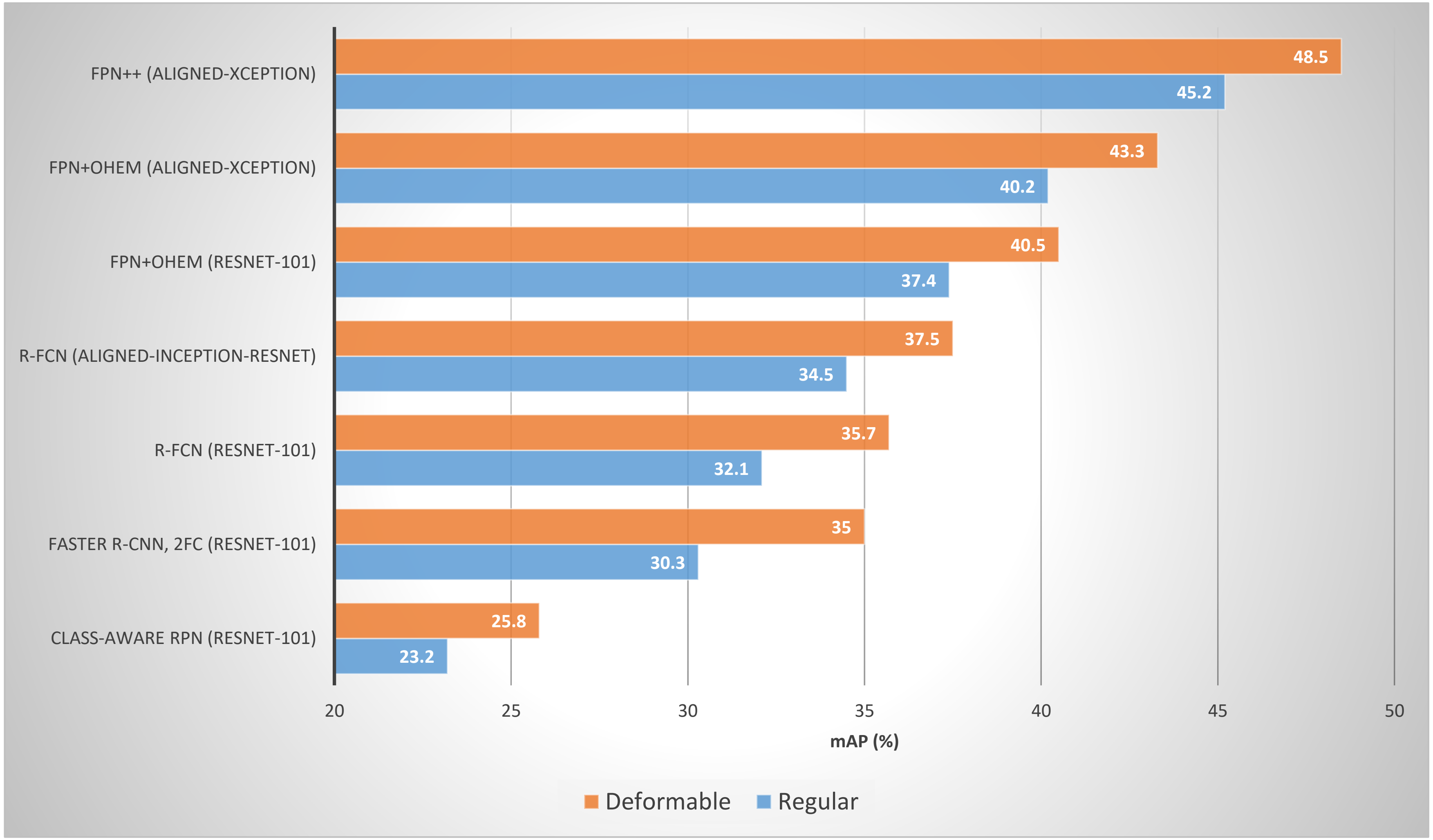
Model Complexity and Runtime on VOC & Cityscapes

- Deformable ConvNets v.s. regular ConvNets

Method	# params	Net forward (sec)	Runtime (sec)
Regular DeepLab @Cityscapes	46.0M	0.610	0.650
Deformable DeepLab @Cityscapes	46.1 M	0.656	0.696
Regular DeepLab @VOC	46.0M	0.084	0.094
Deformable DeepLab @VOC	46.1 M	0.088	0.098
Regular Class-aware RPN	46.0 M	0.142	0.323
Deformable class-aware RPN	46.1 M	0.152	0.334
Regular Faster R-CNN (2fc)	58.3 M	0.147	0.190
Deformable Faster R-CNN (2fc)	59.9 M	0.192	0.234
Regular R-FCN	47.1 M	0.143	0.170
Deformable R-FCN	49.5 M	0.169	0.193

Object Detection on COCO

- Deformable ConvNets v.s. regular ConvNets



Conclusion

- Deformable ConvNets for dense spatial modeling
  - Simple, efficient, deep, and end-to-end
  - No additional supervision
  - Feasible and effective on sophisticated vision tasks for the first time