ICCV 2017 oral

Deformable Convolutional Networks

Jifeng Dai* Haozhi Qi*,† Yuwen Xiong*,† Yi Li*,† Guodong Zhang*,† Han Hu Yichen Wei

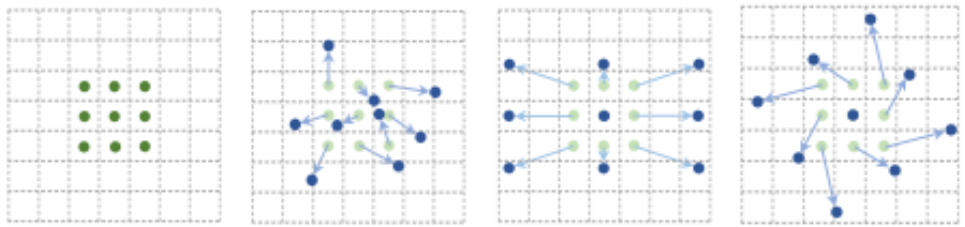
Microsoft Research Asia

{jifdai,v-haoq,v-yuxio,v-yii,v-guodzh,hanhu,yichenw}@microsoft.com

Related Work

- Augment the existing samples by geometric transformations
  - Limitations: assumed fixed and know transformations
- Transformation-invariant features
  - Limitations: difficult or infeasible for overly complex transformations

Different sampling locations



(a) (b) (c) (d)

Figure 1: Illustration of the sampling locations in 3×3 standard and deformable convolutions. (a) regular sampling grid (green points) of standard convolution. (b) deformed sampling locations (dark blue points) with augmented offsets (light blue arrows) in deformable convolution. (c)(d) are special cases of (b), showing that the deformable convolution generalizes various transformations for scale, (anisotropic) aspect ratio and rotation.

Deformable convolution

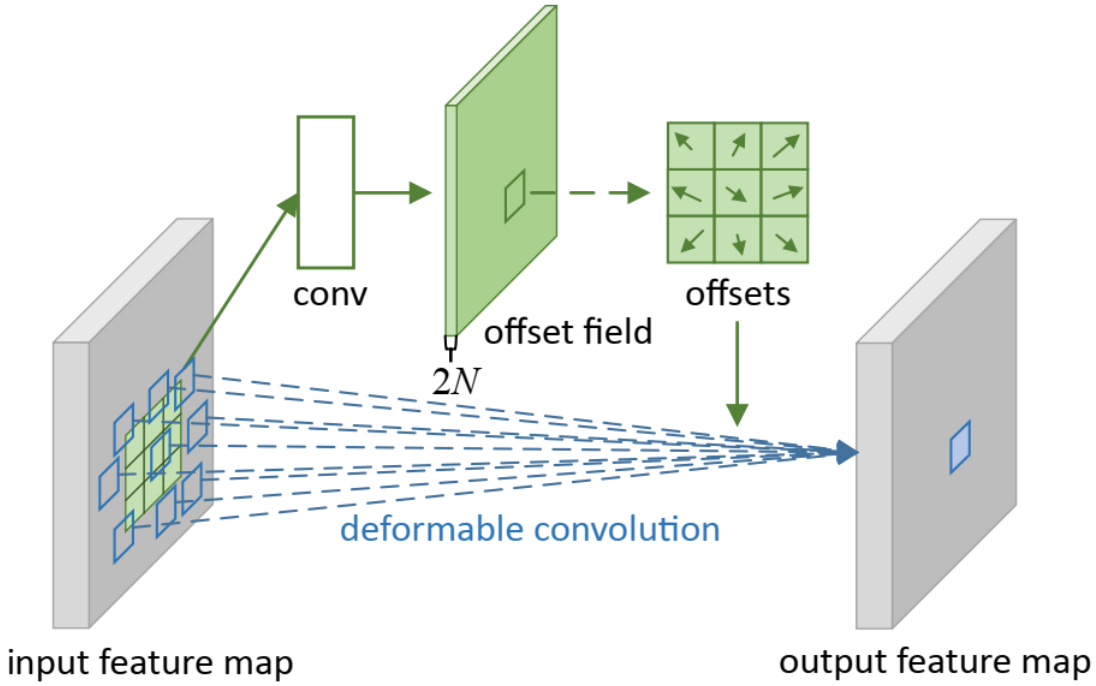


Figure 2: Illustration of 3×3 deformable convolution.

Deformable RoI

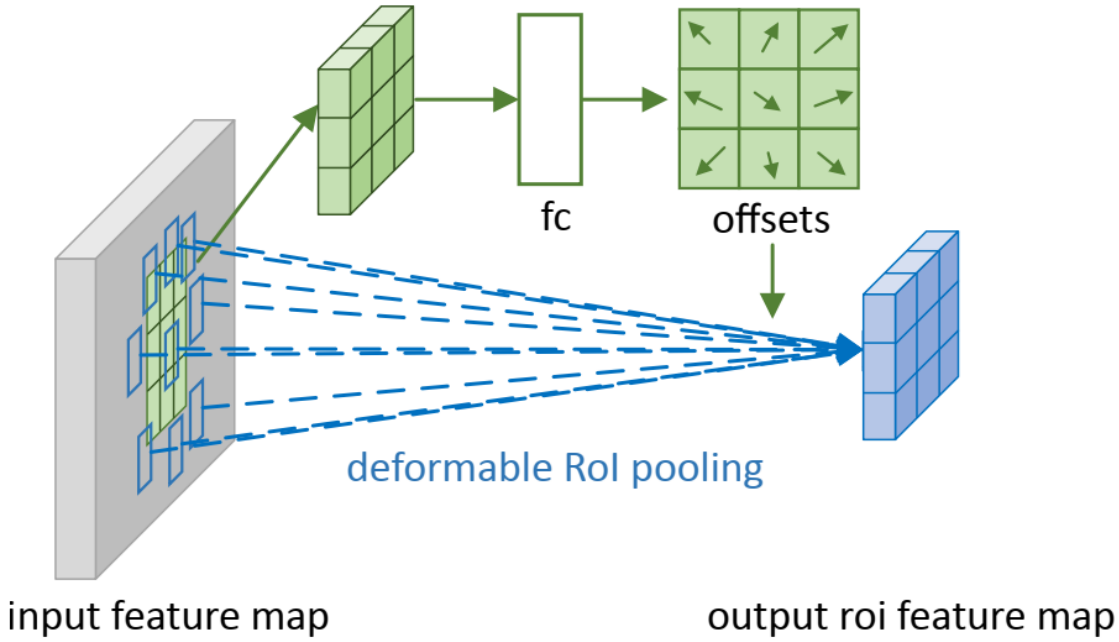


Figure 3: Illustration of 3×3 deformable RoI pooling.

Sampling locations examples

Figure 6: Each image triplet shows the sampling locations ($9^3 = 729$ red points in each image) in three levels of 3×3 deformable filters (see Figure 5 as a reference) for three activation units (green points) on the background (left), a small object (middle), and a large object (right), respectively.

Deformable RoI examples

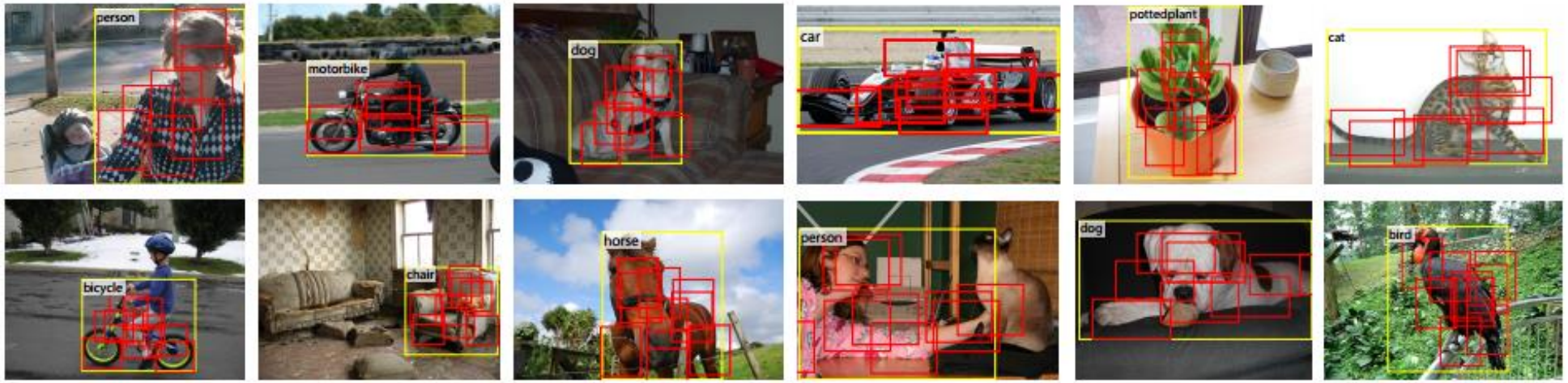


Figure 7: Illustration of offset parts in deformable (positive sensitive) RoI pooling in R-FCN [7] and 3×3 bins (red) for an input RoI (yellow). Note how the parts are offset to cover the non-rigid objects.

Experiments

method	backbone architecture	M	B	mAP@[0.5:0.95]	mAP^r@0.5	mAP@[0.5:0.95] (small)	mAP@[0.5:0.95] (mid)	mAP@[0.5:0.95] (large)
class-aware RPN	ResNet-101			23.2	42.6	6.9	27.1	35.1
Ours				**25.8**	**45.9**	**7.2**	**28.3**	**40.7**
Faster RCNN	ResNet-101			29.4	48.0	9.0	30.5	47.1
Ours				**33.1**	**50.3**	**11.6**	**34.9**	**51.2**
R-FCN	ResNet-101			30.8	52.6	11.8	33.9	44.8
Ours				**34.5**	**55.0**	**14.0**	**37.7**	**50.3**
Faster RCNN	Aligned-Inception-ResNet			30.8	49.6	9.6	32.5	49.0
Ours				**34.1**	**51.1**	**12.2**	**36.5**	**52.4**
R-FCN	Aligned-Inception-ResNet			32.9	54.5	12.5	36.3	48.3
Ours				**36.1**	**56.7**	**14.8**	**39.8**	**52.2**
R-FCN	Aligned-Inception-ResNet	✓		34.5	55.0	16.8	37.3	48.3
Ours		✓		37.1	57.3	18.8	39.7	52.3
R-FCN		✓	✓	35.5	55.6	17.8	38.4	49.3
Ours		✓	✓	**37.5**	**58.0**	**19.4**	**40.1**	**52.5**

Table 5: Object detection results of deformable ConvNets v.s. plain ConvNets on COCO test-dev set. M denotes multi-scale testing, and B denotes iterative bounding box average in the table.

Follow-ups

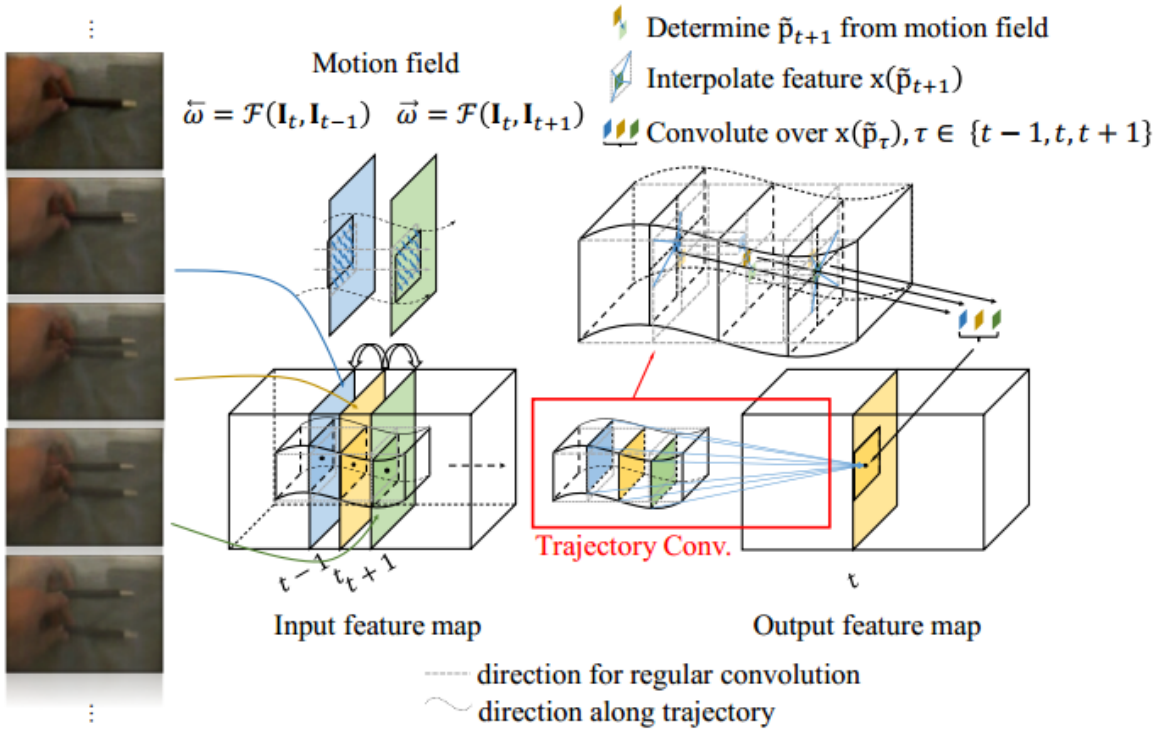


Figure 1: Illustration of our trajectory convolution. Given a sequence of video frames (`left`) and its corresponding input feature map of size $C \times T \times H \times W$ (`bottom-middle`; the dimension of channels C is simplified as one for clarity), in order to calculate the response of a specific point at time step t, we leverage the motion fields $\overleftarrow{\omega}$ and $\overrightarrow{\omega}$ (`top-middle`; the arrows in blue denote the motion velocity) to determine the sampling location at neighboring time step $t-1$ and $t+1$ in the sense of tracking along the motion path. The response is denoted on the output feature map (`bottom-right`). The operation of trajectory convolution (denoted in a red box) is illustrated on the `top-right`. This figure is best viewed in color.

Zhao, Yue, Yuanjun Xiong, and Dahua Lin. "Trajectory Convolution for Action Recognition." NIPS. 2018.