Deformable v2

Lina Hu

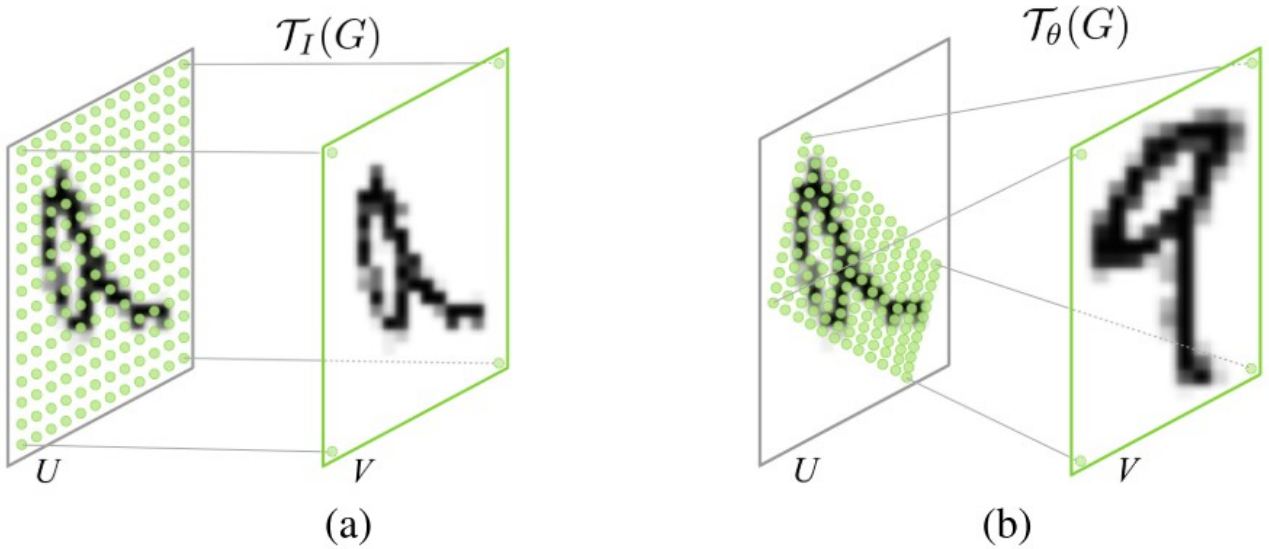
ShanghaiTech University

January 19, 2019

Deformable

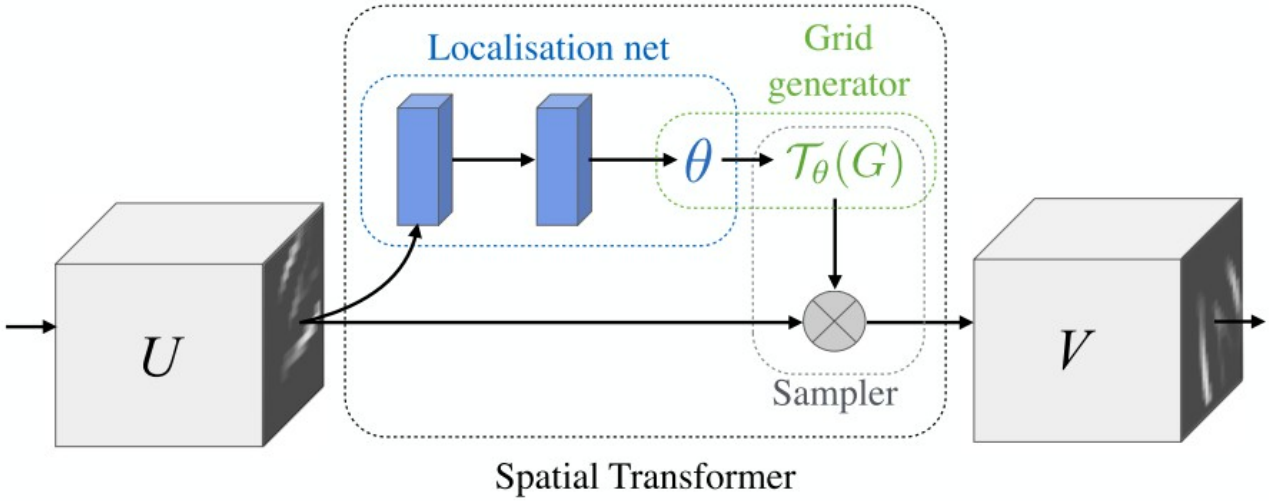
1. Spatial Transform Network (STN) [3]
2. Mask R-CNN [2]
3. Deformable V1 [1]
4. Deformable V2 [4]

STN



$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

STN



Spatial Transformer

Each (x_i^s, y_i^s) coordinate in $\mathcal{T}_\theta(G)$ defines the spatial location in the input where a sampling kernel is applied to get the value at a particular pixel in the output V. This can be written as

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \quad \forall i \in [1 \dots H'W'] \quad \forall c \in [1 \dots C] \tag{3}$$

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

Deform_V1 & V2

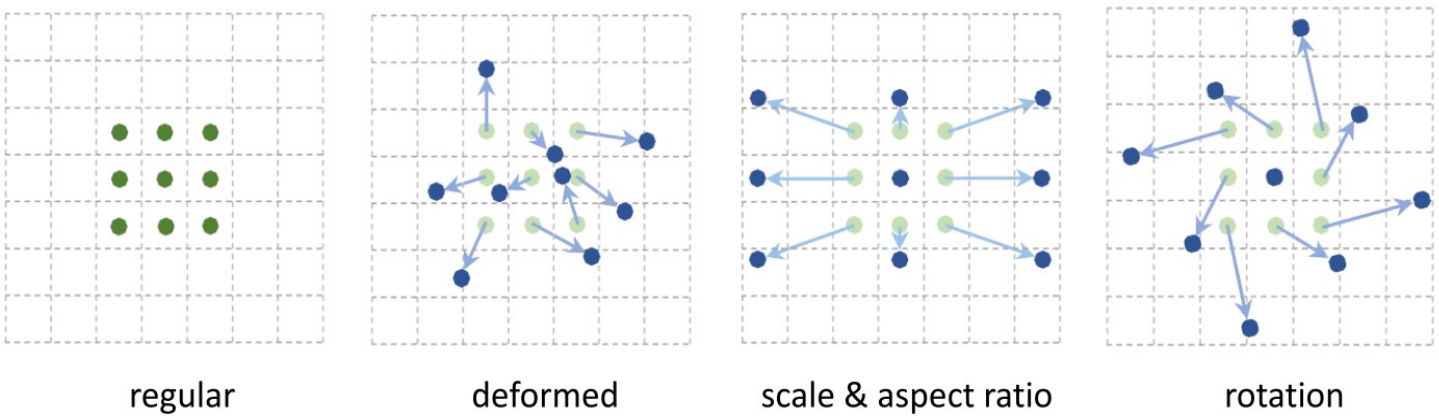
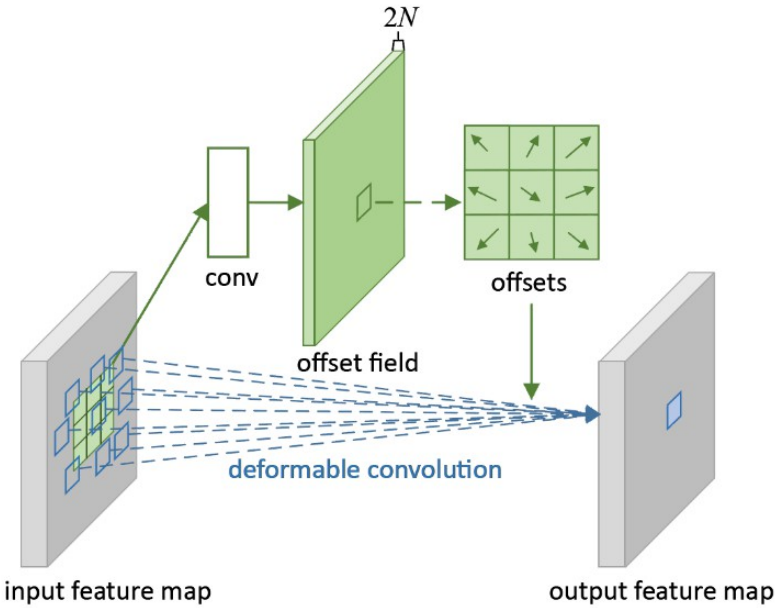


Illustration of the sampling locations in 3x3 standard and deformable convolutions.

Deform_V1 & V2



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

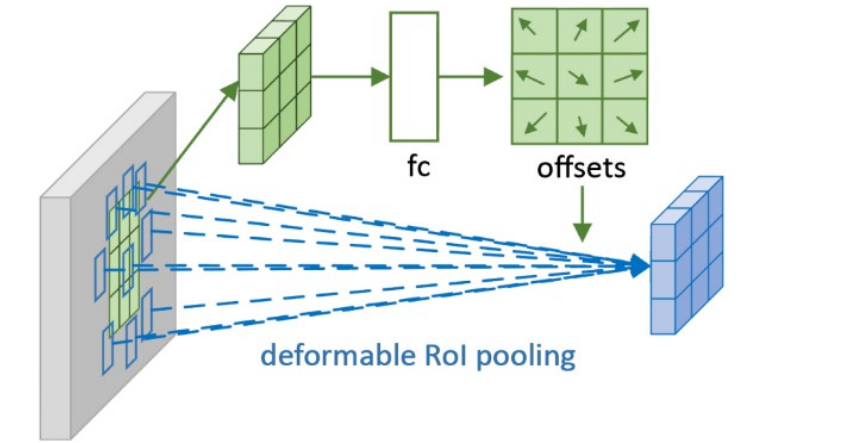
$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)$$

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

Deform_V2 convolution

$$y(p) = \sum_{k=1}^{K} w_k \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k,$$

Deform_V1 & V2



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

Deform_V2 RoI Pooling

$$y(k) = \sum_{j=1}^{n_k} x(p_{kj} + \Delta p_k) \cdot \Delta m_k / n_k,$$

Mask RCNN

ROI Align

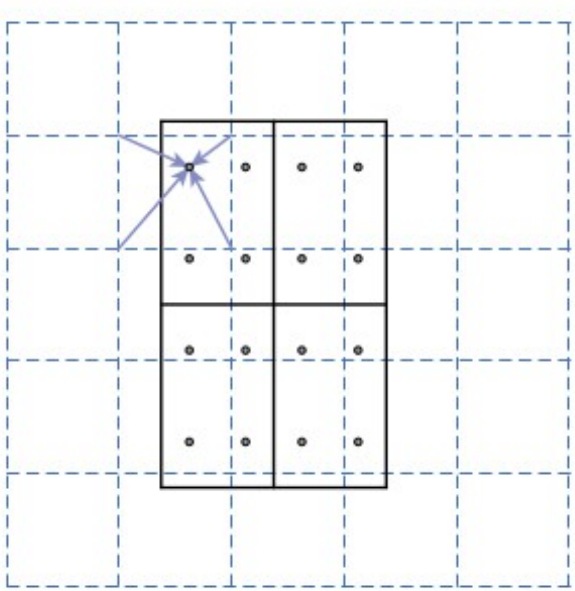


Figure 3. **RoIAlign:** The dashed grid represents a feature map, the solid lines an RoI (with 2×2 bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points.

Deform_V2

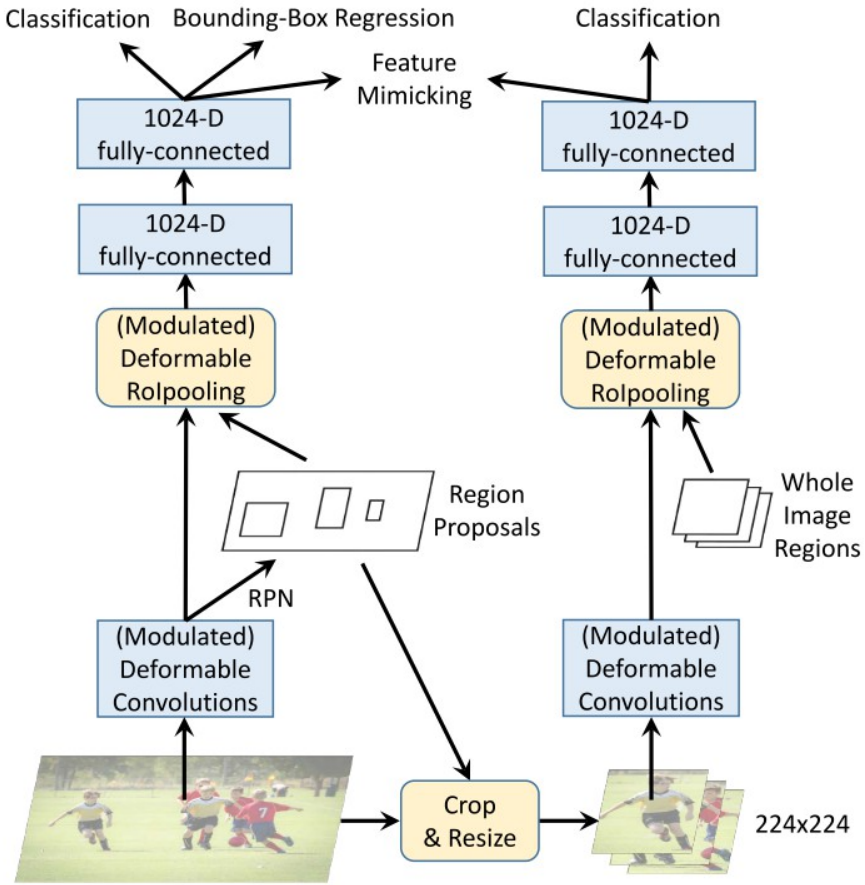


Figure 3. Network training with R-CNN feature mimicking.

$$L_{\text{mimic}} = \sum_{b \in \Omega} [1 - \cos(f_{\text{RCNN}}(b), f_{\text{FRCNN}}(b))],$$

Evaluation

(a) Convolution

(b) RoI Pooling

The regular ConvNet baseline is Faster R-CNN + ResNet-50. In each sub- figure, the effective sampling locations, effective receptive field, and error-bounded saliency regions are shown from the top to the bottom rows.

Reference I

Jifeng Dai et al. "Deformable convolutional networks". In: *CoRR, abs/1703.06211* 1.2 (2017), p. 3.

Kaiming He et al. "Mask r-cnn". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE. 2017, pp. 2980–2988.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. "Spatial transformer networks". In: *Advances in neural information processing systems*. 2015, pp. 2017–2025.

Xizhou Zhu et al. "Deformable ConvNets v2: More Deformable, Better Results". In: *arXiv preprint arXiv:1811.11168* (2018).

Thanks

Thanks for Attention!