可变形卷积V1（DCN v1）

背景

在计算机视觉领域，同一物体在不同场景，角度中未知的几何变换是检测/识别的一大挑战，通常来说我们有两种做法:

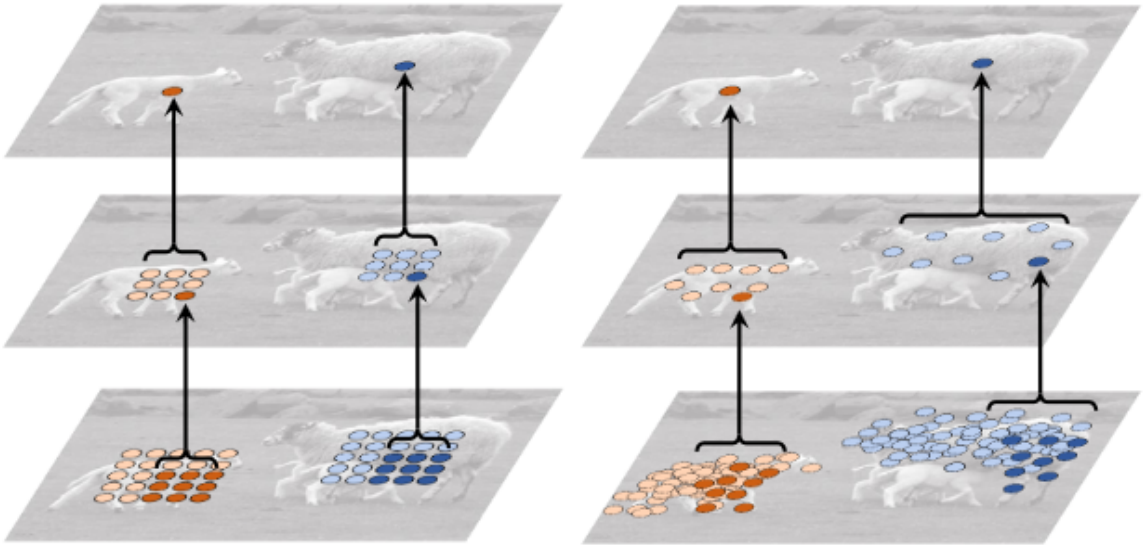
(1)通过充足的数据增强，扩充足够多的样本去增强模型适应尺度变换的能力。

(2)设置一些针对几何变换不变的特征或者算法，比如SIFT和sliding windows。

两种方法都有缺陷，第一种方法因为**样本的局限性显然模型的泛化能力比较低，无法泛化到一般场景中**，第二种方法则因为**手工设计的不变特征和算法对于过于复杂的变换是很难的而无法设计。**所以作者提出了Deformable Conv（可变形卷积）和 Deformable Pooling（可变形池化）来解决这个问题。

简介

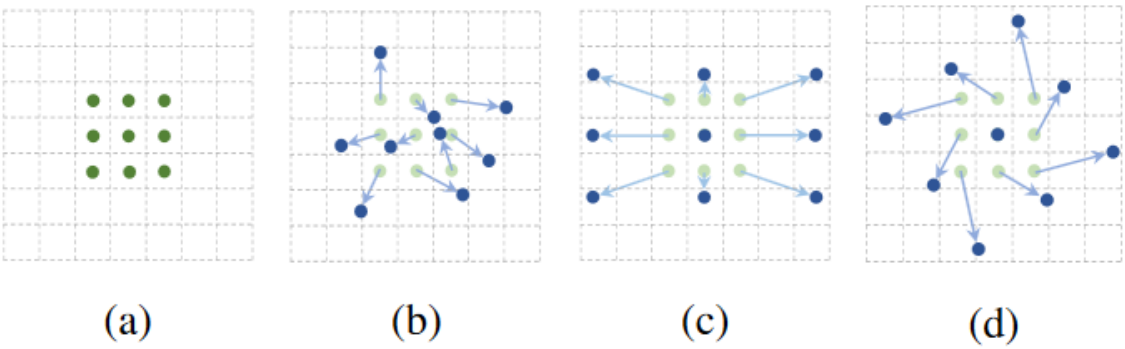
可变形卷积顾名思义就是卷积的位置是可变形的，并非在传统的N × N的网格上做卷积，这样的好处就是更准确地提取到我们想要的特征（传统的卷积仅仅只能提取到矩形框的特征），通过一张图我们可以更直观地了解:



(a) standard convolution　　(b) deformable convolution

在上面这张图里面，左边传统的卷积显然没有提取到完整绵羊的特征，而右边的可变形卷积则提取到了完整的不规则绵羊的特征。

那可变卷积实际上是怎么做的呢？**其实就是在每一个卷积采样点加上了一个偏移量**，如下图所示:



(a) 所示的正常卷积规律的采样 9 个点（绿点），(b)(c)(d) 为可变形卷积，在正常的采样坐标上加上一个位移量（蓝色箭头），其中 (c)(d) 作为 (b) 的特殊情况，展示了可变形卷积可以作为尺度变换，比例变换和旋转变换等特殊情况。

我们先看普通的卷积，以3x3卷积为例对于每个输出y(p0)，都要从x上采样9个位置，这9个位置都在中心位置x(p0)向四周扩散，(-1,-1)代表x(p0)的左上角，(1,1)代表x(p0)的右下角。

$$\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}$$

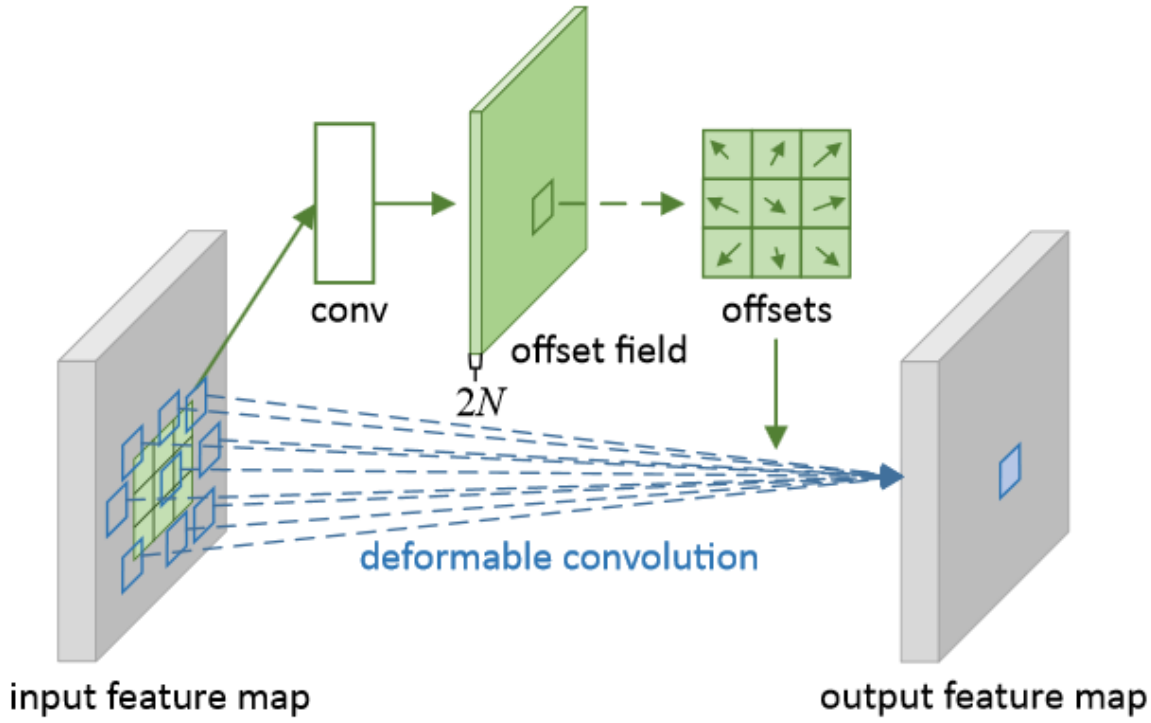
所以传统的卷积输出就是（其中$\boldsymbol{P_n}$就是网格中的n个点）：

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

正如我们上面阐述的可变形卷积，他就是在传统的卷积操作上加入了一个偏移量，正是这个偏移量才让卷积变形为不规则的卷积，这里要注意这个偏移量可以是小数，所以下面的式子的特征值需要通过**双线性插值**的方法来计算。：

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

那这个偏移量如何算呢？我们来看：



对于输入的一张feature map，假设原来的卷积操作是3×3的，那么为了学习偏移量offset，我们定义另外一个3×3的卷积层（图中上面的那层），输出的维度其实就是原来feature map大小，channel数等于2N（分别表示x,y方向的偏移）。下面的可变形卷积可以看作先基于上面那部分生成的offset做了一个插值操作，然后再执行普通的卷积。

可变形卷积V2（DCN v2）

背景

DCN v1听起来不错，但其实也有问题：我们的可变形卷积有可能**引入了无用的上下文（区域）来干扰我们的特征提取**，这显然会降低算法的表现。作者也做了一个实验进行对比说明：

(a) regular conv

(b) deformable conv@conv5 stage (DCNv1)

我们可以看到虽然DCN v1更能覆盖整个物体，但是同时也会引入一些无关的背景，这造成了干扰，所以作者提出了三个解决方法：

（1）**More Deformable Conv Layers（使用更多的可变形卷积）。**

（2）**Modulated Deformable Modules（在DCNv1基础（添加offset）上添加每个采样点的权重）**

（3）**R-CNN Feature Mimicking（模拟R-CNN的feature）。**

使用更多的可变形卷积

在DCN v1中只在conv 5中使用了三个可变形卷积，在DCN v2中把conv3到conv5都换成了可变形卷积，提高算法对几何形变的建模能力。

method	setting (shorter side 1000)	Faster R-CNN						Mask R-CNN			
		AP^{bbox}	AP^{bbox}_S	AP^{bbox}_M	AP^{bbox}_L	param	FLOP	AP^{bbox}	AP^{mask}	param	FLOP
baseline	regular (RoIpooling)	32.1	14.9	37.5	44.4	51.3M	326.7G	-	-	-	-
	regular (aligned RoIpooling)	34.7	19.3	39.5	45.3	51.3M	326.7G	36.6	32.2	39.5M	447.5G
	dconv@c5 + dpool (DCNv1)	38.0	20.7	41.8	52.2	52.7M	328.2G	40.4	35.3	40.9M	449.0G
enriched deformation	dconv@c5	37.4	20.0	40.9	51.0	51.5M	327.1G	40.2	35.1	39.8M	447.8G
	dconv@c4∼c5	40.0	21.4	43.8	55.3	51.7M	328.6G	41.8	36.8	40.0M	449.4G
	dconv@c3∼c5	40.4	21.6	44.2	56.2	51.8M	330.6G	42.2	37.0	40.1M	451.4G
	dconv@c3∼c5 + dpool	41.0	22.0	45.1	56.6	53.0M	331.8G	42.4	37.0	41.3M	452.5G
	mdconv@c3∼c5 + mdpool	41.7	22.2	45.8	58.7	65.5M	346.2G	43.1	37.3	53.8M	461.1G

在DCNv1基础（添加offset）上添加每个采样点的权重

我们知道在DCN v1中的卷积是添加了一个offset$\Delta \mathbf{p}_n$：

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

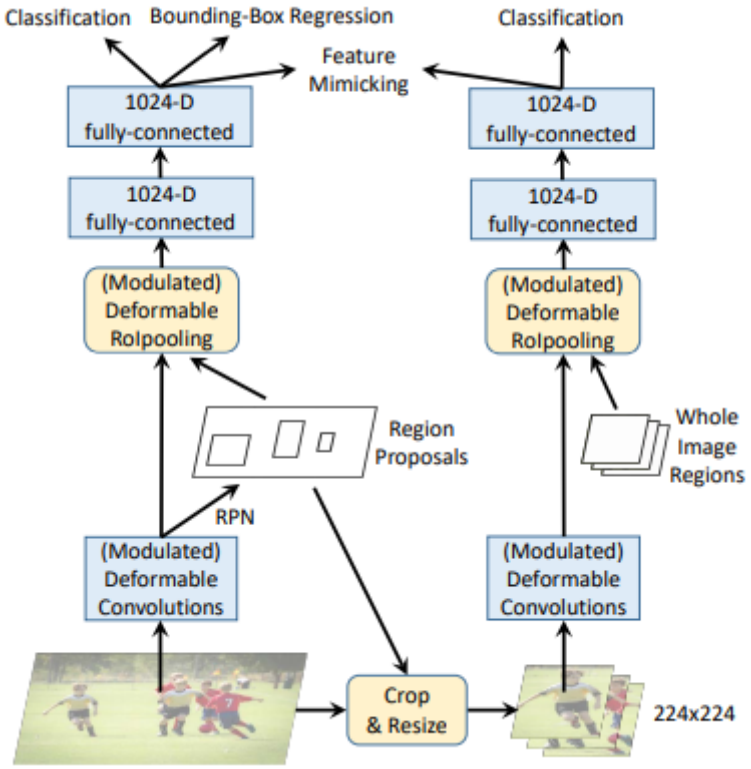
为了解决引入了一些无关区域的问题，在DCN v2中我们不只添加每一个采样点的偏移，还添加了一个权重系数Δm_k，来区分我们引入的区域是否为我们感兴趣的区域，假如这个采样点的区域我们不感兴趣，则把权重学习为0即可：

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

总的来说，DCN v1中引入的offset是要寻找有效信息的区域位置，DCN v2中引入权重系数是要给找到的这个位置赋予权重，这两方面保证了有效信息的准确提取。

R-CNN Feature Mimicking

作者发现把R-CNN和Faster RCNN的classification score结合起来可以提升表现，说明R-CNN学到的focus在物体上的feature可以解决无关上下文的问题。但是增加额外的R-CNN会使inference速度变慢很多。DCNV2里的解决方法是**把R-CNN当做teacher network，让DCN V2的ROIPooling之后的feature去模拟R-CNN的feature**，类似知识蒸馏的做法，下面会具体展开：



左边的网络为主网络（Faster RCNN），右边的网络为子网络（RCNN）。实现上大致是用主网络训练过程中得到的RoI去裁剪原图，然后将裁剪到的图resize到224×224大小作为子网络的输入，这部分最后提取的特征和主网络输出的1024维特征作为feature mimicking loss的输入，用来约束这2个特征的差异（通过一个余弦相似度计算，如下图所示），同时子网络通过一个分类损失进行监督学习，因为并不需要回归坐标，所以没有回归损失。在inference阶段仅有主网络部分，因此这个操作不会在inference阶段增加计算成本。

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

再用直白一点的话说，**因为RCNN这个子网络的输入就是RoI在原输入图像上裁剪出来的图像，因此不存在RoI以外区域信息的干扰，这就使得RCNN这个网络训练得到的分类结果更加可靠，以此通过一个损失函数监督主网络Faster RCNN的分类支路训练就能够使网络提取到更多RoI内部特征，而不是自己引入的外部特征。**

总的loss由三部分组成：mimic loss + R-CNN classification loss + Faster-RCNN loss.