# 模式识别与机器学习组会

# Deformable Convelutional Networks

汇报人：马岩松

2023年11月8日

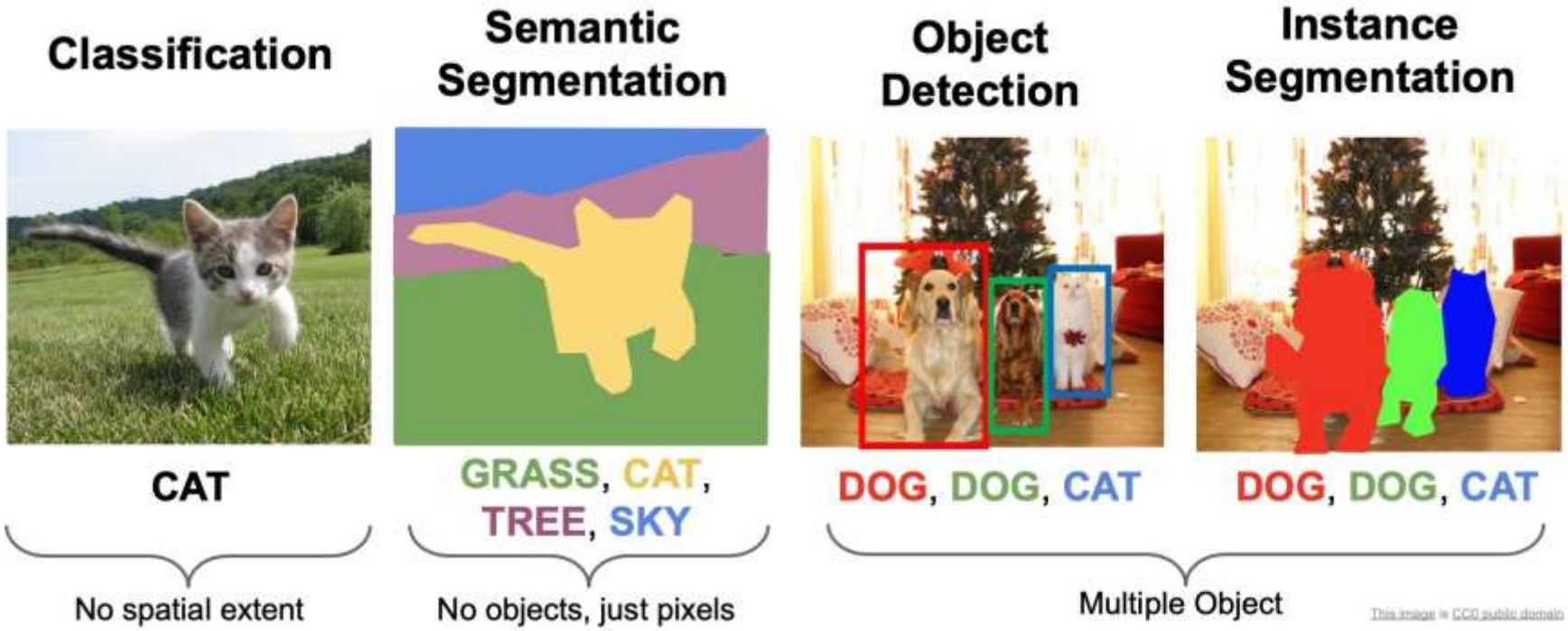
Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, Yichen Wei; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 764-773.

# 汇报提纲

- 论文研究意义
- 论文研究主题
- 该研究主题面临的不足与挑战
- 本文解决思路
- 所提出的模型/方案
- 实验设置与结果分析
- 总结与结论
- 启发与思考

# 论文研究意义

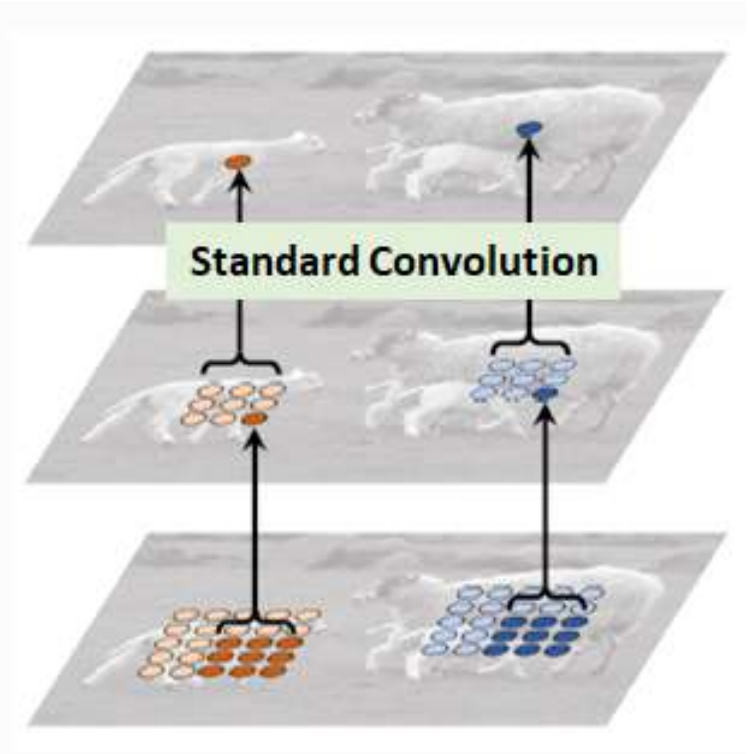
**全**连接**的神**经网络**参数量巨大**，**使得网**络训练耗时甚至难以训练，在数据集有限时会出现过拟合的情况。
**卷**积神经网络则通过局部连接、权值共享等**方法减**轻了训练过程的计算，同时具有良好的平移不变性。
**卷**积神经网络在**下面**这些领域应用广泛



# 论文**研究主**题的不足与挑战

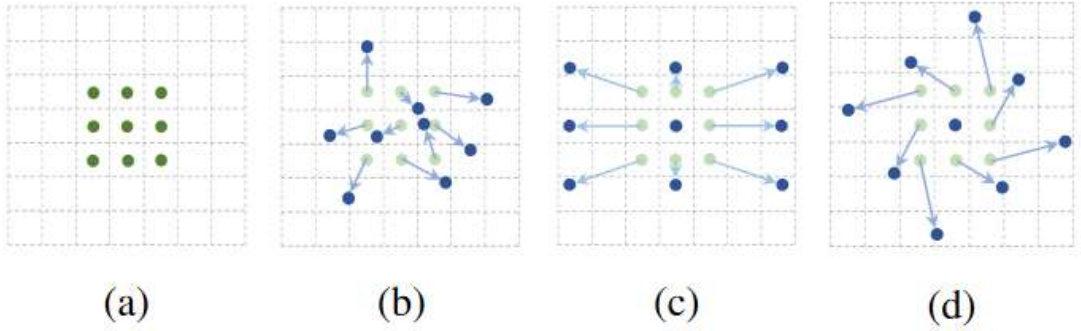
1）**卷**积神经网络的几何结构**是固定的**$n \times n$**矩形，会不可避免地收集到很多背景信息。**

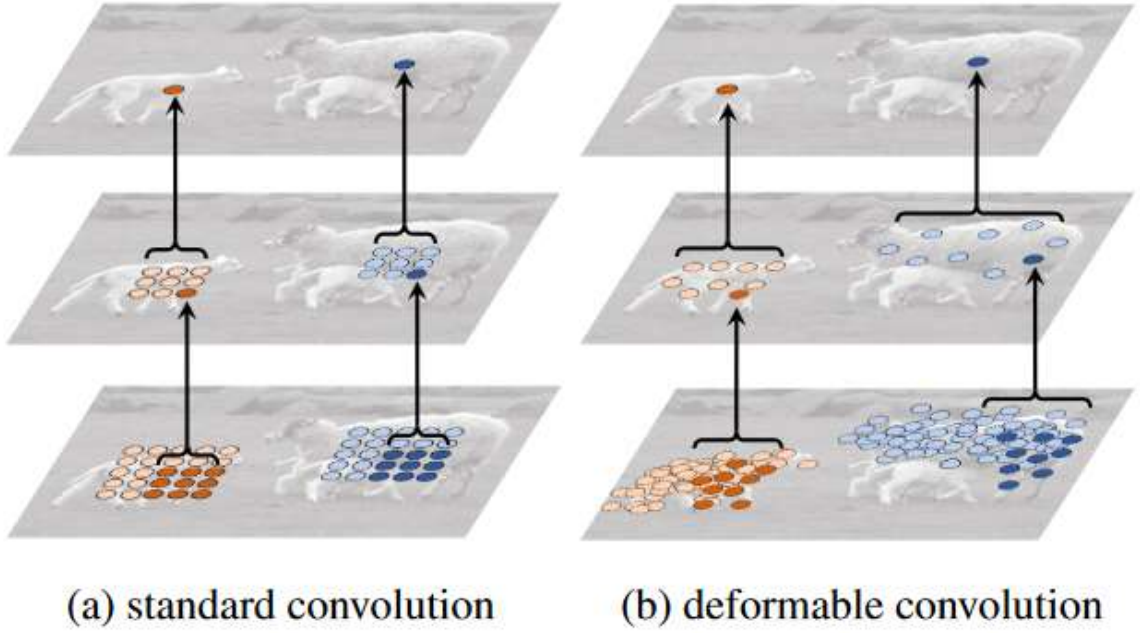
2）**在密集小目**标检测时会框到其他目标影响检测效果。



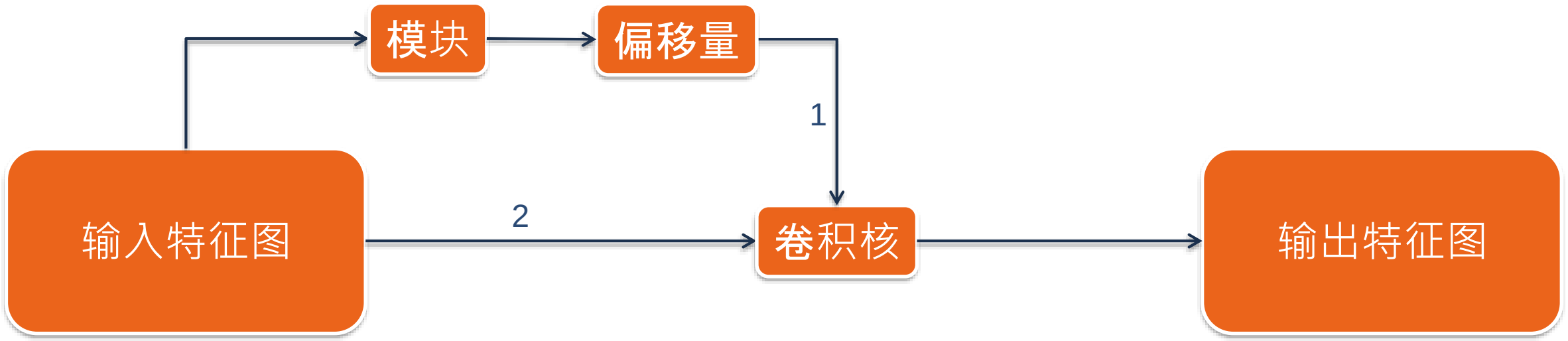
# 本文解决思路

在标准卷积操作中对常规网格采样位置添加了2D偏移量，实现了对采样网格的自由形变使得能够拟合物体的形状，这些偏移量是通过学习得到的，从而可以实现对具有不同形状的对象进行自适应的部分定位。



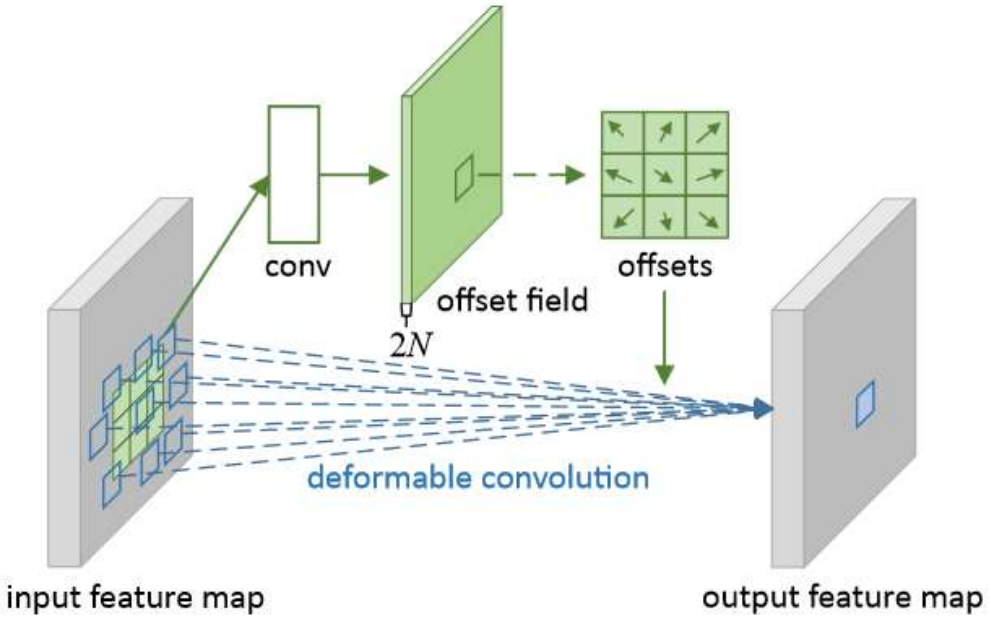



# 所提出的模型/方案-overview



# 所提出的模型/方案--关键（创新）模块

可变形卷积




$$N = |R|$$

对于输出特征图上的每个位置$P_0$有：

$$y(P_0) = \sum_{P_n \in R} w(P_n) \cdot x(P_0 + P_n + \Delta P_n)$$

其中$R = \{(-1,-1),(-1,0),\dots,(0,1),(1,1)\}$，$\Delta P_n$是学到的偏移量，由于$\Delta P_n$通常是分数，因此利用双插值线性函数来求出$x(p)$，其中q为任意位置

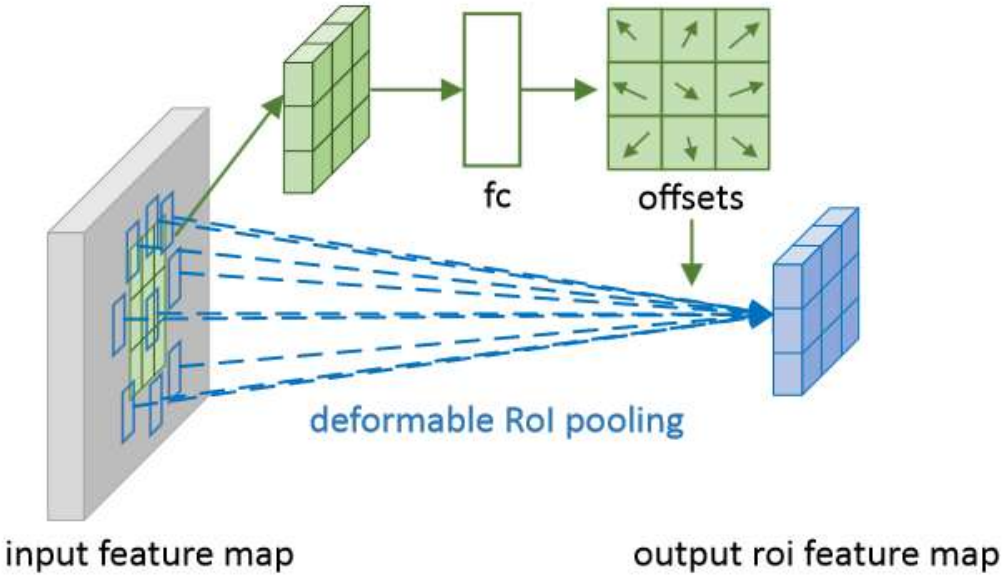
$$x(p) = \sum_q G(q,p) \cdot x(q)$$

$$G(q,p) = g(q_x,p_x) \cdot g(q_y,p_y)$$

$$g(a,b) = \max(0,1-|a-b|)$$

# 所提出的模型/方案--关键（创新）模块

可变形ROI池化层




对于从输入特征图上提出的*ROI*区域，输出的k × k特征图y上的每个位置有

$$y(i,j) = \sum_{p \in bin(i,j)} x\left(P_0 + P + \Delta P_{\mathrm{ij}}\right)/n_{ij}$$

其中$P_0$是*ROI*区域左上角的坐标，$n_{ij}$是该子区域中的像素数，由全连接层生成归一化偏移量$\Delta\hat{P}_{\mathrm{ij}}$后，然后通过逐元素曾轶ROI的宽和高转化为偏移量$\Delta P_{\mathrm{ij}}$，即

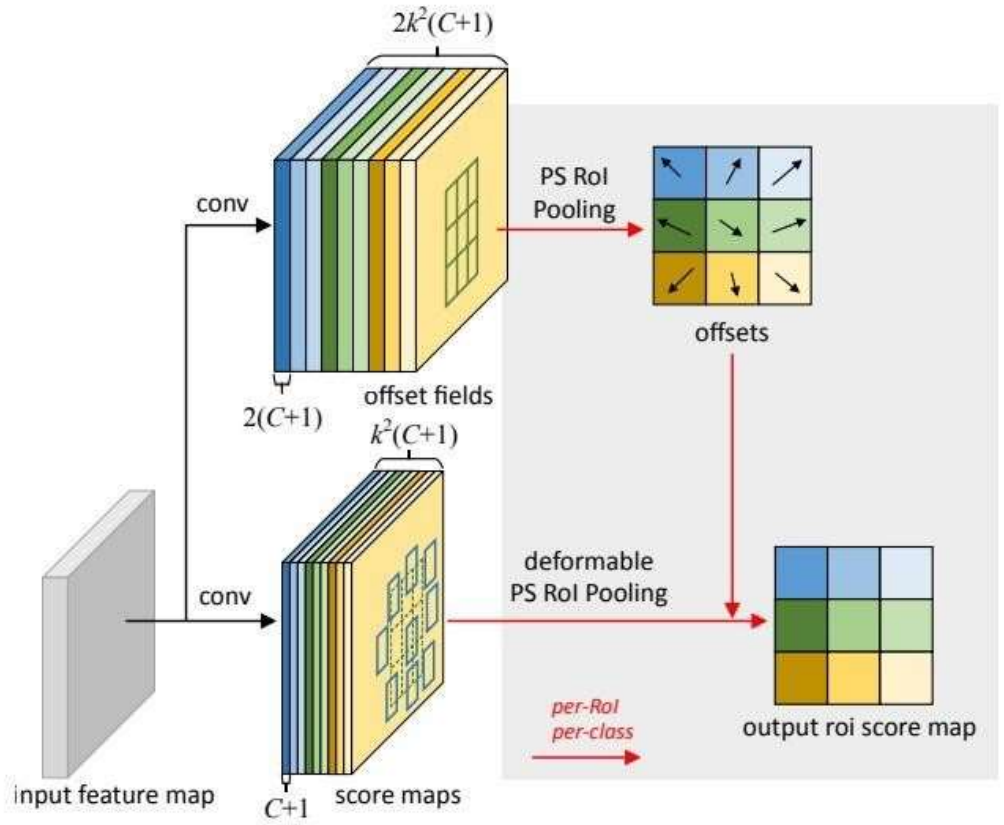
$$\Delta P_{\mathrm{ij}} = \gamma \cdot \Delta\hat{P}_{\mathrm{ij}} \circ (w,h)$$

其中$\gamma$是预定标量用来调节偏移量大小，一般设为0.1

对偏移量的归一化是为了使得偏移量学习不受ROI尺寸大小的影响

# 所提出的模型/方案--关键（创新）模块

## 位置敏感ROI池化层




输出的得分图也是C+1层的

**首先将所有**输入特征图转换为每个对象类别的$k^2$组**得分**图（对于C**个**对象类别来说，一共有C+1（1**代表的是背景**）**个得分**图）
**在**顶部分支中，**用一个卷**积层生偏移**字段**。对**于每个**ROI(**同**样对于每个类别)，**在**这些字段上应用PS ROI**池化**，**得到**归一化偏移量$\Delta\hat{P}_{ij}$，**然后将其**转换为真实偏移量$\Delta P_{ij}$，**方法与可**变性ROI**池化相同**。

# 实验设置与结果分析--Experimental Datasets

**数据集**：

1）PASCAL VOC **数据集包含20个**语义类别，训练集包括10,582张图像，评估则在验证集的1,449张图像上进行。

2）CityScapes **数据集包括19个**语义类别和一个背景类别，训练和评估分别在训练集中的2,975张图像和验证集中的500张图像上进行。

3）COCO **数据集包括33万**张图像，包含**150万个目**标，**80个目**标类别，训练和评估分别在trainval**的**120,000张图像和test-dev**的**20,000张图像上进行。

对于PASCAL VOC 、 CityScapes **和**COCO， 图像的大小都被调整，使得图像的短边分别为360、1024**和**600**像素**

# 实验设置与结果分析--Evaluation Metrics

实验使用的评价指标：**平均精度**(mAP)**和交并比均**值(mIoU)

**定**义了一个称为"**有效膨**胀"**的指**标，用于衡量可变形卷积滤波器的大小。它是滤波器中所有相邻采样位置之间距离的均值。这是对滤波器的感受野大小的粗略度量。

# 实验设置与结果分析--Experimental setting

实验细节：

1）采用SGD进行优化训练，在PASCAL VOC和Cityscapes中分别进行了总共30,000和45,000次迭代在COCO中对RPN进行了与Faster R-CNN和R-FCN的联合训练，并启用了特征共享以进行训练，分别进行了总共30,000和240,000次迭代的训练。

2）网络：默认的ResNet-101模型在最后的三个3×3卷积层中使用了扩张率为2的空洞卷积。我们进一步尝试了扩张率值为4、6和8，实验中还使用了有类别感知的RPN、Faster R-CNN和R-FCN的可变形版本（将Faster R-CNN和R-FCN中的ResNet-101替换为Aligned-Inception-ResNet）

3）实现细节：学习率在前2/3的迭代中为0.001，在最后1/3的迭代中为0.0001

# 实验设置与结果分析--Competing Methods

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

**加入可形**变卷积对目标检测任务模型的**影响**

**其中M表示多尺度**测试，**B表示迭代**边界框平均值，**可**变形**卷**积**性能的提升与**这些附加功能**是互补的**

# 实验设置与结果分析--Experimental results

| method | #params (million) | net. forward (sec) | runtime (sec) |
|---|---|---|---|
| DeepLab@C | 46.0 → 46.1 | 0.610 → 0.656 | 0.650 → 0.696 |
| DeepLab@V | 46.0 → 46.1 | 0.084 → 0.088 | 0.094 → 0.098 |
| class-aware RPN | 46.0 → 46.1 | 0.142 → 0.152 | 0.323 → 0.334 |
| Faster R-CNN | 58.3 → 59.9 | 0.147 → 0.192 | 0.190 → 0.234 |
| R-FCN | 47.1 → 49.5 | 0.143 → 0.169 | 0.170 → 0.193 |

**可形**变卷积对模型复杂度、前向传播和检测运行时间的**影响**

| layer | small | medium | large | background |
|---|---|---|---|---|
| | mean ± std | | | |
| res5c | 5.3 ± 3.3 | 5.8 ± 3.5 | 8.4 ± 4.5 | 6.2 ± 3.0 |
| res5b | 2.5 ± 1.3 | 3.1 ± 1.5 | 5.1 ± 2.5 | 3.2 ± 1.2 |
| res5a | 2.2 ± 1.2 | 2.9 ± 1.3 | 4.2 ± 1.6 | 3.1 ± 1.1 |

R-FCN网络可形变卷积的有效膨胀值

| deformation 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 |
|---|---|---|---|---|
| atrous convolution (2,2,2) (default) | 69.7 / 70.4 | 68.0 / 44.9 | 78.1 / 62.1 | 80.0 / 61.8 |
| atrous convolution (4,4,4) | 73.1 / 71.9 | 72.8 / 53.1 | 78.6 / 63.1 | 80.5 / 63.0 |
| atrous convolution (6,6,6) | 73.6 / 72.7 | 73.6 / 55.2 | 78.5 / 62.3 | 80.2 / 63.5 |
| atrous 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 |
| deformable RoI pooling | N.A | N.A | 78.3 / 66.6 | 81.2 / 65.0 |
| deformable convolution & RoI pooling | N.A | N.A | **79.3 / 66.9** | **82.6 / 68.5** |

**可形**变卷积与空洞卷积的准确率对比

# 实验设置与结果分析--Parameters Analysis

| usage of deformable convolution (# layers) | DeepLab | | class-aware RPN | | Faster R-CNN | | 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** |

**不同网**络的不同层加入可形变卷积对准确率的影响

# 总结与结论

**本文的主要**贡献：

1）**与先前的方法相比，本文通**过设定偏移量并让机器自主学习的方式增强了卷积神经网络的对物体几何变换的建模能力。

2）**新模**块可以很容易的替换原有的普通对应模块，并且可以通过反向传播实现端到端的训练

**本文的不足之**处：

1）**可**变形卷积对小物体的自适应能力较差

2）**偏移的学**习一定程度上还是会受到背景的影响

# 启发与思考

**工作**优点：

**将偏移的学**习交给神经网络而不是设计一种复杂的数学方法，极大程度的减少了运算的复杂度，提高了准确率

**缺点**：

**没有**针对超参数$\gamma$设置实验寻找最佳值，前文中提到的SIFT**方法也没有**进行试验进行比较

**启**发：

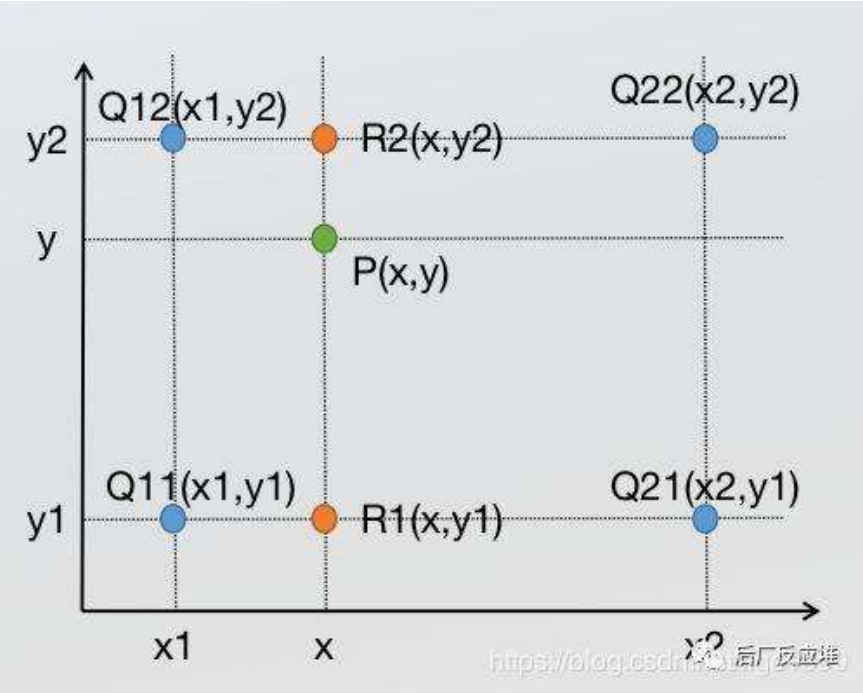
1）**可以**转换思路，将复杂问题简单化，高复杂性的方法未必比简单的方法更高效

2）**在**设计实验时，要尽可能的考虑到该方法带来的所有影响而不是只把眼光放在准确率等指标上

**借**鉴之处：可以尝试将难以进行数学设计的部分交给机器自己学习

# 附录

## 双线性插值




$$f(R_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(P) = \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$

$$f(x,y) = \frac{f(Q_{11})}{(y_2 - y_1)(x_2 - x_1)}(x_2 - x)(y_2 - y) + \frac{f(Q_{21})}{(y_2 - y_1)(x_2 - x_1)}(x - x_1)(y_2 - y) + \frac{f(Q_{12})}{(y_2 - y_1)(x_2 - x_1)}(x_2 - x)(y - y_1) + \frac{f(Q_{22})}{(y_2 - y_1)(x_2 - x_1)}(x - x_1)(y - y_1)$$

# Thank you for listening!