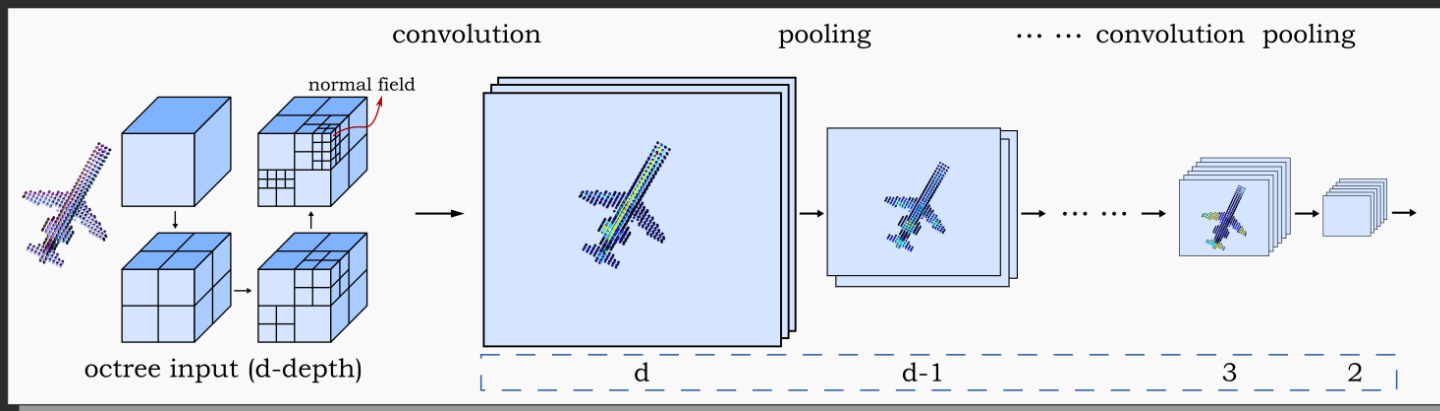


基于八叉树的三维卷积神经网络

刘洋

网络图形组

微软亚洲研究院



O-CNN

- 王鹏帅、刘洋、郭雨潇、孙春宇、童欣， O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis, . ACM Trans. Graph. (SIGGRAPH) 36(4), Article 72 (July 2017), 11 pages.
- 王鹏帅、孙春宇：清华大学高等研究院和微软亚洲研究院联合培养博士生
- 郭雨潇：MSRA 实习生、电子科技大学博士生
- 刘洋、童欣：MSRA 研究员

研究动机：三维世界感知与理解

- 三维理解的契机与推动力
 - 刚需：
 - 人对世界的认识需求
 - 三维世界的复杂度
 - 传统方法的局限性
 - 契机：
 - 数据获取的便捷和日益丰富
 - 机器学习的发展
 - 认知任务的丰富

研究内容：三维形状分析

- 对象：三维形体（surface, point cloud）
- 任务：识别(分类)、搜索、分割、补全、对应、变形、生成等。
- 传统方案：
 - 依靠规则、固定模式、具体特征
 - 在大数据下表现一般。
- 基于数据驱动的方案：
 - 提取内在规律、更好的非手工特征、处理复杂情况
 - 卷积网络等深度学习工具

已有的“三维深度学习”工作

- 表达形式（观察方式有关、考虑的不变量有关、表达的物体多样性有关）
 - 流形 [Boscaini et al 2015, Masci et al. 2015; Sinha et al. 2016...]
 - 点云 [Su et al. 2017, ...]
 - 体素 [Wu et al. 2015, Maturana and Sherer 2015....]
 - 多视图图像 [Bai et al. 2015, Qi et al. 2016; Su et al. 2015...]
 - 图 [Bronstein et al. 2017, ...]
 - 层次结构 [octree, kd-tree....]

已有的“三维深度学习”工作

- 表达形式（观察方式有关、考虑的不变量有关、表达的物体多样性有关）
 - 流形 | 点云 | 体素 | 多视图图像 | 图 | 层次结构
- 信号：示性函数、位置、法向、高阶几何量...
- 计算能力
 - 分辨率 | 计算效率 | 内存开销
- 网络设计

研究目标

- 对输入鲁棒（如不对形状拓扑、曲面光滑程度、物体可见范围等做过多假设）的三维CNN网络，用来处理纷杂的三维数据。
- 这里面最接近我们期望的就是**基于体素的3D CNN**。
 - 表达：物体内部为1，外部为0的示性函数。Pixel (2D) -> voxel (3D)
- 体素 3D CNN:
 - 缺点：存储开销大、计算开销大：立方级别的存储量； 32^3
 - 优点：简单，是图像CNN的直接推广。

观察

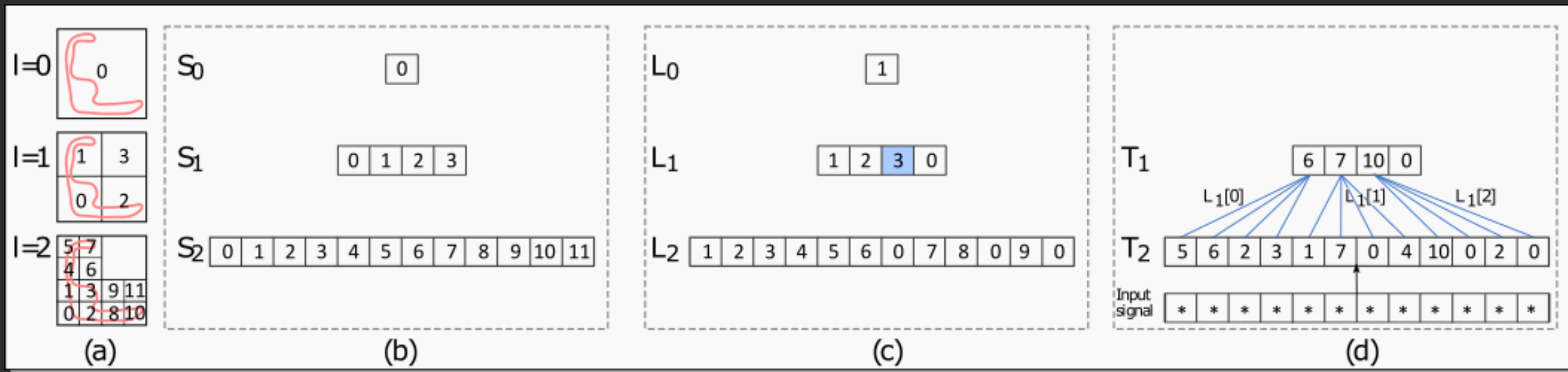
- 体素表达下的特性：
 - 稀疏性 -- 形状 (shape) 在空间中的分布稀疏性。
 - 信息局部有效性： 远离形状表面的区域所含信息对整体贡献不大。
- 已有的稀疏表达下的基于体素的CNN
 - Sparse 3DCNN [Graham 2015]: 随着卷积层的叠加，稀疏性被破坏。
 - OctNet [CVPR2017]: 内存开销在低分辨率很大。
 - ...

核心方案

- 八叉树的稀疏表达与面向CNN的高效访问
 - 稀疏性：八叉树表达自有的属性
 - 原始信号存储：最深的叶节点存储信号：如节点中形状的平均法向
 - 对输入鲁棒性：点云、网格、样条曲面等。
 - 面向CNN和GPU高效访问与操作的八叉树结构

核心方案

支持CNN高效访问的八叉树结构：以二维四叉树为例讲解。



图一：O-CNN数据结构。图(a)中红色部分表示一个二维形状， $l=0,1,2$ 表示了不同深度下四叉树，每个节点的标号用排序过的散乱键标定并记在数组 S_0, S_1, S_2 里(图(b))，每一层非空节点的序号(从1开始计数，0表示空节点)由 L_0, L_1, L_2 数组存储(图(c))。在四叉树每一层都存储CNN网络的特征数组，和 S_i 相对应，比如 T_1, T_2 。当下采样发生的时候，如在 T_2 上，四个同父的子节点通过 L_1 可以迅速把采样结果放在 T_1 中父节点所对应的位置。

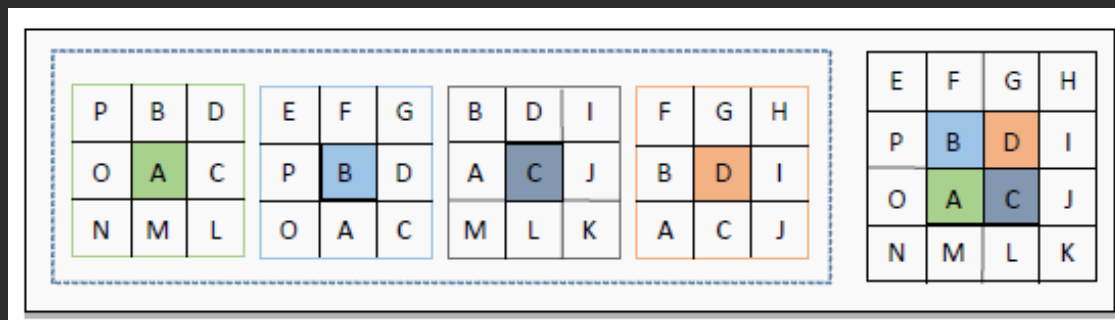
核心方案

- 支持CNN高效访问的八叉树结构

- 快速邻域访问

- 直接方案：局部K邻域预存储。K比较大时，耗费空间。
- 我们方案：构造哈希表以构建散乱键到八叉树节点的映射。

- 减少邻域访问次数

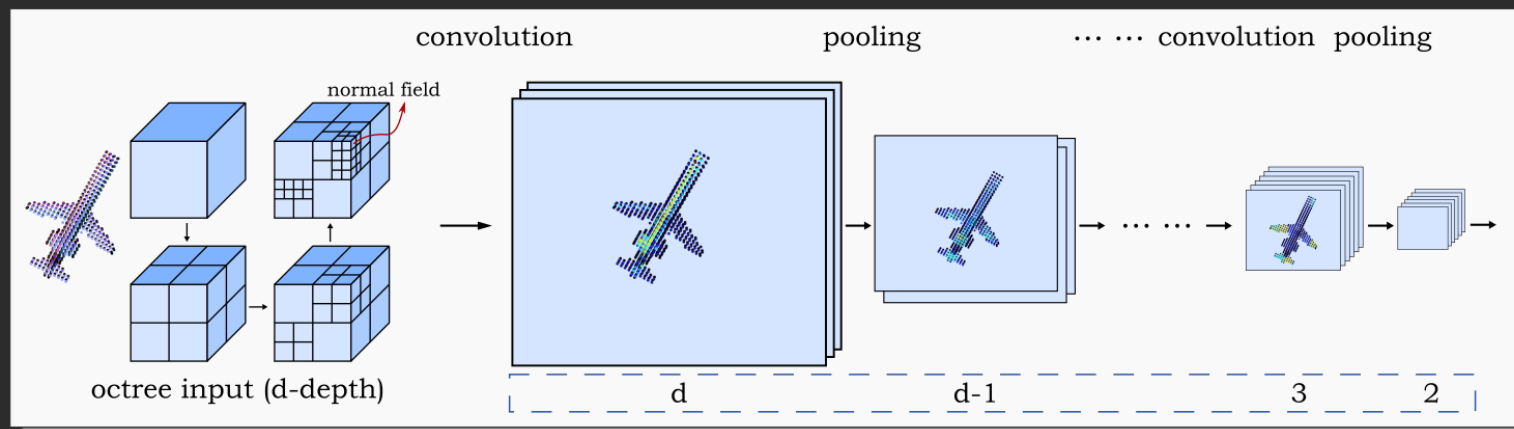


核心方案

- 限制在八叉树节点的CNN计算
 - **存储和计算优势**：计算量只和八叉树节点数目相关。
 - **计算高效**：八叉树的层次结构和便捷的数据组织使得卷积、反卷积、池化、反池化操作与八叉树高效结合。
 - **信息传播优势**：隐式地将网络信息在形状表面附近上传播，更高效地交换和共享信息。

基于八叉树的卷积神经网络 (O-CNN)

- 网络设计：
 - Conv + BN + ReLU + Pooling 作为一个单元 U_l (在八叉树第 l 层)
 - O-CNN的前端构造如下： $input \rightarrow U_d \rightarrow U_{d-1} \rightarrow \dots \rightarrow U_2$ 。
 - U_2 强制八叉树第2层空节点补零，对齐feature长度。
 - Unpooling, Deconv 实现。
 - 简单架构，支持其他架构。



实验：分类问题

- 数据集：Princeton ModelNet40 (12311个模型, 40类)
- 网络：O-CNN 前端 + 两个 (dropout + FC)+ softmax
- O-CNN: 深度d从3到8, 即分辨率从 8^3 到 256^3 , O-CNN(d)

实验：分类问题

- 数据集：Princeton ModelNet40（12311个模型，40类）
- 网络：O-CNN 前端 + 两个（dropout + FC）+ softmax
- O-CNN: 深度d从3到8，即分辨率从 8^3 到 256^3 ，O-CNN(d)

实验：分类问题

网络	识别准确度
VoxNet(分辨率为 32^3) [IROS2015]	82.0%
Geometry image [ECCV2016]	83.9%
SubVolSup(32^3) [CVPR2016]	87.2%
FPNN(64^3) [NIPS2016]	87.5%
FPNN+normal(64^3) [NIPS2016]	88.4%
PointNet [CVPR2017]	89.2%
VRN(32^3) [NIPS workshop 2016]	89.0%
O-CNN(3)	85.5%
O-CNN(4)	88.3%
O-CNN(5)	89.6%
O-CNN(6)	89.9%
O-CNN(7)	89.5%
O-CNN(8)	89.6%

[]括号内注明的是该论文所发表在的会议名称.

实验：O-CNN 效率

- 实验：分类问题。 配置：Geforce 1080 GPU (8GB) , minibatch : 32

方法	16^3	32^3	64^3	128^3	256^3
O-CNN	0.32GB	0.58GB	1.1GB	2.7GB	6.4GB
体素CNN (二值信号)	0.23GB	0.71GB	3.7GB	超过显存大小	超过显存大小
体素CNN (法向信号)	0.27GB	1.20GB	4.3GB	超过显存大小	超过显存大小

显存开销

实验：O-CNN 效率

- 实验：分类问题。 配置：Geforce 1080 GPU (8GB) , minibatch : 32

方法	16^3	32^3	64^3	128^3	256^3
O-CNN	17毫秒	33毫秒	90毫秒	327毫秒	1265毫秒
体素CNN (二值信号)	59毫秒	425毫秒	1648毫秒	-	-
体素CNN (法向信号)	75毫秒	510毫秒	4654毫秒	-	-

平均一次前向和反向消耗时间

实验：O-CNN 信号选取

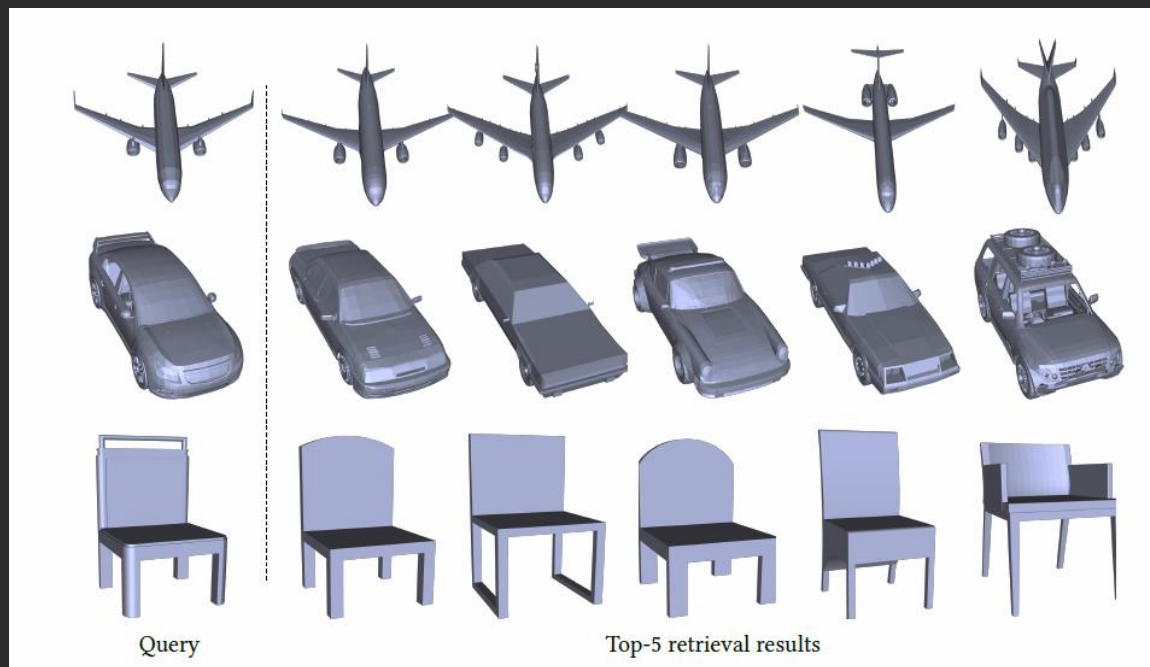
- 法向与 (0, 1) 信号之比较。

数据结构	信号	分类准确度
体素	二值 (内外)	87.9%
体素	法向	88.7%
八叉树	二值 (含数据为1)	87.3%
八叉树	法向	89.6%

分辨率 32^3 ，同样网络构建

实验：检索问题（shape retrieval）

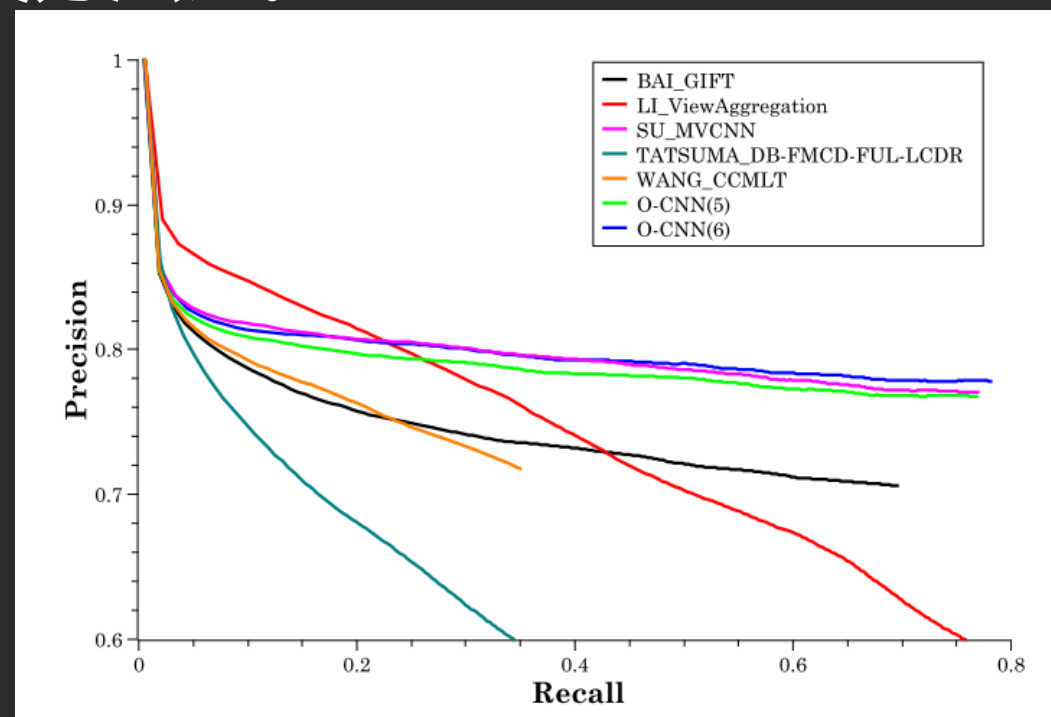
- 数据集：ShapeNet55 Core（51190个模型，55大类，204小类）
- 网络：与分类用的O-CNN一样，用最后输出feature检索。
- 比较：与Shrec2016检索比赛的参赛方法比。



实验：检索问题 (shape retrieval)

- 数据集：ShapeNet55 Core (51190个模型，55 大类，204 小类)
- 网络：与分类用的O-CNN 一样，用最后输出feature检索。
- 比较：如Shrec2016 检索比赛的参赛方法比。

Method	P@N	R@N	F1@N	mAP	NDCG@N
Tatsuma_DB-FMCD-FUL-LCDR	0.427	0.689	0.472	0.728	0.875
Wang_CCMLT	0.718	0.350	0.391	0.823	0.886
Li_ViewAggregation	0.508	0.868	0.582	0.829	0.904
Bai_GIFT	0.706	0.695	0.689	0.825	0.896
Su_MVCNN	0.770	0.770	0.764	0.873	0.899
O-CNN(5)	0.768	0.769	0.763	0.871	0.904
O-CNN(6)	0.778	0.782	0.776	0.875	0.905



实验：分割问题

- 数据集：[Yi et al. 2016] (16,881个模型 , 16类, 2~6 parts)
- 网络：FCN (O-CNN) + CRF



实验：分割问题

- 数据集：[Yi et al. 2016] (16,881个模型, 16类, 2~6 parts)
- 网络：DeConvNet (O-CNN) + CRF

unPooling + DeConv + BN + ReLU作为一个单元 DU_l

DeConvNet : $input \rightarrow U_d \rightarrow U_{d-1} \rightarrow \dots \rightarrow U_2 \rightarrow DU_2 \rightarrow DU_3 \rightarrow \dots \rightarrow DU_d \rightarrow output$

实验：分割问题

- 数据集：[Yi et al. 2016] (16,881个模型, 16类, 2~6 parts)
- 网络：FCN (O-CNN) + CRF

	mean	plane	bag	cap	car	chair	e.ph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
[Yi et al. 2016]	81.4	81.0	78.4	77.7	75.7	87.6	61.9	92.0	85.4	82.5	95.7	70.6	91.9	85.9	53.1	69.8	75.3
PointNet [Qi et al. 2017]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
SpecCNN [Yi et al. 2017]	84.7	81.6	81.7	81.9	75.2	90.2	74.9	93.0	86.1	84.7	95.6	66.7	92.7	81.6	60.6	82.9	82.1
O-CNN(5)	85.2	84.2	86.9	84.6	74.1	90.8	81.4	91.3	87.0	82.5	94.9	59.0	94.9	79.7	55.2	69.4	84.2
O-CNN(6)	85.9	85.5	87.1	84.7	77.0	91.1	85.1	91.9	87.4	83.3	95.4	56.9	96.2	81.6	53.5	74.1	84.4

IoU 指标

小结

- 技术关键点：
 - 八叉树：稀疏存储、高效访问计算
 - 限制在八叉树节点的CNN计算
 - 法向信号

未来工作

- 更多需要高分辨率分析的任务：补全、生成、对应等。
- 场景理解、生成等
- 形体结构和功能的分析与生成

数据、代码

- <https://github.com/Microsoft/O-CNN>





Q & A