

基于 K 近邻卷积神经网络的点云模型识别与分类

于挺, 杨军*

兰州交通大学电子与信息工程学院, 甘肃 兰州 730070

摘要 为了进一步提高大规模多种类点云模型识别与分类的准确率, 提出了一种 K 近邻卷积神经网络模型。首先, 利用最远点采样算法对点云模型均匀采样; 其次, 对采样后的点云模型用 K 近邻算法构建每个点的局部邻域, 为防止信息的非局部扩散, 对卷积层提取的特征也逐个建立局部邻域; 然后, 通过最大池化聚合所有局部特征得到点云模型的全局特征表示; 最后, 用全连接层与 Softmax 函数计算各类别对应的概率并分类。实验结果表明, 本算法在公开数据集 ModelNet40 上的识别准确率为 92%。与已有的点云模型识别与分类算法相比, 能更有效地融合局部结构特征, 提高点云模型识别与分类的准确率。

关键词 机器视觉; 模型识别; K 近邻; 局部特征; 缩放指数线性单元函数

中图分类号 TP 391.4

文献标志码 A

doi: 10.3788/LOP57.101510

Point Cloud Model Recognition and Classification Based on K-Nearest Neighbor Convolutional Neural Network

Yu Ting, Yang Jun*

School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, China

Abstract In order to further improve the recognition and classification accuracy of large-scale multi-category point cloud model, a K-nearest neighbor convolutional neural network is proposed. First, the point cloud model is uniformly sampled with the farthest point sampling algorithm. Second, the K-nearest neighbor algorithm is used to construct the local neighborhood of each point for the sampled point cloud model. In order to prevent the non-local diffusion of information, a local neighborhood is constructed for each feature extracted from the convolution layer. Then, all local features are aggregated to obtain the global feature representation of the point cloud model through the max pooling. Finally, the probabilities corresponding to each category are calculated and classified using the fully connected layer and Softmax function. Experimental results show that the recognition accuracy of this algorithm on the ModelNet40 dataset is 92%. Compared with the current point cloud model recognition and classification algorithms, the proposed algorithm can more effectively fuse local structure features and improve the accuracy of point cloud model recognition and classification.

Key words machine vision; model recognition; K-nearest neighbor; local feature; scaled exponential linear unit function

OCIS codes 150.0155; 150.1135; 100.4996

1 引言

三维模型识别是计算机图形学和视觉领域的重要研究内容之一。目前关于二维图像识别与分类的研究成果较多^[1-3], 二维图像的生成可理解为图像从三维空间到二维空间的映射过程, 该过程容易丢失

大量的结构信息, 导致对物体二维图像的认识无法满足人类需求。随着工业自动化程度的提高, 可大量生产成本低、可靠性强的三维传感器, 如 RGB-D 相机, LiDAR 和 Google Tango, 有效构建三维模型的通用数据表示, 促进了计算机视觉领域的进一步发展, 也为智能机器人在真实环境中工作提供保障。

收稿日期: 2019-09-09; 修回日期: 2019-09-12; 录用日期: 2019-09-17

基金项目: 国家自然科学基金(61862039)

* E-mail: yangj@mail.lzjtu.cn

卷积神经网络(CNN)作为深度学习的一种,在图像处理领域具有明显的优势。标准的CNN模型通常以规则的数据作为输入,如将二维图像变成由像素排列的阵列;体素化模型是将三维多边形网格模型离散化为规则二值化的占用网格形式。点云模型是通过LiDAR直接获得的原始数据,数据点排列具有不规则性。因此,如何利用CNN处理点云模型,是计算机视觉领域的重要研究方向。

受PointNet^[4]网络的启发,本文提出了一种K近邻卷积神经网络(KNN-CNN)模型,能融合点云模型的局部结构信息,增强识别细粒度模型的能力。将点云模型利用K近邻(KNN)算法逐点构建局部邻域,得到每个局部邻域内的 k 个点。通过卷积层的特征提取和最大池化层的特征聚合,得到鲁棒的模型结构特征。在特征提取过程中,利用KNN算法逐特征构建局部邻域,有效防止了信息的非局部扩散,提高了点云模型识别与分类的准确率。在CNN的优化过程中,利用SELU函数和批量标准化(BN),使CNN达到最优识别与分类效果。

2 相关研究工作

物体或场景的识别与分类都离不开特征的提取,点云模型的识别方法多是提取对象的形状、结构特征或者多种特征的组合形式形成特征描述符,再通过对比特征描述符的相似性,完成模型的识别与分类。可分为基于手工提取特征和基于深度学习的模型识别与分类方法。

传统的基于手工提取特征模型的识别与分类方法,如利用二维数据表征三维特征的旋转图像(SI)方法^[5]、三维形状上下文(3DSC)方法^[6]、基于几何特征的点特征直方图(PFH)描述符^[7]以及通过测量密集采样表面点获取的局部形状直径函数描述符^[8]、球谐函数描述符(SPH)^[9]、全局点签名(GPS)^[10]、热核签名(HKS)^[11]以及波核签名(WKS)^[12]等描述符。手工提取的特征描述符虽然在三维模型识别与分类中表现良好,但设计描述符需要先验知识,且只能解决特定问题,普适性较差。因此研究人员结合深度学习提取特征研究了三维模型的识别与分类。

基于深度学习的模型识别与分类方法按照输入数据的不同可分为基于投影图像的方法、基于体素的方法和基于点云的方法。Su等^[13]提出了一种基于多视角CNN的方法,首先获得三维形状在12个不同视点下的投影图,然后采用VGG-M学习各视

点下投影图的特征,最后将所有特征进行池化操作并送入一个CNN模型中得到最终的形状特征。Shi等^[14]沿主轴方向进行圆柱投影将三维形状转化为多个全景图,再利用CNN从全景图中学习特征表示。基于投影图像的方法优势在于利用了图像分类网络,识别准确率高,但并没有利用CNN直接处理三维模型。基于体素的方法是将三维多边形网格模型通过占用网格规范化计算转化为二值三维体素矩阵,为CNN的处理提供了便利。Wu等^[15]将三维形状表示为二值概率分布,并采用卷积深度置信网络学习三维体素与标签之间的联合分布,实现了三维体素模型的识别与分类。Xu等^[16]用定向搜索优化3D ShapeNets网络减少了CNN的参数数量,以提高分类的准确率。文献^[17]对训练集中的体素模型进行旋转,获取到该模型的旋转副本,增加了训练集的数据量,通过扩展训练集降低了网络的过拟合率,同时在训练过程中增强了网络对不同角度三维模型的识别能力。文献^[18]在特征提取过程中,使用堆叠的深度小卷积核CNN,有效提取了模型内部的隐含信息,增强了特征的表达能力和差异性,提高了大规模复杂三维模型识别的准确率。基于体素数据的方法使用三维体素保留三维模型的形状信息,有利于提高特征的鉴别力。但由于输入数据和卷积核维数较高,使计算量和内存都急剧膨胀。虽然使用八叉树^[19]的方法优化算法可以提高计算速度,但处理高分辨率体素模型仍较为困难。

文献^[4]提出结合深度学习处理点云模型的PointNet网络架构。首先从每个点的多层感知机中计算各点的特征,然后应用对称函数将所有特征聚合为点云的一个全局表示,最后通过全连接层得到分类结果。文献^[20]提出了一种类似于文献^[4]的排列等变层,区别在于排列等变层对于非线性输入是最大归一化的。二者都解决了点云的无序性问题,但并未充分利用局部结构信息。Qi等^[21]提出了PointNet的改进方案PointNet++,采用分层应用PointNet的思想,提取局部区域的几何特征。首先将点云分割为较小的点集,并利用PointNet提取这些小点集的特征,然后对更高维的特征迭代重复此过程,从而获得局部结构特征,但会导致体系结构变复杂,降低网络的训练速度。文献^[22]在CNN前端的核相关层计算每个数据点的 k 个最近邻点与 M 个核心点之间的亲和力,获取初始点特征,然后利用最大池化处理获取鲁棒的局部结构特征,最后利用全局最大池化聚合特征。文

献[23]将无序点的特征投影到有序的特征向量序列上,以应用传统的端到端循环神经网络学习(RNN)算法。文献[24]提出了一种树形结构的CNN,为输入点云构建kd树,从枝节点到根节点进行分层特征提取。

综上所述,利用深度学习直接处理点云数据是三维模型识别与分类的发展趋势。在分析和处理点云模型过程中,构建局部邻域弥补多层感知器(MLP)学习局部特征能力的不足,是一种简单、高效的点云模型局部特征学习方法。因此,本文在文献[4]和文献[22]的基础上,结合KNN算法并引入相似性度量,提出了一种KNN-CNN模型,以增强局部特征的学习能力。

3 形状特征提取

点云模型上的每个区域都有相对应的特征描述,局部形状特征能有效刻画三维几何模型的细节信息,弥补全局形状特征描述细节信息的不足。利用CNN学习每个点的局部邻域,形成形状特征,以提高点云模型识别与分类的准确率。

3.1 K近邻构建局部邻域

点云模型上点与点间有交互作用,没有孤立的点,邻域点间往往具有相似的几何结构,因此构建局部邻域至关重要。二维空间中,图像的局部邻域由

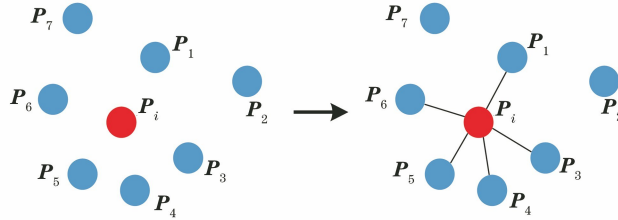


图1 局部邻域的构建过程

Fig. 1 Construction process of the local neighborhood

3.2 局部邻域构建形状特征

由于点云模型是一组没有特定顺序的点,如果将 N 个点输入到CNN中,就有 $N!$ 种排列方式。为了解决点云模型中点的无序性问题,采用逐点学习特征方法,即学习每个点的特征,最后通过最大池化层聚合所有点的特征,可表示为

$$f([\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n]^T) \approx g[h(\mathbf{P}_1), h(\mathbf{P}_2), \dots, h(\mathbf{P}_n)], \quad (2)$$

式中, $[\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n]^T$ 为输入点集, f 为对称函数, g 为与最大池化层近似的对称函数, h 为用多层感知机分别学习每个点的特征。可以发现, h 函数单独得到的每个点特征通过 g 函数聚合为全局形

一定距离度量内的像素阵列索引组成,而三维空间中,点云模型的局部邻域可由点在一定欧氏距离内 k 个点的索引组成。

点云是向量的无序集合,用一个二阶张量 $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n]^T$ 表示三维空间内的点集, \mathbf{P}_i 由该点对应的坐标 (x_i, y_i, z_i) 和特征属性组成,如颜色、法向量等。为计算方便,只使用点的坐标作为输入,为了有效存储点的局部结构信息,计算每个点的局部邻域。首先计算点与点之间的欧氏距离,其次按照欧氏距离的大小选取距离 \mathbf{P}_i 最近的 k 个近邻点,构建局部邻域。点对间的欧氏距离张量 \mathbf{D} 可表示为

$$\mathbf{D} = \mathbf{P}_{\text{sq}} + \mathbf{P}_{\text{sq}}^T - 2\mathbf{P}\mathbf{P}^T, \quad (1)$$

式中, \mathbf{P}_{sq} 为张量 \mathbf{P} 中各行向量模的平方,即 $\mathbf{P}_{\text{sq}} = [\|\mathbf{P}_1\|^2, \|\mathbf{P}_2\|^2, \dots, \|\mathbf{P}_n\|^2]^T$, \mathbf{D} 中的元素 $\mathbf{D}(i, j)$ 为点 \mathbf{P}_i 与点 \mathbf{P}_j 之间的欧氏距离。对每一行元素的 $\mathbf{D}(i, j)$, $0 < i < n, 0 < j < n$ 进行排序,确定每一行元素中 k 个距离最小点的索引。然后按照该索引在 $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n]^T$ 中选出与 \mathbf{P}_i 欧氏距离最小的 k 个点坐标组成一个张量。不同的 k 值表示不同的搜索半径,所得到的局部邻域也不同,因此需要谨慎选择 k 值。图1所示为构建点的局部邻域过程,以8个点为一个点集,建立 \mathbf{P}_i 的局部邻域,设 $k=5$,计算得到 $\mathbf{P}_1, \mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5, \mathbf{P}_6$ 是最接近 \mathbf{P}_i 的5个点,因此建立这5个点关于 \mathbf{P}_i 的索引。

状特征,并未考虑局部结构特征。因此综合局部形状特征与全局形状特征,将所有点与局部邻域点的紧凑度量同时输入到 h 函数中,即输入多层感知机,最后通过 g 函数聚合特征,可表示为

$$f([\mathbf{P}_1, \dots, \mathbf{P}_n]) \approx g[h(\mathbf{P}_1, \mathbf{P}_j - \mathbf{P}_1), \dots, h(\mathbf{P}_n, \mathbf{P}_j - \mathbf{P}_n)], \quad (3)$$

式中, $\mathbf{P}_j, 0 < j < k$ 为分别以点集 $[\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n]^T$ 中每一点为中心计算的 k 个近邻点, $\mathbf{P}_j - \mathbf{P}_n$ 为点 \mathbf{P}_n 的 k 个近邻点与 \mathbf{P}_n 的紧凑度量。此时 h 函数的输入同时包含单个点的特征和近邻点与该点的紧凑度量,经过多层感知机学习形成局部结构特征,利用全局最大池化层将所有局部结构特征聚合,形成点

云模型的形状特征,用于点云模型的识别分类,该过程如图2所示。图中 V_m 表示第 m 个点云模型的特征向量。

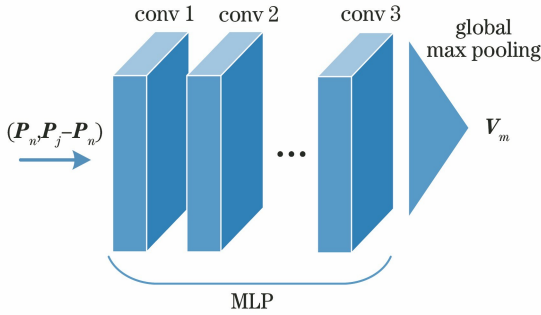


图2 特征提取过程示意图

Fig. 2 Schematic diagram of feature extraction process

(3)式在(2)式的基础上引入了点与近邻点的紧凑度量,紧凑度量(几何相似性度量)^[22]值越小表示在点云模型上该局部邻域描述的几何结构信息越细致,增大了点云模型间的差异,同时提高了点云模型识别与分类的准确率。

4 点云模型的识别与分类

为实现点云模型的识别与分类,需将点云模型对齐到规范空间再进行特征学习。将学习得到的形状特征输入到全连接层,利用Softmax函数计算所属类特征输入到全连接层,利用Softmax函数计算所属类

别的概率值,以此判断该模型的类别。通过选择合适的激活函数优化KNN-CNN模型,以得到最佳性能。

4.1 联合对齐网络

点云模型经几何变换(如旋转,平移,尺度变换)后,必须保持点云的语义信息不变,即所属类别不变。实验使用联合对齐网络(JANet)将点云模型对齐到特定空间。

JANet是一个预测空间变换矩阵的网络,由于输入是 n 个点的三维坐标 (x_n, y_n, z_n) ,因此将JANet的输出维度设置为 3×3 ,从输入数据中学习出一个 3×3 的仿射变换矩阵,并将变换矩阵与输入点的坐标相乘,得到对齐后的点云模型。JANet通过学习每个点的特征和最大池化层聚合所有点的特征,解决了点云的无序性问题,使未输入数据中的每一个点与已输入数据中的点建立关联,即将所有输入点对齐到一个规范空间。JANet模型的基本架构如图3所示,包括三次卷积操作、一次最大池化操作和一个全连接层,三次卷积操作分别输出 64 个 $n \times 1$ 维的特征图($64 @ n \times 1$)、 128 个 $n \times 1$ 维的特征图和 1024 个 $n \times 1$ 维的特征图,全连接层中各层输出维度分别为 512 、 256 。将 256 维的特征乘以 256×9 维的权重(Weights)矩阵,再加上偏置(Biases)得到 1×9 维的特征矩阵,最后转换成 3×3 的仿射矩阵。

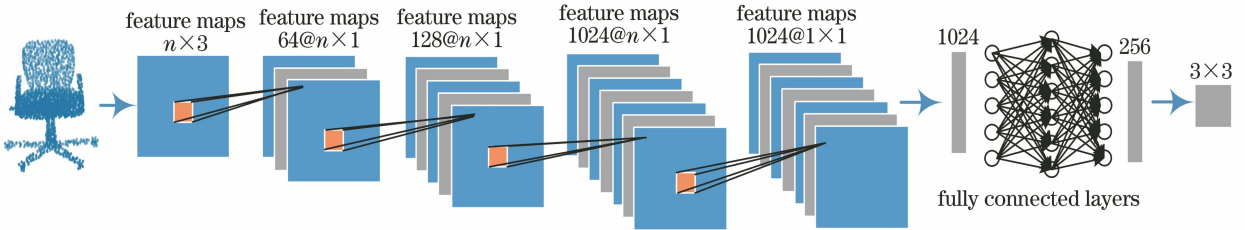


图3 JANet模型

Fig. 3 JANet model

4.2 K近邻卷积神经网络架构

KNN-CNN模型在输入层利用KNN算法构建点的局部邻域,由于卷积操作之后会导致特征的非局部扩散,所以将KNN算法嵌入CNN中构建特征的局部邻域。

用于提取三维点云特征的KNN-CNN模型基本框架如图4所示,其中KNNG为利用KNN算法构建的局部邻域。KNN-CNN包含一个JANet,五次卷积操作,五次最大池化和一个全连接层以及嵌入CNN中的KNN算法。输入坐标为 (x_i, y_i, z_i) 的点云模型,先经过JANet,对齐到规范空间形成 $n \times 3$ 维的张量。构建每个点的局部邻域得到 3 个

$n \times k$ 维的特征图,再经过卷积操作形成 64 个 $n \times k$ 维的特征图。最大池化操作可减少参数维度,以备下一次建立局部邻域。KNN-CNN模型中的最后一次池化操作生成一个 1024×1 维的特征向量,经过全连接层进行分类得分。最后用Softmax函数将全连接层的输出转化为相对应的概率,各类别对应的概率 S_r 可表示为

$$S_r = \frac{\exp(\alpha_r)}{\sum_{r=1}^c \exp(\alpha_r)}, \quad (4)$$

式中, α_r 为全连接层各神经单元的输出, C 为类别数。

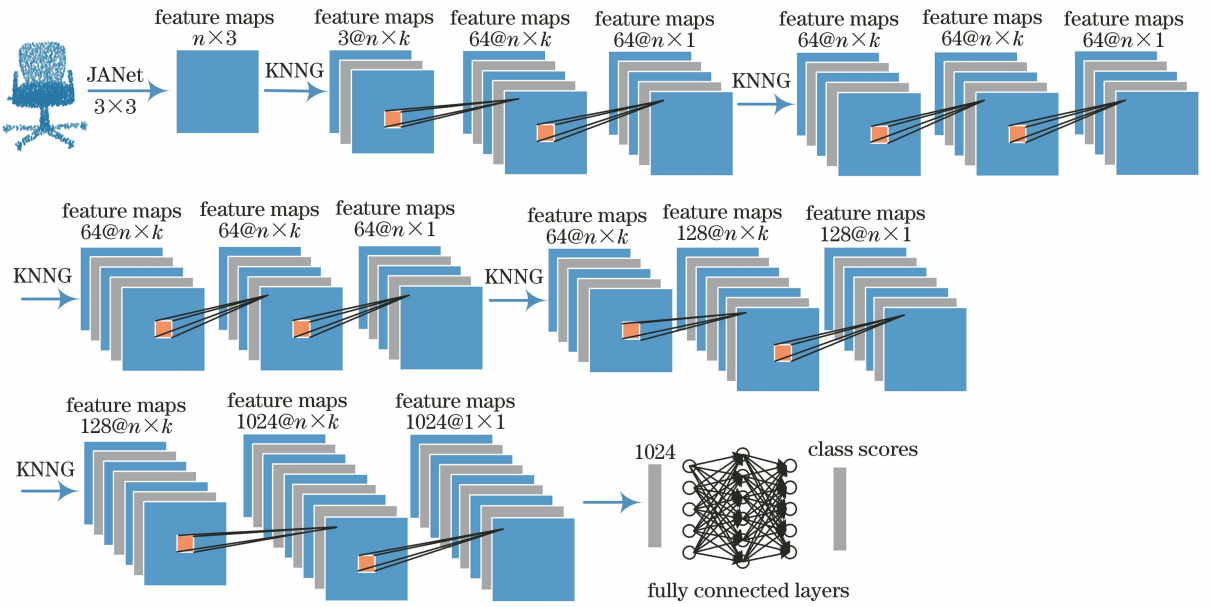


图 4 KNN-CNN 模型

Fig. 4 KNN-CNN model

4.3 激活函数的选择

激活函数将非线性特征引入 CNN 模型中,使 CNN 能够学习输入和响应变量间复杂的非线性映射。目前主流的激活函数有 Sigmoid 函数、双曲正切函数(tanh)和修正线性单元(ReLU)等。Sigmoid 函数与双曲正切函数在 CNN 训练过程易产生梯度消失问题,从而导致 CNN 中某一层网络对输入做线性映射。ReLU 函数是目前使用最广泛的激活函数,实际操作中需要结合批标准化^[25],可表示为

$$\sigma_{ReLU}(x) = \max(0, x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}, \quad (5)$$

式中, \max 为取最大值函数。

实验采用具有缩放因子的缩放指数线性单元函数(SELU)^[26],可表示为

$$\sigma_{SELU}(x) = \lambda \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases}, \quad (6)$$

式中, $\lambda = 1.0507009873554804934193349852946$ 为缩放因子, $\alpha = 1.6732632423543772848170429916717$ 。SELU 激活函数的两个优势:一是当 $x > 0$ 时,SELU 函数的梯度为大于 1 的常数,不存在梯度消失;二是与 ReLU 相比,SELU 在 $x \leq 0$ 时,函数值不为 0,避免了某些神经元不被激活。SELU 能使 CNN 每一层的输出自动归一化到均值接近 0,方差接近 1 的高斯分布,训练效果更佳。另外,当 SELU 函数的输入 x 为负时,存在一个常数使得 SELU 函数的导数趋于 0,即左软饱和特性,可以防止反向传播过程中的噪声传

播到下一层,因此 SELU 函数能使 CNN 模型鲁棒性更强。三种激活函数曲线示意图如图 5 所示,其中 ELU 融合了 sigmoid 和 ReLU。

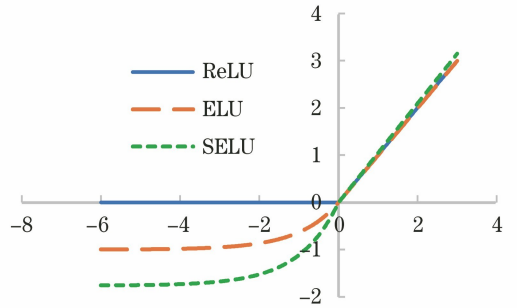


图 5 激活函数曲线图

Fig. 5 Graph of activation function

5 实验结果与分析

实验环境:操作系统为 Linux Ubuntu 16.04,硬件环境为 Intel Xeon E5-2620v4 CPU 和 NVIDIA Quadro M4000 GPU (内存为 8 GB),编程开发环境为 CUDA-Toolkit8.0,编程语言为 Python 2.7,机器学习框架为 Tensorflow。

5.1 实验数据集

为与其他算法进行比较,在 ModelNet40 数据集上进行训练和评估。该数据集包含来自 40 个类别的 12311 个网格 CAD 模型,训练模型为 9843 个,测试模型为 2468 个,包括飞机、浴缸、床、汽车等多种模型。图 6 为 ModelNet40 数据集中汽车、椅子和飞机的渲染模型、网格模型以及包含 RGB 信息的点云模型。

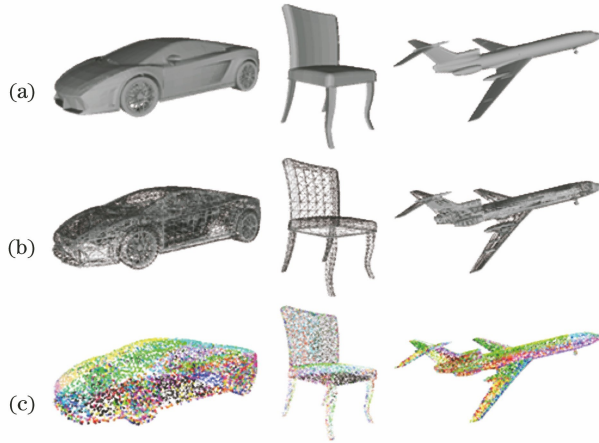


图6 三维模型的三种表示形式。(a)渲染模型;(b)网格模型;(c)3D点云模型

Fig. 6 Three representations of three-dimensional models. (a) Rendered model; (b) mesh model; (c) 3D point cloud model

5.2 参数设置与性能评价

实验使用自适应矩估计优化器(Adam)^[27]优化CNN模型,设置初始学习率为0.001,动量为0.9, Batch Size为32,丢弃率Dropout为0.5。网络训练中初始权重 w_0 初始化为高斯分布的随机数,初始偏差 b_0 设置为0。使用交叉熵损失函数 L 作为分类目标函数,可表示为

$$L = - \sum_{C=1}^M y_T \log p_C, \quad (7)$$

式中, M 为类别数量, y_T 为指示变量(0或1),如果该类别和样本的类别相同 $y_T=1$,否则 $y_T=0$, p_C 为预测样本属于类别 C 的概率。在训练集中对点云模型进行加噪(高斯白噪声)、旋转(任意角度)、尺度变换以提升CNN模型的泛化能力,在测试集中,对训练好的KNN-CNN模型进行测试,总体识别与分类准确率OA可表示为

$$X_{OA} = \frac{X_{\text{correct}}}{X_{\text{seen}}} \times 100\%, \quad (8)$$

式中, X_{correct} 为在测试集上识别分类正确的模型数量, X_{seen} 为测试集模型的数量。

5.3 实验结果分析

将本算法与文献[4]、文献[20]、文献[22]、文献[24]使用的算法在ModelNet40数据集上进行测试,结果如表1所示。其中MA表示各类别识别与分类准确率的平均值,是各类别识别与分类准确率与类别数之比,Input表示输入模型点的数量,normal表示正常点云模型,-表示对应方法未公布该项数据。可以看出相比其他算法,本算法在ModelNet40数据集上的OA有明显提高。原因是本算法利用KNN算法构建了点云模型局部邻域,

有效利用了局部结构信息,因此比文献[4]中的算法OA提高了2.8个百分点。为防止信息在点云特征中的非局部扩散,利用KNN算法逐特征建立局部邻域,因此比文献[22]中只考虑点云模型局部结构信息的算法,OA提高了1.0个百分点。对比文献[20]分组处理点云的PointNet网络,OA也提高了1.3个百分点。由于文献[24]从点云中构建kd树,而kd树的节点必须与输入点数目相同。若不满足此条件,则需要对输入点云做上采样或者下采样处理,导致OA受输入点数和kd树深度影响较大。本算法与文献[4]在ModelNet40的数据集中各类别识别与分类准确率对比如表2所示。可以看出,对于飞机、吉他、键盘等几类具有明显特征的模型,本算法与文献[4]中算法的识别与分类准确率均达到100.0%。相比于文献[4],本文算法对钢琴、电视柜、浴缸等的识别与分类准确率也有很大提高,尤其是对电视柜的识别与分类准确率提高了12.0个百分点。这证明了本算法对特征不明显的模型也能够挖掘其细粒度特征,提高了识别细粒度模型的能力。

表1 不同算法在ModelNet40数据集上的识别准确率

Table 1 Recognition accuracy of different algorithms on ModelNet40 dataset

Method	Input	MA / %	OA / %
PointNet ^[4]	1024	86.2	89.2
PointNet++ ^[20]	1024	-	90.7
PointNet++ ^[20]	normal	-	91.9
KCNet ^[22]	1024	-	91.0
Kd-Net(depth 10) ^[24]	1024	86.3	90.6
Kd-Net(depth 15) ^[24]	32768	88.5	91.8
Ours	1024	89.9	92.0

表 2 ModelNet40 测试集中各类别模型识别准确率对比

Table 2 Comparison of recognition accuracy of each category model in ModelNet40 test set unit: %

Category	PointNet ^[4]	Our model
Airplane	100.0	100.0
Bath tub	86.0	94.0
Bed	97.0	99.0
Bench	70.0	75.0
Bookshelf	91.0	99.0
Bottle	94.0	98.0
Bow	90.0	100.0
Car	98.0	99.0
Chair	97.0	98.0
Cone	95.0	100.0
Cup	80.0	75.0
Curtain	90.0	95.0
Desk	80.2	87.2
Door	80.0	95.0
Dresser	72.1	76.7
Flower pot	20.0	20.0
Glass box	98.0	92.0
Guitar	100.0	100.0
Keyboard	100.0	100.0
Lamp	95.0	90.0
Laptop	100.0	100.0
Mantel	95.0	96.0
Monitor	96.0	99.0
Night stand	72.1	82.6
Person	90.0	100.0
Piano	86.0	95.0
Plant	76.0	82.0
Radio	75.0	90.0
Range hood	92.0	95.0
Sink	80.0	80.0
Sofa	95.0	96.0
Stairs	85.0	95.0
Stool	85.0	85.0
Table	81.0	81.0
Tent	95.0	95.0
Toilet	98.0	98.0
Television stand	79.0	91.0
Vase	80.0	83.0
Wardrobe	60.0	70.0
Xbox	85.0	90.0

利用 KNN 算法构建局部邻域图的过程中,不同的搜索半径 k 下的局部结构信息也不相同,从而影响识别与分类结果。因此在其他条件不变的情况

下,多次设置不同的 k 值进行实验以寻找最优结果。表 3 为选取不同 k 值对实验结果的影响,其中 k_1 表示对点云使用 KNN 算法时设置的近邻点个数, k_2 表示对特征使用 KNN 算法时设置的特征个数。可以看出,随着 k_1 、 k_2 的增加,算法的 OA 也不断提高,这表明在构建局部邻域时,局部邻域内必须包含一定数量的点或特征,避免局部特征数量较少对识别分类结果产生影响。但 k_1 、 k_2 的增大,会导致构建的局部邻域增大,从而增加训练时间。

表 3 k 值对 OA 的影响Table 3 Effect of k value on OA

k_1	k_2	OA / %	Time / h
15	15	88.5	72.0
20	10	91.2	72.1
20	20	91.3	93.3
30	20	91.6	107.3

在 KNN-CNN 训练过程中分别使用了 ReLU、ELU、SELU 等激活函数并结合批标准化处理优化 KNN-CNN 模型,结果如表 4 所示。可以看出,SELU 函数结合批标准化处理在 KNN-CNN 训练过程中的优化效果最好,OA 最高。相比 ReLU 和 ELU 激活函数,SELU 函数能够使 KNN-CNN 每一层的输出归一化为均值接近 0,方差接近 1 的高斯分布,但仍存在提升空间,需要结合批标准化才能得到最优结果。

表 4 不同优化方案的识别准确率

Table 4 Recognition accuracy of different

optimization schemes	unit: %
Activate function	OA
SELU	91.3
ReLU+BN	91.6
ELU+BN	91.8
SELU+BN	92.0

图 7 为本算法部分错误识别与分类的点云模型的三种角度投影图,可以看出,预测结果与真实模型都具有相似特征。如将图 7(c)中的真实模型植物预测为花盆,将图 7(d)中的真实模型花盆预测为植物。原因是 ModelNet40 数据集中一些花盆模型与植物模型具有某些共同特征,导致本算法对这种具有极其相似特征的模型识别准确率较低,这表明本算法虽有效结合了局部结构特征,但在提取模型的细粒度特征上仍存在提升的空间。

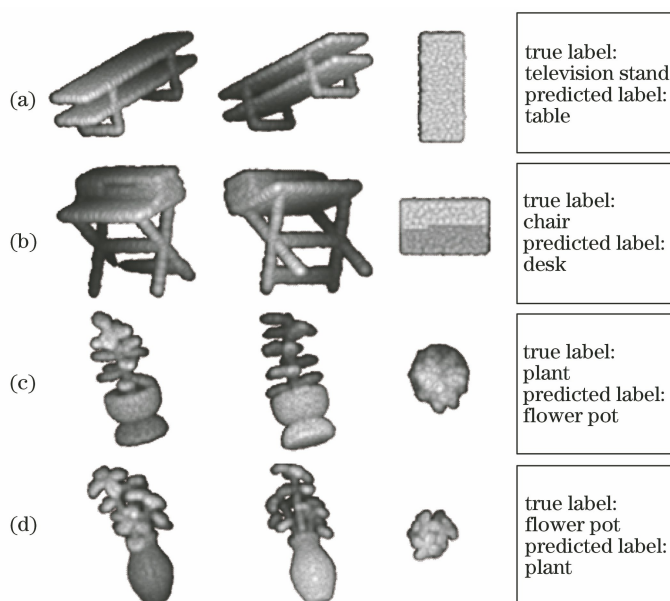


图7 错误分类的点云模型实例。(a)电视柜;(b)椅子;(c)植物;(d)花盆

Fig. 7 Examples of misclassified point cloud models. (a) Television stand; (b) chair; (c) plant; (d) flower pot

6 结 论

针对计算机视觉领域点云模型的识别与分类提出了一种KNN-CNN模型。首先对点云模型中的每个点利用KNN算法构建局部邻域,有效结合了局部结构信息。然后应用JANet将点云模型对齐到特定空间,为防止信息在点云特征中的非局部扩散,同时将KNN算法嵌入CNN模型中对每个特征建立局部邻域。最后经CNN学习后得到模型特征,利用全连接层和Softmax函数预测点云模型所属类别。在公开数据集ModelNet上与其他主流算法进行了对比实验,结果表明本算法的识别准确率有所提高,但仍需要进一步优化。一方面,对具有极其相似特征的三维模型仍不能准确识别;另一方面,CNN模型的层数相对较浅,无法提取更深层次的特征,这些都是未来需要研究的方向。

参 考 文 献

- [1] Seidenari L, Serra G, Bagdanov A D, et al. Local pyramidal descriptors for image recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(5): 1033-1040.
- [2] Timofte R, Zimmermann K, van Gool L. Multi-view traffic sign detection, recognition, and 3D localisation [J]. Machine Vision and Applications, 2014, 25(3): 633-647.
- [3] Yang Z Z, Kuang N, Fan L, et al. Review of image classification algorithms based on convolutional neural

networks[J]. Journal of Signal Processing, 2018, 34(12): 1474-1489.

杨真真, 匡楠, 范露, 等. 基于卷积神经网络的图像分类算法综述[J]. 信号处理, 2018, 34(12): 1474-1489.

- [4] Charles R Q, Hao S, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017. Honolulu, HI, USA. New York: IEEE, 2017: 652-660.
- [5] Johnson A E, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(5): 433-449.
- [6] Frome A, Huber D, Kolluri R, et al. Recognizing objects in range data using regional point descriptors [M] // Pajdla T, Matas J. Computer Vision-ECCV 2004. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, 3023: 224-237.
- [7] Rusu R B, Blodow N, Marton Z C, et al. Aligning point cloud views using persistent feature histograms [C] // 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22-26, 2008. Acropolis Convention Center, Nice, France. New York: IEEE, 2008: 3384-3391.
- [8] Chaudhuri S, Koltun V. Data-driven suggestions for creativity support in 3D modeling [J]. ACM Transactions on Graphics, 2010, 29(6): 183.1-183.9.

- [9] Kobbelt L, Schröder P, Kazhdan M, et al. Rotation invariant spherical harmonic representation of 3D shape descriptors [J]. Eurographics Symposium on Geometry Processing, 2003, 43(2): 156-164.
- [10] Rustamov R M. Laplace-Beltrami eigenfunctions for deformation invariant shape representation [C] // Proceedings of the Fifth Eurographics Symposium on Geometry Processing, July 04-06, 2007, Barcelona, Spain. Berlin: Springer, 2007: 225-233.
- [11] Sun J, Ovsjanikov M, Guibas L. A concise and provably informative multi-scale signature based on heat diffusion [J]. Computer Graphics Forum, 2009, 28(5): 1383-1392.
- [12] Aubry M, Schlickewei U, Cremers D. The wave kernel signature: a quantum mechanical approach to shape analysis [C] // IEEE International Conference on Computer Vision Workshops, November 6-13, 2011, Barcelona, Spain. New York: IEEE, 2011: 1626-1633.
- [13] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3D shape recognition [C] // 2015 IEEE International Conference on Computer Vision, December 7-13, 2015. Santiago, Chile. New York: IEEE, 2015: 945-953.
- [14] Shi B G, Bai S, Zhou Z C, et al. DeepPano: deep panoramic representation for 3-D shape recognition [J]. IEEE Signal Processing Letters, 2015, 22(12): 2339-2343.
- [15] Wu Z R, Song S R, Khosla A, et al. 3D ShapeNets: a deep representation for volumetric shapes [C] // 2015 IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015. Boston, MA, USA. New York: IEEE, 2015: 1912-1920.
- [16] Xu, Todorovic S. Beam search for learning a deep convolutional neural network of 3D shapes [C] // 2016 23rd International Conference on Pattern Recognition, December 4-8, 2016. Cancun, Mexico. New York: IEEE, 2016: 3506-3511.
- [17] Yang J, Wang Y M. 3D model recognition based on depth convolution neural network [J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2019, 31(2): 253-260.
杨军, 王亦民. 基于深度卷积神经网络的三维模型识别 [J]. 重庆邮电大学学报(自然科学版), 2019, 31(2): 253-260.
- [18] Yang J, Wang S, Zhou P. Recognition and classification for three-dimensional model based on deep voxel convolution neural network [J]. Acta Optica Sinica, 2019, 39(4): 0415007.
杨军, 王顺, 周鹏. 基于深度体素卷积神经网络的三维模型识别分类 [J]. 光学学报, 2019, 39(4): 0415007.
- [19] Riegler G, Ulusoy A O, Geiger A. OctNet: learning deep 3D representations at high resolutions [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017. Honolulu, HI, USA. New York: IEEE, 2017: 3577-3586.
- [20] Ravanbakhsh S, Schneider J, Poczos B. Deep learning with sets and point clouds [EB/OL]. [2019-09-01]. <https://arxiv.org/abs/1611.04500>.
- [21] Qi C R, Yi L, Su H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space [C] // Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA. 2017: 5099-5108.
- [22] Shen Y R, Feng C, Yang Y Q, et al. Mining point cloud local structures by kernel correlation and graph pooling [C] // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018. Salt Lake City, UT, USA. New York: IEEE, 2018: 4548-4557.
- [23] Huang Q G, Wang W Y, Neumann U. Recurrent slice networks for 3D segmentation of point clouds [C] // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018. Salt Lake City, UT, USA. New York: IEEE, 2018: 2626-2635.
- [24] Klovov R, Lempitsky V. Escape from cells: deep kd-networks for the recognition of 3D point cloud models [C] // 2017 IEEE International Conference on Computer Vision, October 22-29, 2017. Venice, Italy. New York: IEEE, 2017: 863-872.
- [25] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C] // Proceedings of the 32nd International Conference on Machine Learning, July 6-11, 2015, Lille, France. 2015: 448-456.
- [26] Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks [C] // Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA. 2017: 971-980.
- [27] Kingma D P, Ba J. Adam: a method for stochastic optimization [EB/OL]. [2019-08-28]. <https://arxiv.org/abs/1412.6980>.