DOI：10.13973/j.cnki.robot.2017.0820

一种基于深度学习的机械臂抓取方法

杜学丹[1,2]，蔡莹皓[1]，鲁 涛[1]，王 硕[1]，闫 哲[2]

（1. 中国科学院自动化研究所复杂系统管理与控制国家重点实验室，北京 100190；2. 哈尔滨理工大学自动化学院，黑龙江 哈尔滨 150080）

摘 要：提出了一种基于深度神经网络的机械臂最优抓取位置检测方法．相比传统手工设定的特征，基于深度神经网络的方法学习得到的特征具有较强的鲁棒性和稳定性，能够适应训练集中未曾出现的新物体．本方法首先使用基于深度学习的目标检测算法对图像中的目标物体进行检测，记录目标的类别和位置．然后根据分类检测结果，使用基于深度学习的机械臂抓取方法进行抓取位置学习．仿真实验表明所提方法能对图像中的目标物体进行较为准确的分类，在 Universal Robot 5 机械臂上得到的抓取实验结果证明了所提方法的有效性．

关键词：机械臂抓取；深度学习；目标检测；分类

中图分类号：TP242 **文献标识码：**A **文章编号：**1002-0446(2017)-06-0820-09

A Robotic Grasping Method Based on Deep Learning

DU Xuedan[1,2]，CAI Yinghao[1]，LU Tao[1]，WANG Shuo[1]，YAN Zhe[2]

(1. *The State Key Laboratory of Management and Control for Complex Systems*, *Institute of Automation*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*; 2. *School of Automation*, *Harbin University of Science and Technology*, *Harbin* 150080, *China*))

Abstract: A method is proposed to detect the optimal position of robotic grasping based on deep neural network. Compared with conventional manually-set features, the features learned by deep neural network methods are more robust and stabler, and can be applied to objects outside of the training set. In this method, the object detection algorithm based on deep learning is first used to detect the objects in the image with the classes and locations of the objects recorded. Then, the robotic grasping method based on deep learning is used to learn the grasping positions according to the object classification and detection results. Simulation experiments indicate that the proposed method can classify the objects in the images accurately, and the grasping experimental results on Universal Robot 5 verify the effectiveness of the proposed method.

Keywords: robotic grasping; deep learning; object detection; classification

1 引言（Introduction）

近年来，随着计算机技术的发展，计算机视觉作为人工智能的一个重要研究领域，已经广泛应用于各行各业，其中基于视觉的机械臂抓取也逐渐成为当前的一个研究热点．在机械臂抓取任务中，传统的方法一般采用人工示教的方式，如手掰机械臂，使机械臂到某个固定位置进行抓取．由于抓取位姿凭靠的是记忆且机械臂自身没有感知能力，因而在执行任务时容易受到外界环境中许多不确定因素的影响，如当物体位置发生变化时，机械臂则会抓不到物体．计算机视觉解决机械臂抓取问题的通常做法是，首先利用相机等采集设备对目标物体进行采样，然后结合模式识别、图像处理等方法分析和处理采得的图像数据，获得目标物体的空间位置和姿态等有效信息，最后利用所得信息使机械臂完成抓取动作．计算机视觉使机器拥有了“看”的能力，能够使其有效感知外部环境．

文 [1-4] 介绍了基于视觉的机械臂抓取任务，其共同点是都采用传统的特征提取方法来处理图像信息．这些方法一般由设计者针对特定问题手工设计而成，因受到目标物体的形状、大小、角度变化、外部光照等因素的影响，因而所提取的特征泛化能力不强，鲁棒性较差，难以适应新物体．

深度学习的概念 [5] 由 Hinton 等人于 2006 年首先提出．直到 Krizhevsky 等人使用深度学习的方法 [6] 在 2012 年的 ImageNet 比赛中取得突破性成绩后，深度学习呈现出了爆发式发展．相较传统的手工提取特征的方法，深度学习的优势在于特征提取环节不需要使用者预先选定提取何种特征，而是采

基金项目：国家自然科学基金（61503381）；北京市科技计划（Z171100000817009）．
通信作者：蔡莹皓，yinghao.cai@ia.ac.cn 收稿/录用/修回：2017-03-02/2017-05-22/2017-05-25

用一种通用的学习过程使模型从大规模数据中学习进而学得目标具备的特征[7].

深度学习在机械臂抓取任务中也有其应用. 文 [8] 使用深度神经网络直接从包含稳定抓取的局部目标视图中检测抓手手掌和手指的位置，进而完成抓取任务. 文 [9] 使用抓取函数对所有可能抓到目标的位置作分数预测，通过平滑抓取函数使得模型对不确定的抓取位置有较强的鲁棒性，抓取函数则由卷积神经网络学习获得. 文 [10] 提出一种两步级联检测系统，通过处理 RGB-D 多模态信息对目标物体的最优抓取位置进行检测，并获得了较好的抓取效果.

机械臂对目标物体进行分类并抓取这一工作有其实际应用价值，分拣任务就是其中之一. 传统的分拣工作采用人工分拣方式，劳动强度大且工作重复枯燥. 使用机器人代替人工，不仅可以将工人从繁复的劳动中解放出来，而且可以提高工作效率，降低劳动成本. 分拣机器人应用的较为成功的要数垃圾分拣机器人，它可对不同种类、不同尺寸的垃圾分类，从众多垃圾中辨别出可回收部分并将其归类，从而提高有效材料的回收率和利用率.

本文提出一种基于深度神经网络的机械臂最优抓取位置检测方法. 相比人工设定的特征，基于深度神经网络的方法学习得到的特征能够适应训练集中未曾出现的新物体，具有较强的泛化性和稳定性. 本方法首先使用基于深度学习的目标检测算法[11] 对图像中的目标物体进行检测，记录目标的类别和位置. 其次根据分类检测结果，使用基于深度学习的方法[10] 进行抓取位置学习. 实验结构及算法流程如图 1 所示.

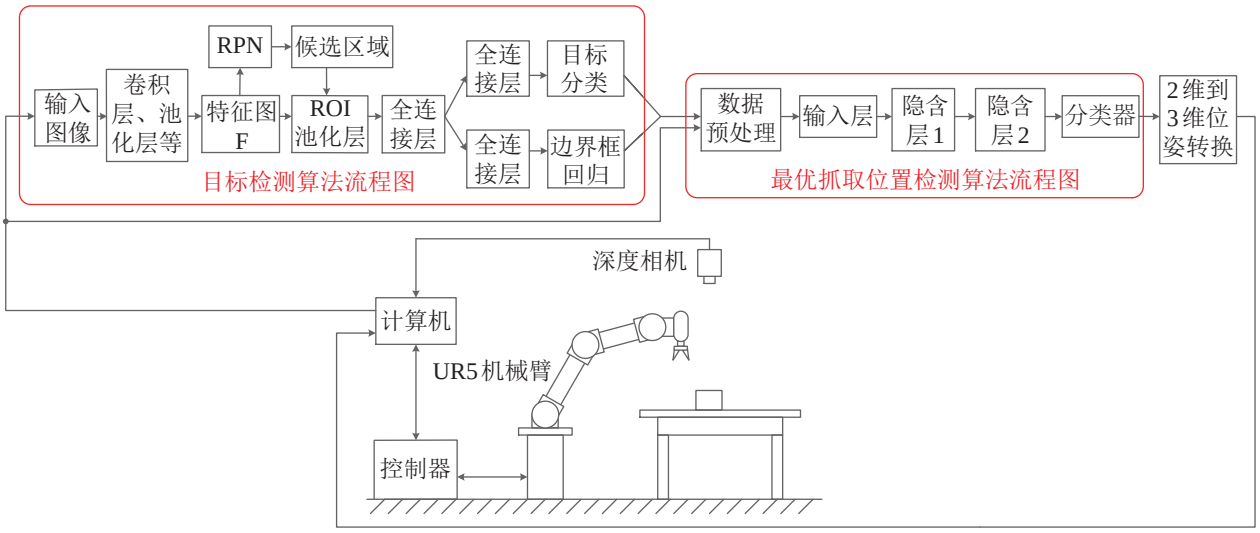


图 1　实验结构及算法流程图

Fig.1　Experimental devices and flow-chart of the proposed algorithm

2　目标检测（Object detection）

深度学习应用在目标检测中的一个经典例子就是基于区域的卷积神经网络（R-CNN）[12]. 文献首先使用选择性搜索（SS）[13] 方法在整个图像中搜索候选区域，然后使用 CNN（卷积神经网络）模型依次在每个候选区域上提取特征并将其转化为固定长度的特征向量，接着将特征向量输入到特定种类的线性支持向量机中分类并得到与候选区域对应的类别分数，最后根据类别分数的高低和非极大值抑制方法选择出最佳边界框. 然而，R-CNN 的不足之处在于模型对每个候选区域都要输入到 CNN 中提取特征，并且物体类别和边界框不是同时生成，因此模型计算量较大，不能达到实时效果.

快速 R-CNN[14] 在 R-CNN 的基础上对检测模型作出了改进. 文 [14] 首先将整个图像输入到卷积层和池化层中生成特征图，接着使用 ROI（感兴趣区域）池化层处理特征图并使其生成一个固定长度的特征向量. 然后将特征向量输入到全连接层分别进行 2 个不同的处理：一个是输入到 softmax 分类器中对候选区域分类；另一个是进行边界框回归，以此产生对应此候选区域的边界框参数. 最后，通过非极大值抑制产生最终结果，快速 R-CNN 的优点是对整个图像只进行一遍卷积操作，并且目标类别和边界框是一同输出的.

虽然快速 R-CNN 的训练过程有所简化，检测效果有所提升，但是仍然达不到实时要求，原因在于提取候选区域的效率仍旧不高，因而更快 R-CNN 模型被提出来解决此问题. 更快 R-CNN 模型引入

区域建议网络（region proposal network，RPN）代替快速 R-CNN 中的 SS 方法提取候选区域．RPN 本质上是一个全卷积神经网络，其输入是共享卷积网络部分输出的特征图，输出为候选区域预测及对候选区域是目标还是背景的判断．RPN 的输出结果与共享网络的特征映射被一同输入到 ROI 池化层为每个候选区域生成固定长度的特征向量，之后的目标分类和边界框回归则与快速 R-CNN 相同．使用 RPN 网络代替 SS 方法，使得目标检测效率明显提高i，在 GPU（图形处理器）加速下基本实现实时检测效果．针对模型的性能，本文最终采用更快 R-CNN 作为目标检测模型加入到机械臂抓取任务中．

之所以采用目标检测方法而不是目标分类，是因为在本文的机械臂抓取任务中采样摄像机被固定在机械臂上方支架上，与操作台有一定距离．这种方式采样所得图像中的目标尺寸较小，很大一部分都是背景，用于目标分类训练时，深度网络容易误将背景当作目标来学习，从而导致分类效果较差．使用目标检测作为分类方法可以有效提高分类准确度．

2.1 目标检测模型

文 [11] 中更快 R-CNN 算法大体由 2 部分构成，一是 RPN 模块，用于提取候选区域；二是摒弃SS 算法的快速 R-CNN 检测模块，用于检测并识别候选区域的目标．文 [11] 分别在 ZFNet[15] 和 VGG-16[16] 模型的基础上做目标检测实验，本文选择使用 ZFNet 模型．

目标检测算法流程图如图 1 中红色框内所示．RPN 与快速 R-CNN 共享的网络结构为 ZFNet 全连接层前的部分．特征图 F 从共享网络中输出并被存储，同时被输入到 RPN 模型的独立结构中用于提取候选区域．候选区域提取出来后，会和特征图 F 一同送入网络剩余部分进行目标分类和边界框回归．

RPN 模型主要由 2 层神经元组成，其框架如图 2 所示．第 1 层为卷积层，将 $n\times n$ 的卷积核视为滑动窗口在卷积特征图 F 上进行卷积运算并获得一个 256 维的特征向量．接着特征向量被分别送入 2 个全连接层中，即 reg 层和 cls 层．reg 层用来预测候选区域的位置，cls 层用来判断当前候选区域是否为目标．图 2 中滑动窗口的中心点对应到原图的相应位置后，可将此位置视为一个锚并在其上框出 k 个不同尺度、不同比例的锚框．滑动窗口在特征图 F 上滑动一遍后可生成一系列锚框，经过 cls 层和 reg 层的处理，即可得到每个锚框的目标得分和回归边界．

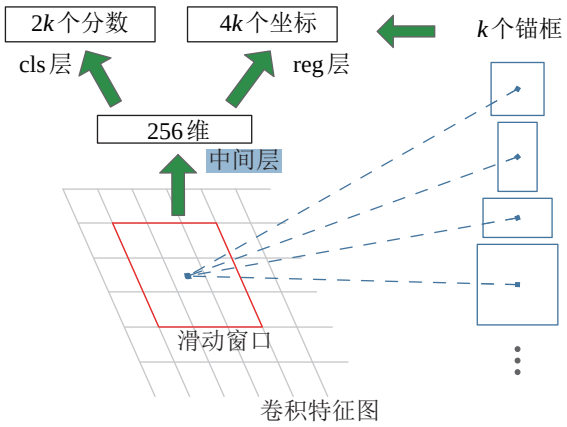


图 2 RPN 框架

Fig.2 The architecture of region proposal network

对于 RPN 的训练，文 [11] 给每个锚框分配了一个二元标签，即此锚框是否为目标．有 2 类锚框可被分配为正标签：1) 与某个或某些真实框有最高的交集并集之比的锚框；2) 与任意真实框的交集并集之比超过 0.7 的锚框．与真实框的交集并集之比小于 0.3 的锚框被分配为负标签，剩余的锚框对模型训练没有贡献．基于这些定义，文 [11] 采用多任务损失来最小化目标函数，并将目标函数定义为

$$L(\{p_i\},\{\boldsymbol{t}_i\})=\frac{1}{N_{\rm cls}}\sum_i L_{\rm cls}(p_i,p_i^*)+\lambda\frac{1}{N_{\rm reg}}\sum_i p_i^* L_{\rm reg}(\boldsymbol{t}_i,\boldsymbol{t}_i^*)$$

其中，i 表示一个小批样本中的锚索引；p_i 表示锚框 i 为目标的概率预测；真实标签 p_i^* 的值为 1 时表示锚框 i 为正标签，为 0 时表示锚框 i 为负标签；$\boldsymbol{t}_i$ 表示预测的边界框的 4 个参数化坐标向量；$\boldsymbol{t}_i^*$ 表示正标签锚框 i 对应真实框的坐标向量；$L_{\rm cls}$ 和 $L_{\rm reg}$ 分别为分类损失和回归损失；$N_{\rm cls}$ 和 $N_{\rm reg}$ 表示归一化参数；λ 表示平衡权重．

文 [11] 采用线性回归方法，目的是微调回归边界，进而能够对目标进行更加准确的定位．其平移缩放参数计算如下：

$$\begin{aligned}
t_x&=(x-x_{\rm a})/w_{\rm a}, & t_y&=(y-y_{\rm a})/w_{\rm a}\\
t_w&=(w-w_{\rm a})/w_{\rm a}, & t_h&=(h-h_{\rm a})/w_{\rm a}\\
t_x^*&=(x^*-x_{\rm a})/w_{\rm a}, & t_y^*&=(y^*-y_{\rm a})/w_{\rm a}\\
t_w^*&=(w^*-w_{\rm a})/w_{\rm a}, & t_h^*&=(h^*-h_{\rm a})/w_{\rm a}
\end{aligned}$$

其中，x,y,w 和 h 分别表示预测边界框的中心坐标、宽度和高度，$x_{\rm a},y_{\rm a},w_{\rm a}$ 和 $h_{\rm a}$ 分别表示锚框的中心坐

标、宽度和高度，x^*, y^*, w^* 和 h^* 分别表示真实边界框的中心坐标、宽度和高度.

在更快 R-CNN 中，虽然 RPN 与快速 R-CNN 是单独训练的，但并不是分别训练出 2 个网络，而是通过共享网络将 2 个训练网络结合起来实现端对端训练. 更快 R-CNN 由 ImageNet 预训练模型进行初始化，训练步骤大致为：1) 使用反向传播算法和随机梯度下降法训练 RPN；2) 使用由步骤 1) 生成的候选区域，在快速 R-CNN 上进行训练；3) 使用上一步中的网络参数初始化 RPN，并微调 RPN 单独的网络部分；4) 使用上一步中的网络参数初始化快速 R-CNN，微调快速 R-CNN 单独的部分. 接下来就是重复 3) ～ 5) 三个步骤，直到训练结束.

3　最优抓取位置检测（Detection of the optimal grasping position）

对于最优位置检测，本文采用文 [10] 中的算法. 该算法是一个由深度网络组成的两步级联系统，第 1 步用于选择一组包含目标物体的候选抓取区域，第 2 步在前一步的基础上在候选区域上进行检测并获取最优抓取框. 最终用于机械臂抓取的数据表示为 (u, v, θ)，其中 (u, v) 为最优抓取框在图像坐标系下的重心位置，θ 为抓取框的长与图像坐标系纵轴的夹角，位置如图 3 所示. 图中的椭圆表示待抓取目标；矩形框表示夹持器位置，其黄色线表示夹持器的开口方向.

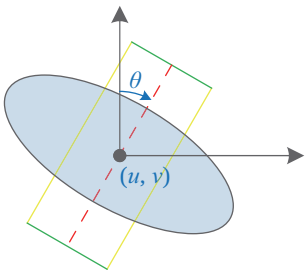
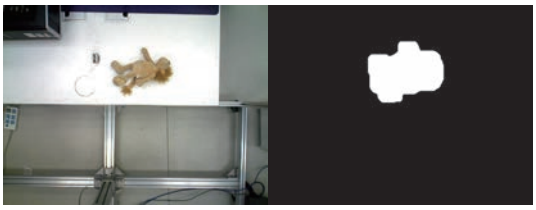


图 3　抓取位置表示

Fig.3　The representation of the grasping position

在检测最优位置之前，首先要获得包含目标的大致区域，实现方法是用含有目标物体的图像减去相应的背景图像，获得一张包含目标的二值图，二值图如图 4(b) 所示. 其中背景为黑色，目标为白色，然后根据颜色信息获得目标的大致位置. 获得大致位置后的下一步做法是将包含目标的最小矩形图像分别从二值图、彩色图、深度图和基于深度图的表面法向量特征图上截取出来，经过旋转、白化数据、保持纵横比等操作，生成若干组搜索框. 每生成一组搜索框时，这组搜索框就被转化成一个 24×24×7 大小的输入特征，24×24 为搜索框的归一化尺寸；7 为特征通道数.



(a)　　(b)

图 4　原图及其二值图

Fig.4　The original image and its binary image

紧接着，输入特征会被送到一个具有 2 个隐含层的神经网络中，网络结构如图 5 所示. 2 个隐含层分别用 $h^{[1]}$ 和 $h^{[2]}$ 表示，每层神经元数量分别为 K_1 和 K_2，神经元的输出形式为 sigmoid 函数. 在第 2 隐含层的顶部为一个逻辑分类器，用于预测. 网络的前向传播过程为

$$h_j^{[1](t)} = \sigma\Big(\sum_{i=1}^{N} x_i^{(t)} W_{i,j}^{[1]}\Big)$$

$$h_j^{[2](t)} = \sigma\Big(\sum_{i=1}^{K_1} h_i^{[1](t)} W_{i,j}^{[2]}\Big)$$

$$P(\hat{y}^{(t)} = 1 | \boldsymbol{x}^{(t)}; \boldsymbol{\Theta}) = \sigma\Big(\sum_{i=1}^{K_2} h_i^{[2](t)} W_{i,j}^{[3]}\Big)$$

其中，N 为特征向量维度；特征 $\boldsymbol{x}^{(t)} \in \mathbb{R}^N$，输出 $\hat{y}^{(t)} \in \{0, 1\}$；sigmoid 函数 $\sigma(a) = 1/(1+\exp(-a))$；$\boldsymbol{\Theta} = (\boldsymbol{W}^{[1]}, \boldsymbol{W}^{[2]}, \boldsymbol{W}^{[3]})$ 为神经网络的权值，由深度学习算法获得. 对于权值训练，文 [10] 的目标是找到一个最优单一抓取框，使得机械臂抓到目标的概率最大，其数学表示如下：

$$G^* = \arg\max_G P(\hat{y}^{(t)} = 1 | \phi(G); \boldsymbol{\Theta})$$

G 表示特定抓取框的位置、方向和大小，G^* 表示最优抓取框；函数 ϕ 用于提取矩形框 G 的恰当输入表示.

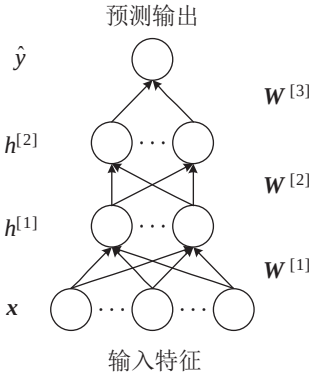


图 5　神经网络结构

Fig.5　The structure of the neural network

权值训练的第 1 阶段是使用无监督特征学习初始化隐含层 $h^{[1]}$ 和 $h^{[2]}$ 的权值，采用稀疏自动编码器的一个变种来预训练网络．稀疏自动编码器要根据以下算法解决初始化权值问题：

$$\boldsymbol{W}^*=\arg\min_{\boldsymbol{W}}\sum_{t=1}^{M}\left(\left\|\hat{\boldsymbol{x}}^{(t)}-\boldsymbol{x}^{(t)}\right\|_2^2+\lambda\sum_{j=1}^{K}g(h_j^{(t)})\right)+\beta f(\boldsymbol{W})$$

$$h_j^{(t)}=\sigma\left(\sum_{i=1}^{N}x_i^{(t)}W_{i,j}\right)$$

$$\hat{x}_i^{(t)}=\sum_{i=1}^{K}h_j^{(t)}W_{i,j}$$

$g(h)$ 为在隐含层单元激活上定义的一个稀疏惩罚，权重为 λ．$f(\boldsymbol{W})$ 为正则化函数，权重为 β．算法首先初始化 $\boldsymbol{W}^{[1]}$ 来重建 $\boldsymbol{x}$，然后固定 $\boldsymbol{W}^{[1]}$ 学习 $\boldsymbol{W}^{[2]}$ 来重建 $h^{[1]}$．权值训练的第 2 个阶段是监督学习，学习算法将分类权值 $\boldsymbol{W}^{[3]}$ 的学习与隐含层权值 $\boldsymbol{W}^{[1]}$ 和 $\boldsymbol{W}^{[2]}$ 的微调相结合，并最大化隐含层中带有正则化惩罚的权值对数似然：

$$\boldsymbol{\Theta}^*=\arg\max_{\boldsymbol{\Theta}}\sum_{t=1}^{M}\log P(\hat{y}^{(t)}=y^{(t)}|\boldsymbol{x}^{(t)};\boldsymbol{\Theta})-\beta_1 f(\boldsymbol{W}^{[1]})-\beta_2 f(\boldsymbol{W}^{[2]})$$

对于 RGB-D 多模态输入，文 [10] 提出了结构正则化特征学习．正则化特征学习有利于消除多模态间的弱连接，提高特征学习的泛化性能，经过实验对比，最终采用的正则化函数如下：

$$f(\boldsymbol{W})=\sum_{j=1}^{K}\sum_{r=1}^{R}I\{(\max_i S_{r,i}|W_{i,j}|)>0\}$$

其中，$\boldsymbol{S}$ 为模态矩阵，大小为 $R\times N$，R 为模态数，N 为特征向量维度；$S_{r,i}$ 表示可见单元 x_i 在模态 r 中的元素．当括号中的值为真时 I 为 1，否则为 0.

4 实验结果及分析（Experiment results and analysis）

机械臂抓取仿真实验算法将更快 R-CNN 算法与文 [10] 提供的抓取算法相结合，实验环境为 Windows 系统，更快 R-CNN 算法的编程环境为 Matlab R2013a，抓取算法的编程环境为 Visual Studio 2013．因此，本实验采用的是 Matlab 与 VS 混编的方式，在 VS 下调用 Matlab 进行实验．

本实验所需数据集为自建数据集和文 [10] 中提供的抓取数据集．其中，自建数据集按照 PASCAL VOC 数据集格式建立，用于目标检测和最优位置检测，抓取数据集用于最优位置检测．UR5 机械臂支架上方固定的深度相机采集的图像仅用于实验测试阶段，不参与模型的训练．在对比实验时如未说明，实验过程均保持在其他条件相同的情况下，对比某一特殊条件所产生的结果．

在目标类别检测阶段，使用的数据集包含 7 类共 1400 张图片，每类 200 张．其中，7 个类别分别为“bottle”、“hedgehog”、“polarbear”、“squirrel”、“umbrella”、“box”和“scoop”．用于模型训练的图片每类各随机选 160 张，剩余的 280 张用于模型测试．

4.1 仿真实验

4.1.1 目标检测

在对目标检测网络进行训练时，实验环境为 Ubuntu 系统，并使用显卡 GTX980 对训练过程进行加速．更快 R-CNN 模型所依赖的深度学习框架为 caffe，模型训练所需的超参数设置如下：1) 基础学习率 base_lr 为 0.001；2) 学习率的衰减策略 lr_policy 为“step”，步长 stepsize 为 30 000；3) 学习率的变化比率 gamma 为 0.1；4) 屏幕显示间隔 display 为 20；5) 最大迭代次数 max_iter 为 40 000；6) 动量 momentum 为 0.9；7) 权重衰减项 weight_decay 为 0.000 5．

在 Ubuntu 系统环境下，调用 GPU 时的目标检测算法的检测速度约为 0.047 秒/张．在 Windows 系统环境下，由于没有使用 GPU，目标检测速度约为 1.036 秒/张．目标检测实验结果如图 6(b)(e)(h) 所示．

本文将“hedgehog”、“polarbear”和“squirrel”归到“toy”一类，“bottle”为另一类．然而网络训练完成之后，模型检测结果显示，“toy”类的准确率明显低于“bottle”类．将“hedgehog”、“polarbear”和“squirrel”分开各为一类后，每类的检测准确率明显上升．原因是这 3 种物体的个体差异性较大，模型不容易学得较好的特征，“bottle”的个体差异性小，特征学习效果较好．

本文采用不同尺度大小的样本来验证模型的检测效果，以此判断检测模型对小目标的辨别能力．当样本图像中的目标尺寸较大时，检测效果相对较好；当目标尺寸较小时，检测效果相对较差，实验结果如图 7 所示．

经过分析，原因在于小目标的特征不太明显，模型不易进行特征提取．影响检测效果的另一个因素还有输入到特征提取网络的图像归一化尺度．当归一化尺度较小时，模型的检测效果较差；当归一化尺度较大时，模型的检测效果较好．原因是当输

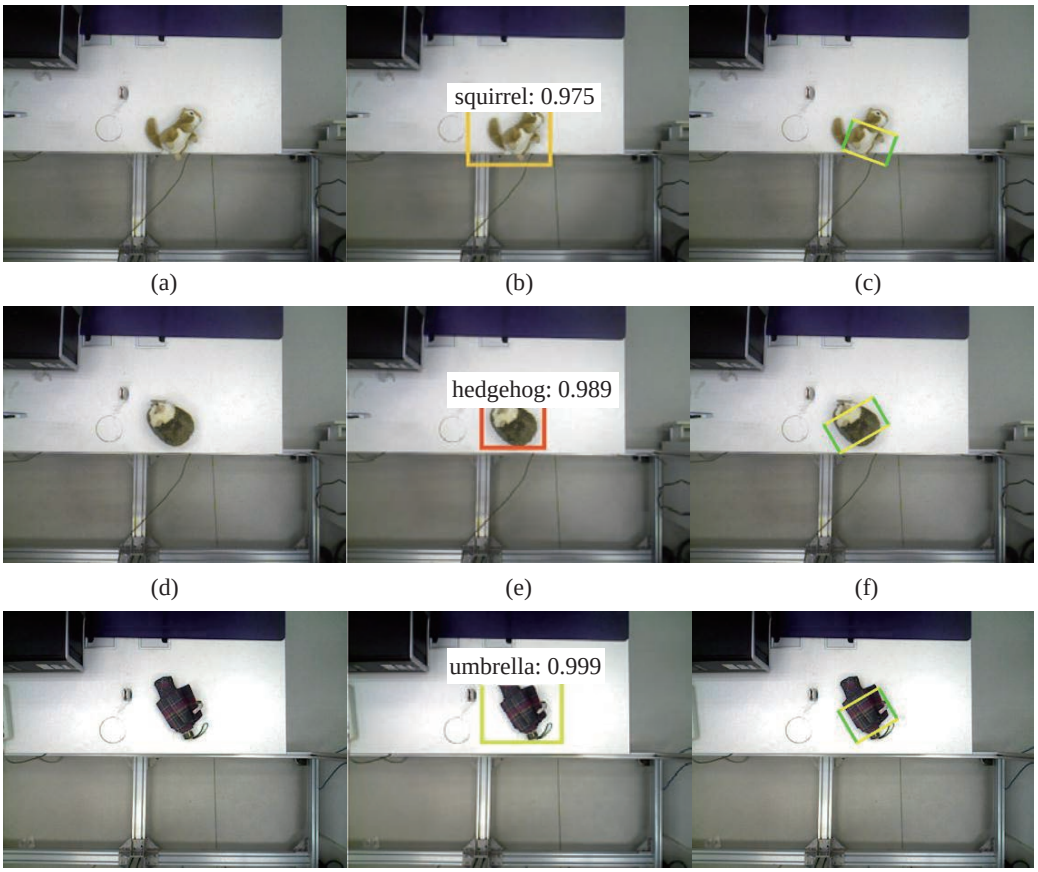


图 6　目标检测及最优抓取位置检测实验结果

Fig.6　Experimental results of the object detection and the optimal grasping position detection

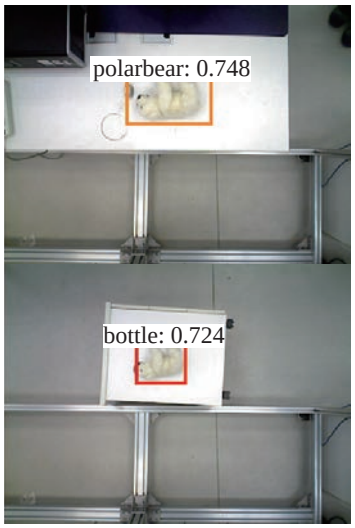


图 7　不同尺度下的目标检测结果

Fig.7　Detection results of the small object under different scales

入图像的归一化尺度变大时，图像中的目标尺寸也相应变大，因而使得目标特征变得较为明显，进而提高检测准确度．不同归一化尺度下的目标检测结果如图 8 所示．其中图 8(a) 为较小归一化尺度下的检测结果，图 8(b) 为较大归一化尺度下的检测结果．

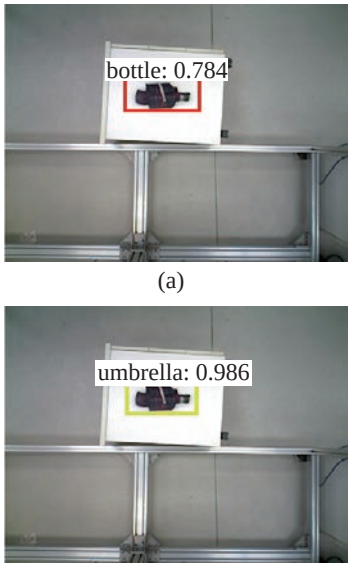


图 8　不同归一化尺度下的目标检测结果

Fig.8　Detection results at different normalized scales

在本文实验条件下，表 1 列出了不同目标检测

实验的准确率. 其中 1 表示默认条件下的实验组; 2 表示较小目标尺度组, 其对比组为 1; 3 表示较大归一化输入尺度组, 其对比组为 1; 4 表示较大归一化输入尺度组, 其对比组为 2.

表 1 目标检测实验对比结果

Tab.1 The comparison result of object detection experiments

类别	1 /%	2 /%	3 /%	4 /%
bottle	91.7	85.0	92.9	87.6
hedgehog	100	87.2	100	97.4
polarbear	77.6	40.9	83.6	70.5
squirrel	100	93.1	100	100
umbrella	98.1	71.0	79.6	90.3
box	75.5	–	81.1	–
scoop	98.8	–	97.7	–

4.1.2 最优抓取位置检测

最优抓取位置检测结果如图 6(c)(f)(i) 所示.

实验发现, 寻找最优抓取位置的搜索框起始大小、搜索框的数量以及搜索步长均会影响最终检测出的抓取框位置, 起始大小的影响最大, 其次是搜索框的数量, 最后是搜索步长. 对于同一图像中相同的目标物体, 不同的搜索框配置产生的结果如图 9 所示.

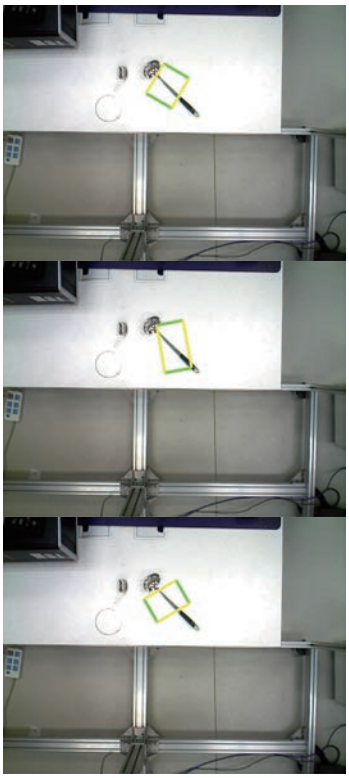


图 9 不同搜索框配置下的抓取检测结果

Fig.9 Grasping detection results under different search box configurations

由于每检测一张图像都要调整一次搜索框配置, 因而实际操作的大部分时间都会花在搜索框的参数选择上, 这导致了不同图像的最优位置检测时间出现明显差异. 在配置搜索框时, 事先并不知道物体的大小, 这就增加了配置难度. 搜索框的起始尺寸选择过小, 就会增大搜索时间; 起始尺寸过大以至于超过了候选抓取区域的尺寸, 就会导致检测失败. 针对这一点, 本文在文 [10] 算法的基础上进行了改进, 当模型第一步检测出包含目标的候选区域后, 根据候选区域的尺寸来确定搜索框的起始大小, 搜索框的起始宽、高分别为候选区域宽、高的一半左右. 改进后的检测算法在检测抓取位置时, 速度最快的为 4.4808 秒/张, 最慢的为 137.982 秒/张.

本文采用传统的位置检测方法与深度学习抓取检测方法作对比. 传统位置检测方法的大致步骤为首先获得包含目标信息的二值图, 其中目标物体的颜色为白, 背景的颜色为黑. 然后根据二值图求出目标物体的重心位置 (u,v)、与目标物体具有相同标准 2 阶中心矩的椭圆的长轴、短轴长度 l_1,l_2 及长轴与图像坐标系横轴的夹角 α. 最后由 (u,v)、l_1、l_2 和 α 即可求出抓取位置. 传统位置检测方法的检测结果如图 10 所示.

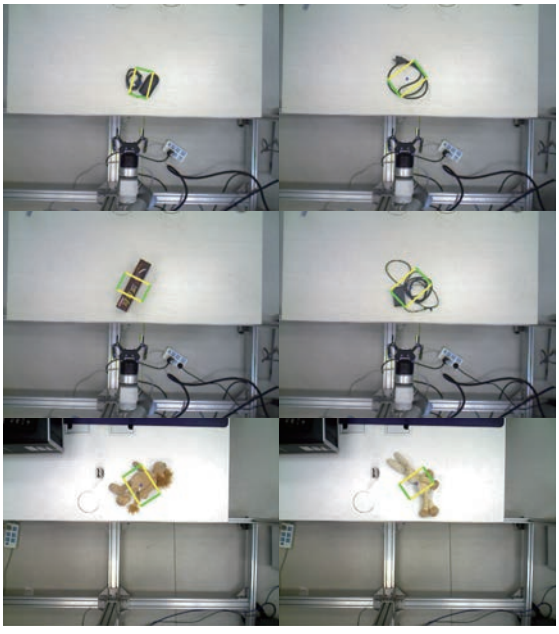
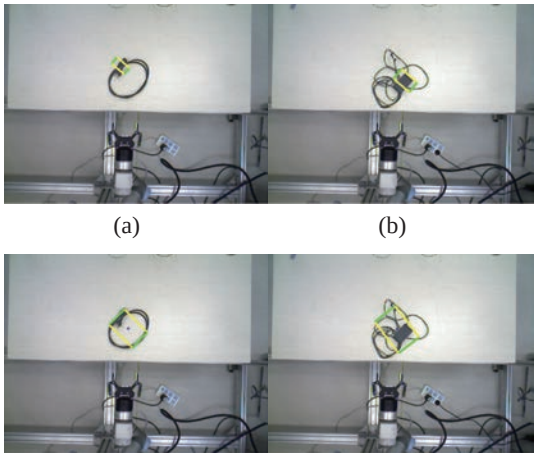


图 10 传统位置检测方法检测结果

Fig.10 The detection results of the traditional position detection method

对传统的位置检测方法进行分析, 发现其存在较大问题. 由于仅是通过使用重心和夹角 2 种数据作出抓取判断, 因此算法对重心位于目标物体上的

情形有较高的抓取率．但是受限于机械臂末端夹持器的大小及其开口角度，对于目标区域面积较大、重心区域较为平坦或整体不规则的情形，则不能进行抓取．而对于重心不在目标物体上的情形，当目标区域较集中、面积较小时，可用于目标抓取，否则会导致抓取失败．另外，因为每个目标只有一个重心，所以单个目标对应的抓取位置只有一个，较为单一．而对于深度学习抓取检测方法，模型先从人工标定的抓取位置中学得特征，之后才用于位置检测，因此目标区域面积对抓取检测的影响比传统方法要小．由于深度学习抓取检测方法不需要目标物体的重心，因而不论物体重心附近形状如何、重心是否在物体上，都对位置检测没有影响．只要目标区域中有符合的特征，都可被模型视为抓取位置，因此深度学习抓取检测可给出不同的抓取位置．深度学习抓取方法与传统位置检测方法的对比结果如图 11 所示，其中图 11(a)(b) 为深度学习方法的检测结果，图 11(c)(d) 为传统方法的检测结果．



(a)　(b)

(c)　(d)

图 11　深度学习抓取方法 (a, b) 与传统方法 (c, d) 对比结果

Fig.11　The comparison results between the deep learning method (a, b) and the conventional method (c, d)

假设搜索框的大小为 $m\times n$，搜索区域的大小为 $M\times N$，搜索步长为 s，则每个搜索框在搜索区域内寻找最优位置时大约要计算 $[(M-m)/s+1][(N-n)/s+1]$ 次．当搜索框的宽有 a 个取值，高有 b 个取值，则 $a\times b$ 个搜索框在搜索区域内进行一次搜索大约要计算 p_1 次，其中 p_1 的计算方法为

$$p_1=\sum_{i=1}^{a}\sum_{j=1}^{b}[(M-m_i)/s+1][(N-n_j)/s+1]$$

由于目标物体的摆放姿态不定，最优抓取框的角度也不定，因而需要搜索框从不同角度搜索最优位置．文 [10] 的做法是将搜索区域进行旋转，假设每进行一次搜索后搜索区域旋转 10°，旋转 18 次共转 180°．旋转 18 次时检测算法在搜索区域内大约要计算 p_2 次，p_2 的计算方法为

$$p_2=\sum_{k=1}^{18}\sum_{i=1}^{a}\sum_{j=1}^{b}[(M_k-m_i)/s+1][(N_k-n_j)/s+1]$$

而在传统的位置检测方法中，目标的重心只有一个，位置检测只进行一次．因此深度学习抓取检测方法的检测速度明显慢于传统的位置检测方法．

为了提高深度学习抓取方法的检测速度，本文在其基础上使用了与目标物体具有相同标准 2 阶中心矩的椭圆的长轴与图像坐标系横轴间的夹角 α 这一数据，预判断目标物体的走势并作为搜索区域旋转的依据．假设夹角 α 取 α_0 时，搜索区域从 $\pi/2-\alpha_0-\delta$ 角度位置开始旋转，其中 δ 为旋转间隔．此时，取搜索区域的旋转角度为 $\pi/2-\alpha_0-\delta$、$\pi/2-\alpha_0$、$\pi/2-\alpha_0+\delta$ 和 $\pi/2-\alpha_0+2\delta$ 这 4 个值，这种方法使得搜索次数由原来的 18 次变为 4 次，明显提高了算法的检测速度．由于检测是在牺牲潜在最优抓取位置的条件下进行的，因而对于一些目标物体，其最优位置可能被过滤掉．

4.2　机械臂抓取实验

本文中仿真实验获得的抓取数据仅是最优抓取框的重心在 2 维图像坐标系下的位置及抓取框在图像中的方位角，这些信息不足以使机械臂完成抓取任务．抓取动作执行前，必须向机械臂提供目标物体的抓取点在机械臂基坐标系下的 3 维位姿信息，进而机械臂才能成功抓取目标．

本文实验使用的图像采集设备为深度相机，相机被固定在工作台上方，且镜头平面基本平行于地面．使用深度相机可获得抓取框重心位置在深度相机下的深度信息，通过深度相机和机械臂的坐标关系，可将抓取框重心的 2 维坐标点及坐标点对应的深度信息转换为实际抓取点在机械臂基坐标系下的 3 维位置信息．本文假设在待抓取目标的 3 维姿态中，目标物体的 Z 轴都垂直于地面，X、Y 轴的方向由抓取框的长和宽所在的直线确定，由此即可获得目标物体的 3 维姿态信息．

抓取实验基于 6 自由度人机协作型工业机械臂 UR5，机械臂实物图如图 12 所示．机械臂包含 6 个关节，图 12 中的①～⑥分别为机座关节、肩关节、肘关节、腕关节 1、腕关节 2 和腕关节 3；⑦为末端夹持器，采用 Robotiq 二指夹手．基座是机械臂的安装位置，夹持器安装在腕关节 3 上．机械臂可

实现在 850 mm 可达范围内随意移动.

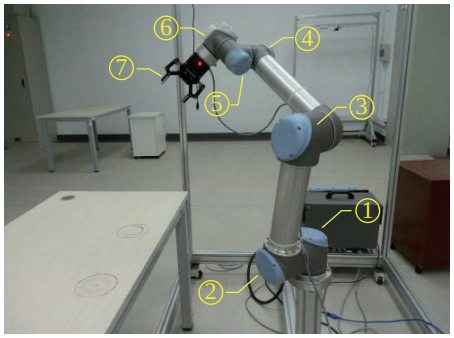


图 12 UR5 机械臂实物图

Fig.12 The appearance of the UR5 robot

实验前，深度相机与机械臂的坐标关系已标定，即可获得实际抓取点在机械臂基坐标系下的 3 维位置信息. 首先，计算抓取框重心的像素坐标 (u,v) 在深度相机坐标系下对应的坐标 (x_c,y_c,z_c). 坐标转换为

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = z_c \boldsymbol{M}_{in}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \boldsymbol{M}_{in} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

其中 $\boldsymbol{M}_{in}$ 为深度相机的内参数，z_c 为坐标点 (u,v) 对应的深度值. 其次，计算深度相机坐标系下的坐标 (x_c,y_c,z_c) 在机械臂基坐标系下对应的坐标 (x,y,z). 坐标转换为

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \boldsymbol{T}_m \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

其中 $\boldsymbol{T}_m$ 为标定得到的手眼关系矩阵. (x,y,z) 即为目标物体在机械臂基坐标系下的位置.

抓取数据转换完成后，即可控制机械臂执行抓取任务，目标物体的抓取过程如下：

1) 初始化机械臂位姿. 假设机械臂在笛卡儿空间下的初始位姿为 $(x_0,y_0,z_0,rx_0,ry_0,rz_0)$，腕关节 3 初始夹角为 θ_0.

2) 控制腕关节 3 转动 θ，使末端夹持器转至抓取任务要求的姿态.

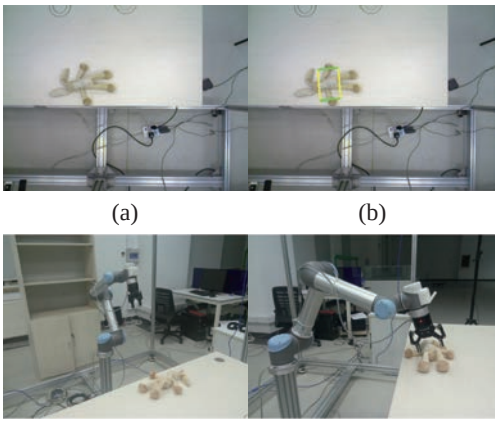
3) 移动机械臂，使其末端从初始位置 (x_0,y_0,z_0) 平移至目标位置 (x,y,z).

4) 控制末端夹持器闭合，将目标物体抓起.

5) 控制机械臂将目标物体放置在指定位置，并返回初始姿态.

6) 若要进行连续抓取，则返回第 1) 步；否则，结束抓取任务.

UR5 机械臂抓取实验结果如图 13 所示. 其中图 13(a) 为深度相机获取的原始图像，图 13(b) 为算法检测出的抓取框位置，图 13(c) 为机械臂的初始位姿，图 13(d) 为机械臂平移到目标位置的位姿.



(a) (b)

(c) (d)

图 13 UR5 机械臂抓取实验结果

Fig.13 The grasping experiment result of UR5 robot

5 结论（Conclusion）

本文基于深度学习实现了对不同种类、不同尺寸的物体分类并抓取. 通过仿真实验验证了本文方法能对图像中的目标物体进行较为准确的分类. 在 UR5 机械臂上进行了抓取实验，实验结果证明了目标检测方法和抓取位置检测方法的有效性.

未来的研究方向是进一步优化本文方法，使得分类抓取方法更为连贯，能够将 2 种网络模型更加有效地结合成一个整体，最终同时进行目标检测和最优抓取位置检测.

参考文献（References）

[1] Maitin-Shepard J, Cusumano-Towner M, Lei J, et al. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2010: 2308-2315.

[2] Ramisa A, Alenya G, Moreno-Noguer F, et al. Using depth and appearance features for informed robot grasping of highly wrinkled clothes[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2012: 1703-1708.

[3] Jiang Y, Moseson S, Saxena A. Efficient grasping from RGB-D images: Learning using a new rectangle representation[C]// IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2011: 3304-3311.

[4] Lin Y, Sun Y. Robot grasp planning based on demonstrated grasp strategies[J]. International Journal of Robotics Research, 2015, 34(1): 26-42.

（下转第 837 页）

[12] 张丹凤．基于能量平衡的蛇形机器人被动蜿蜒步态研究[D]．沈阳：中国科学院沈阳自动化研究所，2015.
Zhang D F. Study on the passive creeping of a snake-like robot based on energy balance[D]. Shenyang: Shenyang Institute of Automation, Chinese Academy of Sciences, 2015.

[13] 郭宪，马书根，李斌，等．基于动力学与控制统一模型的蛇形机器人速度跟踪控制方法研究 [J]．自动化学报，2015，41(11)：1847-1856.
Guo X, Ma S G, Li B, et al. Velocity tracking control of a snake-like robot with a dynamics and control unified model[J]. Acta Automatica Sinica, 2015, 41(11): 1847-1856.

[14] Saito M, Fukaya M, Iwasaki T. Serpentine locomotion with robotic snakes[J]. IEEE Control Systems Magazine, 2002, 22(1): 64-81.

[15] Date H, Takita Y. Control of 3D snake-like locomotive mechanism based on continuum modeling[C]//ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. New York, USA: ASME, 2005: 1351-1359.

[16] Liljeback P, Pettersen K Y, Stavdahl O, et al. Fundamental properties of snake robot locomotion[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2010: 2876-2883.

[17] Guo Z V, Mahadevan L. Limbless undulatory propulsion on land[J]. Proceedings of the National Academy of Sciences of the United States of America, 2008, 105(9): 3179-3184.

[18] Watanabe K, Iwase M, Hatakeyama S, et al. Control strategy for a snake-like robot based on constraint force and verification by experiment[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2008: 1618-1623.

[19] Yamada H, Hirose S. Steering of pedal wave of a snake-like robot by superposition of curvatures[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2010: 419-424.

[20] Ma S G. Analysis of creeping locomotion of a snake-like robot [J]. Advanced Robotics, 2001, 15(2): 205-224.

作者简介：

张丹凤（1984 –），女，博士，讲师．研究领域：仿生机器人，机器人控制方法与路径规划．

李　斌（1963 –），男，硕士，研究员．研究领域：仿生机器人，移动机器人，机器人控制．

王立岩（1980 –），女，博士，讲师．研究领域：优化算法．

（上接第 828 页）

[5] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[6] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[M]//Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2012: 1097-1105.

[7] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[8] Varley J, Weisz J, Weiss J, et al. Generating multi-fingered robotic grasps via deep learning[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2015: 4415-4420.

[9] Johns E, Leutenegger S, Davison A J. Deep learning a grasp function for grasping under gripper pose uncertainty[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2016: 4461-4468.

[10] Lenz I, Lee H, Saxena A. Deep learning for detecting robotic grasps[J]. International Journal of Robotics Research, 2015, 34(4/5): 705-724.

[11] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [M]//Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2015: 91-99.

[12] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2014: 580-587.

[13] Uijlings J R R, van de Sande K E A, Gevers T, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154-171.

[14] Girshick R. Fast R-CNN[C]//IEEE International Conference on Computer Vision. Piscataway, USA: IEEE, 2015: 1440-1448.

[15] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]//13th European Conference on Computer Vision. Berlin, German: Springer, 2014: 818-833.

[16] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[A/OL]. (2015-04-10) [2017-04-18]. https://arxiv.org/abs/1409.1556v6.

作者简介：

杜学丹（1991 –），女，硕士生．研究领域：控制工程，计算机视觉．

蔡莹皓（1983 –），女，博士后，副研究员．研究领域：机器学习，模式识别，计算机视觉．

鲁　涛（1979 –），男，博士，副研究员．研究领域：智能机器人，机器学习，机器人控制．