计算机科学
COMPUTER SCIENCE

基于 YOLOv4 的目标检测知识蒸馏算法研究

楚玉春，龚航，王学芳，刘培顺

引用本文

楚玉春，龚航，王学芳，刘培顺. 基于YOLOv4的目标检测知识蒸馏算法研究[J]. 计算机科学, 2022, 49(6A): 337-344.

CHU Yu-chun, GONG Hang, Wang Xue-fang, LIU Pei-shun. Study on Knowledge Distillation of Target Detection Algorithm Based on YOLOv4[J]. Computer Science, 2022, 49(6A): 337-344.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

指静脉识别技术研究综述

Survey on Finger Vein Recognition Research

计算机科学, 2022, 49(6A): 1-11. https://doi.org/10.11896/jsjkx.210400056

基于多尺度特征的脑肿瘤分割算法

Brain Tumor Segmentation Algorithm Based on Multi-scale Features

计算机科学, 2022, 49(6A): 12-16. https://doi.org/10.11896/jsjkx.210700217

基于 Transformer 和 LSTM 的药物相互作用预测

Drug-Drug Interaction Prediction Based on Transformer and LSTM

计算机科学, 2022, 49(6A): 17-21. https://doi.org/10.11896/jsjkx.210400150

基于深度学习的黑色素瘤智能诊断多模型算法

Multi Model Algorithm for Intelligent Diagnosis of Melanoma Based on Deep Learning

计算机科学, 2022, 49(6A): 22-26. https://doi.org/10.11896/jsjkx.210500197

基于注意力机制和多任务学习的阿尔茨海默症分类

Alzheimer's Disease Classification Method Based on Attention Mechanism and Multi-task Learning

计算机科学, 2022, 49(6A): 60-65. https://doi.org/10.11896/jsjkx.201200072

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210600204

基于 YOLOv4 的目标检测知识蒸馏算法研究

楚玉春[1]　龚　航[1]　王学芳[2]　刘培顺[1]
1 中国海洋大学计算机科学与技术学院　山东 青岛 266100
2 中国海洋大学数学科学学院　山东 青岛 266100
(540857379@qq.com)

摘　要　知识蒸馏作为一种基于教师-学生网络思想的训练方法,它通过复杂的教师网络来引导网络结构相对简单的学生网络进行训练,使得学生网络获得与教师网络相媲美的精度。知识蒸馏在自然语言处理和图像分类领域均有广泛的研究,而在目标检测领域的研究则相对较少且实验效果有待提升。目标检测的蒸馏算法主要是在特征提取层进行,而单一的特征提取层的蒸馏方式易导致学生不能充分学习教师网络知识,使得模型的精度较差。针对上述问题,通过在特征提取和目标分类与边框预测上都利用了教师网络的"知识"来指导学生网络进行训练,并提出了一种基于多尺度注意力机制的蒸馏算法,使得教师网络的知识更好地流向学生网络。实验分析表明,在 YOLOv4 基础上提出的蒸馏算法可有效地提高学生网络的检测精度。

关键词:知识蒸馏;深度学习;模型压缩;YOLOv4;注意力机制
中图法分类号　TP391

Study on Knowledge Distillation of Target Detection Algorithm Based on YOLOv4

CHU Yu-chun[1],GONG Hang[1],Wang Xue-fang[2] and LIU Pei-shun[1]
1 School of Computer Science and Technology,Ocean University of China,Qingdao,Shangdong 266100,China
2 School of Mathematical Sciences,Ocean University of China,Qingdao,Shangdong 266100,China

Abstract　Knowledge distillation,as a training method based on the teacher-student network,guides the relatively simple student network to be trained through the complex teacher network,so that the student network can obtain the same precision as the teacher network. It has been widely studied in the field of natural language processing and image classification,while the research in the field of object detection is relatively less,and the experimental effect needs to be improved. The Distillation Algorithm of object detection is mainly carried out in the feature extraction layer,and the distillation method of single feature extraction layer will cause students can't learn the teacher's network knowledge fully,which makes the accuracy of the model poorly. In view of the above problem,this paper uses the "knowledge" in feature extraction,target classification and border prediction of teacher network to guide student network to be trained,and proposes a multi-scale attention Distillation Algorithm to make the knowledge of teacher network influence student network. Experimental results show that the distillation algorithm proposed in this paper based on YOLOv4 can effectively improve the detection accuracy of the original student network.

Keywords　Knowledge distillation,Deep learning,Model compression,YOLOv4,Attention mechanism

1　引言

目标检测技术是数字图像处理和计算机视觉的热门研究方向,Girshick 等于 2014 年首次提出 R-CNN[1] 二阶段检测模型,用于目标检测,紧接着受到 SPPnet[2] 启发,在 R-CNN 基础上引入 ROI 层,同年提出了 Fast R-CNN[3] 检测模型,与 R-CNN 相比,Fast R-CNN 保留了模型的精度且大大缩短了训练时间。Ren 等于 2016 年将选择性搜索算法替换为 RPN 候选生成框算法,提出了 Faster R-CNN[4] 模型。针对单个高层特征对小物体检测能力不足的问题,Lin 等提出使用特征金字塔(FPN[5])。为了提高检测精度,解决单纯提高 IOU 阈值带来的过拟合和误匹配问题,Cai 等使用 cascade 回归作为复杂采样机制,提出了 Cascade R-CNN[6] 结构。2016 年 Redmon 等提出了 YOLO 一阶段检测模型,将预测框的位置信息和所属类别直接在输出层输出,大大提高了预测速度;在此基础上,又引入批归一化以及先验框机制,并用聚类算法提取先验框尺度,提出了 YOLOv2[7] 算法,接着使用更强的特征提取网络并利用多尺度特征进行检测提出了 YOLOv3[8]。Bochkovskiy 等在特征提取网络中融入 CSP 结构,并将 PAN 结构使用在特征融合层,提出了 YOLOv4[9]。Liu 等提出了 SDD[10] 模型,通过在检测阶段引入不同尺度的卷积特征图,使得对小目标的检测更加精确。一阶段的目标检测往往存在正负样本比例失衡严重的问题,为了解决上述问题,Lin 等通过引入损失函数 Focalloss,提出了 RetinaNet[11] 结构。

目标检测速度和精度的不断提高,为工业级的应用提供了可能,为了使其能够在移动设备上平稳地运行,需要更加轻量级的目标检测方法。知识蒸馏作为一种深度学习模型压缩方法开始被越来越多的人研究。

知识蒸馏的本质就是通过神经网络迁移学习训练的方式,将复杂网络学到的"知识"转移到简单网络中。2006 年,

通信作者:王学芳(wangxuefang@ouc.edu.cn)

Bucila 等[12]首次利用知识迁移的方式来压缩网络模型，通过集成强分类器标注的伪数据去训练一个压缩模型，再现了原始复杂网络的输出结果。随后，Hinton 等[13]在已有研究的基础上提出了知识蒸馏(Knowledge Distillation，KD)算法的概念，该算法将教师模型和学生模型共同的输出作为目标函数，引入一个温度参数 T 用于引导学生模型的训练，实现了知识从教师模型到学生模型的迁移，但该方法并没有对学生网络的中间隐藏层进行约束。2016 年，Zagoruyko 等[14]对知识蒸馏算法进行了改进，提出将注意力机制引入知识蒸馏，并利用教师模型生成的注意力特征图去引导学生模型的训练，有效地提高了学生网络模型的分类准确度，但考虑到目标检测中的多尺度问题，直接将该方法应用于目标检测任务时检测效果不佳。2019 年，Hou 等[15]在 Zagoruyko 等的研究基础上，提出了一种自注意力蒸馏算法，通过引导浅层网络学习深层网络的特征表达来加强学生网络的特征提取能力。2019 年，He 等[16]提出了一种基于知识蒸馏的高效语义分割算法，在不改变网络结构的前提下，使学生模型的性能提升了 2%。Wang 等[17]提出了一种细粒度的特征模仿方法，通过利用特征响应的跨位置差异来实现知识的蒸馏。2020 年，Gao 等[18]提出了一种残差知识蒸馏算法，通过引入一个学习教师和学生网络之间残差的助理网络，来帮助学生网络从教师网络中获取更多的知识，但该算法设计过于复杂，在训练过程中会占用较大的显存空间。Wang 等[19]通过将教师和学生网络生成的特征图分别作为真假样本来进行对抗训练，以提高学生网络的检测性能。2021 年 Chawla 等[20]针对目标检测提出了一种无数据的知识蒸馏方法。Dai 等[21]充分利用了特征、关系以及基于反应的知识，提出了通用的实例选择模块。

为了进一步提升蒸馏算法在一阶段目标检测的效果，本文以 YOLOv4 作为教师网络，提出了一种多尺度注意力目标识别蒸馏算法。本文的贡献如下：1)提出多尺度的注意力蒸馏算法，指导学生网络的输出使其接近于教师网络的输出，进一步对学生网络生成的特征图进行指导训练，使其能够学习到与教师网络相似的、不同尺度的特征编码方式，从而提高学生网络的检测性能；2)在特征提取层、目标分类与边框预测上都利用教师网络的“知识”指导学生网络训练。

2 知识蒸馏算法

2.1 知识蒸馏的概念

知识蒸馏是一种模型压缩算法，最先应用于分类网络模型。该算法首先训练一个复杂度高、特征提取能力强的教师网络，然后将该网络的输出作为软目标监督信息，用于引导一个网络结构更为简单、推理速度更快的学生网络的训练，从而实现教师网络的“知识”向学生网络的迁移。在传统的目标分类或检测任务中，通常采用“0”和“1”硬标签方式标注数据，但硬标签包含的信息量有限，没有包含类与类之间的关系。与其相比，知识蒸馏采用了教师网络的输出概率(取值介于 0 和 1 之间)来标注数据，可以将不同类别之间的相似性很好地表示出来。

为了获取更加“软”的分类输出，知识蒸馏算法在 Softmax 中引入了一个温度系数 T 来软化目标的类别概率，其计算式如式(1)所示：

$$q_i=\frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

其中，z_i 为神经网络输出层的内容。

温度系数 T 的取值大小直接关系到神经网络 Softmax 输出结果的分布情况，T 取值越大，输出的目标类别概率分布就越平滑，标签的软化程度也就越高；而 T 取值越小，输出的错误分类概率就越大，模型的训练就越困难。

图 1 为知识蒸馏算法的流程图。对于同一个输入向量，算法首先计算教师网络模型的预测输出，然后将其输出除以温度 T，经过 Softmax 变换后得到一个软化的概率分布(软目标)。与硬目标相比，该软目标隐藏了不同目标类别之间的关系信息。接着，计算学生网络模型的预测输出，将其除以温度 T，经 Softmax 变换后，再与软目标计算交叉熵损失。最后计算学生网络模型的预测输出与硬目标之间的交叉熵损失。因此，训练一个学生网络，知识蒸馏算法的整体损失包含了两部分，其计算式如下：

$$L_{\text{total}}=\alpha L_{\text{soft}}(p_s,p_t)+(1-\alpha)L_{\text{hard}}(p_s,y) \tag{2}$$

$$L_{\text{soft}}(p_s,p_t)=-\sum p_s\log p_t \tag{3}$$

$$L_{\text{hard}}(p_s,y)=-\sum p_s\log y \tag{4}$$

其中，p_s 和 p_t 分别为学生网络模型和教师网络模型的预测输出，y 为真实标签，α 为加权系数。在训练过程中，α 可用于调整软目标和硬目标对整体损失的贡献，前期适当加大 α 的值，可以提高教师网络模型输出的贡献，使学生网络快速掌握教师网络的“知识”，增强学生网络识别简单样本的能力。而在训练后期，为了提高学生网络识别困难样本的能力，需要结合真实标注去帮助学生网络模型进行识别，因此可以适当减小 α 的值。教师网络的分类或检测能力通常优于学生网络，而且教师网络的分类或检测精度越高，就越有利于学生网络的学习。

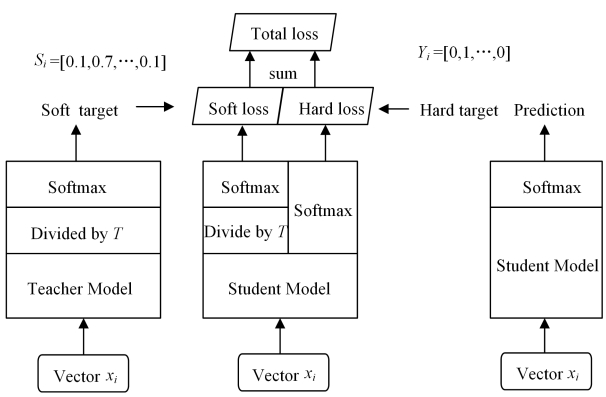


图 1 知识蒸馏算法的流程

Fig. 1 Flow chart of knowledge distillation algorithm

2.2 教师网络和学生网络设计

针对现阶段的研究成果，本文选取表现效果较好的 YOLOv4 目标检测算法作为教师网络，并在此基础上实现蒸馏算法的设计。

相比教师网络，学生网络是一个参数更少、计算速度更快的网络，其总体网络结构与教师网络相似，我们将基础的学生网络宽度设计为教师网络宽度的 1/4，即将 YOLOv4 网络每层卷积层的通道数减少至原来的 1/4。在此基础上分别进行教师网络、学生网络的蒸馏算法设计。图 2 为教师网络和学生网络的特征提取网络对比图。

Layer	Teacher:YOLOv4	Student:¼ YOLOv4
Conv1	32, 3×3, stride 1	8, 3×3, stride 1
DownSample1	64, 3×3, stride 2	16, 3×3, stride 2
CSPResNet1	CSP 32, 1×1, stride 1 64, 3×3, stride 1 Residual ×1	CSP 8, 1×1, stride 1 16, 3×3, stride 1 Residual ×1
DownSample2	128, 3×3, stride 2	32, 3×3, stride 2
CSPResNet2	CSP 64, 1×1, stride 1 64, 3×3, stride 1 Residual ×2	CSP 8, 1×1, stride 1 8, 3×3, stride 1 Residual ×2
DownSample3	256, 3×3, stride 2	64, 3×3, stride 2
CSPResNet3	CSP 128, 1×1, stride 1 128, 3×3, stride 1 Residual ×8	CSP 32, 1×1, stride 1 32, 3×3, stride 1 Residual ×8
DownSample4	512, 3×3, stride 2	128, 3×3, stride 2
CSPResNet4	CSP 256, 1×1, stride 1 256, 3×3, stride 1 Residual ×8	CSP 64, 1×1, stride 1 64 3×3, stride 1 Residual ×8
DownSample5	1 024, 3×3, stride 2	256, 3×3, stride 2
CSPResNet5	CSP 512, 1×1, stride 1 512, 3×3, stride 1 Residual ×4	CSP 128, 1×1, stride 1 128, 3×3, stride 1 Residual ×4

图 2 师生特征提取网络对比图

Fig. 2 Network comparison chart of feature extraction between teachers and students

3 目标检测知识蒸馏算法设计

在 YOLOv4 目标检测算法中，网络的特征提取能力直接关系到最终算法模型的分类和定位能力，将特征子网络融合可以获得更好的深度特征信息[22]。因此，在训练过程中，本文在特征提取和目标分类与边框预测上都利用了教师网络的“知识”来指导学生网络训练。

本文提出的知识蒸馏 YOLOv4 检测算法的训练流程如图 3 所示。首先在网络特征提取阶段，为了使教师网络学习到的“知识”能更好地流向学生网络，本文提出了一个多尺度像素注意模块（Multi-scale Pixel Attention Module，mPAM）来构造特征提取网络的特征注意力图，并利用了最小均方误差损失（L2 损失）函数来计算教师网络和学生网络特征图之间的损失。然后，在目标分类和定位阶段，本文沿用了传统的知识蒸馏算法来引导学生网络训练一个强大的目标分类回归器，在计算软损失时，本文将教师网络的编码层输出作为软目标与学生网络的编码层输出进行计算。

本文提出的知识蒸馏算法的损失函数由真实标注作为硬目标产生的硬损失和教师网络输出作为软目标产生的软损失组成，具体表达式如下：

$$L_{\text{total}}=\alpha L_{\text{soft}}t+(1-\alpha)L_{\text{hard}}(p_s,y) \tag{5}$$

$$L_{\text{soft}}=L_{\text{output}}(p_s,p_t)+\lambda L_{\text{pam}}(A_s,A_t) \tag{6}$$

$$L_{\text{output}}(p_s,p_t)=\beta L_{cls}+\gamma L_{\text{box}} \tag{7}$$

其中，$L_{\text{hard}}(p_s,y)$为硬损失，L_{soft}为软损失，它由网络的输出损失 $L_{\text{soft}}(p_s,p_t)$与特征注意力损失 $L_{pam}(A_s,A_t)$组成，其中 $L_{\text{output}}(p_s,p_t)$又由目标分类损失 L_{cls}和边框回归损失 L_{box}组成。

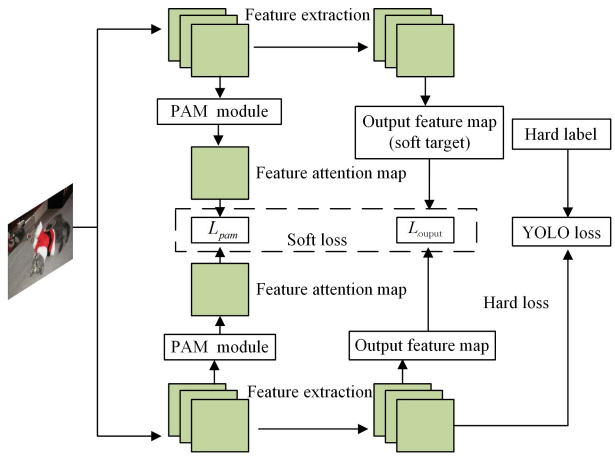


图 3 YOLOv4 算法的蒸馏学习

Fig. 3 Distillation learning of YOLOv4 algorithm

3.1 基于注意力特征图的蒸馏学习

YOLOv4 目标检测算法采用了多尺度的预测方法，在不同尺度的特征图上分别预测目标的类别和边框位置，因此学生网络学习教师网络的特征编码方式对提高学生网络的性能有着重要的帮助。在计算教师网络与学生网络特征图之间的距离时，Romero 等[23]采用了最小均方误差损失函数进行计算，其计算式如式(8)所示：

$$\boldsymbol{L}(\boldsymbol{V}_s,\boldsymbol{V}_t)=\|\boldsymbol{V}_s-\boldsymbol{V}_t\|^2 \tag{8}$$

其中，$\boldsymbol{V}_s$ 表示学生网络特征图向量，$\boldsymbol{V}_t$ 表示教师网络特征图向量。$\boldsymbol{V}_s$ 和 $\boldsymbol{V}_t$ 具有相同的尺寸和通道数，因此在特征图向量距离的计算上，Romero 等是采用逐像素计算方式进行计算的。本文设计的教师网络和学生网络卷积层通道数各不相同，上述方法并不适合本文学生网络特征图的蒸馏学习，而且随着网络层级的加深，输出特征图向量的通道数逐渐增多，逐像素计算特征图之间的距离会带来较大的计算量。

注意力机制是目标检测的一个重要研究方向，它可以使模型更加关注于目标的重要特征信息，提高模型预测结果的准确性。近年来，一些基于注意力机制的神经网络的应用也证明了注意力的有效性，可用于提升网络的性能。因此，学生网络学习教师网络的特征注意力，可以有效地提高学生网络的特征提取能力。

为了构造这样一个注意力特征图，文献[14]提出了一种基于激活函数值的注意力迁移方式，通过计算激活函数输出的绝对值统计量来衡量神经元对输入的关注程度。该文设计了 3 种不同的方案来生成空间注意力特征图。

(1)计算特征图向量在维度 C 上的绝对值和。

$$F_{sum}^{p}(\boldsymbol{V})=\sum_{i=1}^{c}|\boldsymbol{V}_i| \tag{9}$$

(2)计算特征图向量在维度 C 上的绝对值的 P 次方和。

$$F_{sum}^{p}(V)=\sum_{i=1}^{c}|V_i|^p \tag{10}$$

(3)计算特征图向量在维度 C 上的绝对值的 P 次方最大值。

$$F_{\max}^{p}(V)=\underset{i=1,c}{\text{Max}}|V_i|^p \tag{11}$$

上述公式中取绝对值的操作也是逐像素计算的，但与直接计算整个特征图向量的 L2 距离方式相比，该方法的计算量更小。

在上述研究基础上，本文借鉴了 YOLOv3-SPP 算法的

SPP 模块设计思想，提出了一种基于多尺度像素注意力的特征图蒸馏学习算法，通过构建一个特征映射函数，将教师网络输出的三维特征张量转换成一个二维的特征注意力图，用于指导学生特征提取网络的训练。

本文提出的多尺度像素注意力模块（Multi-scale Pixel Attention Module，mPAM）的计算流程如图 4 所示。

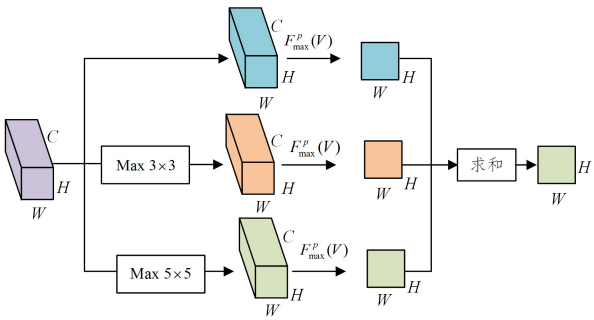


图 4 多尺度像素注意力模块计算流程图

Fig. 4 Calculation flow chart of multi scale pixel attention module

首先使用 1×1，3×3 和 5×5 这 3 种不同尺度的最大池化，分别获取特征图向量 $\boldsymbol{R}^{W\times H\times C}$ 对应尺度下的特征；然后分别计算 3 个特征图向量在维度 C 上的绝对值的 P（本文选取了 $P=1$）次方最大值，得到输入特征图向量 $\boldsymbol{R}^{W\times H\times C}$ 不同尺度的注意力特征图；最后将 3 个特征注意力图对应位置的像素相加，得到一个单通道的多尺度注意力特征图。该注意力特征图 $\boldsymbol{R}^{W\times H}$ 不仅反映了特征图向量和图像中预测物体的空间关系，而且增强了简单样本的特征，这也证明了注意力机制在目标检测中的重要性。

mPAM 模块采用了 3 种不同尺度的最大池化对输入特征图进行池化操作，使最终得到的多尺度注意力特征图上的每个像素点都可以综合其周围某个大小区域内的响应值，可以表示以该像素点为中心，在其周围某个大小区域内包含目标的概率。由于检测目标的多样性以及物体大小的不同，本文使用了 3 种不同尺度的最大池化来覆盖不同大小的区域。实验结果证明，这个改进能有效地提高学生网络模型的平均准确率。

在获取教师网络和学生网络多尺度的注意力特征图之后，本文采用了最小均方误差损失函数来计算两者之间的 L2 距离，计算式如式(12)所示：

$$L_{\text{pam}}(A_s, A_t)=\frac{1}{w\times h}\times \| A_s-A_t \|^2 \tag{12}$$

其中，w 和 h 分别表示特征图的宽和高，A_t 表示教师网络的注意力特征图，A_s 表示学生网络的注意力特征图。由于本文以 YOLOv4 网络为基础，可以获取 5 种不同尺度的特征图，并且学生网络与教师网络结构相同。因此，在知识蒸馏算法设计中，本文将基于多尺度像素注意力的损失模块分别插入 5 组 CSP 残差模块结束的位置，使学生网络不同层级的卷积网络都能学习到与教师网络相似的特征提取能力。如图 5 所示，本文一共插入了 5 个注意力模块将教师网络与学生网络连接起来，涵盖了浅层到深层 5 种不同层级的特征。

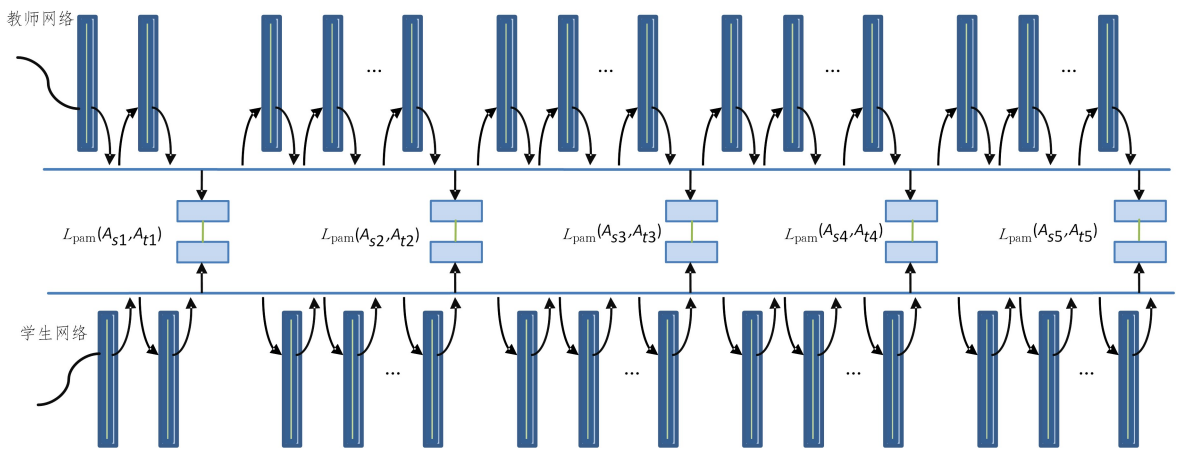


图 5 注意力特征图蒸馏学习

Fig. 5 Distillation learning of attention characteristics

3.2 分类与回归蒸馏学习

基于 YOLO 系列的单阶段目标检测算法，将目标分类和边框回归问题相结合，使用同一个网络进行训练。

对于分类蒸馏学习，本文将教师网络 YOLOv4 的输出概率分布作为软标签用于指导学生网络的训练，使学生网络的输出概率分布尽可能地接近于教师网络。为了软化网络的输出内容，使其包含更多的信息，本文在师生网络的输出层中引入温度参数 T。在训练学生网络时，可以将网络的输出层内容除以一个较高的温度 T，使输出结果的分布更加缓和，其中训练所用的分类蒸馏损失函数如下：

$$C_t=\text{softmax}(z_t/T) \tag{13}$$

$$C_s=\text{softmax}(z_s/T) \tag{14}$$

$$L_{cls}=T^2*KLdiv(C_s^t, C_t^T) \tag{15}$$

其中，z_t 和 z_s 分别表示教师网络和学生网络的分类输出。通过上述计算公式，可以得到软化的标签 C_t 和 C_s。在计算分类损失上，本文借鉴了文献[24]的计算方式，采用了 KL 散度来度量教师网络与学生网络分类输出之间的差异，而且 KL 散度越小，两者输出的概率分布越相似。

对于回归蒸馏学习，为了提高学生网络目标定位的精度，本文对学生网络的学习对象做了一些调整，先分别计算出教师网络和学生网络预测框与真实框之间的 L2 距离，然后根据两者距离的大小选择学生网络的学习对象。当学生网络预测框与真实框距离更大时，则让学生网络预测框向真实框学习，而不是向教师网络输出学习。学生网络训练所用的回归蒸馏损失函数如下：

$$L2_t-dis=\| b_t-b_{\text{truth}} \|^2 \tag{16}$$

$$L2_s-dis=\|b_s-b_{truth}\|^2 \tag{17}$$

$$L_{box}=\begin{cases}L2_s, & L2_s-dis>L2_t-dis\\ L2_t, & L2_t-dis>L2_s-dis\end{cases} \tag{18}$$

其中，b_t 和 b_s 分别为教师网络和学生网络预测边框坐标，b_{truth} 为真实框坐标，$L2_t-dis$ 为教师网络预测框与真实框的距离，$L2_s-dis$ 为学生网络预测框与真实框的距离，而且 L_{box} 输出越小，学生网络边框回归位置越准确。

4 实验与结果分析

4.1 实验数据集和评价指标

本文使用的数据集为 VOC 数据集[25]，数据集共分为 20 类，训练集选取了 VOC2007 和 VOC2012 的训练集和验证集的融合数据，共计 16551 张图片，测试集选取了 VOC2007 中的测试集，共计 4952 张图片。

本文采用的具体评价指标包括分类准确率（Precision，P）、召回率（Recall，R）、F1 分数（F1-score）、平均准确率（mean Average Precision，mAP@0.5）、模型参数量大小（Params）以及单张图片的平均检测时间（Mean Time to Detect，MT），其中：

$$Precision=\frac{TP}{TP+FP} \tag{19}$$

$$Recall=\frac{TP}{TP+FN} \tag{20}$$

$$F1\text{-}score=\frac{2*Precision*Recall}{Precision+Recall} \tag{21}$$

其中，TP（True Positive，TP）表示正样本被正确预测的个数，FP（False Positive，FP）表示负样本被错误预测的个数，FN（False Negative，FN）表示正样本被错误预测的个数。

4.2 实验环境和训练细节

本文知识蒸馏算法是在开源的 PyTorch1.7.1 框架上实现的，训练所用的计算机 CPU 配置为 Intel xeon E5-2630 v4，GPU 型号为 TITAN xp，显存为 12 GB，搭配的操作系统为 Ubuntu 16.04。

表 1 实验环境

Table 1 Experimental environment

深度学习框架	PyTorch1.7.1
CPU	Intel xeon E5-2630 v4
GPU	TITAN xp，显存为 12GB
操作系统	Ubuntu 16.04

在训练阶段，本文设置的相关实验参数如表 2 所列，其中训练所采用的学习率为余弦学习率，具体的计算式如式（22）所示：

$$LR=lr\times\left(\left(1+\cos\left(epoch\times\frac{\pi}{epoch}\right)\right)\times0.95+0.05\right) \tag{22}$$

其中，初始 lr 为 0.01，$epoch$ 为当前迭代代数，$epochs$ 为总的迭代代数。

表 2 相关实验参数

Table 2 Relevant experimental parameters

参数名	参数值
Batch	16
Width×Height	416×416
Momentum	0.937
Decay	0.0005
IOU_Thresh	0.213
T	3.0
Epochs	300

4.3 模型评价

4.3.1 知识蒸馏算法的性能对比

为验证上述提出的基于知识蒸馏的 YOLOv4 算法的性能，本文进行了消融实验。本文将设计的学生网络命名为 YOLOv4-S，基础参数信息与教师网络相同，经分类与回归蒸馏学习的网络命名为 YOLOv4-S(CR)，经分类与回归和单尺度注意力特征图蒸馏学习的网络命名为 YOLOv4-S(FM)，经分类与回归和多尺度注意力特征图（mPAM 模块）蒸馏学习的网络命名为 YOLOv4-S(KD)。算法的测试结果如表 3 所列。

表 3 知识蒸馏算法的性能对比（$IoU=0.5$）

Table 3 Performance comparison of knowledge distillation algorithm（$IoU=0.5$）

Net	P/%	R/%	mAP/%	$Params$/M	MT/ms
YOLOv4-S	78.86	82.03	80.82	16.5	17
YOLOv4-S(CR)	81.03	84.12	82.58	16.5	17
YOLOv4-S(FM)	83.94	86.35	84.49	16.5	17
YOLOv4-S(KD)	**84.15**	**85.98**	**86.57**	**16.5**	**17**
YOLOv4	86.98	89.76	88.63	257.0	58

由表 3 可以看出：1）在未引入知识蒸馏算法之前，YOLOv4-S 模型的相关性能指标（MT 除外）与 YOLOV4 模型相比，都有明显的下降，尤其是 mAP@0.5 下降了 7.81%，说明原 YOLOv4-S 模型对物体的检测效果较差；2）在引入知识蒸馏算法之后，经分类与回归蒸馏学习的 YOLOv4-S(CR) 模型的 mAP@0.5 由原来的 80.82%提高到了 82.58%，说明知识蒸馏算法能有效地提升 YOLOv4-S 模型对物体检测的能力；3）在 YOLO-S(CR)的基础上，引入单尺度注意力特征图蒸馏学习之后可以发现，YOLO-S(FM)模型的 mAP@0.5 值提高了 1.91%，说明基于注意力特征图的蒸馏学习是非常有效的，能提升学生模型对物体检测的准确率和定位精度；4）在 YOLO-S(CR)的基础上，继续引入多尺度注意力特征图蒸馏学习之后可以发现，YOLOv4-S(KD) 模型的 mAP@0.5 值与 YOLOv4-S(FM)模型相比，又提高了 2.08%，说明本文提出的基于多尺度注意力特征图的蒸馏学习算法性能更好；5）与 YOLOv4 模型相比，YOLOv4-S(KD)模型检测性能稍有下降，但是实现了推理速度的大幅度提高，并且压缩后的参数量大小仅为 16.5×10^6，说明本文提出的知识蒸馏算法能有效地对 YOLOv4 模型进行压缩和推理加速。

4.3.2 Feature map 对比分析

本文在特征提取层选取了前 12 个通道蒸馏前后的特征图并进行了可视化，如图 6 和图 7 所示。

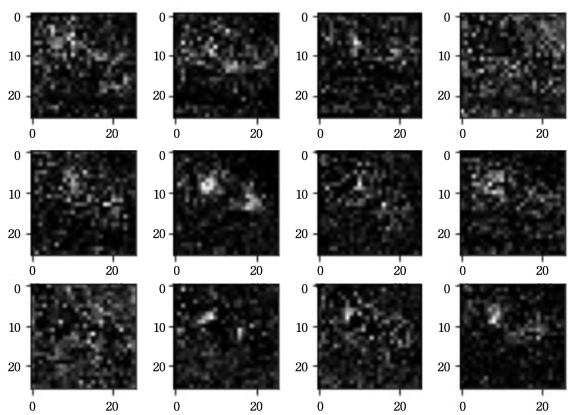


图 6 蒸馏后的特征图

Fig. 6 Characteristics after distillation

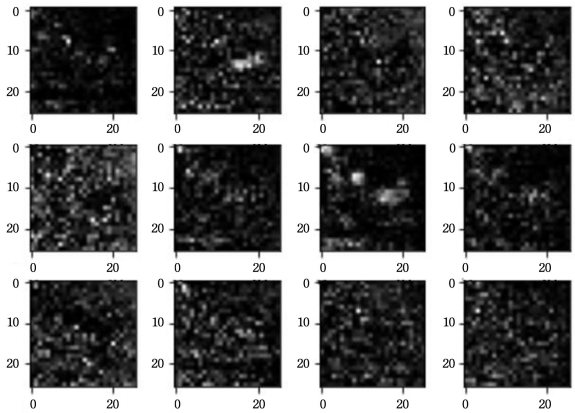


图 7 蒸馏前的特征图

Fig. 7 Characteristics before distillation

从图 6、图 7 中的第二行和第三行可以看出，在进行蒸馏之前，学生网络对物体特征的关注度较少，受背景干扰严重，学习到的信息比较杂乱。通过蒸馏的学习，学生网络对目标物体的关注度更好，受背景的干扰性变小，学习到的特征信息有明显的改善，由此可以看出，本文提出的多尺度注意力蒸馏算法对提高学生网络的特征学习具有很好的指导意义。

4.3.3 YOLO Head 特征可视化对比分析

为了更加直观地体现实验结果的有效性，我们分别对 YOLOv4，YOLOv4-S(KD)以及 YOLOv4-S 中 YOLO Head 层输出的 Person 类别进行了可视化，可视化结果如图 8—图 10 所示。

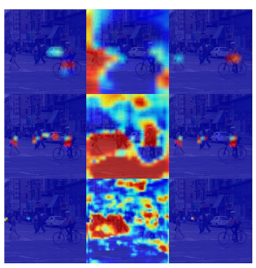


图 8 YOLOv4 的 YOLO Head 层可视化结果

Fig. 8 Visualization results of YOLO Head layer of YOLOv4

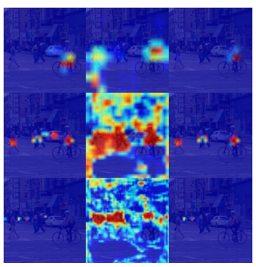


图 9 YOLOv4-S(KD)的 YOLO Head 层可视化结果

Fig. 9 Visualization results of YOLO Head layer of YOLOv4-S(KD)

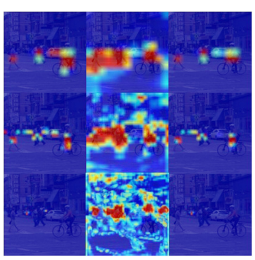


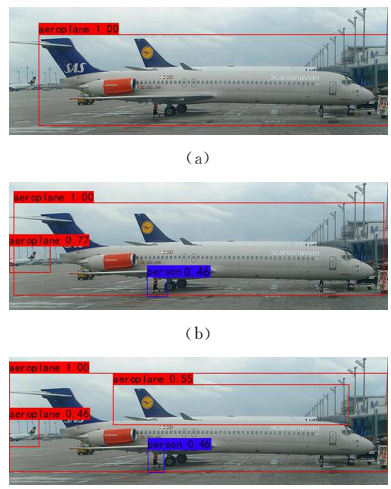
图 10 YOLOv4-S 的 YOLO Head 层可视化结果

Fig. 10 Visualization results of YOLO Head layer of YOLOv4-S

图 8 给出了 YOLOv4 网络结构中 YOLO Head 层的可视化结果，图 9 给出了 YOLOv4-S(KD)中 YOLO Head 层的可视化结果，图 10 给出了 YOLOv4-S 中 YOLO Head 层的可视化结果。每张图的第一行代表的是 13×13 网格的可视化结果，第二行代表的是 26×26 网格的可视化结果，第三行代表的是 52×52 网格的可视化结果。每张图中的第一列代表的是置信度，即代表此网格内是否存在物体，第二列代表的是对所属类别的打分，第三列则是对前两列的求和，即判断为是物体同时又是 Person 类的结果。

由图 10 可知，在未进行蒸馏学习之前，学生网络自己学习的结果比较杂乱，同时对较远的目标物体存在漏检的情况。由图 9 可以看出，通过蒸馏的学习，学生网络对特征进行了更加充分的学习，对于原来一些没有检测到的小目标进行了更加精确的检测，同时对物体类别实现了更精准的划分，达到了与 YOLOv4 网络结果相差不大的可视化输出。

最后为了验证实际检测结果的有效性，我们对比了蒸馏前后以及 YOLOv4 的检测效果，实验效果如图 11 所示。



(a)

(b)

(c)

图 11 蒸馏前后以及 YOLOv4 的检测效果

Fig. 11 Results of YOLOv4 before and after distillation

图 11(a)是 YOLOv4-S 的检测结果，检测到一个目标，图 11(b)是 YOLOv4-S(KD)的检测结果，检测到 3 个目标，图 11(c)是 YOLOv4 网络的检测结果，检测到 4 个目标。从图中可以看出，在进行知识蒸馏后学生网络的检测精度有所提高，检测结果更加接近教师网络。

最后，为了验证模型的有效性，我们对检测结果进行了部分提取，检测结果如图 12 所示。

图 12 YOLOv4-S(KD)检测结果

Fig. 12 Result of YOLOv4-S(KD)

4.3.4 与其他模型的性能对比

实验最后，基于 VOC 数据集，本文进行了与其他 YOLO 轻量级检测算法的对比实验，实验结果如表 4 所列。

表 4 YOLO 轻量级算法的性能对比（$IoU=0.5$）

Table 4 Performance comparison of YOLO lightweight algorithm ($IoU=0.5$)

Net	P/%	R/%	mAP/%	Params/M	MT/ms
YOLOv3-tiny	61.53	55.12	58.4	33.4	16
YOLOv4-tiny	79.26	76.02	77.8	22.6	18
YOLOv4-mobilenetv2	78.69	81.65	80.12	46.8	18
YOLOv4-S(KD)	**84.15**	**85.98**	**86.57**	**16.5**	**17**

由表 4 可知，与 YOLOv3-tiny 相比，本文提出的基于知识蒸馏的 YOLOv4 算法虽然在平均检测时间上不如其检测速度，但模型的 mAP@0.5 值提高了 28.17%，在牺牲较短时间的情况下获得了较大的精度。

与 YOLOv4-tiny 相比，本文提出的基于知识蒸馏的 YOLOv4 算法模型的 mAP@0.5 值提高了 8.77%，模型参数缩小了近 27%，平均检测时间提高了近 5.6%。

与表现效果较好的 YOLOv4-mobilenetv2 相比，本文提出的基于知识蒸馏的 YOLOv4 算法的 mAP@0.5 值提高了 6.45%，模型大小缩小为原来的 1/3，同时在平均检测时间上也有所提升。

综上可知，本文提出的基于知识蒸馏的 YOLOv4 算法虽然减小了网络的宽度，使得学生网络学习到的每一层特征有所减少，但通过本文添加的多尺度像素注意力模块以及分类回归蒸馏对学生网络的引导学习，使得学生网络学习到了更加丰富的特征，达到了较好的检测精度，同时减少了模型参数，提升了推理速度，在检测精度和速度上达到了一个很好的平衡。

结束语 本文基于 YOLOv4 和知识蒸馏算法进行改进，提出多尺度像素注意模块（mPAM）来构造特征提取网络的特征注意力图，并同时在目标分类与边框预测上展开蒸馏。多尺度注意力蒸馏算法对学生网络生成的特征图进行指导训练，使得学生网络可以学到与教师网络相似的、不同尺度的特征编码方式。通过在目标分类与定位阶段引入知识蒸馏，使得学生网络学习到教师网络强大的分类回归功能。实验结果表明，本文提出的知识蒸馏算法在不改变学生网络结构的基础上，获得了较好的精度，模型参数更小，可以更好地满足实时性要求，为移动设备的部署提供了可行方案。但其仍然存在提升的空间，如何更好地兼顾速度和效率是下一步需要考虑的问题，同时如何更好地在实际场景中进行应用也是下一步的研究方向。

参 考 文 献

[1] GIRSHICK R J, DONAHUE T, DARRE L, et al. Region Based Convolutional Networks for Accurate Object Detection and Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38: 142-158.

[2] HE K, ZHANG X, REN S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9): 1904-1916.

[3] GIRSHICK R. Fast R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision. 2015: 1440-1448.

[4] REN S Q, HE K M, GIRSHICK R, et al, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39: 1137-1149.

[5] LIN T Y, PIOTOR DOLLA R, GIRSHICK R, et al. Feature Pyramid Networks for Object Detection[C]// Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2117-2125.

[6] CAI Z W, VASCONCELOS N. Cascade R-CNN: Delving into High Quality Object Detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recogntion. 2018: 6154-6162.

[7] REDMON J, FARHADI A. YOLO9000: Better, Faster, Stronger [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 6517-6525.

[8] REDMON J, FARHADI A. YOLOv3: AnIncremental Improvement[J]. arxiv: 1804.02767, 2018.

[9] BOCHKOVSKIY A, WANG C Y, LIAO H. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. arXiv: 2004.10934, 2020.

[10] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single Shot MultiBox Detector[C]// Proceedings of European Conference on Computer Vision. 2016: 21-37.

[11] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal Loss for Dense Object Detection[C]// Proceedings of IEEE International Conference on Computer Vision. 2017: 2980-2988.

[12] BUCILA C, CARUANA R, NICULESCU-MIZIL A. Model Compression[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06). Computer Science Cornell University, 2006: 535-541.

[13] HINTON G, VINYALS O, DEAN J. Distilling the Knowledge in a Neural Network[J]. Computer Science, 2015, 14(7): 38-39.

[14] ZAGORUYKO S, KOMODAKIS N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer[J]. arXiv: 1612.03928, 2016.

[15] HOU Y, MA Z, LIU C, et al. Learning Lightweight Lane Detection CNNS By Self AttentionDistillation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1013-1021.

[16] HE T, SHEN C, TIAN Z, et al. KnowledgeAdaptation for Efficient Semantic Segmentation[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 578-587.

[17] WANG T, YUAN L, ZHANG X, et al. Distilling Object Detectors With Fine-Grained Feature Imitation[C]// 2019IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[18] GAO M, SHEN Y, LI Q, et al. Residual Knowledge Distillation [J]. arXiv: 2002.09168, 2020.

[19] WANG W, HONG W, WANG F, et al. Gan-knowledge distillation for one-stage object detection[J]. IEEE Access, 2020, 8: 60719-60727.

[20] CHAWLA A, YIN H, MOLCHANOV P, et al. Data-Free Knowledge Distillation for Object Detection[C]// Proceedings of

the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021:3289-3298.
[21] DAI X, JIANG Z, WU Z, et al. General Instance Distillation for Object Detection[C]// Computer Vision and Pattern Recognition (CVPR). IEEE, 2021.
[22] ZHOU D W, MA L Y, TIAN J Y, et al. Super-resolution Reconstruction of Images Based on Feature Fusion Attention Networks[J]. Acta Automatica Sinica, 2019, 57:1-9.
[23] ROMERO A, BALLAS N, KAHOU S E, et al. Fitnets: Hints for Thin Deep Nets[J]. arXiv:1412.6550, 2014.
[24] LI H. Exploring Knowledge Distillation of Deep Neural Nets for Efficient Hardware Solutions[C]// CS230 Report. 2018.
[25] EVERINGHAM M, ESLAMI S, GOOL L V, et al. The PASCAL Visual Object Classes Challenge: A Retrospective[J]. International Journal of Computer Vision, 2015, 111(1):98-136.

CHU Yu-chun, born in 1996, postgraduate. His main research interests include information security and object detection.

WANG Xue-fang, born in 1975, Ph. D, lecturer. Her main research interests include artificial intelligence and deep learning.

(上接第 271 页)

[11] Facebook App Links[EB/OL]. https:// developers. facebook. com/docs/applinks.
[12] MA Y, HU Z, LIU Y, et al. Aladdin: Automating release of deep-link APIs on Android[C]// Proceedings of the 2018 World Wide Web Conference. 2018:1469-1478.
[13] AZIM T, RIVA O, NATH S. uLink: Enabling user-defined deep linking to app content[C]// Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. 2016:305-318.
[14] Android Accessibility API[EB/OL]. http:// www. androiddoc. com/guide/topics/ui/accessibility/index. html/.
[15] Wandoujia[EB/OL]. https://www. wandoujia. com/.
[16] SHIN C, HONG J H, DEY A K. Understanding and prediction of mobile application usage for smart phones[C]// Proceedings of the 2012 ACM Conference on Ubiquitous Computing. 2012:173-182.
[17] XU Y, LIN M, LU H, et al. Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns[C]// Proceedings of the 2013 International Symposium on Wearable Computers. 2013:69-76.
[18] LEE J, LEE K, JEONG E, et al. Context-aware application scheduling in mobile systems: What will users do and not do next? [C]// Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2016:1235-1246.
[19] LIAO Z X, LEI P R, SHEN T J, et al. Mining temporal profiles of mobile applications for usage prediction[C]// 2012 IEEE 12th International Conference on Data Mining Workshops. IEEE, 2012:890-893.
[20] RAHNAMOUN R, RAWASSIZADEH R, MASKOOKI A. Learning mobile app usage routine through learning automata [J]. arXiv:1608.03507, 2016.
[21] PICHILIANI M C, HIRATA C M. Evaluation of the Android accessibility API recognition rate towards a better user experience[C]// International Conference on Universal Access in Human-Computer Interaction. Cham: Springer, 2015:340-349.
[22] FERNANDES E, RIVA O, NATH S. Appstract: On-the-fly app content semantics with better privacy[C]// Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking. 2016:361-374.
[23] Standford NER[EB/OL]. https://nlp. stanford. edu/software/CRF-NER. html.
[24] SHI C Y, XU C J, YANG X J. TFIDF algorithm research project [J]. Computer Applications, 2009, 29(B6):167-170.
[25] RUAN D, PAN H, GAO K. Similar words analysis based on POS-CBOW language model[J]. Journal of Hebei University of Science & Technology, 2015, 36(5):532-538.
[26] HUNG P T, YAMANISHI K. Word2vec skip-gram dimensionality selection via sequential normalized maximum likelihood [J]. Entropy, 2021, 23(8):997.
[27] Tencent AI Lab[EB/OL]. https://ai. tencent. com /ailab/zh/paper/?page=1/.
[28] YAN P M, TANG W Q. Chinese text classification based on improved BERT[J]. Industrial Control Computers, 2020, 33(7):108-110.
[29] ZHOU X, WU K, CAI H, et al. LogPruner: detect, analyze and prune logging calls in Android apps[J]. Science China Information Sciences, 2018, 61(5):050107:1-050107:3.

WANG Yi, born in 1996, postgraduate. His main research interests include Android application service generation and Android application service adaptation.

CHEN Xing, born in 1985, Ph.D, professor, Ph. D supervisor, is a member of China Computer Federation. His main research interests include system software, software self-adaptation and cloud computing.