文章编号:1001－9383(2020)04－0001－08

基于卷积神经网络的图像检测识别算法综述

曾文献,张淑青,马 月,李伟光

(河北经贸大学 信息技术学院,河北 石家庄 050061)

摘 要:介绍了卷积神经网络与图像识别的前置技术,对主流图像检测算法进行了综述,对比了主流算法模型在voc2007＋2012和COCO数据集中的性能,接着讨论了图像识别算法的部署及应用,最后对卷积神经网络下的图像检测与识别算法进行了总结与展望。

关键词:卷积神经网络;图像检测;图像识别

中图分类号:TP391 文献标识码:A

DOI:10.16191/j.cnki.hbkx.2020.04.001

Overview of image detection and recognition algorithms based on convolutional neural network

ZENG Wen-xian, ZHANG Shu-qing, MA Yue, LI Wei-guang

(*College of Information Technology, Hebei University of Economics and Business, Shijiazhuang Hebei* 050061, *China*)

Abstract: This article introduces the front-end technology of convolutional neural networks and image recognition, reviews mainstream image detection algorithms, compares the performance of mainstream algorithm models in the voc2007＋2012 and COCO datasets, and then discusses the deployment and application of image recognition algorithms. Finally, the image detection and recognition algorithm under convolutional neural network is summarized and prospected.

Keywords: Convolutional neural network; Image detection; Image recognition

0 引言

随着人工智能技术的不断发展,图像的数量呈指数函数型增长,图像检测识别技术的应用也愈加广泛,如何在深度学习基础上对海量图像数据进行智能处理成为计算机视觉领域研究的热点之一[1]。

近年来,深度学习技术越来越成熟,计算机视觉的应用范围也逐渐扩大,在很多领域都取得了比较良好的成果,给图像检测技术的发展带来更多元的研究方向[2]。卷积神经网络下的

收稿日期:2020－09－23

基金项目:河北省科技厅科技计划项目(199676265D)

作者简介:曾文献(1971－),男,陕西旬阳人,硕士,教授,研究领域为计算机视觉及物联网技术.

图像检测算法主要包括基于快速 CNN 的检测算法和基于回归学习的检测算法。本文主要对这两种方法中的主流算法进行了总结分析。

1 卷积神经网络与图像识别前置技术

1.1 CNN 网络模型

在图像检测识别领域，最常用的网络模型为 CNN 网络模型，如图 1 所示。在该模型中，可将输入信息在不同阶层不同结构中根据需求进行平移不变分类[3]。

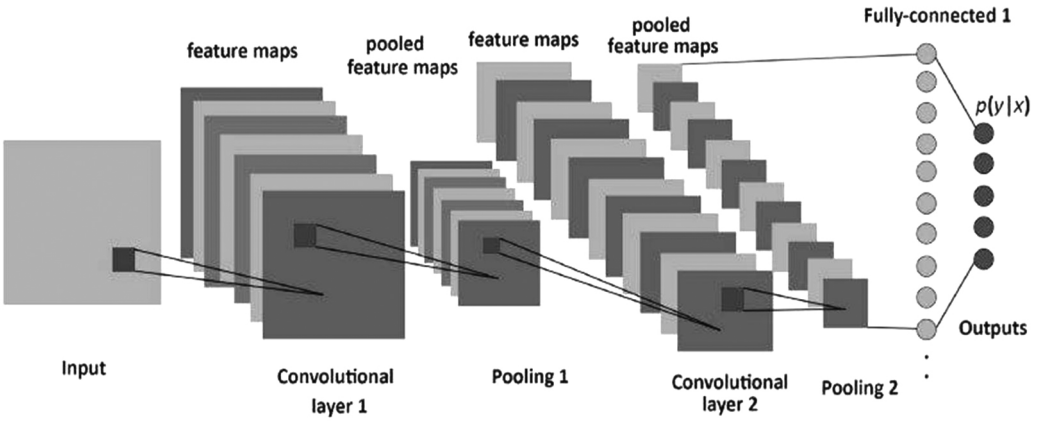

图 1 CNN 网络模型

1.2 图像识别前置技术

1.2.1 深度学习框架

近年来，深度学习技术在理论和基础框架方面都取得了重大的突破，本节主要对主流的深度学习框架进行综述。

深度学习框架包括主要包括 Theano、TensorFlow、MXNet、Keras、PyTorch 和 Caffe。其中，Theano 是一个 python 库，可理解为数学表达式的编译器，是早期深度学习开发与研究的行业标准。TensorFlow 是现阶段使用最多的深度学习框架。MXNet 支持多种语言，比较适用于云平台。Keras 是一个高层神经网络 API。PyTorch 可使用 GPU 加速，相较于其他框架具有实现动态神经网络的优点。Caffe 可支持 Matlab 和 python 等接口，是现今较为流行的框架之一。

1.2.2 Numpy 包

图像处理过程中，大多数场景都需要先将图像转换成矩阵或向量，再进行图像识别，因此 Numpy 包是图像识别算法中非常重要的前置技术。Numpy 矩阵对图片进行科学计算，将图片处理过程简化为空间向量计算，进而实现图像的识别。

2 基于卷积神经网络的图像检测算法

2.1 基于快速 CNN 的检测算法

基于快速 CNN 的检测算法主要包括 R-CNN、SPP-Net、Fast RCNN 、Faster RCNN、R-FCN、Mask R-CNN、Cascade R-CNN 和 TridentNet 等，本节主要分析了几种主流的基于快速 CNN 的检测算法。

2.1.1　R-CNN

RCNN 算法[4]是 2013 年由 Grishick 等人提出,以 Alexnet 为主干网络,算法的网络模型如图 2 所示。在 RCNN 中,将 CNN 与候选框推荐方法进行结合,检测过程分为了特征提取+SVM 分类两部分,在精度方面有了较大的提升,但检测速度比较缓慢。

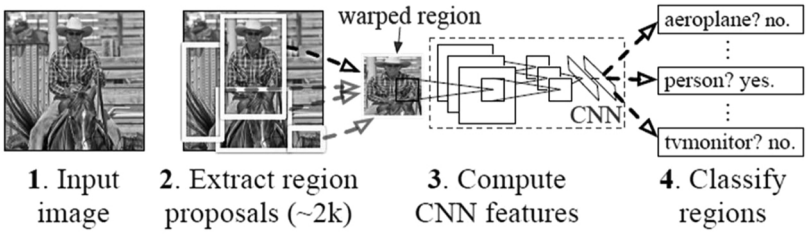

图 2　RCNN 网络模型

2.1.2　SPP-Net

SPP-Net 算法[5]是 2014 年由 He 等人提出,它是以 ZF-5 为主干网络的算法,可对任意大小的图像进行池化。SPP-Net 在最后一层添加了 SPP 层,每一张图像仅进行一次卷积,当输入任意尺寸图片(w,h)时,想要输出 21 个神经元时,我们可以利用空间金字塔对图片进行尺度划分完成特征提取。

2.1.3　Faster RCNN

Faster RCNN 算法[6]主要是对 FastCNN 算法[7]中一些不足地方进行的改进,2015 年由 Ren 等人提出,是以 VGG-16 为主干网络的算法,网络模型如图 3 所示。算法的损失函数分为 RPN 分类损失、RPN 位置回归损失、ROI 分类损失和 ROI 位置回归损失四个部分,具体的计算公式如公式(1)所示。

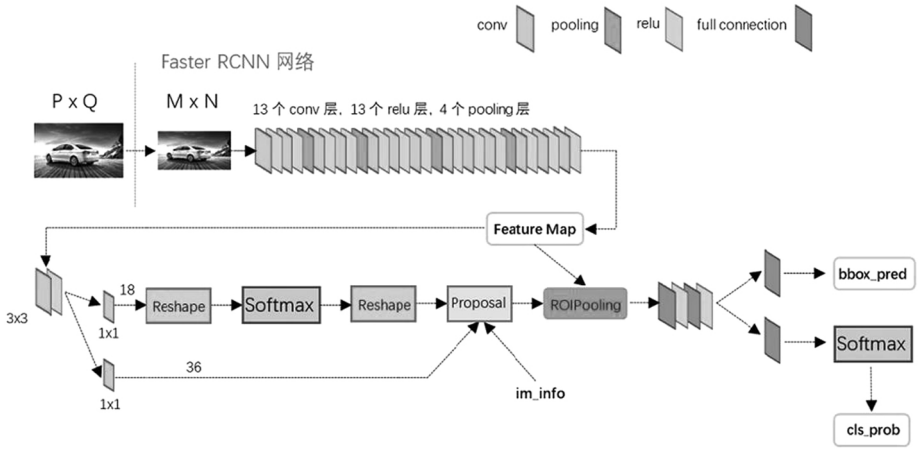

图 3　Faster RCNN 网络模型

$$L(p_i,t_i)=\frac{1}{N_{cls}}\sum\nolimits_i L_{cls}(p_i,p_i^*)+\lambda\frac{1}{N_{reg}}\sum\nolimits_i p_i^* L_{reg}(t_i,t_i^*) \tag{1}$$

式中,p_i 表示正样本预测分类概率,当为正样本时,$p_i^*=1$,否则 $p_i^*=0$;t_i 为正样本时预测的 Bounding Box 的参数化坐标,t_i^* 为样本的 Ground Truth 的 Bounding Box 的参数化坐标,N_{cls} 表示 mini-batch size,N_{reg} 表示 Anchor Location,$p_i^* L_{reg}(t_i,t_i^*)$ 表示只有预测为正样本时回归 Bounding Box。L_{cls} 为分类误差,是两个类别的对数损失,如公式(2)所示,L_{reg} 是检测误差。

$$L_{cls}(p_i,p_i^*)=-\log[p_ip_i^*+(1-p_i)(1-p_i^*)] \tag{2}$$

2.1.4 Mask R-CNN

Mask R-CNN[8]是 2017 年由 He 等人[8]提出，选择 ResNET-101-FPN 为主干网络，FCN 网络模型如图 4 所示。它在 Faster RCNN 网络基础上加入了一个预测目标掩饰的并行分支，不但提高了检测精度，还可以利用 FCN 网络实现实例分割，具有高速、高准确率、简单易用等特点，并在损失函数中加入了 mask 分支，计算公式如公式 3 所示。

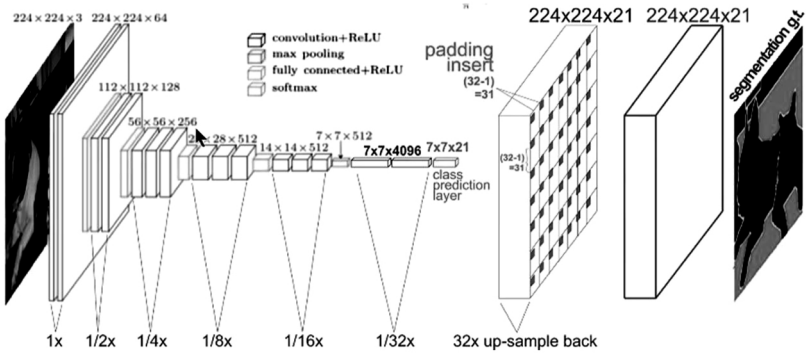

图 4 FCN 网络模型

$$L=\overline{L}_{cls}+L_{box}+L_{mask} \tag{3}$$

式中 L_{cls} 为分类误差，L_{box} 是检测误差，L_{mask} 为分割误差。

2.2 基于回归学习的检测算法

基于回归的检测算法又称为单阶段检测算法，主流算法包括 YOLO 系列、SSD 系列、RetinaNet、CornerNet、CenterNet、EfficientDe 等。本节对主流的 YOLO 系列、SSD 系列、RetinaNet 和 CenterNet 进行介绍。

2.2.1 YOLO 系列

YOLO 系列从 2015 发展至今共有四个版本，本节从算法原理分别对其介绍。

YOLOV1 算法[9]是 2016 年由 Redmond 等人提出，在它的实验过程中使用了一个单独的 CNN 模型，实现了端到端的检测，网络模型如图 5 所示。各边界框的类别置信度如公式 4 所示。

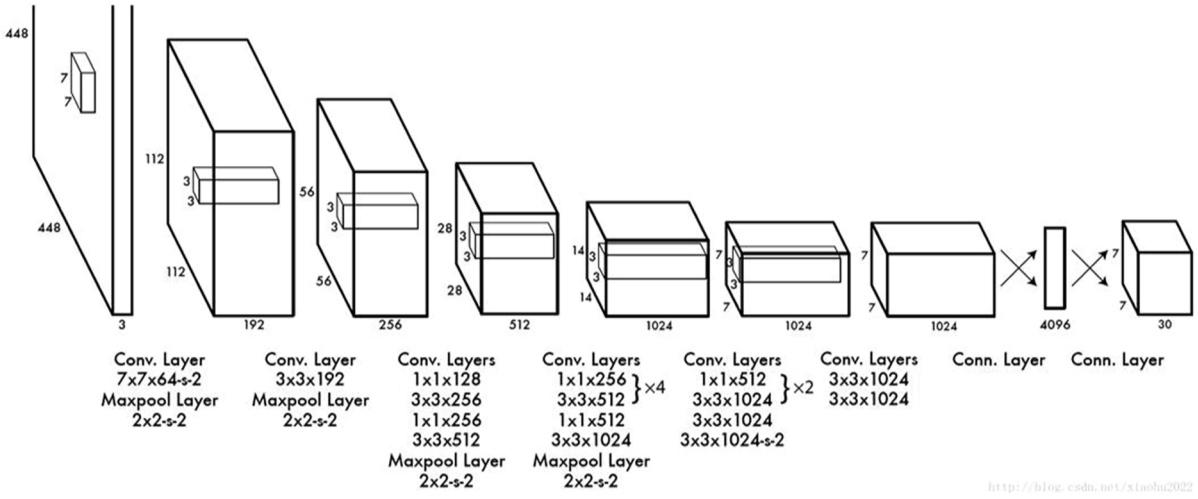

图 5 YOLOV1 网络模型

$$Pr(class_i \mid object) * Pr(object) * IOU_{pred}^{truth} = Pr(class_i) * IOU_{pred}^{truth} \tag{4}$$

式中,$Pr(object) * IOU_{pred}^{truth}$ 为 $Confidence$,IOU_{pred}^{truth} 为预测框和实际框的 IOU,是描述两个区域重合度的评价指标,重合度越高,代表模型的定位越准确。

YOLOV2 算法[10]是 2017 年 Redmon 等在 V1 基础上进行了 BN 操作,选择 Darknet-19 作为主干网络,提升了模型的收敛速度。使用 anchor boxes 来预测边界框相对于先验框的 offsets,位置预测公式和边界框预测公式如公式(5)－(6)所示。

$$x=(t_x * w_a)+x_a,\ y=(t_y * h_a)+y_a \tag{5}$$

$$b_x=\sigma(t_x)+c_x, b_y=\sigma(t_y)+c_y, b_w=p_w e^{t_w}, b_h=p_h e^{t_h} \tag{6}$$

式中,x、y 为预测边框中心,x_a、y_a 为先验框中心点坐标,w_a、h_a 为宽和高,t_x、t_y 为要学习参数。b_x、b_y、b_w、b_h 分别为预测边框的中心和宽高。

YOLOV3 算法[11]是 Choi 等人利用 Darknet-53 作为主干网络,在网络中加入了残差模块,在检测精度和速率方面都得到很大的提升。

YOLOV4 算法[12]是 2020 年 Bochkovskiy 等人提出,主要由 CSP Darknet53 主干网络和 SPP、PAN 结构组成,网络模型如图 6 所示。

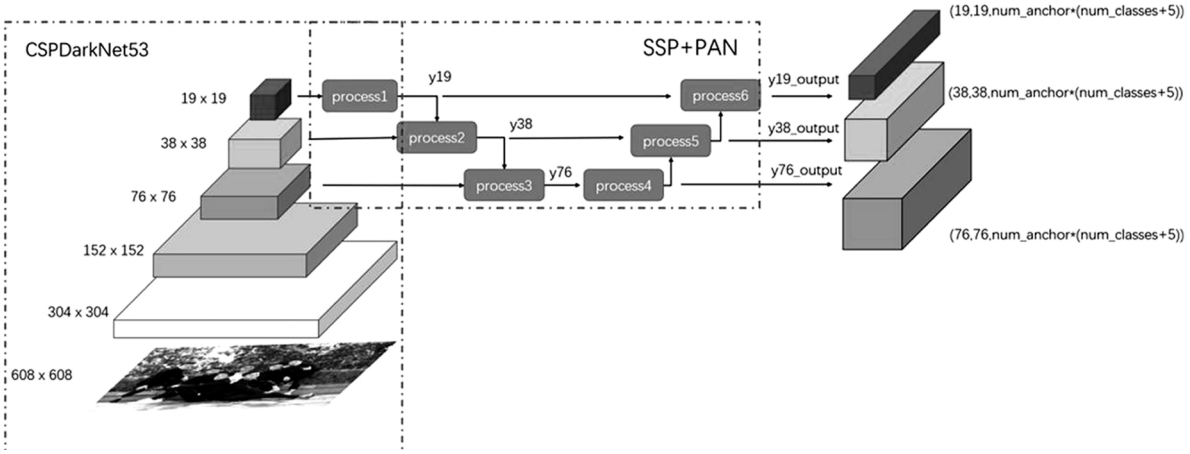

图 6　YOLOV4 网络模型

YOLOV4 算法选择 Mish 作为激活函数,如公式(7)所示。式中,$tanh(x)=\dfrac{e^{2x}-1}{e^{2x}+1}$,$softplus(x)=log(1+e^{2x})$。

$$y=x * tanh(softplus(x)) \tag{7}$$

2.2.2　SSD 系列

SSD[13]系列网络是 2016 年由 Liu 等人提出,主要利用不同层特征完成检测,实验结果在检测精度和准确率等方面有着显著提升,网络模型如图 7 所示。它使用 default box 来预测生成边界框,计算公式如公式(8)所示。

$$S_k=S_{min}+\frac{S_{max}-S_{min}}{m-1}(k-1),\ \text{其中}\ k\in(0,1) \tag{8}$$

SSD 改进算法主要包括 DSSD[14]、DSOD[15] 和 FSSD[16]。其中,DSSD 算法[14]选择 Resnet-101作为主干网络,使用多个反卷积层来扩展低维度信息的上下文信息。FSSD 算法[16]将 FPN 和 SSD 进行特征融合,把网络中某些 feature 调整为同一尺寸再融合。

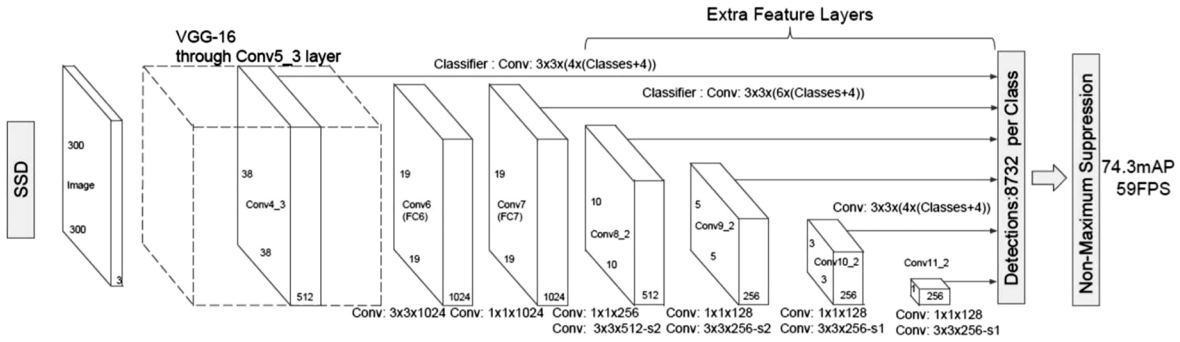

图 7　SSD 网络模型

2.2.3　RetinaNet

RetinaNet **算法**[17]**是** 2017 **年** Lin **等人针对样本类别不平衡导致单阶段检测算法精度低问题所提出，利用** FPN **作为主干网络，网络模型如图** 8 **所示。**

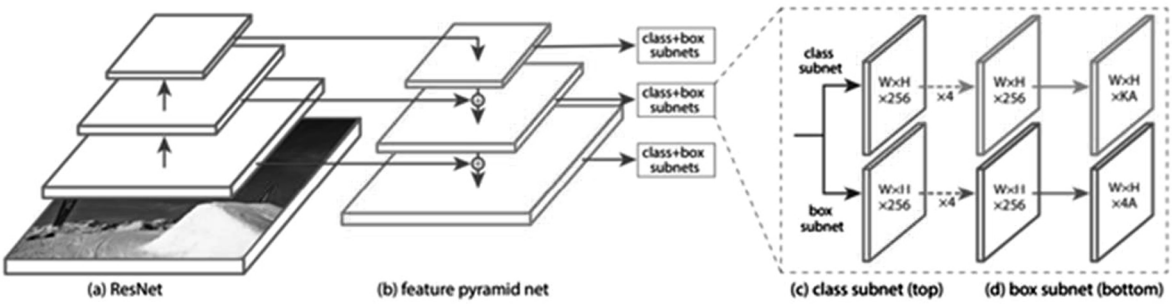

图 8　RetinaNet 网络模型

该模型使用 focal loss **作为训练过程的损失函数，计算公式如公式**(9)**所示。**

$$FL(p_t)=-(1-p_t)^{\gamma}\log(p_t) \tag{9}$$

式中，$(1-p_t)^{\gamma}$ **为调节因子，当** $\gamma=0$ **时，**FL **等价于** CE**，在论文**[17]**的实验中，当** $\gamma=2$ **时实验效果最好。**

2.2.4　Centnet

Centnet **算法**[18]**是** 2019 **年** Duan **等人提出，使用关键点来确定目标，将检测问题转换成中心点估计思想。该算法选择** Hourglass **作为主干网络进行提取特征，以** center pooling **和** cascade corner pooling **进行** key point forecast**，网络模型如图** 9 **所示。**

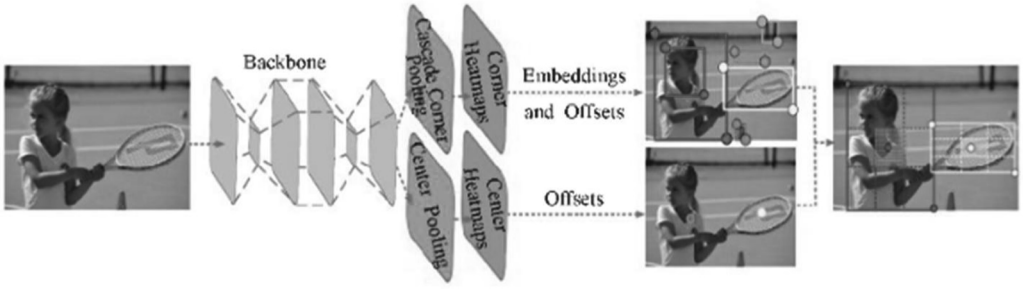

图 9　Centnet 网络模型

2.3　算法模型性能对比

在深度学习检测识别算法中,常用的性能评价指标包括检测速度(FPS,每秒处理图片的数量)、交并比(IOU)、准确率(Accuracy)、查准率(P)、召回率(R)平均精确度(AP)和平均精确率均值(mAP,由 P-R 曲线决定)等。计算如公式(10)所示。

$$P=\frac{TP}{TP+FP}\ ,\ R=\frac{TP}{TP+FN}=\frac{TP}{T}$$

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN},\ IOU=\frac{TP}{TP+FP+FN} \tag{10}$$

2.3.1　VOC2007+2012 数据集上算法模型性能对比

VOC2007 和 VOC2012 数据集是计算机视觉领域的标准数据集,主要包括 20 类不同的数据,是图像检测领域使用最多的数据集,是检验算法性能好坏的评价基准之一。上述主流算法在以 VOC2007+2012 为训练集,VOC2007 为测试集的实验过程中,性能对比如图 10 所示。

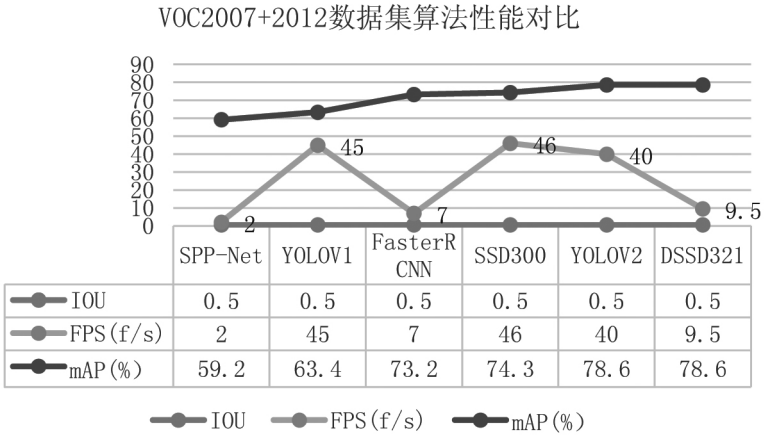

图 10　VOC2007+2012 数据集上算法模型性能对比

2.3.2　COCO 数据集上算法模型性能对比

Microsoft COCO 数据集是微软团队收集用来进行图像识别、语义分割等任务的数据集,包含 91 个物体类别,可应用于多种不同场景,是现如今图像分割领域最大的数据集。上述算法在 COCO 数据集上实验的模型性能如图 11 所示。

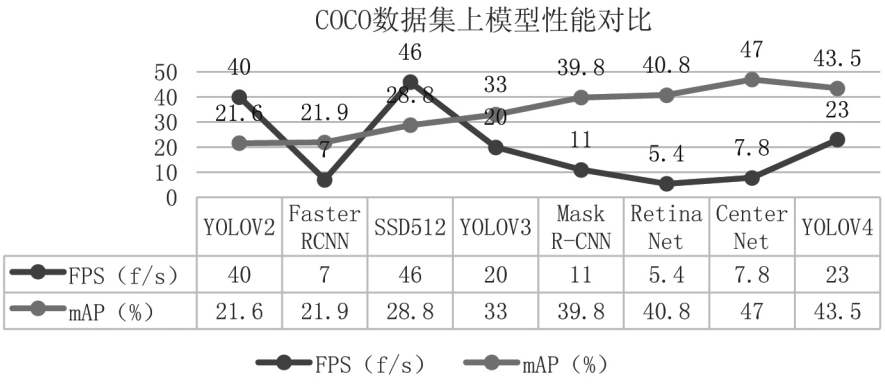

图 11　COCO 数据集上算法模型性能对比

3 图像识别算法的部署及应用

3.1 图像识别算法部署模式

研究表明，人类对世界的感知 80％来自视觉，因此图像识别技术变得越来越重要。图像识别算法的部署模式主要可分为以下三类：

(1)基于公共云云计算的计算机集群，利用公共 API 服务的形式提供接口，用户通过调用 API 接口进行图像的识别，可在实时性要求较低的场景中使用。

(2)基于私有云云计算的计算机集群，这种方式需要用户自己搭建私有云架构系统，将图像识别算法打成容器镜像等方式进行部署。

(3)基于 ARM 的硬件部署，这种方式不仅需要将算法和硬件耦合，还需要算法的开发人员和硬件的生产商共同设计开发，以达到图像识别的目的。

3.2 实际应用场景

图像识别算法的应用场景非常广泛，在不同部署模式下都有其相对应的应用场景。公共云部署模式主要针对图像识别实时性要求不高的场景，如人脸识别，车牌识别，垃圾广告识别等。私有云计算部署模式主要针对国家安监部门、刑侦机关及一些大型企业等应用场景。ARM 的硬件部署主要将图像识别算法应用到一些专用硬件中，使算法在硬件内部完成计算并输出识别结果。

4 总结与展望

本文首先介绍了 CNN 和图像识别前置技术，然后综述了主流目标检测算法，对各算法在 VOC2007＋2012 和 coco 数据集中的实验结果进行了对比，最后总结了算法的部署及应用。通过综述发现，如何进一步简化图像检测识别过程、如何进一步提升检测精度和速率等，将会是未来目标检测领域发展的重中之重。

参考文献：

[1] 王颢. 深度学习在图像识别中的研究与应用[J]. 科技视界，2020，(24)：37－38.

[2] 黄健，张钢. 深度卷积神经网络的目标检测算法综述[J]. 计算机工程与应用，2020，56(17)：12－23.

[3] Zhang，W. Shift-invariant pattern recognition neural network and its optical architecture. In Proceedings of annual conference of the Japan Society of Applied Physics，1988.

[4] AGRAWAL P，GIRSHICK R，MALIK J. Analyzing the performance of Multilayer Neural Networks for Object Recognition[J]. Lecture Notes in Computer Science，2014.

[5] He K，Zhang X，Ren S，et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2015，37(9)：1904－1916

[6] Ren S，He K，Girshick R，et al. Faster R－CNN：towards real-time object detection with region proposal networks[C]// Conference on Neural Information Processing Systems，2015：91－99.

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

[8] He K，Gkioxari G，Dollar P，et al. Mask R－CNN[C]//Proceedings of the IEEE International Conference on Computer Vision，2017：2980－2988.

[9] Redmon J，Divvala S，Girshick R，et al. You only look once：unified，real-time object detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2016：779－788.

（下转第 28 页）

[13] M. Yousefi, S. Golmohammady, A. Mashal, F. D. Kashani. Analyzing the propagation behavior of scintillation index and bit error rate of a partially coherent flat-topped laser beam in oceanic turbulence[J]. J. Opt. Soc. Am, 2015, 32: 1982－1992.

[14] M. Yousefi, F. D. Kashani, S. Golmohammady. Scintillation and bit error rate analysis of a phase-locked partially coherent flat-topped array laser beam in oceanic turbulence[J]. J. Opt. Soc. Am, 2017, 34: 2126－2131.

[15] X. Yi, Z. Li, Z. Liu. Underwater optical communication performance for laser beam propagation through weak oceanic turbulence[J]. Appl. Opt, 2015, 54: 1273－1278.

[16] M. C. Gökçe, Y. Baykal. Aperture averaging and BER for Gaussian beam in underwater oceanic turbulence[J]. Opt. Commun, 2018, 410: 830－835.

[17] M. C. Gökçe, Y. Baykal. Aperture averaging in strong oceanic turbulence[J]. Opt. Commun, 2018, 413: 196－199.

[18] R. Griffis, J. Howard. Oceans and Marine Resources in a Changing Climate: A Technical Input to the 2013 National Climate Assessment, 2013.

[19] O. I. Mamayev. Temperature-Salinity Analysis of World Ocean Waters, 1975.

[20] G. Einsele. Sedimentary Basins: Evolution, Facies, and Sediment Budget, 2000.

[21] M. Elamassie, M. Uysal, Y. Baykal. Effect of eddy diffusivity ratio on underwater optical scintillation index[J]. J. Opt. Soc. Am, 2017, 34: 1969－1973.

[22] L. C. Andrews, R. L. Phillips, C. Y. Hopen. Laser Beam Scintillation with Applications, 2001.

[23] L. C. Andrews, R. L. Phillips. Laser Beam Propagation through Random Media, 2nd ed, 2005.

[24] E. B. Kraus, J. A. Businger. Atmosphere-Ocean Interaction, 1994.

（上接第 8 页）

[10] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6517－6525.

[11] Redmon J, Farhadi A. YOLOv3: an incremental improvement[J]. ar Xiv: 1804. 02767, 2018.

[12] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. [2020－02－17]. https://arxiv. org/pdf/2004. 10934. pdf.

[13] Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector[C]//European Conference on Computer Vision, 2016: 21－37.

[14] Fu C Y, Liu W, Ranga A, et al. DSSD: deconvolutional single shot detector[EB/OL]. (2017－12－10). https://arxiv. org/pdf/1701. 06659. pdf.

[15] Shen Z, Liu Z, Li J, et al. DSOD: learning deeply supervised object detectors from scratch[C]//IEEE International Conference on Computer Vision, 2017: 1937－1945.

[16] Li Z, Zhou F. FSSD: feature fusion single shot multibox detector[J]. ar Xiv: 1712. 00960, 2017.

[17] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//International Conference on Computer Vision, 2017: 2999－3007.

[18] Duan K, Bai S, Xie L, et al. Center Net: keypoint triplets for object detection[J]. ar Xiv: 1904. 08189, 2019.

[19] Donggeun YOO, Sunggyun Park, Joon-Young Lee, et al. AttentionNet: Aggregating Weak Directions for Accurate Object Detection[J]. arXiv: 1506. 07704v2, 2015