第 58 卷 第 2 期/2021 年 1 月/激光与光电子学进展 研究论文

激光与光电子学进展 先进成像

一种基于多尺度特征融合的目标检测算法

张涛，张乐*

天津大学电气自动化与信息工程学院，天津 300072

摘要 基于深度学习的目标检测器 RetinaNet 和 Libra RetinaNet 均是使用特征金字塔网络融合多尺度特征，但上述两个检测器存在特征融合不充分的问题。鉴于此，提出一种多尺度特征融合算法。该算法是在 Libra RetinaNet 的基础上进一步扩展，通过建立两条自底向上的路径构建两个独立的特征融合模块，并将两个模块产生的结果与原始预测特征融合，以此提高检测器的精度。将多尺度特征融合模块与 Libra RetinaNet 结合构建目标检测器并在不同的数据集上进行实验。实验结果表明，与 Libra RetinaNet 检测器相比，加入模块后的检测器在 PASCAL VOC 数据集和 MSCOCO 数据集上的平均精度分别提高 2.2 个百分点和 1.3 个百分点。

关键词 机器视觉；卷积神经网络；目标检测；特征金字塔；特征融合

中图分类号 TP391.4 **文献标志码** A **doi**：10.3788/LOP202158.0215003

□Multiscale Feature Fusion-Based Object Detection Algorithm

Zhang Tao, Zhang Le*

School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, *China*

Abstract The RetinaNet and Libra RetinaNet object detectors based on deep learning employ feature pyramid networks to fuse multiscale features. However, insufficient feature fusion is problematic in these detectors. In this paper, a multiscale feature fusion algorithm is proposed. The proposed algorithm is extended based on Libra RetinaNet. Two independent feature fusion modules are constructed by establishing two bottom-up paths, and the results generated by the two modules are fused with the original predicted features to improve the accuracy of the detector. The multiscale feature fusion module and Libra RetinaNet are combined to build a target detector and conduct experiments on different datasets. Experimental results demonstrate that the average accuracy of the added module detector on PASCAL VOC and MSCOCO datasets is improved by 2.2 and 1.3 percentage, respectively, compared to the Libra RetinaNet detector.

Key words machine vision; convolution neural network; object detection; feature pyramid; feature fusion

OCIS codes 150.0155; 150.1135; 100.4996

1 引 言

目标检测是机器视觉领域中的一个基础任务，而且已成为该领域中的一个研究热点。相较于图像分类[1]而言，基于卷积神经网络(CNN)[2-3]的目标检测算法需要使用一个网络来完成分类和定位两个任务。随着研究的深入，许多高效的检测算法已应用在日常生活中，如行人检测、自动驾驶和实时视频分析等。

目前，基于 CNN 的目标检测算法根据模型结构主要分为两类。第一类是单阶段检测算法，包含 YOLO[4]、YOLO9000[5]、YOLOv3[6]、DSSD[7]、FSSD[8]和 SPAD[9]等，这类算法利用 CNN 来获取一系列不同尺度的特征，再利用这些特征检测多个

收稿日期：2020-06-02；**修回日期**：2020-06-18；**录用日期**：2020-07-03

* **E-mail**：Polaris963@163.com

尺寸的目标。第二类是双阶段检测算法，包含 R-CNN[10]、Fast R-CNN[11]、Faster R-CNN[12]、Mask R-CNN[13]、Cascade R-CNN[14] 和 Libra R-CNN[15] 等，这类算法在输入的图像上生成一系列的候选区域，再对每个候选区域进行类别分类和位置回归。

在单阶段检测算法中，RetinaNet[16] 和 Libra RetinaNet 因具有较高的检测精度与效率而受到了广泛的关注。然而，上述两个检测器的网络结构存在两个问题：1）残差网络（ResNet）[1] 第一级输出的特征未输入特征金字塔网络（FPN）[17] 中进行融合；2）为了检测大尺寸目标，在分类网络最深层特征的基础上额外生成两个特征，这两个特征在未与其他特征融合的情况下直接用于预测。若特征融合得不充分，则导致检测器精度受限。

为了解决上述两个问题，本文提出一种多尺度特征融合算法。相关研究表明，深层特征中含有较多的语义信息，浅层特征中含有较多的细节信息[18]。鉴于此，本文设计两个独立的特征融合模块，即 LFF（Low-level Feature Fusion）模块和 HFF（High-level Feature Fusion）模块。LFF 模块的作用是增加浅层预测特征中的细节信息，HFF 模块的作用是融合两个具有强语义信息的深层预测特征。首先将残差网络第一级输出的特征输入 LFF 模块中，输出的两个特征与原始特征中的两个浅层特征相加，将原始预测特征中尺度最大的特征输入 HFF 模块中，输出的两个特征与原始特征中的两个深层特征相加，从而生成新的预测特征，然后将新的预测特征送入分类和回归网络中得到最终的预测结果，最后在 PASCAL VOC 数据集和 MSCOCO 数据集[19] 上进行对比实验。

2　多尺度特征融合算法

2.1　特征融合

基于多尺度特征融合的网络结构如图 1 所示，其中⊕为融合操作。假设 $X_i(i=1,2,3,\cdots,n)$ 和 $X_f(f=1,2,3,\cdots,n)$ 分别表示原始特征和融合特征，特征融合算法[8] 可以表示为

$$X_f=\phi_f[\varphi_i(X_i)],\tag{1}$$

$$S_{\mathrm{Loc(Class)}}=\psi_g[\theta_p(X_f)],\tag{2}$$

式中：$p=1,2,3,\cdots,n$；$g=1,2,3,\cdots,n$；$\varphi_i(\cdot)$ 表示转换函数；$\phi_f(\cdot)$ 表示特征融合函数；$\theta_p(\cdot)$ 表示预测特征的生成函数；$\psi_g(\cdot)$ 表示最终的预测函数；$S_{\mathrm{Loc(Class)}}$ 表示位置预测结果（类别预测结果）的得分。

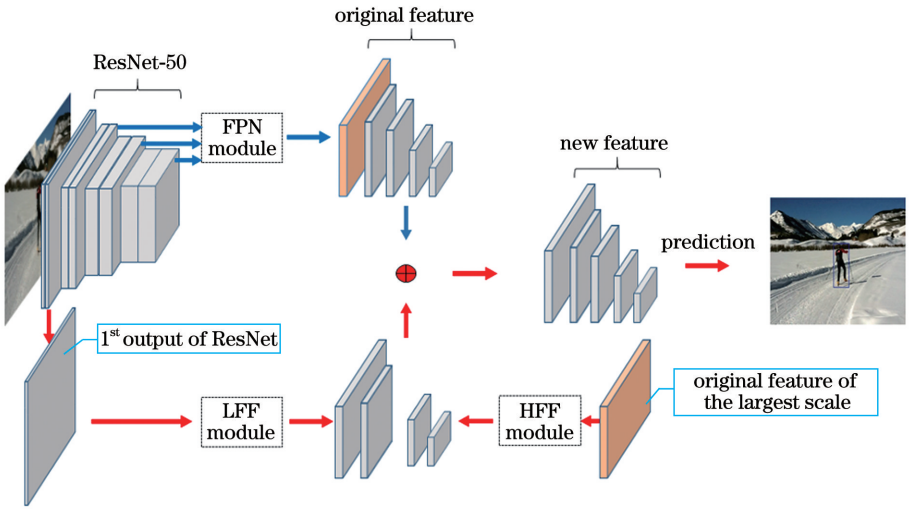

图 1　基于多尺度特征融合的网络结构

Fig. 1　Network structure based on multi-scale feature fusion

多数特征融合算法致力于寻找 φ_i 和 ϕ_f，使检测器的性能达到最优。φ_i 用来调整特征的尺寸，一般使用卷积或插值操作。ϕ_f 用来融合不同尺度的特征，主要使用拼接或相加操作。FPN 中，φ_i 使用的是 1×1 卷积，ϕ_f 可表示为

$$\phi_f=A_i\varphi_i(X_i)+B_{i+1}\xi[\varphi_{i+1}(X_{i+1})],\tag{3}$$

式中：ξ 表示最近邻插值；A 和 B 表示权重参数。

FPN、LFF 和 HFF 模块的网络结构，如图 2 所示。在 RetinaNet[16] 和 Libra RetinaNet[15] 检测器中，设 $X_1\sim X_4$ 为从残差网络中提取的 4 个不同尺度的特征，将 X_2、X_3 和 X_4 特征输入 FPN 中得到 P_2、P_3 和 P_4 三个融合特征，分别经过一个 3×3 卷积操作后生成预测特征，同时在 X_4 特征上经过一个 3×3 卷积操作后得到 P_5 融合特

征，在 P_5 融合特征上经过一个 3×3 卷积操作后得到 P_6 融合特征，最终有 5 个不同尺度的特征用于预测，如图 2(a)所示。上述设计过程会导致特征融合不充分，具体原因：1）X_1 特征未输入 FPN 中，由文献[17]可知，当 X_1～X_4 特征均输入 FPN 中时，检测器的性能达到最佳；2）P_5 和 P_6 两个融合特征仅由 X_4 特征生成，信息过于单一。

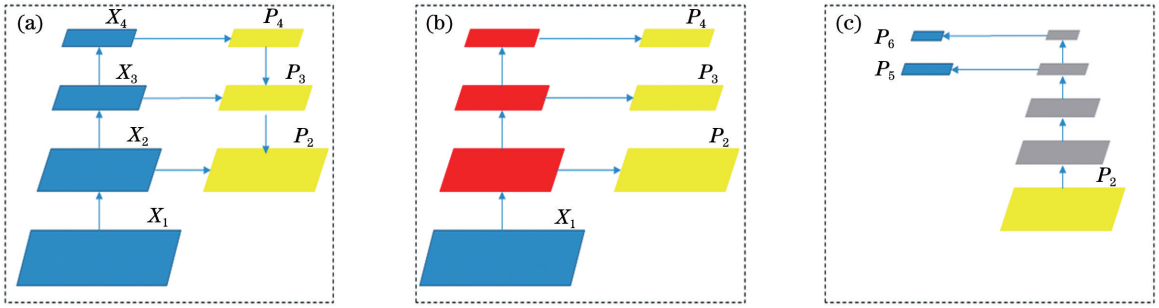

图 2　不同特征融合模块的网络结构。(a)FPN 模块；(b)LFF 模块；(c)HFF 模块

Fig. 2　Network structure of different feature fusion modules. (a) FPN module; (b) LFF module; (c) HFF module

从图 2 可以看到，FPN 模块采用一种自上而下和侧面连接的方式来融合相邻尺度的特征，LFF 模块和 HFF 模块均采用两条全新的自底向上的方式来融合特征，区别是 LFF 模块以残差网络第一级输出的特征为起点，每次卷积后的结果分别加到由 FPN 模块得到的融合特征上，HFF 模块则以最大尺度的融合特征为起点，将最后两次卷积得到的结果加到额外的两个深层特征上。

为了最优化上述设计，本文将注意力放在很少被关注的特征选择上。LFF 模块的作用是增加预测特征上的细节信息，最终用于预测的特征多达 5 个。通过研究发现，从 5 个特征中选择不同的组合方式并与 LFF 模块中的特征进行融合，这对检测器的性能有很大的影响。本文选择 $\{P_2, P_3\}$ 与 LFF 模块中的特征进行融合，原因在于使用 HFF 模块将浅层特征的信息融合到 $\{P_5, P_6\}$ 上，如果将 LFF 模块的特征再加到这两个特征上会造成浅层特征被重复融合，这会丢失特征上的语义信息。

2.2　训练细节

随着神经网络的加深，张量尺寸会持续减小。为了弥补细节上的损失，张量的通道数量会随之增加，如 VGG(Visual Geometry Group)[20] 网络最后一层通道的数量为 512，而 ResNet[1] 则达到 1024。检测器中，除特征提取网络(骨干网络)之外，将其他模块的通道数量作为一种超参数，这是影响检测器性能的因素之一。通道数量过少，则检测器无法充分学习到整个数据集的特征。通道数量过多，则导致整个网络发生过拟合，降低检测器的泛化能力。通过不同的对比实验发现，对于规模较小的数据集，参数设为 256，对于大型数据集，参数设为 512，这可以使检测器的性能达到最优。为了简单起见，特征融合模块的权重参数全部设为 1，所有的对比实验均以 Libra RetinaNet 为基准。

3　实验结果及分析

为了验证多尺度特征融合算法的有效性，本文在 PASCAL VOC 数据集和 MSCOCO 数据集上进行对比实验。实验环境选择基于 PyTorch 的 MMDetection[21] 目标检测工具箱，以及 Nvidia RTX2080TI GPU。为了公平比较，所有预训练模型均由 PyTorch 官方提供。另外，本文通过消融学习来分析每个模块对检测精度的影响。

3.1　在 PASCAL VOC 数据集上的实验

PASCAL VOC 数据集是一个经典的目标检测数据集。本文使用 trainval-2007 和 trainval-2012 的数据训练模型，使用 test-2007 的数据对模型进行测试。实验过程中，有 16551 张图片用于训练，有 4952 张图片用于测试。PASCAL VOC 数据集的评价指标为均值平均精度(mAP)和平均精度(AP)。

FPN 模块和预测模块的通道数量均设为 256。在 1 块 GPU(骨干网络为 ResNet-50)上训练模型，迭代次数设为 30 个周期(epoch)，初始学习率设为 0.001，迭代 22 个和 26 个周期后的学习率分别降低 10%。训练过程中，批大小(batch size)设为 4。每张图像的尺寸为 300 pixel×300 pixel。其他的超参数均是 MMDetection 的默认设置。采用图像旋转和颜色变化等方法对数据进行增强。图 3 为在 Libra RetinaNet 中加入 HFF 模块和 LFF 模块前后的损失函数曲线。从图 3 可以看到，经过 1.2×10^5 次迭代(30 个周期)后，损失值趋于稳定。

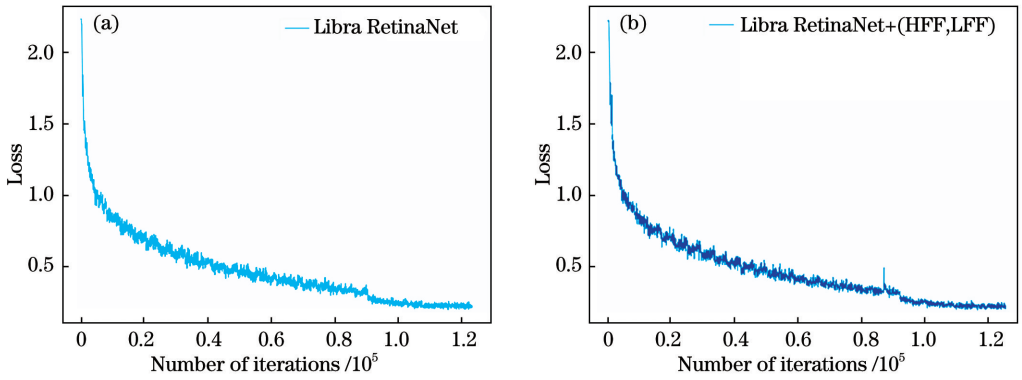

图 3　加入特征融合模块前后的损失函数曲线。(a)加入前;(b)加入后

Fig. 3　Loss function curves before and after adding feature fusion module. (a) Before adding; (b) after adding

表 1 为 4 种目标检测算法的对比结果。从表 1 可以看到，SSD[22]、RetinaNet[16] 和 Libra RetinaNet[15] 的 mAP 值分别为 69.7%、70.2% 和 70.4%，在 Libra RetinaNet 中加入 HFF 模块和 LFF 模块后，检测器的 mAP 值达到 72.6%，说明加入模块后的 mAP 有 2.2 个百分点的提升。

表 1　不同检测算法在 PASCAL VOC 数据集上的 mAP 值

Table 1　mAP values of different detection algorithms on PASCAL VOC dataset

Algorithm	Backbone network	mAP /%
SSD300	VGG-16	69.7
RetinaNet	ResNet-50	70.2
Libra RetinaNet	ResNet-50	70.4
Proposed algorithm	ResNet-50	72.6

表 2 为每一类目标具体的 AP 值。从表 2 可以看到，除了 Bike、Mbike 和 Car 三类目标外，其他目标的检测精度均有所提升。

表 2　不同类别的 AP 值

Table 2　AP values of different categories　　unit: %

Category	SSD300	RetinaNet	Libra RetinaNet	Proposed algorithm
Aero	75.7	75.4	75.4	**77.4**
Bike	78.4	80.2	79.6	80.1
Bird	67.2	72.1	72.1	**73.5**
Boat	64.4	60.7	64.8	**67.3**
Bottle	38.9	39.4	37.9	**43.1**
Bus	79.9	78.9	79.4	**81.1**
Car	83.3	79.0	79.0	80.3
Cat	84.6	86.4	85.1	**87.4**
Chair	49.5	52.5	52.6	**54.9**
Cow	67.1	64.1	68.0	**74.1**
Table	63.7	67.7	66.5	**68.5**
Dog	79.2	80.1	80.5	**82.2**
Horse	80.5	79.4	78.9	**81.4**
Mbike	79.7	78.2	78.1	79.7
Person	75.2	74.2	74.5	**75.6**
Plant	37.2	43.0	42.3	**45.3**
Sheep	69.9	68.4	70.4	**70.9**
Sofa	68.5	71.3	71.3	**72.0**
Train	81.6	83.1	83.7	**84.1**
TV	69.3	70.3	68.8	**72.7**

为了分析 HFF 模块和 LFF 模块分别对检测精度的影响，本文以 Libra RetinaNet 检测器为基础进行消融学习，实验过程中，每次仅添加一个模块。表 3 为消融学习的结果。从表 3 可以看到，加入 HFF 模块和 LFF 模块后的 mAP 值分别为 72.0% 和 72.2%，分别提升 1.6 和 1.8 个百分点。

表 3　消融学习的结果

Table 3　Results of ablation study

HFF module	LFF module	mAP /%
		70.4
√		72.0
	√	72.2
√	√	72.6

表 4 为消融学习后每个类别的 AP 值。从表 2 和表 4 可以看到，加入 HFF 模块和 LFF 模块可以检测更多的大目标和小目标，如 Aero 的 AP 值从 75.4% 提升到了 77.9%，Bottle 的 AP 值从 37.9% 提升到了 40.7%。

表 4　消融学习后每个类别的 AP 值

Table 4　AP values of each category after ablation learning

unit：%

Category	LFF module	HFF module
Aero	76.5	77.9
Bike	80.2	80.5
Bird	74.1	72.6
Boat	66.0	66.6
Bottle	40.7	41.9
Bus	78.7	82.4
Car	80.2	79.4
Cat	86.1	86.8
Chair	54.9	53.2
Cow	70.9	69.2
Table	70.2	68.3
Dog	82.2	82.2
Horse	82.5	82.5
Mbike	79.2	80.5
Person	75.0	75.0
Plant	44.7	45.0
Sheep	69.8	70.7
Sofa	73.5	73.3
Train	83.9	83.4
TV	70.5	71.6

图 4 为 Libra RetinaNet 中加入特征融合模块前后在 PASCAL VOC 数据集上的可视化检测结果对比。图 4(a)为原始 Libra RetinaNet 的检测结果，图 4(b)为在 Libra RetinaNet 中加入特征融合模块后的检测结果。从图 4 可以看到，特征融合模块不仅可以解决漏检和错检的问题，而且对目标的定位更准确。

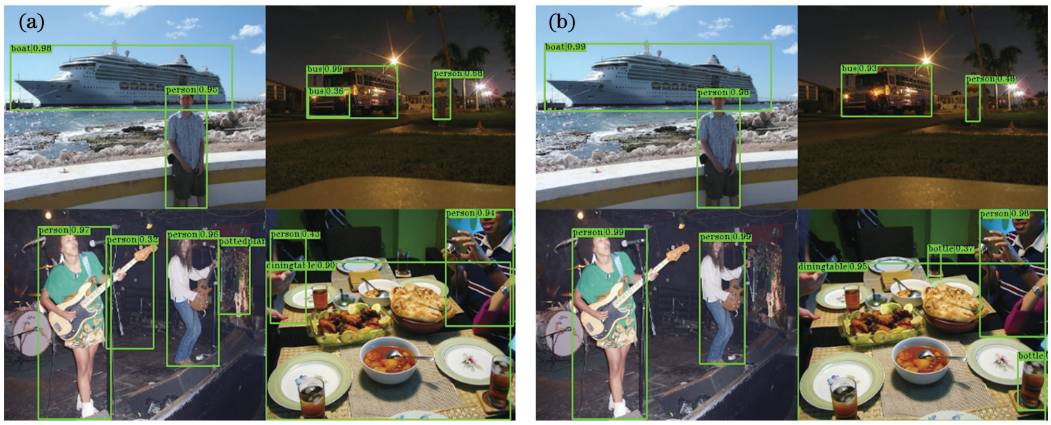

图 4　加入特征融合模块前后的检测结果。(a)加入前；(b)加入后

Fig. 4　Detection results before and after adding feature fusion module. (a) Before adding; (b) after adding

3.2　在 MSCOCO 数据集上的实验

为了进一步验证融合模块的有效性，在 MSCOCO 2017 数据集上进行对比实验。该数据集包含 80 类目标，1.15×10^5 张训练图像(train-2017 数据集)，5×10^3 张验证图像(val-2017 数据集)。为了综合评价算法的性能，该数据集根据交并比(IoU)和目标尺寸对评价指标进行进一步划分。根据 IoU 阈值的不同，划分为 AP_{50}(IoU 阈值为 0.5 的 AP)，AP_{75}(IoU 阈值为 0.75 的 AP)；根据目标尺寸的不同，划分为 AP_S(检测小目标的 AP)，AP_M(检测中等目标的 AP)，AP_L(检测大目标的 AP)。

FPN 模块和预测模块的通道数量均设为 512。训练过程中，批大小设为 16。在 4 块 GPU(每块 GPU 上 4 张图像，骨干网络为 ResNet-50)和 8 块 GPU(每块 GPU 上 2 张图像，骨干网络为 ResNet-101)上训练模型，迭代次数设为 12 个周期，初始学习率设为 0.01，在 8 个和 11 个周期后的学习率分别降低 10%。每张图像在保持长宽比的情况下，将尺寸缩放为 1333 pixel×800 pixel。其他的超参数均是 MMDetection 的默认设置。

表 5 为 4 种检测算法的结果。从表 5 可以看到，当以 ResNet-50 为骨干网络时，加入 HFF 模块和 LFF 模块后的 AP 值达到 38.8%，与 Libra RetinaNet 相比提升 1.3 个百分点，AP_S 值达到了 22.9%；当以 ResNet-101 为骨干网络时，加入 HFF 模块和 LFF 模块后的 AP 值达到 40.3%，与 Libra RetinaNet 相比提升 1.2 个百分点；不同骨干网络的 AP 值均有所提升，说明加入两个特征融合模块后的检测器具有较强的泛化能力。

表 5 不同检测算法在 MSCOCO 数据集上的 AP 值

Table 5 AP values of different detection algorithms on MSCOCO dataset unit: %

Algorithm	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SSD512	VGG-16	25.7	44.1	26.6	9.2	29.0	39.0
RetinaNet	ResNet-50	35.6	55.6	38.1	20.8	39.5	46.1
Libra RetinaNet	ResNet-50	37.5	56.9	39.9	22.4	41.4	49.2
Proposed algorithm	ResNet-50	**38.8**	**58.5**	**41.3**	**22.9**	**42.6**	**50.4**
RetinaNet	ResNet-101	37.8	57.5	40.8	20.9	42.1	49.6
Libra RetinaNet	ResNet-101	39.1	58.6	41.7	22.6	43.8	51.4
Proposed algorithm	ResNet-101	**40.3**	**59.9**	**42.9**	**23.1**	**44.8**	**53.3**

表 6 为在 MSCOCO 数据集上消融学习后的结果，实验采用与 4.1.3 节相同的方式进行。从表 6 可以看到，加入 HFF 模块和 LFF 模块后，检测器的 AP 值都达到 38.5%，与未加入模块相比提升 1.0 个百分点；HFF 模块将 AP_L 值从 49.2%提升到 50.2%；LFF 模块将 AP_S 值从 22.4%提升到 23.8%，说明两个模块能够分别提高对大目标和小目标的检测精度。

表 6 在 COCO val-2017 上消融学习后的结果

Table 6 Results after ablation learning on COCO val-2017 unit: %

HFF module	LFF module	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
		37.5	56.9	39.9	22.4	41.4	49.2
√		38.5	58.3	41.0	22.4	42.6	50.2
	√	38.5	58.4	40.9	23.8	42.4	49.7
√	√	38.8	58.5	41.3	22.9	42.6	50.4

4 结 论

为了解决 RetinaNet 和 Libra RetinaNet 检测器中特征融合不充分的问题，本文提出一种基于多尺度特征融合的目标检测算法，该算法通过建立两个独立的特征融合模块并分别融合深层特征和浅层特征。在 PASCAL VOC 和 MSCOCO 数据集上进行对比实验，实验结果表明，两个多尺度特征融合模块能够使检测器获得更高的检测精度。

参 考 文 献

[1] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 770-778.

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// Proceedings of 2012 Neural Information Processing Systems (NIPS), December 3-6, 2012, Lake Tahoe, Nevada, United States. 2012: 1097-1105.

[3] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation, 1989, 1(4): 541-551.

[4] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 779-788.

[5] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 6517-6525.

[6] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08)[2020-06-01]. https://arxiv.org/abs/1804.02767.

[7] Fu C Y, Liu W, Ranga A, et al. DSSD: deconvolutional single shot detector[EB/OL]. (2017-01-23)[2020-06-01]. https://arxiv.org/abs/1701.06659.

[8] Li Z X, Zhou F Q. FSSD: feature fusion single shot multibox detector[EB/OL]. (2018-05-17)[2020-06-01]. https://arxiv.org/abs/1712.00960.

[9] Zhu C C, Chen F Y, Shen Z Q, et al. Soft anchor-point object detection[EB/OL]. (2019-11-27)[2020-06-01]. https://arxiv.org/abs/1911.12448.

[10] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE, 2014: 580-587.

[11] Girshick R. Fast R-CNN[C]// 2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 1440-1448.

[12] 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(6): 1137-1149.

[13] He K M, Gkioxari G, Dollár P, et al. Mask R-CNN [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386-397.

[14] Cai Z W, Vasconcelos N. Cascade R-CNN: delving into high quality object detection[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 6154-6162.

[15] Pang J M, Chen K, Shi J P, et al. Libra R-CNN: towards balanced learning for object detection[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 821-830.

[16] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.

[17] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 936-944.

[18] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[M]// Fleet D, Pajdla T, Schiele B, et al. Computer vision-ECCV 2014. Lecture notes in computer science. Cham: Springer, 2014, 8689: 818-833.

[19] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: common objects in context[M]// Fleet D, Pajdla T, Schiele B, et al. Computer vision-ECCV 2014. Lecture notes in computer science. Cham: Springer, 2014, 8689: 740-755.

[20] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2015-04-10)[2020-06-01]. https://arxiv.org/abs/1409.1556.

[21] Chen K, Pang J Q, Wang J M, et al. MMDetection [EB/OL]. [2020-06-01]. https://github.com/open-mmlab/mmdetection.

[22] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[EB/OL]. (2016-12-29)[2020-06-01]. https://arxiv.org/abs/1512.02325.