计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200400090

# 一种新颖的单目视觉深度学习算法:H_SFPN

石先让[1] 宋廷伦[1,2] 唐得志[2] 戴振泳[1]
1 南京航空航天大学能源与动力学院 南京 210001
2 奇瑞前瞻与预研技术中心 安徽 芜湖 241006
(nuaasxr@163.com)

**摘 要** 针对单目视觉目标检测,提出了一种基于 single-stage 深度学习的 H_SFPN 算法。该算法与现有的 YOLOv3 和 CenterNet 算法相比,在保证实时性能的条件下,可有效提高小目标检测的准确度。首先设计了一种新的网络架构(backbone),这种架构通过改进的沙漏(Hourglass)网络模型来提取特征图,以便充分利用底层特征的高分辨率以及高层特征的高语义信息。然后在特征图融合阶段提出了基于 SFPN 的特征图加权融合方法。最后,H_SFPN 算法对目标位置和大小的损失函数进行了改进,可有效降低训练误差,并加快收敛速度。由 MSCOCO 数据集上的实验结果可知,所提 H_SFPN 算法明显优于 Faster-RCNN,YOLOv3 以及 EfficientDet 等现有的主流深度学习目标检测算法,其中对小目标的检测指标 $AP_s$ 最高,达到了 32.7。

**关键词**:深度卷积神经网络;目标检测;加权融合;网络架构;损失函数
**中图法分类号** TP391.41;TN219

## Novel Deep Learning Algorithm for Monocular Vision:H_SFPN

SHI Xian-rang[1],SONG Ting-lun[1,2],TANG De-zhi[2] and DAI Zhen-yong[1]
1 College of Energy and Power Engineering,Nanjing University of Aeronautics and Astronautics,Nanjing 210001,China
2 Chery Advanced Engineering & Technology Center,Wuhu,Anhui 241006,China

**Abstract** This paper proposes a single-stage deep learning based H_SFPN algorithm for monocular visual object detection. Compared with the existing YOLOv3 and CenterNet algorithms,the proposed algorithm can effectively improve the accuracy of small object detection without sacrificing the real-time performance. This paper designs a new network architecture (backbone),which uses an improved Hourglass network model to extract feature maps in order to make full use of the high resolution of the underlying features and the high semantic information of the high-level features. Then in the feature map fusion stage,a method SFPN based on the weighted fusion of feature maps is proposed. Finally,the proposed H_SFPN algorithm improves the loss function of the object position and size,which can effectively reduce the training error and accelerate the convergence speed. According to the experimental results on the MSCOCO data set,the proposed H_SFPN algorithm is significantly better than the existing mainstream deep learning object detection algorithms such as Faster-RCNN,YOLOv3 and EfficientDet. Among them,the small object detection index $AP_s$ of this algorithm is the highest,reaching 32.7.

**Keywords** Deep convolutional neural network,Object detection,Weighted fusion,Backbone,Loss function

## 1 引言

目标检测是计算机视觉领域中的一个重要研究课题,被广泛用于智能视频监控、自动驾驶、智能机器人等领域。目标检测是在单帧图像或者一段视频帧中快速准确地识别出目标的类型、位置等属性。准确性和实时性是目标检测算法的重要评价指标,如何有效提高目标检测算法的准确性和实时性是目前的研究重点和难点。

传统的目标检测方法通过滑动窗口提取候选区域,并通过人工设计的方法提取特征[1-2],最后通过分类器[3]进行分类识别。但传统的目标检测方法存在一些不可避免的问题[4]:1)基于滑动窗口的区域选择方法是一种穷举的策略,时间复杂度高,窗口冗余;2)人工提取特征的方法对于目标的多样性变化没有很好的鲁棒性。这些弊端极大地限制了传统目标检测算法的实际应用。

深度神经网络的出现有效解决了人工提取特征效率低的问题,原因是它可以通过大量的数据训练自主地学习有用的图像特征[5]。2012 年,Krizhevsky 等[6]首先将深度神经网络运用在 ImageNet 数据集[7]上,取得了当时最好的成绩。自此,通过深度神经网络进行目标检测引起了广泛的关注。

到稿日期:2020-04-20 返修日期:2020-07-29 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:安徽省发改委重大研发项目
This work was supported by the Anhui Provincial Development and Reform Commission's Major R&D Project.
通信作者:宋廷伦(songtinglun@nuaa.edu.cn)

近年来，随着深度神经网络算法和硬件的不断发展，目标检测算法也取得了长足的发展。目前基于深度神经网络的目标检测算法可分为两类：1）two-stage 算法，如 Faster r-cnn[8-10]等；2）one-stage 算法，如 SSD[11]，Yolo[12-14]等。one-stage 算法直接对输入图像进行计算，输出类别和相应的定位，因此其实时性明显优于其他算法，这在对实时性要求高的自动驾驶领域十分重要。2018 年，Redmon 等[14]提出了 YOLOv3 算法，该算法实现了端到端（end to end）训练，是当时综合性能最好的 one-stage 算法。Tan 等[15]于 2019 年提出了 EfficientDet 算法，使目标检测能力得到了进一步的提高。

目前，基于深度学习的目标检测算法虽然取得了一定的进展，但仍存在一些不足。例如，one-stage 算法虽然实时性较好，但是对目标检测的精度不够理想，对小目标检测的准确性也有待进一步提高。为了增强对小目标的检测能力，一些算法运用特征金字塔（Feature Pyramid Network，FPN）[16]来进行特征图的融合，但牺牲了部分实时性。

综上，现有算法在目标检测的准确性和实时性上还有待进一步提高，以满足自动驾驶等领域的需求。本文在深入研究 Yolo 系列算法的基础上，基于 one-stage 深度学习算法的模型框架，通过对网络架构（Backbone）的巧妙设计和对损失函数的改进，提出了综合性能更好的 H_SFPN 算法。下面将从网络架构（Backbone）设计和损失函数改进这两个维度来详细介绍 H_SFPN 算法。

## 2 网络架构设计

### 2.1 Hourglass 的改进

目标检测算法所采用的 Backbone 的性能对其检测的实时性和准确性有着很大的影响。文献[13]设计了 DarkNet-19 深度网络作为 Yolov2 算法的 Backbone，使其性能相比 Yolov1[11]有很大提升。在 YOLOv3[14]中，Redmon 等在 Darknet-19 网络的基础上进行了拓展，设计了 DarkNet-53 深度网络，该深度网络的综合性能优于 ResNet-101 和 ResNet-152 残差网络[17]。对 Zhou 等[18]提出的 CenterNet 算法采用不同的 Backbone，其目标检测的准确性和实时性会有很大的差异。采用 Hourglass-104[19]作为 Backbone 时，算法目标检测的准确性最高，实时性最低；采用 DLA-18[20]网络时，算法目标检测的准确性最低，实时性最高。本节以 Hourglass-104 中的 Hourglass 模型为基础，通过优化其架构提出一种改进的 Hourglass 模型，并利用改进的 Hourglass 模型设计本文算法 H_SFPN 的网络架构（Backbone）。

Hourglass 模型[19]最初是用于对网络最终输出的汇集和后续采样步骤的可视化。与许多产生像素级输出的卷积方法一样，通过 Hourglass 模型将卷积提取的特征汇聚到一个非常低的分辨率，然后对多个分辨率的特征进行上采样和组合。本节在深入研究该模型的基础上，优化和改进了原 Hourglass 模型的内部结构，提出了参数更少、计算效率更高的改进的 Hourglass 模型。

图 1 为本文提出的改进的 Hourglass 模型。其中，红色矩形框代表一个卷积层（Conv）；浅灰色矩形框代表 BN 层（Batch Normalization）；黄色矩形框代表非线性激活函数 Leaky relu 的激活层；橙色矩形框代表上采样层（Upsample）；白色矩形框代表神经网络层的相加（Add）；浅绿色矩形框代表 1×1 卷积层（1×1 Conv）；绿色圆形代表特征图拼接（Concat）后进行卷积操作；蓝色矩形表示由卷积层、BN 层和激活层构成的卷积块；深蓝色矩形表示由卷积块构成下采样的残差网络模块；紫色矩形表示由上采样和卷积块构成的残差网络模块；淡蓝色矩形表示由 1×1 卷积层和卷积块构成的旁路卷积。

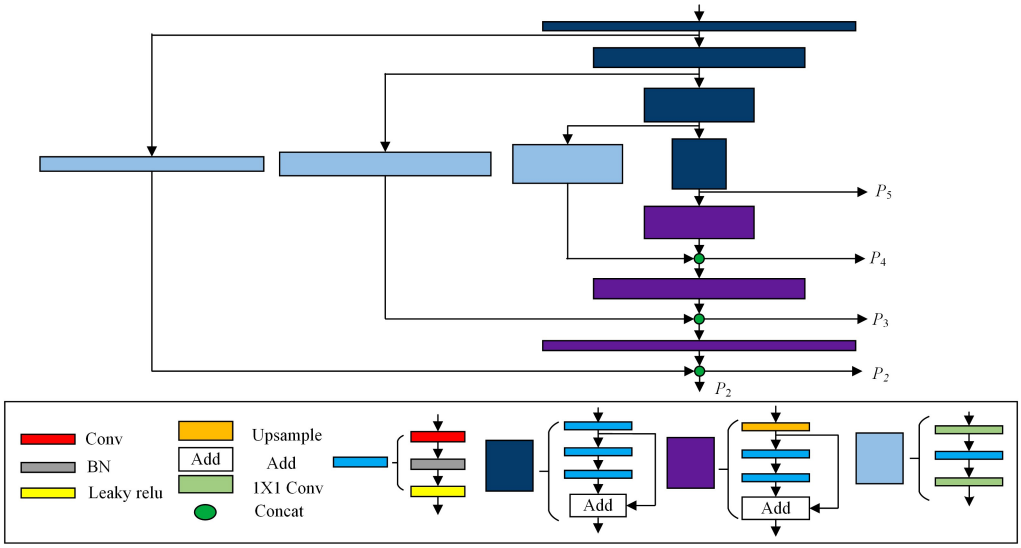



图 1 改进的 Hourglass 模型（电子版为彩色）

Fig. 1 Improved Hourglass model

在改进的 Hourglass 模型中，主要的卷积核都采用 3×3 大小，在旁路中的卷积核采用 1×1 Conv 卷积核进行调整。在卷积运算中，通过卷积步长 $stride=2$ 来实现降采样，舍去池化层。

在原 Hourglass 模型中[19]，输入为 4 倍下采样，依次经过 8 倍、16 倍、32 倍、64 倍和 128 倍下采样，再经过上采样进行输出。本文改进的 Hourglass 模型舍去了 64 倍、128 倍下采样，只采用 4 倍、8 倍、16 倍和 32 倍下采样，再经过上采样进

行输出。这种方法有效减少了参数量和计算量，从而提高了模型提取特征的实时性。

在输入图像大小为 416×416×3(宽($w$)×高($h$)×通道数($c$))时，改进的 Hourglass 模型的输入张量大小为 208×208×128，通过一系列的残差块实现特征提取和下采样，然后输出 $P_5$ 特征图，再经过一系列的上采样和残差块的特征提取，分别输出 $P_4$，$P_3$，$P_2$ 特征图。其中 $P_2$ 的输出有两支：一支作为下一个 Hourglass 模型的输入；另一支将同 $P_5$，$P_4$，$P_3$ 进行特征图的融合。

每个 Hourglass 模型均输出 4 个特征图 $P_2$，$P_3$，$P_4$，$P_5$，它们的下采样倍数分别为 $2^2$，$2^3$，$2^4$，$2^5$。换句话说，当输出的特征图为 $P_i$ 时，输出特征图的长宽是原输入图像的 $1/2^i$。

这里，$P_5$，$P_4$，$P_3$，$P_2$ 特征图的尺寸大小分别为：13×13×1 024，26×26×512，52×52×512，104×104×256。

### 2.2 特征图的融合策略

现有的目标检测算法大多只使用最后一层的特征图来进行目标检测，这样容易造成特征丢失。本文采用多个特征图进行多尺度的目标检测，尤其加强了对小目标的检测。借鉴 Tan[15] 提出的 BiFPN 特征图融合策略，本文设计了基于 SFPN 架构的特征图融合算法。

图 2 为本文所设计的 SFPN 架构原理图。图中蓝色箭头为上采样，红色箭头为下采样，紫色箭头为残差连接，圆形为卷积操作。本文的 SFPN 特征图融合算法只采用了 4 个特征图进行融合，在特征图融合的同时采用双向交叉连接和快速归一化融合，但是为了提高对小目标的检测能力，对 $P_2$ 特征图增加了一层卷积计算，同时增加了 $P_3$ 到 $P_2$ 的上采样融合。

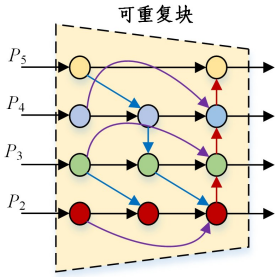



图 2 SFPN 架构(电子版为彩色)

Fig. 2 Structure of SFPN

特征图的融合方法类比 BiFPN。在图 2 中：

$$P_4^{td}=Conv\left(\frac{\omega_1\cdot P_4^{in}+\omega_2\cdot Resize(P_5^{in})}{\omega_1+\omega_2+\varepsilon}\right)$$

$$P_3^{td}=Conv\left(\frac{\omega_1'\cdot P_3^{in}+\omega_2'\cdot Resize(P_4^{td})}{\omega_1'+\omega_2'+\varepsilon}\right)$$

$$P_3^{out}=Conv\left(\frac{\omega_1''\cdot P_3^{in}+\omega_2''\cdot P_3^{td}+\omega_3''\cdot Resize(P_2^{out})}{\omega_1''+\omega_2''+\omega_3''+\varepsilon}\right) \tag{1}$$

最后，对两个 Hourglass 模型输出的特征图进行可学习的权重融合，其权重值的大小由网络学习得到，融合方式如图 3 所示。

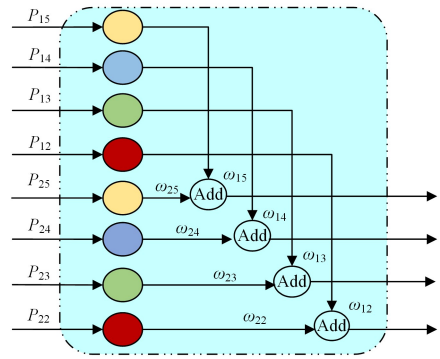



图 3 对输出的特征图进行权重融合

Fig. 3 Weight fusion of output feature map

在图 3 中，$P_{ij}$ 代表第 $i$ 个 Hourglass 模型中第 $j$ 个特征图；$\omega_{ij}$ 代表第 $i$ 个 Hourglass 模型中第 $j$ 个特征图的权重；Add 代表特征图相加。

对两个 Hourglass 模型的特征图进行权重融合时，采用快速归一融合法，其公式如下：

$$P_{out}=\frac{\sum\omega_{ij}\cdot P_{ij}}{\sum\omega_{ij}+\varepsilon} \tag{2}$$

式(2)中，可用 ReLU 来保证 $\omega\geqslant 0$，设置一个极小值 $\varepsilon=0.0001$，避免分母为 0 时计算结果不稳定的状况。

### 2.3 网络架构(Backbone)的构建

本文设计了新的特征提取的网络架构，即 Backbone，它以 Hourglass 模型为基础，主要由 2 个 Hourglass 模型堆栈而成。为了测试该 Backbone 特征提取的性能，在 ImageNet 数据集[7]上进行了分类实验测试。

在该实验中，将输入图像的大小调整为 416×416，输入图像首先经过 32 个 3×3 的卷积核提取特征，卷积步长 $stride=1$；接着经过 64 个 3×3 的卷积核提取特征，卷积步长 $stride=1$；然后经过 64 个 3×3 的卷积核提取特征，卷积步长 $stride=2$；再经过 128 个 3×3 的卷积核提取特征，卷积步长为 $stride=1$，输入到 Hourglas 模块中，经过一个 Hourglass 模块，输出 4 个特征图；最后将这 4 个特征图按照本文的特征图融合策略(SFPN)进行特征图融合，其融合方式如图 2 和式(1)所示。

文中对这 4 个特征图进行了两次融合，如图 4 所示。此外，继续对 $P_2$ 特征图进行卷积操作，生成 208×208×128 的张量，然后再次输入到下一个 Hourglass 模型，经过处理后，得到 4 个特征图并进行输出，同样按照本文提出的特征图融合策略(SFPN)进行融合。

在该 Backbone 中，经过堆栈的两个 Hourglass 模型通过本文设计的特征图融合策略(SFPN)融合后，输出 8 个处理后的特征图，然后对这 8 个处理后的特征图进行加权融合，如式(2)所示。最后经卷积层、平均池化层和全连接层输出分类。

由于本文中的 Backbone 采用了 2 个 Hourglass 进行堆栈，因此称其为 H_2。为了测试采用 2 个 Hourglass 模型堆栈和重复 2 个 SFPN 的合理性，将在 4.1 节进行实验验证。

H_2 整体模型的结构如图 4 中的虚线框所示。

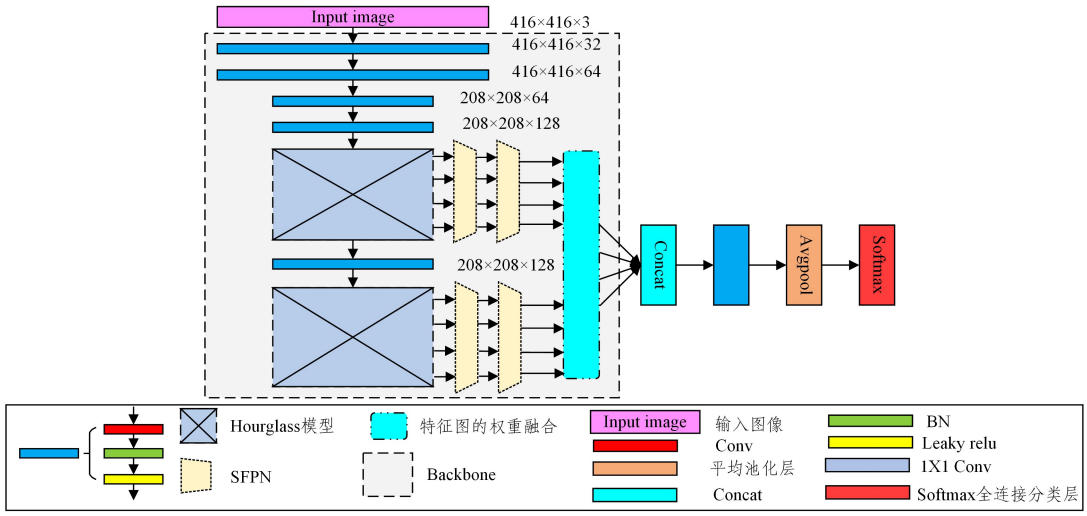



图 4　基于 H_2 Backbone 的分类模型

Fig. 4　Classification model based on H_2 Backbone

## 3　H_SFPN 算法

### 3.1　整体算法

受文献[14]中 YOLOv3 的启发，本文在目标检测中对于目标的定位还是基于锚框(anchor)来实现，其锚框定位原理与 YOLOv3 基本相同。

在 YOLOv3 中，Redmon 等[14]引入了 FPN[16]网络，目的是充分利用底层特征的高分辨率以及高层特征的高层特征丰富的语义信息。本文通过提出的 SFPN 特征图融合策略来进行不同特征层的融合，在 4 个不同尺度的特征图上检测目标。本文的目的是提高算法目标检测的准确性，同时加强对小目标的检测能力。

本文将原 YOLOv3 中的 Backbone 改用为 H_2 模型，同时对尺度检测模块进行了改进，将原有的 3 个尺度检测扩展为 4 个尺度检测，为特征图分配准确的锚点框，进而提高目标检测的精度。本文提出的 H_SFPN 目标检测算法如图 5 所示。

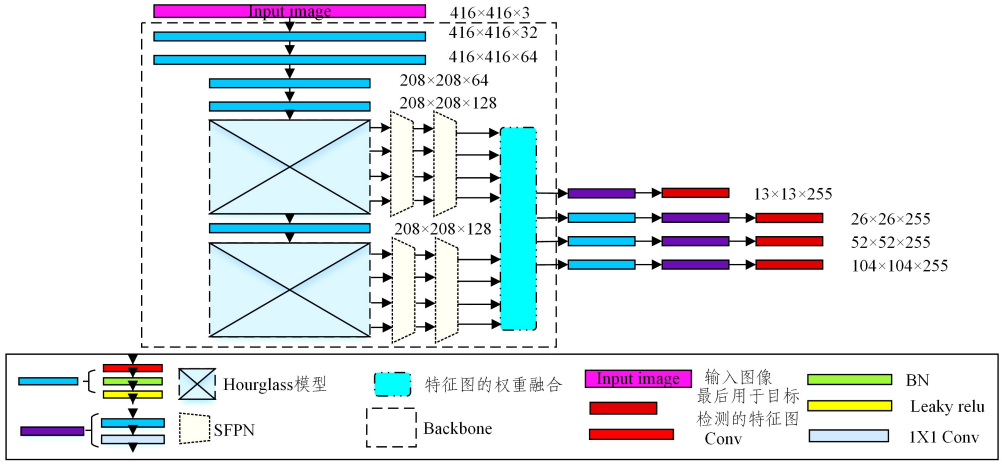



图 5　H_SFPN 算法

Fig. 5　H_SFPN algorithm

由图 5 可知，在特征图的权重融合之前 H_SFPN 采用了 H_2 模型，与图 4 相同。在对两个 Hourglass 输出的特征图进行权重融合之后，重新进行卷积操作，以便进行目标检测。

本文将权重融合之后的特征图再次进行卷积操作，然后通过 1×1 Conv 卷积操作来调整输出维度，最后输出 4 个待进行目标检测的特征图，维度分别为 104×104×255，52×52×255，26×26×255，13×13×255，其原理同 YOLOv3[14]。

### 3.2　锚框(anchor)的生成

锚框是目标边界框的先验，不同于 FasterRCNN[9]的锚框机制。锚框的参数为($p_w$，$p_h$)，代表锚框的宽和高。在 MSCOCO 数据集[22]上，本文采用 $k$-means 聚类算法对训练集的边界框进行聚类，取 $k=12$，生成恰当的边界框先验，即为锚框。

### 3.3　目标边框的预测

YOLOv3[14]中的每个单元格预测目标的位置($x$，$y$)、长宽($w$，$h$)、置信度 $p$，输出 $t_x$，$t_y$，$t_w$，$t_h$，$p$，然后通过以下坐标变换公式[13]得到最终的目标检测的边框 $b_x$，$b_y$，$b_w$，$b_h$。

$$\begin{aligned} b_x &= \sigma(t_x)+c_x \\ b_y &= \sigma(t_y)+c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{3}$$

其中，$c_x$ 和 $c_y$ 代表目标中心在单元格中相对于图像左上角的偏移，$p_w$ 和 $p_h$ 为定位锚框的宽和高。坐标变换示意图如图 6 所示。

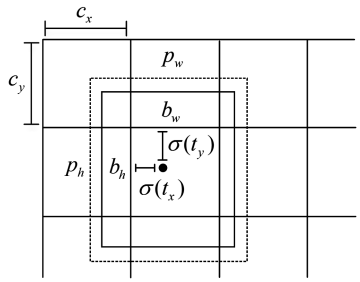



图 6 边框预测的坐标变换[13]

Fig. 6 Coordinate transformation of border prediction[13]

### 3.4 损失函数

损失函数(loss function)也称作代价函数(cost function)。损失函数用以评价模型的预测值和真实值不一样的程度，损失函数越小，则模型拟合得越好。

在 YOLOv3[14] 中，$x,y,w,h$ 的训练损失函数采用均方误差损失函数(Mean Square Error，MSE)，其中的类(class)和置信度(confidence)采用交叉熵损失函数。本文对该均方误差的损失函数进行了改进。

文中先定义一种损失函数，如式(4)所示：

$$loss=A\cdot|x_i-\hat{x}_i|^e+b \tag{4}$$

其中，$A$ 为系数；$e$ 为待定指数；$b$ 为常数项，目的是使函数连续。

然后，根据算法 1 来确定系数 $A$、指数 $e$ 和常数 $b$。

**算法 1** 确定系数 $A$、指数 $e$ 和常数 $b$

输入：$|x_i-\hat{x}_i|$

输出：A，e，b

1. if $|x_i-\hat{x}_i|\geqslant 1$：

$A=\frac{1}{2}$；

$e=2$；

$b=\frac{1}{2}$；

2. else：

$A=1$；

$e=1$；

$b=0$.

对于形如 $f(x)=A\cdot|x|^e+b$ 的函数，对指数 $e$、系数 $A$ 和常数 $b$ 进行确定之后(系数 $A$ 取 1/2 是为了使 $x=1$ 时 $f(x)$ 可微)，函数图像如图 7 所示。

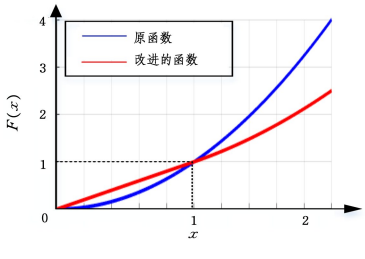



图 7 改进的函数和原均方误差函数的比较

Fig. 7 Comparison of the improved function and original MSE function

从图 7 可知，与原均方误差函数相比，当自变量小于 1 时，改进的函数值大于原均方误差函数。在这种情况下，在训练时，当 $loss_{x,y,w,h}$ 值减小到一定程度后，改进的函数的梯度大于原均方误差函数，在训练深度神经网络时有利于模型加速收敛。

因此，本文目标检测中的损失函数 $loss_{x,y,w,h}$ 如式(5)所示：

$$\begin{aligned}loss_{x,y,w,h}=&\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}(F(x_i,y_i))+\\&\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}(G(w_i,h_i))\end{aligned} \tag{5}$$

其中，$F(x_i,y_i)$ 和 $G(w_i,h_i)$ 根据式(4)分别表示如下：

$$F(x_i,y_i)=A\cdot|x_i-\hat{x}_i|^e+A\cdot|y_i-\hat{y}_i|^e+2\cdot b \tag{6}$$

$$G(w_i,h_i)=A\cdot|w_i-\hat{w}_i|^e+A\cdot|h_i-\hat{h}_i|^e+2\cdot b \tag{7}$$

其中，$x_i,y_i,w_i,h_i$ 为实际值，$\hat{x}_i,\hat{y}_i,\hat{w}_i,\hat{h}_i$ 为预测值。

关于类的损失和置信度的损失，采用 Lin 等[21] 提出的 Fcoal loss 函数，如式(8)所示：

$$loss_{\text{class,confidence}}=\frac{-1}{N}\begin{cases}(1-\hat{y}_i)^{\alpha}\log(\hat{y}_i), & \text{如果 } y_i=1\\(1-y_i)^{\beta}(\hat{y}_i)^{\alpha}\log(1-\hat{y}_i), & \text{其他}\end{cases} \tag{8}$$

其中，$y_i$ 为期望值，$\hat{y}_i$ 为预测值；$N$ 为每张图片上的目标个数；$\alpha$ 和 $\beta$ 为待定指数，根据 Foc-alloss[21]，取 $\alpha=2,\beta=4$。

最终，改进的总损失函数表示如下：

$$loss=loss_{x,y,w,h}+loss_{\text{class,confidence}} \tag{9}$$

## 4 实验

### 4.1 Backbone 的性能研究与测试

本文的实验都是在 Ubuntu16.04 系统下进行，显卡采用英伟达的 TITAN XP 12G 独立显卡，安装 CUDA8.0，CUDNN5.1 以及 OpenCV3.2 等。

首先，对改进后的 Hourglass 与未改进的 Hourglass 进行学习能力的比较。在实验中，分别对改进后的 Hourglass 与未改进的 Hourglass 模型进行学习训练，即只采用一个 Hourglass 模型进行堆栈。为评价模型的优劣，将其在 ImageNet 数据集[7] 上进行分类实验，比较准确率(Top-1/%)和运算时间。实验结果如表 1 所列。

表 1 改进后的 Hourglass 与原 Hourglass 的性能比较

Table 1 Performance comparison between the improved Hourglass and original Hourglass

| | 改进后的 Hourglass | 原 Hourglass |
|---|---|---|
| Top-1 | 61.2 | 75.8 |
| $t$/ms | 14 | 23 |

在表 1 中，Top-1 表示图像分类的准确率，$t$ 表示识别 1 张图像所需的时间。可以看出，改进后的 Hourglass 的准确率虽然下降了 19.3%，但是其识别 1 张图像的时间却减少了 39.1%。检测速度的提高对于提高目标检测的实时性至关重要，因此综合来看，改进后的 Hourglass 模型有利于提高目标检测的实时性。虽然其分类结果的准确率有所降低，但可以通过 SFPN 等方法进行弥补和改进。

然后，探索最佳的 Hourglass 模型的堆栈个数，以及 SFPN 的重复个数，以实现最佳的特征提取。同样，通过在 ImageNet 数据集上进行分类实验，比较准确率（Top-1），来设计最佳的 Backbone。本实验不采用 SFPN 特征图融合策略，只考虑 Hourglass 个数对特征提取的影响。实验结果如图 8 所示。

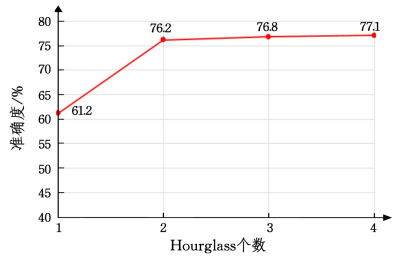



图 8 Hourglass 对图像分类准确率的影响

Fig. 8 Impact of Hourglass on accuracy of image classification

由图 8 可知，当 Hourglass 的个数由 1 变为 2 时，准确率提升很明显，从 61.2% 提升到 76.2%；当 Hourglass 的个数为 3 时，图像分类的准确率提升不明显，只提升了 0.6 个百分点；当 Hourglass 的个数为 4 时，准确率的提升只有 0.3 个百分点。当 Hourglass 的个数大于 2 时，该模型的学习能力已经达到瓶颈，因此不需要再增加 Hourglass 的数目。

下面分析不同 Hourglass 数目对单张图片的识别时间的影响，结果如表 2 所列。

表 2 不同 Hourglass 数目下一张图片的识别时间

Table 2 Recognition time of a picture under different numbers of Hourglass

| Hourglass | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $t$/ms | 14 | 19 | 23 | 28 |

由表 2 可知，当 Hourglass 的数目增加时，单张图片的识别时间也呈线性增加。考虑到目标识别的实时性要求，结合图 8，本文选择 2 个 Hourglass 模型堆栈。

在 2 个 Hourglass 模型堆栈的基础上，考虑 SFPN 特征图融合策略的重复个数对特征提取的影响。实验结果如图 9 所示。

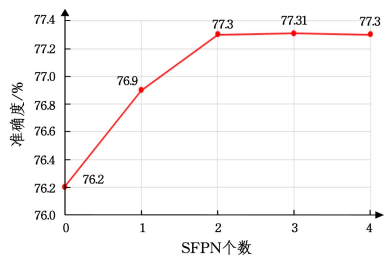



图 9 SFPN 对图像分类准确率的影响

Fig. 9 Impact of SFPN on accuracy of image classification

通过图 9 可知，当 SFPN 的重复个数为 3 时，准确率达到峰值，但是相比 2 个 SFPN，只有 0.1 个百分点的提升，由此可知当 SFPN 的个数大于 2 时，该模型的学习能力已达到瓶颈。另一方面，当 SFPN 的重复个数为 4 时，模型出现了过拟合，准确率稍微有所下降。

下面分析 SFPN 数量对单张图片识别时间的影响，结果如表 3 所列。

表 3 不同 SFPN 数目下单张图片的识别时间

Table 3 Recognition time ofapicture with different numbers of SFPN

| SFPN | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $t$/ms | 19 | 21 | 22 | 23.5 | 25 |

由表 3 可知，当 SFPN 的数目增加时，单张图片的识别时间也呈线性增加。考虑到目标识别的实时性要求，结合图 9，本文选择以 2 个 SFPN 模型堆栈。

最后分析加权融合对 Backbone 性能的影响。在实验中，比较采用特征图加权融合和直接相加（非加权）这两种方法对图像分类性能（准确度（Top-1）和运算时间（$t$））的影响。结果如表 4 所列。

表 4 加权融合和直接相加对结果的影响

Table 4 Effect of weighted fusion and direct addition on results

| | 加权融合 | 直接相加 |
|---|---|---|
| Top-1 | 77.3 | 74.6 |
| $t$/ms | 22 | 22 |

通过表 4 可以看出，当对特征图采用加权融合后，其准确率比直接相加提高了 2.7%，而运算时间却相同，因此采用加权融合可以提高模型的准确率。

通过以上实验，本文的 H_2 模型决定采用 2 个 Hourglass 模型堆栈，然后重复 2 个 SFPN 特征图融合策略，输出时对特征图进行加权融合，如图 4 所示。最后对 H_2 模型和部分 Backbone 模型进行比较，图像分类的准确率如表 5 所列。

表 5 ImageNet 数据集上分类准确率的比较

Table 5 Comparison of classification accuracy on ImageNet dataset

| Backbone | Top-1 | Top-5 |
|---|---|---|
| ResNet-101[17] | 76.3 | 91.5 |
| ResNet-152[17] | 77.1 | 92.1 |
| Darknet-53[14] | 77.0 | 92.2 |
| H_2 | **77.3** | **93.4** |

由表 5 可知，本文设计的 H_2 的 Backbone，其特征提取的性能不但超过了经典的残差网络，也超过了 YOLOv3 中提出的 DarkNet-53。

### 4.2 H_SFPN 目标检测实验

本文中的训练数据集使用 MSCOCO 数据集[22]，在 MSCOCO2017 数据集上进行训练和测试，该数据集包含 118000 幅训练图像（train2017）、5000 幅验证图像（val2017）和 20000 幅支持测试图像（test-dev）。

实验条件同 4.1 节的 Backbone 测试实验。本文在该数据集上采用了数据增强技术，输入图像的大小调整为 416×416。学习率初始值设置为 0.003，采用指数下降法，每进行 1000 次迭代（Iteration），学习率就下降 4%。Batch size 的大小设为 1，表示该模型在训练图像时一次只训练一张图像。具体实验参数如表 6 所列。

表 6 实验参数

Table 6 Experimental parameters

| Name | Parameters |
|---|---|
| Size of image | 416×416 |
| Initial learning rate | 0.003 |
| Decay iterations | 1000 |
| Decay | 0.96 |
| Optimizer | Adam |
| Batch size | 1 |

在实验中先对改进的损失函数和原损失函数的训练收敛过程行比较。在本次实验中，只选取损失函数小于 1 的值进行比较，结果如图 10 所示。其中，*loss* 表示损失函数的值，*epoch* 表示训练集中所有图像的训练次数。从图 10 可以看出，当训练到大约 800 个 epoch 时，改进的损失函数值开始小于改进前的值。可以看出，改进的损失函数相比原损失函数收敛速度更快。

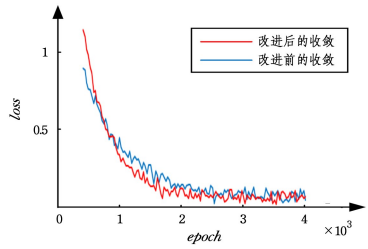



图 10 收敛比较

Fig. 10 Convergence comparison

本文在 MSCOCO 数据集上对所提 H_SFPN 模型进行了评价，同时在相同的条件下将所提模型与其他模型进行了比较，结果如表 7 所列。

表 7 几种目标检测算法的比较(电子版为彩色)

Table 7 Comparison of several object detection algorithms

| Method | Backbone | *FPS* | *AP* | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|
| Faster R-CNN[9] | ResNet-101-FPN | 10 | 35.8 | 61.5 | 40.0 | 21.3 | 36.2 | 49.7 |
| YOLOv3[14] | DarkNet-53 | 31 | 30.2 | 51.9 | 31.3 | 14.1 | 24.8 | 41.3 |
| CenterNet[18] | Hourglass-104 | 14 | 43.3 | 63.2 | 50.7 | 25.1 | 44.2 | 51.8 |
| EfficientDet-D0[15] | EfficientNet-B0 | **47** | 30.4 | 52.1 | 34.7 | 21.8 | 38.7 | 48.3 |
| EfficientDet-D1[15] | EfficientNet-B1 | 33 | 34.6 | 57.3 | 37.4 | 23.2 | 39.5 | 49.8 |
| EfficientDet-D2[15] | EfficientNet-B2 | 31 | 38.8 | 60.1 | 44.7 | 27.6 | 41.1 | 51.4 |
| EfficientDet-D3[15] | EfficientNet-B3 | 28 | **44.1** | **63.5** | **51.2** | 31.4 | **45.3** | **55.6** |
| H_SFPN_r | H_2 | 5 | 40.1 | 61.3 | 45.8 | 31.8 | 42.4 | 52.3 |
| H_SFPN | H_2 | 35 | 41.3 | 62.3 | 46.1 | **32.7** | 42.8 | 52.6 |

在表 7 中，FPS(Frame per second)表示每秒检测图像的帧数，是实时性的衡量参数。$AP$，$AP_{50}$，$AP_{75}$，$AP_S$，$AP_M$，$AP_L$ 为对 MSCOCO 数据集中图像的目标检测准确性的评价指标，其中 $AP$(Average Precision)为目标检测的平均精度，$AP_S$ 为小目标的检测精度，$AP_L$ 为大目标的检测精度[1)]。加粗数字代表排名第一，斜体数字代表排名第二，加下划线数字代表排名第三。H_SFPN_r 代表没有采用改进的损失函数。由表 7 可知：

1)H_SFPN 算法在小目标上的检测性能较好，其中 $AP_S$ 达到了 32.7。

2)虽然 H_SFPN 算法的 *FPS* 值低于 EfficientDet-D0 的 47，但是也达到了 35，排名第二，具有实时性。

3)H_SFPN 算法的目标检测的各项指标均高于 EfficientDet-D2，虽然其目标检测准确性低于 EfficientDet-D3，但 *FPS* 值却较高。

4)从综合性能上来看，H_SFPN 的各项指标都位列前三，其中 $AP_S$ 达到了 32.7，排名第一，而 *FPS* 和 $AP_L$ 均排名第二。

综上所述，H_SFPN 算法为综合性能最佳的算法。

**结束语** 为提高目标检测的准确性与实时性，本文提出了一种基于改进的 Hourglass 框架和 SFPN 的特征图加权融合方法的目标检测 H_SFPN 算法。该算法利用改进的 Hourglass 框架提取特征图，通过 SFPN 特征图加权融合策略来实现小目标的准确识别，利用新颖的损失函数实现快速的误差收敛，最终实现高效准确的目标检测。

实验结果表明，本文所提算法比 YOLOv3 方法的检测效果更好，同时在综合性能上优于 CenterNet 以及 EfficientDet 等目标检测算法。其 *FPS* 值也达到了 35，达到了自动驾驶等实际运用的实时性要求，尤其对小目标的检测效果优于现有的深度学习算法，可有效支持中远距离摄像头的目标检测。

但 H_SFPN 算法中，其网络架构的构建和设计部分主要是依据经验和实验，下一步将融合数学理论算法对网络架构和损失函数等进行优化，以进一步提高算法的实时性和准确性。

## 参 考 文 献

[1] DALAL N. Histograms of Oriented Gradients for Human Detection[C]// IEEE Conference on Computer Vision & Pattern Recognition. San Diego, 2005: 886-893.

[2] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[3] CORTES C, VAPNIK V N. Support-Vector Networks[J]. Machine Learning, 1995, 20(3): 273-297.

[4] ROSENBERG C, HEBERT M, SCHNEIDERMAN H. Semi-Supervised Self-Training of Object Detection Models[C]// IEEE

1) http://cocodataset.org/#detections-eval

Workshops on Application of Computer Vision. Breckenridge, 2005:29-36.

[5] HINTON G E,OSINDERO S,TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation,2006,18(7):1527-1554.

[6] KRIZHEVSKY A,SUTSKEVER I,HINTON G E. Image Net-classification with deep convolutional neural networks[J]. Communications of the ACM,2017,60(6):84-90.

[7] DENG J,DONG W,SOCHER R,et al. ImageNet:A large-scale hierarchical image database[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2009:248-255.

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

[9] REN S,HE K,GIRSHICK R,et al. Faster R-CNN:Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2017,39(6):1137-1149.

[10] KAIMING H,GEORGIA G,PIOTR D,et al. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2017:2961-2969.

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

[12] REDMON J,DIVVALA S,GIRSHICK R,et al. You Only Look Once:Unified,Real-Time Object Detection[C]// IEEE Conference on Computer Vision& Pattern Recognition. 2016:779-788.

[13] REDMON J,FARHADI A. YOLO9000:Better,Faster,Stronger [C]// IEEE Conference on Computer Vision & Pattern Recognition. 2017:6517-6525.

[14] REDMON J,FARHADI A. YOLOv3:An Incremental Improvement[J]. arXiv:1804.02767,2018.

[15] TAN M,PANG R,LEQ V. EfficientDet:Scalable and Efficient Object Detection[J]. arXiv:1911.09070.

[16] LIN T Y,DOLLAR,PIOT R,et al. Feature Pyramid Networks for Object Detection[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2017:4-9.

[17] HE K,ZHANG X,REN S,et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2016:770-778.

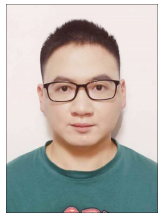
[18] ZHOU X Y,WANG D Q,KRHENBUHL P. Objects as Points [J]. arXiv:1904.07850,2019.

[19] NEWELL A,YANG K,DENG J. Stacked Hourglass Networks for Human Pose Estimation[C]// European Conference on Computer Vision. Springer,Charm,2016:483-499.

[20] YU F,WANG D,SHELHAMER E,et al. Deep Layer Aggregation[J]. arXiv:1707.06484,2017.

[21] LIN T Y,GOYAL P,GIRSHICK R,et al. Focal Loss for Dense Object Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2017(99):2999-3007.

[22] LIN T Y,MAIRE M,BELONGIE S,et al. Microsoft COCO:Common Objects in Context[C]// European Conference on Computer Vision. 2014:740-755.



**SHI Xian-rang**, born in 1996, postgraduate. His main research interests include autonomous driving, object detection and pattern recognition.

**SONG Ting-lun**, born in 1965, Ph. D, professor, Ph. D supervisor. His main research interests include simulation driven vehicle architecture design and development, autonomous driving vehicles, and data driven energy management strategies for new energy vehicles.