

基于通道变换轻量化Yolov5s交通标志识别算法

冷晨, 王萍*

东华大学信息科学与技术学院, 上海

收稿日期: 2022年5月6日; 录用日期: 2022年6月6日; 发布日期: 2022年6月13日

摘要

为满足车载终端设备对于交通标志识别模型占用内存的需求, 本文针对计算资源受限的嵌入式平台, 提出一种轻量交通标志识别算法Yolov5s-lite。通过在Yolov5s中引入Fire Module结构进行通道变换、降低残差模块数量, 减少了其模型占用内存。在TT100K数据集上实验结果表明, Yolov5s-lite相比于Yolov5s, 模型参数量下降22.8%、计算量下降27.9%、实际模型内存下降21.7%、mAP仅下降0.5%, 在检测准确率相当的前提下有效压缩了模型大小。

关键词

交通标志识别, Fire Module, Yolov5s-Lite, 轻量, 残差模块

Lightweight Yolov5s Traffic Sign Recognition Algorithm Based on Channel Transformation

Chen Leng, Ping Wang*

School of Information Science and Technology, Donghua University, Shanghai

Received: May 6th, 2022; accepted: Jun. 6th, 2022; published: Jun. 13th, 2022

Abstract

In order to meet the memory requirements of the vehicle terminal equipment for the traffic sign recognition model, this paper proposes a lightweight traffic sign recognition algorithm Yolov5s-lite for the embedded platform with limited computing resources. By introducing Fire Module structure in Yolov5s to transform channels and reduce the number of residual modules,

*通讯作者。

the memory occupied by the model is reduced. The experimental results on the TT100K data set show that compared with Yolov5s, Yolov5s-lite reduces the number of model parameters by 22.8%, the amount of computation by 27.9%, the actual model memory by 21.7%, and the mAP by only 0.5%. On the premise of the same detection accuracy, this effectively compresses the model size.

Keywords

Traffic Sign Recognition, Fire Module, Yolov5s-Lite, Lightweight, Residual Module

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

交通标志识别[1] (Traffic Sign Recognition, TSR)是智能交通系统中的一个重要分支,在辅助驾驶、无人驾驶等多个安全应用领域有着重要意义。主要实现方法有基于云端服务器和基于终端嵌入式设备。云端检测方法受限于网络条件,无法应用于网络信号较差的交通环境。基于终端嵌入式设备的交通标志识别很好地解决了这一类问题,但受限于计算资源,大多数基于深度学习的算法无法有效部署,所以本文对交通标志识别算法轻量化进行研究,在有效压缩模型大小的同时保证其检测准确率不受较大影响。

传统的交通标志识别方法有基于颜色、基于形状和基于机器学习[2]的方法,此类方法的特征提取较弱,无法适应较为复杂的交通场景。随着基于深度学习的目标检测算法的不断发展,其拥有特征提取能力强、检测准确率高等优势。以 R-CNN 系列为代表的两阶段算法,该类算法先进行区域推荐,然后利用卷积神经网络进行分类。因此这类算法检测准确率较高,但是其模型内存大、识别速度较慢。以 SSD [3]、Yolo 系列为代表的一阶段算法,其主要基于回归的思想,不生成候选区域,直接通过卷积神经网络进行分类,这类算法检测速度较快。在交通标志识别领域中,Zuo Z 等人[4]采用 Faster R-CNN 对交通标志进行检测,并优化其检测性能。汪辉辉[5]在 Yolov3 中引入深度可分离卷积,以提高检测速度。陈梦涛[6]在 Yolov4 中融合注意力机制和 RFB 模块提升网络的特征融合能力,最后 mAP 指标提升近 4%。闫志峰[7]将 MobileNet 引入 SSD 算法中以提升实时性。Tabernik D 等人[8]通过卷积神经网络对交通标志进行端到端的检测识别,取得了较好的效果。Yolov5s [9]作为 Yolov5 系列中最小的模型,对比大部分模型在内存上已拥有轻量的优势,但为了更好地适应嵌入式交通标志识别的应用环境,实现更为高效的运行,还需对其模型大小进行压缩处理。

为了在计算资源受限的嵌入式设备更高效运行交通标志识别算法,降低模型占用内存,本文提出一种轻量交通标志识别算法 Yolov5s-lite。新算法通过在 Yolov5s 中使用 Fire Module 结构进行通道变化,以降低模型参数数量和计算量;其次降低网络中的残差模块数量,以抑制无用背景信息的多次叠加。在检测准确率相当的前提下,有效地压缩了模型大小。

本文后续内容安排如下:第 2 部分为 Yolov5 算法原理;第 3 部分为 Yolov5s-lite 的优化设计;第 4 部分为实验结果与分析;第 5 部分为结论。

2. Yolov5 算法原理

Yolov5 算法网络结构如图 1 所示,Yolov5 共推出四个版本,模型从小到大依次为 Yolov5s、Yolov5m、

Yolov5l、Yolov5x。Yolov5 系列算法大小不同在于网络整体通道数和 C3 中的残差模块数量, 网络整体结构基本一致。本文考虑嵌入式应用场景, 以其中模型最小的 Yolov5s 为基础进行改进。

在输入端, Yolov5 采用了 Mosaic 数据增强、自动计算锚框等方式对输入图像进行处理。主干网络[10] (Backbone)的作用是对图片的特征进行提取。主要组件包括: Focus, Conv, C3, SPP。网络的第一层采用 Focus 结构, 是一种特殊的下采样操作, 其操作过程如图 2 所示。Focus 将特征图每隔一个像素取一个值, 即通道维数扩大为原来的 4 倍。Conv 是 Yolov5 网络中的基本单元, 由卷积、BN 以及激活函数(Silu)组成。C3 由多个 Conv 和多个残差模块组成。其思想为将梯度的变化从头到尾地集成到特征图中, 减少网络参数和运算量, 既保证速度和准确率, 也减少模型尺寸大小。SPP 是空间金字塔, 采用四种不同尺度的卷积核做最大池化操作, 将所得结果进行拼接, 以获得不同尺度的感受野。

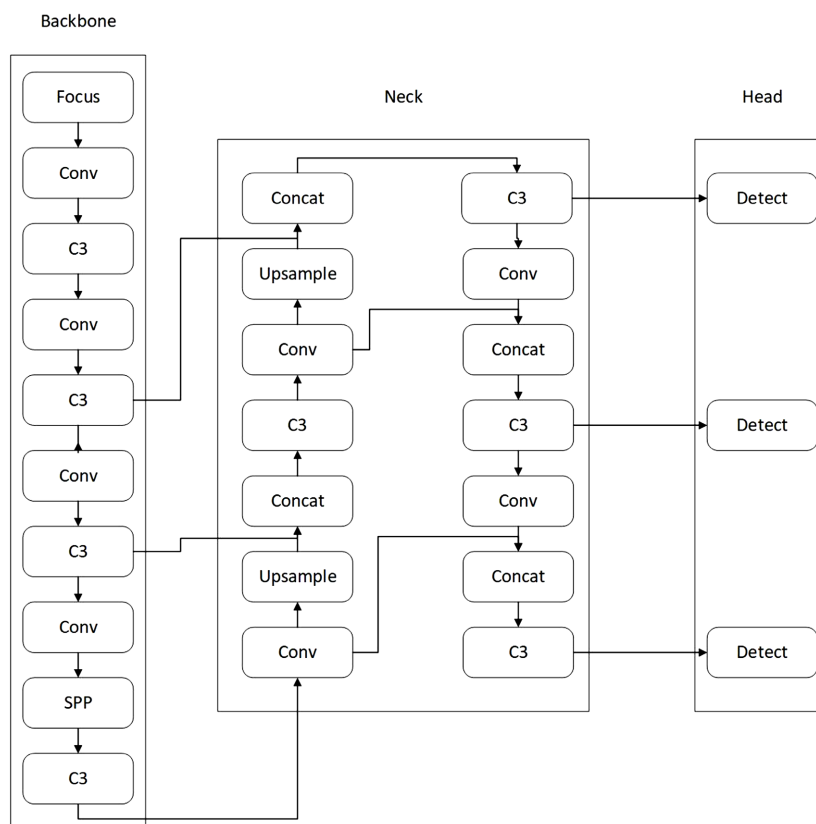


Figure 1. Yolov5 network structure
图 1. Yolov5 网络结构

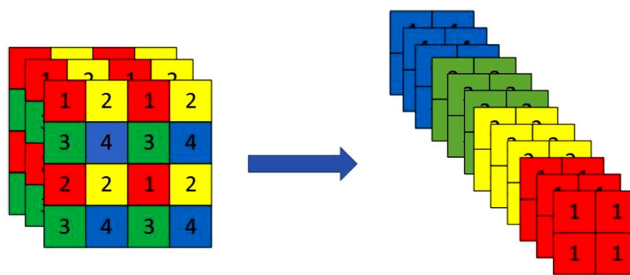


Figure 2. Schematic diagram of Focus operation
图 2. Focus 操作示意图

特征增强网络(Neck)由特征金字塔[10] (Feature Pyramid Networks, FPN)和路径聚合网络(Path Aggregation Network, PAN)组成。FPN 和 PAN 结构实现高层特征与低层特征融合互补。FPN 结构将高层的大目标的类别特征向低层传递, PAN 结构将低层的大目标的位置特征和小目标的类别、位置特征向上传递, 两者互补并克服各自局限性, 强化模型特征提取能力。

不同于传统的目标检测算法只利用高层信息对图像进行检测, Yolov5s 检测网络(Head)设置了大、中、小三个检测层, 满足了不同大小物体对于检测网络的要求。

3. Yolov5s-Lite 交通标志识别算法

3.1. Fire Module 结构

Iandola 等人[11]提出的 SqueezeNet 网络模型将 3×3 卷积使用 1×1 卷积和较低通道数量 3×3 卷积来代替, 通过分解成多个卷积, 降低通道数可以有效的减少模型大小。对于 3×3 卷积核, 输入通道数为 M , 输出通道数为 N , 其参数数量为 $3 \times 3 \times M \times N$, 通过减少 M 和 N 的数量可以有效的降低网络模型的参数量。

SqueezeNet 网络由卷积层、池化层、全连接层以及若干个 Fire Module 模块组成。其中 Fire Module 结构是 SqueezeNet 网络中提出的一个用于减少卷积层参数数量的创新点。Fire Module 由两部分组成, 分别为 Squeeze 结构和 Expand 结构, 其总体结构如图 3 所示。其中 Squeeze 层包含低通道数量的 1×1 卷积层, 负责压缩模型参数量; Expand 层则由一组 1×1 卷积层和一组 3×3 卷积层分两路组成, 负责对模型进行扩充。激活函数选择简单快速的 ReLU 函数, 最后的输出为两路卷积后结果的拼接。

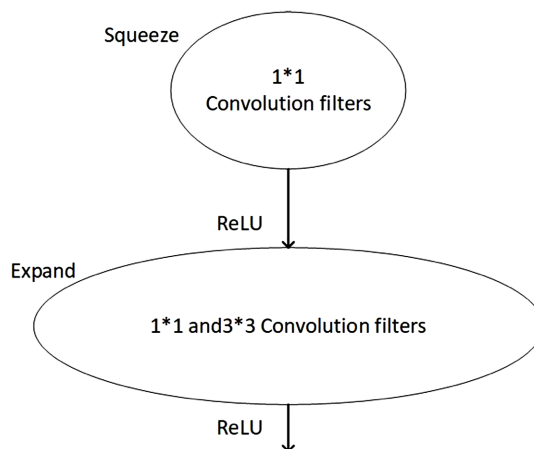


Figure 3. Fire Module structure diagram

图 3. Fire Module 结构图

Fire Module 结构中 Squeeze 层 1×1 的卷积通道数记为 s_1 , Expand 层中 1×1 和 3×3 卷积通道数分别记为 e_1 和 e_3 。在 Fire Module 中, 作者建议 $s_1 < e_1 + e_3$ 。在 SqueezeNet 网络模型中, 作者使用的策略为 $4 \times s_1 = e_1 = e_3$ 。在本文中采用同样的策略。为详细解释 Fire Module 的结构, 以输入特征图为 $20 \times 20 \times 128$, 卷积核大小为 $3 \times 3 \times 256$, 步长为 1 为例, 其输出特征图为 $20 \times 20 \times 256$ 。如图 4 所示, 使用 Fire Module 代替 $3 \times 3 \times 256$ 卷积核, 则输入图像首先经过 Squeeze 层 $1 \times 1 \times 32$ 的卷积核进行通道变换, 输出特征图为 $20 \times 20 \times 32$ 。接着该图像再分别与 Expand 层中的 $1 \times 1 \times 128$ 和 $3 \times 3 \times 128$ 的卷积核进行卷积, 得到两个 $20 \times 20 \times 128$ 的特征图。最后将这两个特征图进行拼接, 得到最终输出 $20 \times 20 \times 256$ 的特征图, 与 $3 \times 3 \times 256$ 卷积核, 步长为 1 的输出结果相同。

为了体现 Fire Module 压缩的有效性, 分别计算两种方法的参数量, 在使用 $3 \times 3 \times 256$ 卷积核时, 参数量计算公式如下:

$$3 \times 3 \times 256 \times 128 = 294912 \quad (1)$$

其中 128 为输入通道数, 256 为输出通道数。其次使用 Fire Module 结构时计算参数量为:

$$1 \times 1 \times 32 \times 128 + 1 \times 1 \times 128 \times 32 + 3 \times 3 \times 128 \times 32 = 45056 \quad (2)$$

Fire Module 结构参数量相较于原卷积层下降了约 85%。由此可见, Fire Module 结构可以有效地压缩模型参数量。为兼顾检测速度与检测精度, 本文将 YOLOv5s 网络中 3×3 , 步长为 1 的卷积层使用 Fire Module 结构代替, 以用于压缩模型大小。

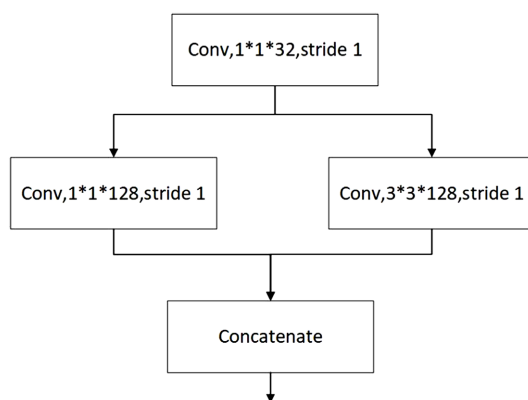


Figure 4. Detailed connection diagram of Fire Module

图 4. Fire Module 详细连接图

3.2. 降低残差模块深度

为了提取更深层次的目标特征信息, CNN 的深度也越来越大。但是当网络深度增加到一定程度时, 出现了梯度消失和网络退化现象。残差模块的出现很好地解决了这些问题, YOLOv5s 的残差模块 (Bottleneck) 的结构如图 5 所示。由于在上节中, 使用 Fire Module 结构代替了 3×3 , 步长为 1 的卷积层, 所以改进后的 Bottleneck 如图 6 所示。

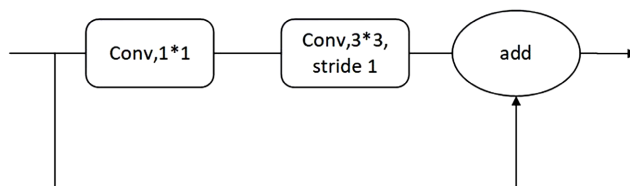


Figure 5. Bottleneck structure diagram

图 5. Bottleneck 结构图

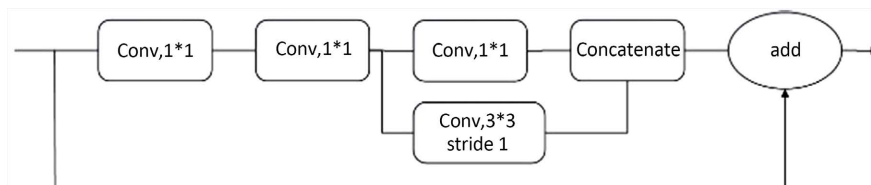


Figure 6. Structure diagram of Bottleneck after improvement

图 6. 改进后的 Bottleneck 结构图

Bottleneck 对特征的深度传递有着积极影响。由于改进后的 Bottleneck 相较于之前, 进行了卷积分解, 卷积层次变得更深。过深的残差模块容易造成无关背景的多次

叠加, 小目标的语义信息容易消失且增加模型大小。Yolov5s 网络中, Bottleneck 模块位于 C3 模块之中, 主干网络中共有 4 个 C3 模块, 分别有 1, 3, 3, 1 个 Bottleneck 模块。重新适配 Bottleneck 模块深度后, 降低为 1, 2, 2, 1 个。

4. 实验结果和分析

4.1. 计算机实验环境

本文计算机硬件环境使用的显卡型号为 NVIDIA GEFORCE RTX 2070 (super), 8G 运行内存, 整个实验框架都基于 Window10 操作系统实现。表 1 是实验环境的具体配置。

Table 1. Experimental environment

表 1. 实验环境

操作系统	Window 10
系统类型	64 位
GPU	NVIDIA GEFORCE RTX 2070(super)
运行内存	8G
编程语言	Python3.9
编译器	Pycharm
深度学习框架	Pytorch1.9.0
GPU 加速工具	CUDA11.0+ CUDNN8.0.5

4.2. 交通标志数据集

本文使用的交通标志数据集来源于 TT100K 数据集[11]。由于数据集中交通标志种类相差过大, 所以挑选出具有重要语义信息的 16 类交通标志, 共计训练集 2725 张, 测试集 624 张。16 类交通标志依次命名为: i4, i5, il60, io, p11, p26, pl100, pl30, pl40, pl5, pl50, pl60, pl80, pn, pne, po, 如图 7 所示。



Figure 7. 16 categories of traffic signs

图 7. 16 类交通标志

4.3. 性能评价指标

在分类任务中, TP (True Positives), 预测为正样本, 实际也为正样本的个数。FP (False Positives), 预测为正样本, 实际为负样本的个数 FN (False Negatives), 预测为负样本, 实际为正样本的个数。在目标检测领域中, 常用以下几种指标评价算法性能。

精确率(Precision, P)定义为所有预测值为 Positive 的样本中真实值为 True 的样本所占比例, 精确率直接反映了模型的错检程度, 其计算公式如下:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

召回率(Recall, R)定义为所有真实值为 True 的样本中预测值为 Positive 的样本所占比例, 召回率直接反映了模型的漏检程度, 其计算公式如下:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

平均精确率(Average Precision, AP)是将 P 和 R 相结合的指标。将 P 和 R 分别当作横纵坐标轴, 然后得到一条曲线 P - R , P 和 R 是不可兼得的指标, 一方的上升往往伴随着一方的下降, PR 曲线下面所包围的面积为 AP, 采用 AP 能够更好地评估模型的检测性能, 计算公式如下:

$$\text{AP} = \int_0^1 R d(P) \quad (5)$$

在目标检测领域中, 通常不是简单的二分类任务。对于多分类任务, 只需将所有类别的 AP 值取平均值, 即为 mAP, 其数学公式如公式 6 所示, 其中 C 代表类别的数量。

$$\text{mAP} = \frac{\sum_{j=1}^C \text{AP}_j}{C} \quad (6)$$

4.4. 训练过程

本文实验过程在超参数文件中设置训练参数: 训练批次数量为 16, 线程数为 4, 初始学习率为 0.01, 循环学习率为 0.2, 采用随机梯度下降法(SGD)作为训练优化函数。在正式训练之前, 首先进行 3 轮预热学习, 其中预热学习动量为 0.8, 预热初始学习率为 0.1, 目的是为模型经过预热学习趋于稳定后, 再进行正式训练效果更佳。

由图 8 可知, 三种损失函数在前期均处于下降趋势, 当 epoch 达到 300 时, 三种损失函数在测试集上基本不再变化, 即代表训练已经趋于稳定, 可终止训练。

4.5. 算法对比

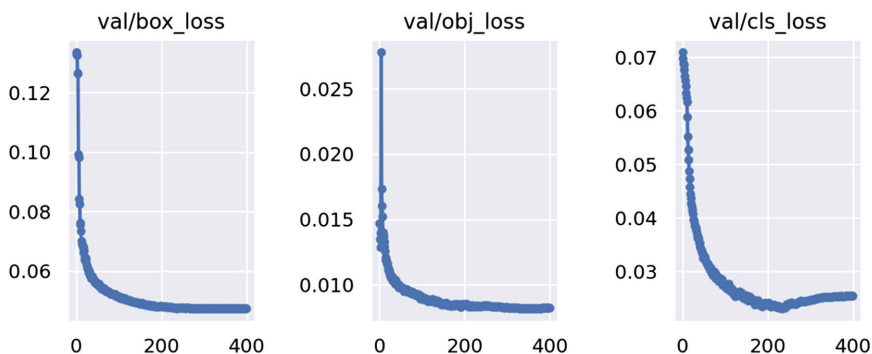
如表 2 所示, Yolov5s-lite 相比于 Yolov5s, 模型参数量下降 22.8%、计算量下降 27.9%、实际模型内存下降 21.7%, mAP 仅下降 0.5%。Yolov5s-lite 模型内存仅有 10.8M, 更好地满足了嵌入式交通标志识别的应用场景。

4.6. 测试结果展示

图 9 给出 Yolov5s-lite 不同交通场景下的检测效果。图 9(a)表明可以在交通标志侧面准确识别。图 9(b)表明可以准确在阴影和强光处识别交通标志。这验证了 Yolov5s-lite 在不同交通环境下拥有较强的鲁棒性。

Table 2. Algorithm comparison**表 2.** 算法对比

算法	参数量	计算量	模型内存	mAP
Yolov5s	7103997	16.5 GFLOPs	13.8M	76.2%
Yolov5s-lite	5481781	11.9 GFLOPs	10.8M	75.7%

**Figure 8.** Loss function curve**图 8.** 损失函数曲线图

(a) 交通标志侧面

(b) 阴影和强光下交通标志

Figure 9. Test results**图 9.** 测试结果

5. 结论

本文针对计算资源受限的嵌入式交通标志识别应用场景, 提出一种轻量交通标志识别算法 Yolov5s-lite。新算法通过使用 Fire Module 结构和降低残差模块深度, 有效压缩了模型大小。在 TT100K 数据集的实验结果表明, Yolov5s-lite 相较于 Yolov5s, 模型参数量下降 22.8%、计算量下降 27.9%、实际模型内存下降 21.7%、mAP 仅下降 0.5%, 改进后的模型内存仅有 10.8 M。

综上, 本文所提的 Yolov5s-lite 交通标志算法具有轻量化的优势。未来可进一步结合实际交通应用场景展开研究。

参考文献

- [1] Fleyeh, H. (2018) Traffic Sign Detection and Recognition. In: Loce, R.P., Bala, R. and Trivedi, M., Eds., *Computer Vision and Imaging in Intelligent Transportation System*, John Wiley & Sons Ltd., 343-383.

-
- <https://doi.org/10.1002/9781118971666.ch14>
- [2] Huang, Z.Y., Yu, Y.L., Gu, J. and Liu, H.P. (2017) An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine. *IEEE Transactions on Cybernetics*, **47**, 920-933.
<https://doi.org/10.1109/TCYB.2016.2533424>
- [3] 孙超, 温蜜, 景俐娜. 改进 SSD 算法在交通标志检测中的应用[J/OL]. 计算机工程与应用, 2021: 1-10.
<http://kns.cnki.net/kcms/detail/11.2127.TP.20210928.0236.006.html>, 2021-09-28.
- [4] Zuo, Z.R., Yu, K., Zhou, Q., Wang, X. and Li, T. (2017) Traffic Signs Detection Based on Faster R-CNN. 2017 *IEEE 37th International Conference on Distributed Computing Systems Workshops*, Atlanta, 5-8 June 2017, 286-288.
<https://doi.org/10.1109/ICDCSW.2017.34>
- [5] 汪辉辉. 基于嵌入式的交通标志识别器的设计[D]: [硕士学位论文]. 成都: 电子科技大学, 2020.
- [6] 陈梦涛, 余粟. 基于改进 YOLOV4 模型的交通标志识别研究[J/OL]. 微电子学与计算机, 2021: 1-10.
<http://kns.cnki.net/kcms/detail/61.1123.TN.20210923.0115.001.html>, 2021-09-23.
- [7] 闫志峰. 面向小型嵌入式设别的基于深度学习的交通标志检测方法研究[D]: [硕士学位论文]. 北京: 中国科学院大学, 2020.
- [8] Tabernik, D. and Skocaj, D. (2019) Deep Learning for Large-Scale Traffic Sign Detection and Recognition. *IEEE Transactions on Intelligent Transportation Systems*, **21**, 1427-1440.
- [9] Luo, Y. (2021) Occlusion Face Detection Is Based on Yolov5s. 2021 *IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, Chongqing, 17-19 December 2021, 978-981.
<https://doi.org/10.1109/ICIBA52610.2021.9688319>
- [10] Wang, C.-Y., Liao, H.-Y.M., *et al.* (2019) CSPNet: A New Backbone That Can Enhance Learning Capability of CNN.
<https://doi.org/10.48550/arXiv.1911.11929>
- [11] Iandola, F.N., Han, S., Moskewicz, M.W., *et al.* (2016) SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5 MB Model Size. <https://doi.org/10.48550/arXiv.1602.07360>