DOI：10.13890/j.issn.1000-128X.2024.01.120

基于低秩分解的YOLO轻量化目标检测模型

林德铝[1]，刘　畅[1]，陈　琦[1]，曾　阳[2]，何　琨[1]
（1.华中科技大学 计算机科学与技术学院，湖北 武汉 430074；
2.中车株洲电力机车研究所有限公司，湖南 株洲 412001）

摘　要：随着列车智能化程度的不断提高，许多研究探索了车载设备目标检测模型的轻量化技术，以满足在资源有限情况下的高效计算。针对当前YOLO（You Only Look Once）系列目标检测模型轻量化方法通用性不够强的问题，文章提出了一种针对YOLO系列的低秩分解参数压缩算法。首先通过预设的低秩比例系数和卷积单元的输入/输出通道数量计算低秩，然后通过对目标结构的卷积层进行Tucker分解，得到新的卷积序列，最后融合新的卷积序列，取代原有卷积层。使用公开数据集对所提出的基于低秩分解的参数压缩方法进行试验，选用了YOLOv5-l、YOLOv8-x和YOLOX-x这3种模型，在保证低秩分解后的模型检测平均精度为原模型96%的前提下，模型参数量和浮点计算量均减少了约40%，同时图像检测速度能达到原模型的150%左右。此外，可视化结果显示，该方法压缩过的模型与原模型在相同图像上的关注区域基本相同。试验结果表明，文章提出的方法可以有效地对单阶段YOLO系列目标检测模型进行轻量化压缩，提高模型在车载设备上的可用性；同时，所做工作对轨道交通领域自动驾驶场景下的其他模型的轻量化处理也具有重要的借鉴意义。

关键词：深度学习；低秩分解；轨道交通；自动驾驶；模型轻量化；目标检测

中图分类号：TP18　　**文献标识码：**A　　**文章编号：**1000-128X（2024）01-0138-07

0　引言

随着经济水平的提升与智能交通系统的快速发展，列车的智能化升级是实现更高水平的轨道交通智能化重要的一步。对于轨道交通的列车无人驾驶，需要通过复杂的系统执行多项任务，其中最具挑战性的任务之一就是对列车周围的环境感知。车载目标检测技术在列车自动驾驶、智能列车安全等领域中发挥着关键的作用。但车载目标检测系统仍面临着诸多挑战，如处理实时视频流时的计算复杂性、数据模型在资源受限环境下的性能发挥，以及在车辆上部署的实用性等问题。因此，开展轻量化车载目标检测模型的研究已成为当前研究的热点之一。

目前，目标检测的轻量化技术在公路道路的无人驾驶方面得到了广泛应用。在公路道路上，无人驾驶汽车通过感知系统获取环境信息，如检测道路上的各种目标——车辆、行人、交通标志等，这种环境感知是无人驾驶汽车实现安全、高效行驶的关键，而轨道交通通常是在固定的轨迹上运行，相对于公路道路上车辆的自由行驶，轨道交通的运动轨迹变化相对较小。在收集数据的传感器仅使用高清摄像头的前提下，视觉目标检测及其轻量化技术在公路道路无人驾驶和轨道交通的环境感知问题上基本通用。因此，将公路道路无人驾驶方面的轻量化技术迁移到轨道交通目标检测技术是一个重要的解决思路。

基金项目：国家自然科学基金项目（62076105）
通信作者：何　琨（1972—），女，博士，教授，博士生导师，主要从事深度学习、机器学习、算法设计与优化方面的研究；
E-mail: brooklet60@hust.edu.cn
引用格式：林德铝, 刘畅, 陈琦, 等. 基于低秩分解的YOLO轻量化目标检测模型[J]. 机车电传动, 2024(1): 138-144.DOI:10.13890/j.issn.1000-128X.2024.01.01.120
Citation：LIN Delü, LIU Chang, CHEN Qi, et al. Low-rank decomposition based lightweight YOLO model for the object detection[J]. Electric drive for locomotives,2024(1): 138-144.DOI:10.13890/j.issn.1000-128X.2024.01.120.

近年来，卷积神经网络在计算机视觉领域取得了巨大的成功，凭借其强大的识别能力和高效快速的处理图像能力在工业界中得到广泛应用。目前，基于卷积神经网络的主流目标检测模型基本可以划分为两阶段和单阶段2种类型。其中，以Faster R-CNN[1]为代表的两阶段目标检测网络主要包括2个关键步骤：①网络通过卷积层和特征提取模块对输入图像进行特征提取，并利用候选区域生成算法生成候选目标区域；②针对每个候选区域进行目标分类和精确的位置回归，以实现目标的准确定位和分类。在具体应用上，文献[2]使用Faster R-CNN实现了铁路接触网鸟窝的检测。文献[3]提出了一种利用Mask-RCNN模型诊断牵引设备故障问题的方法。然而，两阶段模型的网络权重较大，对车载的边缘设备而言存在很大的计算资源压力和存储压力。因此，在列车智能交通系统中，针对资源受限的车载设备，单阶段目标检测网络成为更高效和实用的选择。

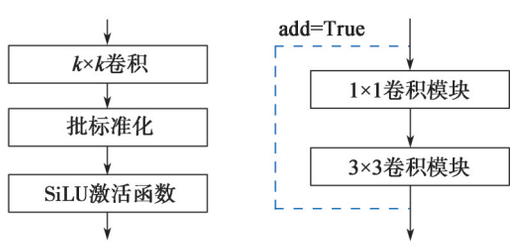
在单阶段目标检测模型的发展中，YOLO（You Only Look Once）系列模型[4-9]凭借其实时性和高效性而备受关注。YOLO模型以其单阶段的目标检测策略和端到端的训练方式在列车智能化目标检测应用中取得了显著的成果。文献[10]介绍使用YOLOv5模型在矿山实际场景数据集中用于识别石头、矿卡和卡车等目标。文献[11]使用YOLOv4模型针对异常弓网的检测问题设计了一套系统方案。然而，由于车载设备的计算资源有限，传统的YOLO系列模型在车载环境下的实时性和准确性之间往往存在一定的矛盾。因此，如何对YOLO系列模型进行轻量化压缩，以同时满足在车载环境下实时性和准确性的需求，成为一个重要而具有挑战性的问题。文献[12]通过对YOLOv5模型进行了轻量化改进实现对轨道障碍物检测，改进方法包括直接修改数据增强方案，并采用深度可分离卷积层替代普通卷积层，以减少计算开销；同时，通过通道剪枝使模型更加轻量化，但这种针对特定模型的改进方法缺少通用性。文献[13]对YOLOv3-tiny模型同样改进了深度可分离卷积层，并采用具有反向残差的线性瓶颈结构设计主干网络，实现对轨道紧固件进行检测，但这种改进方法每遇到一个新场景就需要重新设计轻量化网络，缺少可拓展性。

针对上述问题，低秩分解作为一种有效的压缩参数技术，可以通过将原始权重矩阵分解为2个低秩矩阵的乘积，实现对模型参数的精减。通过低秩分解，可以在不显著损失模型性能的情况下，减少模型的计算复杂度，降低模型的存储需求，从而提高机车车载YOLO系列模型的实用性和可扩展性。本文针对车载YOLO系列模型的轻量化问题，使用低秩分解技术对模型压缩，在保证精度的同时有效减少了模型参数，利于模型在计算资源有限的情况下的部署，为实际列车车载设备的YOLO系列模型的轻量化处理提供一种可行的通用解决方案。

1　YOLO系列检测算法

YOLO系列算法在目标检测领域有着显著的表现，尤其在实时目标检测中展现出卓越的性能。这一系列模型主要由Backbone、Neck和Head这3部分构建，而且在设计过程中始终强调轻量化和计算效率。

Backbone是模型的基础部分，负责从输入图像中提取特征。在YOLO的不同版本中，采用了不同的Backbone网络结构。早期的YOLO选择了Darknet19网络，而后续的版本如YOLOv3和YOLOv4则转向使用CSPDarknet53网络。Darknet是一种轻量级的卷积神经网络，由Darknet Bottleneck堆叠而成；Darknet Bottleneck是由2个卷积单元和1个残差连接组成的基本单元；每个卷积单元通常包含1×1或3×3的卷积核，1×1的卷积核负责降低或增加特征图的维度，3×3的卷积核复制提取空间信息；卷积后紧跟着批标准化层（BN层）和SiLU激活函数层，残差连接可以选择性地打开和关闭，如图1所示。CSP是Cross Stage Partial Networks[14]的简称，它将每一层的特征图划分为2部分，一部分直接传递到下一层，另一部分则进行卷积操作。这样可以减少信息流在网络中的冗余，使网络参数能更有效地被利用，从而提高模型的性能。



(a) $k\times k$卷积模块　　(b) Darknet Bottleneck

图1　卷积模块与Darknet Bottleneck模块结构

Fig. 1　Structure of convolution module and Darknet Bottleneck module

Neck部分负责整合和融合来自Backbone的多尺度特征图。常见的Neck架构为FPN（Feature Pyramid Networks）结构，其中高层特征图丰富的语义信息可被传递到底层特征图，为识别不同尺度的目标提供全面的上下文信息。新版YOLO模型YOLOv5还引入了PANet（Path Aggregation Network）结构，该结构可将高分辨特征图的语义信息传递到上层的低分辨率的特征图，进一步增强顶层特征图的语义信息。Head部分负责生成最终的目标检测结果，YOLO系列模型通常包括多个并行的输出层，用于同时检测不同尺度的目标。每一个输出层都会利用1个预定义的网格划分原始图像，并在每个网格单元上预测若干个边界框及其类别和置信度。例如，YOLOv3和YOLOv4设定了3个输出层，分别对应大、中、小3个尺度的目标。

2 基于低秩分解的YOLO系列轻量化方法

2.1 低秩分解轻量化方法

目前，YOLO系列模型已取得了令人瞩目的成绩，但在轻量化设计方面仍存在进一步的探索空间。因此，本研究致力于探讨如何使用低秩分解方法对YOLO系列模型进一步轻量化改进，在保留其优异的检测性能的同时，进一步减小模型规模和计算量。

低秩分解是一种常用的矩阵分解技术，它将一个大的矩阵分解为多个较小的矩阵乘积，而这些较小的矩阵通常拥有较低的秩。低秩矩阵有多个优点：①由于低秩矩阵中许多元素都是0，所以计算时只需要考虑那些非零元素，可以显著降低计算复杂度，从而提高计算效率；②压缩数据规模：在处理大量数据时，使用低秩矩阵可以有效压缩数据规模，这对于存储和传输大数据集尤其重要，特别是在资源受限的设备上；③特征提取：利用低秩矩阵表示，可以帮助提取出数据的主要特征，使得模型能更专注于最关键的特征，从而提高性能。这种分解方法在降维、压缩数据和简化计算中都非常有效，在机器学习领域被广泛应用。

奇异值分解（Singular Value Decomposition, SVD）是一种经典的低秩分解方法。对于一个给定的$m\times n$的矩阵$\boldsymbol{A}$，本文使用SVD将其分解为3个矩阵的乘积：$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$，其中$\boldsymbol{U}$和$\boldsymbol{V}$是正交矩阵，$\boldsymbol{\Sigma}$是由奇异值构成的对角矩阵，其对角线上的元素从大到小排列。如果只保留前r个最大的奇异值和对应的$\boldsymbol{U}$和$\boldsymbol{V}$的列，就得到了原矩阵的一个近似分解，即

$$\boldsymbol{A}\approx\boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^{\mathrm{T}} \tag{1}$$

式中：$\boldsymbol{U}_r$, $\boldsymbol{V}_r$分别为由$\boldsymbol{U}$和$\boldsymbol{V}$的前r列组成的矩阵；$\boldsymbol{\Sigma}_r$是$\boldsymbol{\Sigma}$的左上角$r\times r$子矩阵。

对于一个线性层$\boldsymbol{W}$，其权重为$m\times n$的矩阵，SVD分解后取前k个最大特征值，该权重矩阵可分解为$m\times k$和$k\times n$的矩阵，线性层的计算变为

$$\boldsymbol{Y}=\boldsymbol{X}\boldsymbol{W}=\boldsymbol{X}\boldsymbol{W}_1\boldsymbol{W}_2=\left(\boldsymbol{X}\boldsymbol{W}_1\right)\boldsymbol{W}_2 \tag{2}$$

式中：$\boldsymbol{X}$为线性层$\boldsymbol{W}$的输入；$\boldsymbol{Y}$为线性层$\boldsymbol{W}$的输出；$\boldsymbol{W}_1$,$\boldsymbol{W}_2$为线性层$\boldsymbol{W}$低秩分解后所得的矩阵。

矩阵分解前的时间复杂度为$O(m\times n)$，低秩分解的时间复杂度变为$O(m\times k+k\times n)$。当$k<\dfrac{m\times n}{m+n}$，则时间复杂度降低。

Tucker分解是一种更为通用的多线性分解方法，能将1个高维的张量分解为1个核心张量与若干个矩阵的乘积。Tucker分解允许对每一个维度单独设置秩，相对于SVD，提供了更高的灵活性，能有效对高阶张量分解。以卷积层的权重分解为例，不考虑输入样本数量，假设某个卷积层有输入张量$\boldsymbol{X}\in\mathbb{R}^{C\times H\times W}$，输出张量$\boldsymbol{Y}\in\mathbb{R}^{O\times H'\times W'}$，卷积核权重$\boldsymbol{W}\in\mathbb{R}^{O\times C\times K_1\times K_2}$，对于卷积输出张量在输出通道$o$、高度位置$h'$和宽度位置$w'$的值$Y_{o,h',w'}$由输入张量中的中心坐标位于$i$、$j$的对应区域计算。该过程可描述为：

$$Y_{o,h',w'}=\sum_{c=1}^{C}\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\left(\boldsymbol{W}_{o,c,k_1,k_2}\boldsymbol{X}_{c,h_i,w_i}\right) \tag{3}$$

$$h_i=(h'-1)S+i-P \tag{4}$$

$$w_i=(w'-1)S+j-P \tag{5}$$

式中：S为卷积步长；P为输入图像边缘填充的像素数量；$\boldsymbol{W}_{o,c,k_1,k_2}$为卷积核在输出通道$o$、输入通道$c$、高度位置$k_1$和宽度位置$k_2$的相应权重；$\boldsymbol{X}_{c,h_i,w_i}$为输入张量在通道$c$、高度位置$h_i$和宽度位置$w_i$的值。

Tucker分解对4个维度单独设置秩R_1、R_2、R_3、R_4，卷积核权重可近似为

$$\boldsymbol{W}_{o,c,i,j}=\sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3}\sum_{r_4=4}^{R_4}\left(\boldsymbol{C}_{r_1,r_2,r_3,r_4}\boldsymbol{U}_{o,r_1}^{1}\boldsymbol{U}_{c,r_2}^{2}\boldsymbol{U}_{i,r_3}^{3}\boldsymbol{U}_{j,r_4}^{4}\right) \tag{6}$$

式中：$\boldsymbol{C}\in\mathbb{R}^{R_1\times R_2\times R_3\times R_4}$，为核张量；$\boldsymbol{U}^1$, $\boldsymbol{U}^2$, $\boldsymbol{U}^3$, $\boldsymbol{U}^4$为4个分解矩阵，且$\boldsymbol{U}^1\in\mathbb{R}^{O\times R_1}$，$\boldsymbol{U}^2\in\mathbb{R}^{C\times R_2}$，$\boldsymbol{U}^3\in\mathbb{R}^{K_1\times R_3}$，$\boldsymbol{U}^4\in\mathbb{R}^{K_2\times R_4}$。

由于K_1、K_2维度本身很小，不需要进一步压缩，因此实际应用的对卷积核权重的张量估计可以进一步简化为

$$\boldsymbol{W}_{o,c,i,j}=\sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\left(\boldsymbol{C}_{r_1,r_2,i,j}\boldsymbol{U}_{o,r_1}^{1}\boldsymbol{U}_{c,r_2}^{2}\right) \tag{7}$$

式中：$\boldsymbol{C}$为核张量，$\boldsymbol{C}\in\mathbb{R}^{R_1\times R_2\times K_1\times K_2}$。

将式(7)代入式(1)，可得

$$\boldsymbol{Y}_{o,h',w'}=\sum_{c=1}^{C}\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\left(\boldsymbol{C}_{r_1,r_2,i,j}\boldsymbol{U}_{o,r_1}^{1}\boldsymbol{U}_{c,r_2}^{2}\boldsymbol{X}_{c,h_i,w_i}\right) \tag{8}$$

式中：$\boldsymbol{C}_{r_1,r_2,i,j}$为核张量$\boldsymbol{C}$中的第$r_1$、$r_2$、$i$、$j$位置的元素值；$\boldsymbol{U}_{o,r_1}^{1}$, $\boldsymbol{U}_{c,r_2}^{2}$分别为分解矩阵$\boldsymbol{U}_1$、$\boldsymbol{U}_2$相应位置的权重。

令

$$\boldsymbol{Z}_{r_2,h,w}=\sum_{c=1}^{C}U_{c,r_2}^{2}X_{c,h,w} \tag{9}$$

$$\boldsymbol{Z}'_{r_1,h,w}=\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\sum_{r_2=1}^{R_2}\left(\boldsymbol{C}_{r_1,r_2,i,j}\boldsymbol{Z}_{r_2,h,w}\right) \tag{10}$$

式中：$\boldsymbol{Z}_{r_2,h,w}$为中间结果$\boldsymbol{Z}\in\mathbb{R}^{R_2\times H\times W}$在相应位置的输出值，可以由$1\times1$卷积实现；$\boldsymbol{Z}'_{r_1,h,w}$是中间结果$\boldsymbol{Z}'\in\mathbb{R}^{R_1\times H\times W}$相应位置的输出值，可以由$k\times k$卷积实现。则

$$\boldsymbol{Y}_{o,h',w'}=\sum_{r_1=1}^{R_1}\left(\boldsymbol{U}_{o,r_1}^{1}\boldsymbol{Z}'_{r_1,h,w}\right) \tag{11}$$

上述的式(9)、式(10)和式(11)可以分别由1×1，$k\times k$，1×1卷积实现，从而实现对原始卷积权重的分解，通过合理地设置R_1、R_2将分解得到2个小矩阵，以这2个小矩阵取代原来的1个大矩阵，从而达到减少模型参数、降低计算复杂度的目的。

2.2 YOLO模型的低秩分解

对于YOLO系列模型的低秩分解，主要关注其

Backbone 和 Neck 内的结构。Backbone 和 Neck 包含大量卷积层，每个卷积层的权重都可以视为 1 个待分解的矩阵或张量。通过应用上述低秩分解策略，能有效地简化模型结构。以 YOLOv5 为例，YOLOv5 的 Backbone 和 Neck 由大量 DarknetBottlenet 组成，而 DarknetBottlenet 则由 2 个卷积单元构成，其中每个卷积单元包含 1 个卷积层、批标准化层和 SiLU 激活函数层。对 YOLOv5 的低秩分解具体是通过对 DarknetBottlenet 内的 3×3 卷积层的分解实现，具体过程如下：

①设置分解后矩阵的秩 r。通过预设的低秩比例系数 r_{lr}（low-rank rate）和输入输出通道数量来计算新的低秩，具体计算如下：

$$r=\left\lfloor\frac{2r_{\mathrm{lr}}C_{\mathrm{in}}C_{\mathrm{out}}}{C_{\mathrm{in}}+C_{\mathrm{out}}}\right\rfloor \tag{12}$$

式中：C_{in}, C_{out} 分别为卷积的输入、输出通道数。

②对 DarknetBottleneck 中的 3×3 卷积单元的卷积层执行 Tucker 分解，得到 1 个核心张量和 2 个因子矩阵，创建 1 个新的卷积序列，包含 3 个卷积层：1×1 卷积层、3×3 卷积层和 1×1 卷积层，以此替换原有的 3×3 卷积层。

③进行卷积融合。由于 Tucker 分解将原有的 3×3 卷积层分解为 3 个 1×1、3×3 和 1×1 卷积层序列，可以通过卷积融合进一步精简 DarknetBottleneck 中的卷积层。对 DarknetBottleneck 第一个 1×1 卷积单元中的卷积层和批标准化层进行融合。令 $\boldsymbol{X}$、$\boldsymbol{Y}$ 分别为卷积的输入和输出，$\boldsymbol{b}$ 为偏置系数，μ、σ 分别为输入的均值和标准差，γ、β 分别为批标准化的缩放系数和平移系数，则卷积和批正则化的计算如下：

$$\boldsymbol{Y}=\boldsymbol{W}\boldsymbol{X}+\boldsymbol{b} \tag{13}$$

$$\boldsymbol{Z}=\frac{\gamma\left(\boldsymbol{Y}-\mu\right)}{\sigma}+\beta \tag{14}$$

令新的卷积权重 $\boldsymbol{W}'=\frac{\gamma}{\sigma}\boldsymbol{W}$，新的偏置项 $\boldsymbol{b}'=\gamma\left(-\frac{\mu}{\sigma}\right)+\beta-\boldsymbol{b}$，取代原有的权重和偏置成为融合后的卷积层；接着将 Tucker 分解的第一个 1×1 卷积融合到 DarknetBottleneck 的第一个卷积单元，2 个卷积层的操作公式为

$$\boldsymbol{Y}_1=\boldsymbol{K}_1\boldsymbol{X} \tag{15}$$

这 2 个操作融合，可以得到以下等价形式：

$$\boldsymbol{Y}_2=\boldsymbol{K}_1\boldsymbol{K}_2\boldsymbol{X} \tag{16}$$

式中：$\boldsymbol{K}_1$、$\boldsymbol{K}_2$ 分别为 2 个卷积的权重。

用 $\boldsymbol{K}_1$、$\boldsymbol{K}_2$ 取代原有 1×1 卷积层的参数，将两者合并为 1 个 1×1 卷积层。同理，将 Tucker 分解的后 2 个卷积层（3×3 和 1×1）合并，即可得到最终的分解模型。

3　试验结果与分析

3.1　数据集与评价指标

本文使用开源的 COCO 目标检测数据集[15]与 OSDaR23 数据集[16]进行训练和测试。其中，COCO 数据集包含大约 118 000 张图像和 5 000 张验证集，涵盖人物、动物、交通工具在内的 80 个不同类别的目标；OSDaR23 数据集是 2021 年 9 月在德国汉堡市使用轨道作业车辆捕获的 21 个序列组成的一个多传感器数据集，传感器包括摄像机、激光、雷达、位置和加速度传感器。本研究在试验中仅使用轨道作业车辆前部安装的高分辨率摄像机采集的图像数据，共选取 780 张高清图像。试验采用平均精度 mAP（mean Average Precision, mAP）、每秒检测帧（Frames Per Second, FPS）、模型参数量（Params）、测试时的前向推理浮点计算量（FLOPs）指标作为评判算法的性能。其中，mAP 的计算公式为

$$\bar{\alpha}=\frac{1}{n}\sum_{i=1}^{n}\alpha_{\mathrm{AP}\,i} \tag{17}$$

式中：$\bar{\alpha}$ 为 mAP 的数值；$\alpha_{\mathrm{AP}\,i}$ 为某个目标类的平均精度。

对于推理阶段实时性要求，FPS 要求达到 100 以上。

3.2　试验环境与训练参数

试验使用的环境是 Intel（R） Xeon（R） Silver 4314 的处理器和 NVIDIA GeForce RTX 3090 显卡，操作系统为 Ubuntu 18.04.4，深度学习框架为 PyTorch 1.12.1，TensorRT 版本为 8.5.1.7。选用 YOLOv5-l、YOLOv8-x 和 YOLOX-x 作为试验模型。其中，YOLOv5-l 和 YOLOX-x 的训练次数 Epoch 设置为 300，YOLOv8-x 的训练次数 Epoch 设置为 500；3 个模型的训练批大小 batch size 都设置为 32；训练优化器为 SGD 优化器；学习率在从 0 开始，经过 5 个 Epoch 的指数预热后从 0.01 余弦退火公式调整至 0；对于低秩分解的超参数低秩比例系数，将其统一设置为 0.3。相关验证及检测结果如表 1~表 4。

3.3　试验结果

3.3.1　YOLO 系列模型低秩分解的试验分析

对 YOLOv5-l、YOLOv8-x 和 YOLOX-x 这 3 种模型进行了低秩分解操作，并将模型转化为 TensorRT 格式以方便车载设备部署。模型各项试验结果的评价指标如表 1 所示。表 1 数据表明，所使用的低秩分解方法在维持目标检测模型检测效果的同时，提升了推理速度，减少了模型参数。具体来说，3 个模型在经过低秩分解后的 mAP 都保持在原 mAP 的 96% 以上，这表明低秩分解方法对 YOLO 系列模型的效果影响很小；同时，模型的参数量和浮点计算量都减少了约 40%，即低秩分解压缩了模型的大小，并降低了模型的计算复杂度。在 TensorRT 框架下，低秩分解将 3 个模型的 FPS 提升到了原模型的近 1.5 倍，尤其是 YOLOv8-x 和 YOLOX-x，在低秩分解后模型的 FPS 均超过了 100，满足模型在后续推理使用时的实时性要求。

表 1　COCO 数据集上低秩分解验证试验

Table 1　Verification experiment of low-rank decomposition on COCO dataset

参数值	YOLOv5-l		YOLOv8-x		YOLOX-x	
	基线模型	低秩分解	基线模型	低秩分解	基线模型	低秩分解
平均精度/%	0.478	0.464	0.526	0.508	0.506	0.492
FPS	221	321	93	140	71	106
参数量 /MiB	46.56	27.49	68.22	41.85	99.07	58.74
GFLOPs	54.79	30.42	129.27	76.56	141.22	86.07

此外，本文还对低秩分解后的 3 个模型在转化为 TensorRT 时都进行了量化处理，以便更加适用于部署在车载设备上，量化的速度如表 2 所示。由表 2 可知，低秩分解后进行量化的操作对于速度的提升十分明显，可以更好满足车载设备推理的实时性要求。

表 2　COCO 数据集上低秩分解与量化后验证试验

Table 2　Verification experiment of lightweight based on low-rank decomposition on COCO dataset

模型	FPS	
	基线模型	低秩分解+量化
YOLOv5-l	221	1 681
YOLOv8-x	116	829
YOLOX-x	71	798

对于 OSDaR23 数据集，随机挑选了若干场景进行试验，3 个检测模型分别加载 COCO 数据集上预训练的权重，对代表性关键类别“行人”进行推理测试，试验结果如表 3 所示。由表 3 可知，对于真实的轨道列车数据，经过低秩分解方法处理后的模型依然具有很强

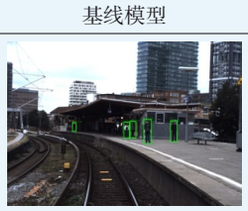
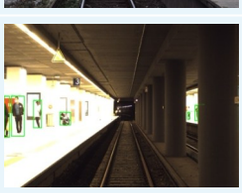
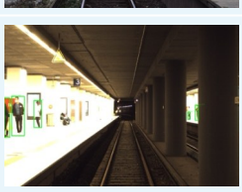
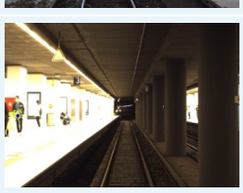
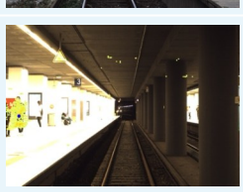
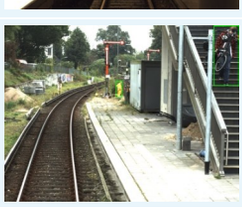
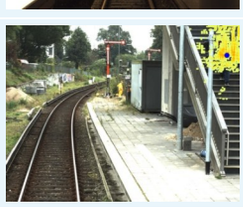
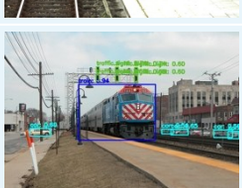
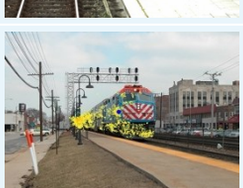
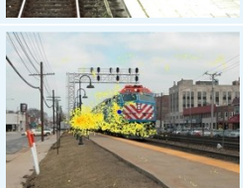
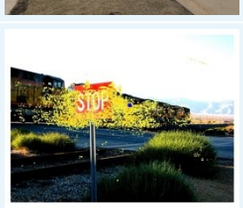
表 3　OSDaR23 数据集低秩分解验证试验

Table 3　Verification experiment of low-rank decomposition on OSDaR23 dataset

模型	平均精度/%	
	基线模型	低秩分解+量化
YOLOv5-l	0.362	0.354
YOLOv8-x	0.384	0.379
YOLOX-x	0.350	0.346

表 4　YOLOv8-x 的基线模型与低秩分解模型的可视化检测结果

Table 4　Visual detection results of baseline model and low-rank decomposition model of YOLOv8-x

试验	检测结果		感受野可视化	
	基线模型	低秩分解模型	基线模型	低秩分解模型
试验一				
试验二				
试验三				
试验四				
试验五				

的识别能力。

3.3.2　可视化分析

为进一步验证低秩分解的有效性，随机在数据集中抽取了5张图片，用YOLOv8-x对其进行了检测试验，并画出其中1个目标的感受野，检测结果如表4所示。由表4可知，经过低秩分解后的模型与基线模型在车载设备检测的实际场景下的表现基本一致，并且感受野分布也基本相同，这进一步说明低秩分解后的模型的实用性。

4　结论

在轨道交通的自动驾驶场景中，车载设备的有限算力是提高目标检测模型性能的瓶颈。在保持模型检测效果的前提下对目标检测模型的参数进行压缩是提升模型在实际场景下可用性的重要手段。本文针对单阶段YOLO系列模型，设计了一种基于低秩分解的轻量化方法，主要针对YOLO模型Backbone和Neck部分的卷积层参数进行了压缩。试验结果验证了所提方法的有效性。此外，该方法还有很强的通用性，对其他目标检测模型的轻量化有着重要的指导意义。如何结合其他方法（例如剪枝等）以进一步对模型进行轻量化压缩是下一阶段的重点研究方向。

参考文献：

[1] REN Shaoqing, HE Kaiming, 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.

[2] 贺德强, 江洲, 陈基永, 等. 基于深度卷积神经网络的铁路接触网鸟窝检测方法研究[J]. 机车电传动, 2019(4): 126-130.
HE Deqiang, JIANG Zhou, CHEN Jiyong, et al. Research on detection of bird nests in overhead catenary based on deep convolutional neural network[J]. Electric drive for locomotives, 2019(4): 126-130.

[3] 林珊, 邓树, 谌小莉, 等. 基于Mask-RCNN的牵引供电设备红外智能诊断[J]. 机车电传动, 2022(4): 55-61.
LIN Shan, DENG Shu, CHEN Xiaoli, et al. Intelligent infrared diagnosis method for traction power supply equipment based on Mask-RCNN[J]. Electric drive for locomotives, 2022(4): 55-61.

[4] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//IEEE. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016: 779-788.

[5] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//IEEE. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017: 6517-6525.

[6] REDMON J, FARHADI A. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08) [2023-07-10]. https://arxiv.org/abs/1804.02767.

[7] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. (2020-04-23) [2023-07-10]. https://arxiv. org/abs/2004.10934.

[8] JOCHER G, STOKEN A, BOROVEC J, et al. Ultralytics/yolov5: v4.0-nn. SiLU()activations, weights & biases logging, pytorch hub integration[EB/OL]. (2021-01-05) [2023-10-22]. https://zenodo.org/record/4418161.

[9] GE Zheng, LIU Songtao, WANG Feng, et al. YOLOX: exceeding YOLO series in 2021[EB/OL]. (2021-08-06) [2023-10-13]. https://arxiv.org/abs/2107.08430.

[10] 林军, 潘文波, 游俊, 等. 基于多传感器融合的视觉感知困难样本挖掘方法[J]. 机车电传动, 2021(6): 93-99.
LIN Jun, PAN Wenbo, YOU Jun, et al. Hard example mining method for visual perception based on multi-sensor fusion[J]. Electric drive for locomotives, 2021(6): 93-99.

[11] 周兆安, 李树枝. 基于深度学习的弓网异常状态在线检测研究[J]. 机车电传动, 2022(5): 135-143.
ZHOU Zhaoan, LI Shuzhi. Research on online inspection of pantograph and catenary based on deep learning[J]. Electric drive for locomotives, 2022(5): 135-143.

[12] 李昂, 孙士杰, 张朝阳, 等. 改进YOLOv5s的轨道障碍物检测模型轻量化研究[J]. 计算机工程与应用, 2023, 59(4): 197-207.
LI Ang, SUN Shijie, ZHANG Chaoyang, et al. Research on lightweight of improved YOLOv5s track obstacle detection model[J]. Computer engineering and applications, 2023, 59(4): 197-207.

[13] 卢艳东, 李积英, 王筱婷. 一种基于改进YOLOv3-tiny的轻量级轨道紧固件检测算法[J]. 铁道标准设计, 2022, 66(7): 41-47.
LU Yandong, LI Jiying, WANG Xiaoting. A lightweight track fastener detection algorithm based on improved YOLOv3-tiny[J]. Railway standard design, 2022, 66(7): 41-47.

[14] WANG C Y, MARK LIAO H Y, WU Y H, et al. CSPNet: a new backbone that can enhance learning capability of CNN [C]//IEEE. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Seattle: IEEE, 2020: 1571-1580.

[15] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: common objects in context[C]//Springer. Computer Vision-ECCV 2014. Zurich: Springer, 2014: 740-755.

[16] TAGIEW R, KÖPPEL M, SCHWALBE K, et al. OSDaR23: open sensor data for rail 2023[EB/OL]. (2023-05-04) [2023-10-13]. https://arxiv.org/abs/2305.03001.

Lightweight YOLO models for object detection based on low-rank decomposition

LIN Delü[1], LIU Chang[1], CHEN Qi[1], ZENG Yang[2], HE Kun[1]

(1. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China; 2. CRRC Zhuzhou Institute Co., Ltd., Zhuzhou, Hunan 412001, China)

Abstract: As train intelligence continues to advance, numerous studies have emerged to explore lightweight techniques for object detection models on onboard equipment, for the purpose of improving calculation efficiency amidst limited resources. This paper proposed a parameter compression algorithm based on low-rank decomposition for the You Only Look Once (YOLO) series of object detection models, aiming to overcome the limited versatility of current lightweight treatment methods for these models. Initially, calculations were conducted to determine a low rank using the preset low-rank ratio coefficient along with the number of input and output channels of convolution units. Subsequently, a new convolution sequence was obtained by performing the Tucker decomposition to the convolutional layer of the target structure. Lastly, a new convolution sequence was fused to replace the original convolutional layer. Experiments were conducted on the proposed parameter compression method based on low-rank decomposition, using public datasets and three models, i.e., YOLOv5-l, YOLOv8-x, and YOLOX-x. While ensuring an average detection accuracy of the models based on low-rank decomposition over 96% compared to the original models, the number of parameters and float-point calculations of these models reduced by about 40%. The experiments resulted in a nearly 50% improvement in image detection speed. Further visual displays illustrated similar receptive fields on the same images between the compressed and original models. The experimental results show that the proposed method is effective in lightweight compression of single-stage object detection models of the YOLO series and can enhance their usability for onboard equipment. In addition, this study serves as a valuable reference for the lightweight treatment of other models in the automatic train operation scenarios of rail transit field.

Keywords: deep learning; low-rank decomposition; automatic train operation; lightweight treatment of model; object detection

收稿日期：2023-08-21；修回日期：2024-01-01