



计算机科学

COMPUTER SCIENCE

基于Light-YOLOv8的围棋棋谱识别

张雷, 武文喆, 白雪媛

引用本文

张雷, 武文喆, 白雪媛. 基于Light-YOLOv8的围棋棋谱识别[J]. 计算机科学, 2024, 51(11A): 230900037-7.

ZHANG Lei, WU Wenzhe, BAI Xueyuan. [Go Chessboard Recognition Based on Light-YOLOv8](#)[J]. Computer Science, 2024, 51(11A): 230900037-7.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于多模态对比学习的场景图生成方法](#)

Multimodal Contrastive Learning Based Scene Graph Generation

计算机科学, 2024, 51(11A): 231200185-5. <https://doi.org/10.11896/jsjcx.231200185>

[基于双重标签分配的遥感有向目标检测方法](#)

Remote Sensing Oriented Object Detection Method Based on Dual-label Assignment

计算机科学, 2024, 51(11A): 240100058-9. <https://doi.org/10.11896/jsjcx.240100058>

[一种改进的基于YOLOv5s的轻量化航拍目标检测模型](#)

Improved Lightweight Aerial Photography Object Detection Model Based on YOLOv5s

计算机科学, 2024, 51(11A): 231100119-8. <https://doi.org/10.11896/jsjcx.231100119>

[基于YOLOv8改进的脑癌检测算法](#)

Enhanced Brain Cancer Detection Algorithm Based on YOLOv8

计算机科学, 2024, 51(11A): 231100122-7. <https://doi.org/10.11896/jsjcx.231100122>

[PS-YOLOv8:增强电力线路检测中的小规模损坏检测](#)

PS YOLOv8:Enhancing Detection of Small-scale Damage in Power Lines Inspection

计算机科学, 2024, 51(11A): 240100003-6. <https://doi.org/10.11896/jsjcx.240100003>

基于 Light-YOLOv8 的围棋棋谱识别

张雷¹ 武文喆¹ 白雪媛²

1 沈阳航空航天大学电子信息工程学院 沈阳 110136

2 沈阳航空航天大学理学院 沈阳 110136

(rd_zhangl@126.com)

摘要 为实现围棋对弈过程中的高精度实时记谱,提出了一种基于结合三维注意力机制与轻量化卷积的实时检测算法 Light-YOLOv8。在 YOLOv8 模型的基础上,使用 PWConv+PConv 替换主干网络中跨阶段局部网络的 3×3 卷积,大幅减少模型计算量与参数规模;加入 CARAFE 上采样算子与 SimAM 三维注意力机制,提高对围棋目标的检测能力;使用 Wise-IOU 损失函数提高模型定位能力与收敛速度,提高了对棋子粘连、棋子重叠与光照不均匀情况下的检测能力。在自定义围棋数据集上进行对比训练表明,改进后的算法实现了检测精度的提升与推理速度的提高。针对移动端设备部署需求对模型进行优化与压缩,并在不同安卓设备部署,图像分辨率为 640×480 的情况下,结合图像预处理与后处理操作,拍照检测平均时间为 89 ms,平均模型推理帧率为 37.6 fps。进行 50 轮记谱实验,平均记谱准确率高于 97%,平均胜负判别准确率到达 100%,能够实现稳定的围棋记谱功能。

关键词: 目标检测;棋局识别;实时记谱;YOLOv8;轻量化网络;移动设备

中图分类号 TP391

Go Chessboard Recognition Based on Light-YOLOv8

ZHANG Lei¹, WU Wenzhe¹ and BAI Xueyuan²

1 Department of Electronic Information Engineering, Shenyang Aerospace University, Shenyang 110136, China

2 Department of Science, Shenyang Aerospace University, Shenyang 110136, China

Abstract A real-time detection algorithm Light-YOLOv8 based on a combination of three-dimensional attention mechanism and lightweight convolution is proposed to achieve high-precision real-time chessboard recording during Go games. On the basis of YOLOv8, PWConv+PConv is used to replace the 3×3 convolution of the cross stage local network in the backbone network, which greatly reduce the computational complexity of the model. Adding CARAFE upsampling operator and SimAM three-dimensional attention mechanism to improve the detection ability of small Go targets. The use of the Wise-IOU loss function improves the model's localization ability and convergence speed, and improves its detection ability in cases of chess piece adhesion, chess piece overlap, and uneven lighting. Optimize and compress the model for mobile deployment and deploy it on different Android devices, with an image resolution of 640×480 . The average single detection time combined with image preprocessing and post-processing operations is 89ms, and the average detection frame rate is 37.6 fps. Conduct 50 rounds of score recording experiments, with an average score recording accuracy of over 97% and an average winner/loser discrimination accuracy of 100%, which can achieve stable go chess score recording function.

Keywords Object detection, Chess game recognition, Real time notation, YOLOv8, Lightweight network, Mobile devices

1 引言

围棋棋谱记录与胜负判别在围棋教学、围棋竞赛等应用场景有十分重要的意义,传统围棋记谱与胜负判别方式主要依赖人工,难以保证效率与准确性。计算机视觉与图像处理高速发展,围棋识别受到国内外学者的广泛研究。

一些学者采用基于形态学与图形学的传统目标检测算法进行棋子检测。Liang 等^[1]利用灰度级插值与空间变换分割出棋盘区域,解决了不同棋盘放置角度的影响,通过灰度直方图确定黑白棋子的阈值,提取出黑白棋子区域。Mao 等^[2]提出了一种棋盘分割方法,通过边沿检测获得棋盘外边沿的若干采样点,使用平均值法与极值法确定棋盘 4 个角点的位置

以获得棋盘边沿。Gui 等^[3]提出了一种基于霍夫圆检测的围棋棋子定位与识别方法,通过霍夫圆特征对棋子进行定位,使用阈值分割进行棋子分类。Wang 等^[4]提出了一种针对光照不均匀下的通用棋子定位方法,通过凸包检测分割棋盘区域,通过内轮廓检测与多边形拟合进行棋盘分类,通过图像掩膜进行棋子定位。Zhang 等^[5]提出了一种基于霍夫变换的围棋棋谱生成方法,通过霍夫线检测定位棋盘的交叉线,对交叉点进行邻域像素检测以判断棋子的有无或种类。

传统算法普遍对光照、旋转、尺度、形变、噪声和遮挡敏感,识别精度受环境、图像采集质量与预处理方法等多方面影响,一些学者开始采用深度学习算法进行研究。深度学习方法主要有单阶段检测与双阶段检测,前者主要有 SSD^[6],

SqueezeDet^[7], EfficientNet 系列^[8], YOLO 系列^[9]等;后者主要有 RCNN 系列^[10-11]、SPPNet^[12]等。Kong 等^[13]提出了一种针对中国象棋的分类检测方法,使用基于 Inception 模块的卷积神经网络进行棋子分类。Liu 等^[14]提出了一种基于 EfficientNet 的端到端国际象棋检测方法,采用 k-means 聚类对锚框(anchor boxes)尺寸进行设定,实现了端到端的棋子分类与定位。

卷积操作是卷积神经网络最耗时的操作,Howard 等^[15]提出了深度可分离卷积,该卷积由逐深度卷积(DWConv)与逐点卷积(PWConv)组成。由于卷积核尺寸远小于输出通道数,因此深度可分离卷积有效减少了卷积操作的计算参数。网络剪枝是减少模型计算量最有效的方法,Fang 等^[16]提出了基于非深度图算法的结构化剪枝方法(DepGraph),通过依赖图解决了参数分组问题,实现了通用性的深度网络剪枝。

围棋棋谱识别系统应满足计算快速、定位准确、无漏检与误检的需求,并且能够在光照不均匀与非固定背景环境下稳定工作。为实现高效且准确的围棋棋谱识别,并且易于部署在移动设备中,本文提出了一种轻量化的围棋目标检测算法 Light-YOLOv8。算法使用 PWConv+PConv 卷积与 CARAFE 上采样算子替换标准卷积与最邻近上采样,增加 SimAM 注意力机制并使用 Wise-IoU 损失函数提高对围棋的检测能力。使用安卓设备进行实时图像的采集,对棋盘区域进行裁切与矫正以去除无效背景,通过棋子检测模型检测棋子的位置与种类,对比棋子中心点与棋盘交叉点的空间分布,确定棋子的坐标,完成棋局检测。当围棋棋盘中出现遮挡时停止棋子检测,当棋盘完整时重新进行棋子检测,以应对对弈过程中的落子环节对棋局检测的影响。当棋局发生变动时确定最新落子的种类与位置更新棋谱,当终局确定时,根据规则判定棋局胜负,完成记谱功能。

2 围棋棋子检测算法设计

围棋棋谱识别与记谱流程如图 1 所示。通过对采集到的图像进行目标检测来初步确定棋盘与棋子的位置,裁切棋盘位置来排除棋盘外的棋子或其他物体的干扰,通过透视变换检测精确棋子中心点的位置,将棋盘交叉点与棋子中心点的坐标进行比对,将最邻近交叉点的二维棋盘位置确定为棋子位置,生成当前棋局。记谱是连续的棋局判断,棋局变化主要由落子与提子两种操作导致,由于 sgf 棋谱格式中不记录提子操作,因此记谱过程即找到每回合最新落子的坐标。

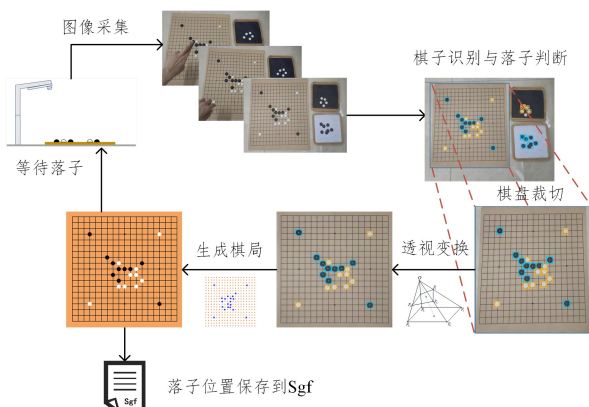


图 1 围棋棋谱识别整体流程

Fig. 1 Overall process of go chess score recognition

2.1 棋子检测模型

YOLOv8 网络由主干网络(Backbone)与检测头(Head)两部分组成,主干部分由 Conv 模块、C2f 模块与 SPPF 模块组成,主要功能为特征提取;检测头部分由上采样模块与 C2f 模块组成,主要功能为特征重组与多尺度目标检测。为实现轻量化的目标并提高模型对于小目标的检测能力,Light-YOLOv8 网络在 YOLOv8 的基础上,将主干部分的跨阶段局部网络中的 3×3 卷积变为轻量化卷积 PConv+PWConv,将 FPN 部分的最邻近上采样算子变为动态轻量化上采样算子 CARAFE,引入无参注意力机制 SimAM 以及将位置损失函数变为 Wise IOU 等。Light-YOLOv8 网络结构如图 2 所示。

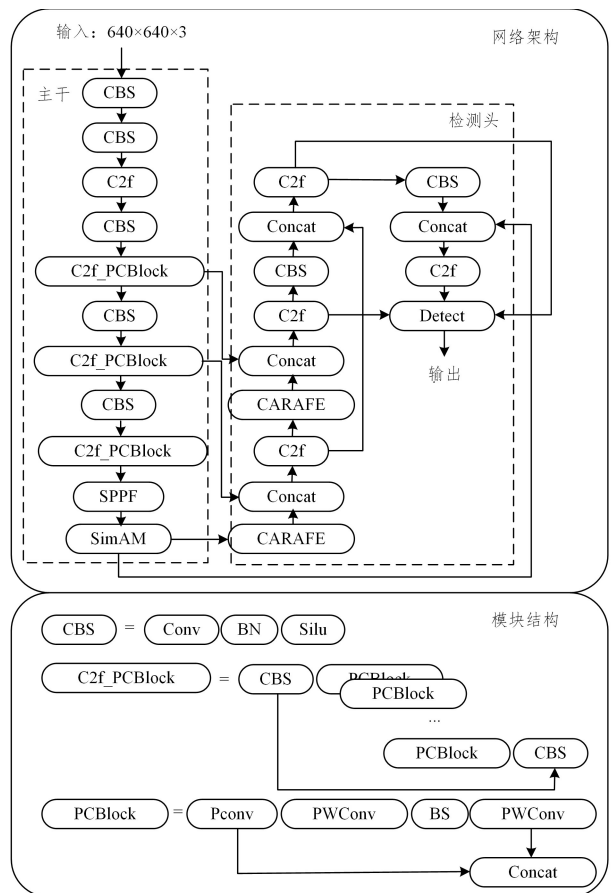


图 2 Light-YOLOv8 整体网络架构

Fig. 2 Overall network architecture of Light-YOLOv8

2.1.1 PConv+PWConv 卷积

PConv 是一种新型的卷积方法,能够有效地减少冗余计算与内存访问,在 x86 架构与 ARM 架构的处理器(CPU)以及图形处理器(GPU)上均有着计算快速的特点。特征图的不同通道之间存在很高的相似性,这在计算上则体现了冗余性^[7]。因此 PConv 引入掩码来决定一部分通道需要进行进一步的卷积操作,而被掩码遮盖的通道则保持不变,这样的卷积操作所需要的参数量从标准卷积的 $H \times W \times C^2 \times k^2$ 降至 $H \times W \times C_p \times k^2$,其中 H 与 W 为特征图的高与宽, C 为总通道数, C_p 为需要进行卷积操作的通道数,采样比 $r = C_p / C$ 通常为 $1/4$ 。

PWConv 常置于 PConv 后,用于融合通道之间的特征^[7]。PWConv 由一组 1×1 的卷积核组成,在不引入额外计算量的同时进行通道特征的融合与输出通道的扩展。对

PWConv的输出进行批标准化处理(Batch Normalization)操作后通过激活函数,综合考虑精度与速度,选择SiLu作为激活函数。PConv+PWConv卷积操作如图3所示。

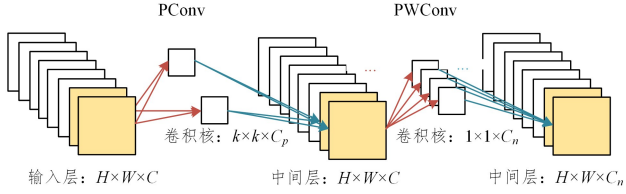


图3 PConv+PWConv示意图

Fig. 3 Schematic diagram of PConv+PWConv

2.1.2 CARAFE上采样算子

CARAFE算子是应用于上采样的卷积核,相比线性插值或反卷积上采样操作,CARAFE具有更大的感受野,能够在引入较少计算量的同时更好地利用特征图上下文信息^[8]。基于“编码器-解码器”结构的思想,CARAFE上采样过程分为构建上采样卷积核与特征重组两个阶段。

第一阶段,首先通过 1×1 卷积将输入通道数 C 压缩至 C_m 以减少计算量。随后通过 $k_{enc} \times k_{enc}$ 卷积将原始特征图映射到 $\sigma \times \sigma \times k_{up} \times k_{up}$ 个 $H \times W$ 特征图的特征空间,将特征图通过图4所示的方式拼接,获得尺寸为 $\sigma H_{in} \times \sigma W_{in}$ 的特征空间。在 $\sigma H_{in} \times \sigma W_{in}$ 中,某一位置 (i', j') 是一组 $1 \times 1 \times k_{up}^2$ 的特征,包含了 $H_{in} \times W_{in}$ 特征图中 (i, j) 位置的信息。使用softmax对 (i', j') 进行归一化,实现对原始特征的重组,将 $1 \times 1 \times k_{up}^2$ 的特征变换为 $k_{up} \times k_{up}$,得到了重组核。

第二阶段,找到重组核 (i', j') 在原始图像中的对应位置 (i, j) , (i', j') 与 (i, j) 的映射关系为 $i = \text{floor}(i'/\sigma)$, $j = \text{floor}(j'/\sigma)$ 提取以 (i, j) 为中心的 $k_{up} \times k_{up}$ 大小的邻域,使其与重组核进行点积,获得输出特征图中 (i', j') 位置的值,对于输入特征图的不同通道,相同位置采用相同的重组核进行重建。

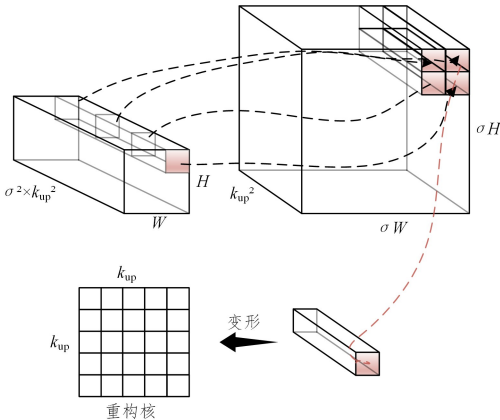


图4 CARAFE上采样算子示意图

Fig. 4 Schematic diagram of CARAFE upsampling operator

2.1.3 三维注意力机制

SimAM是无参数的三维注意力模块,在不引入额外计算量的同时融合了一维通道注意力与二维空间注意力,将注意力细化到像素级别,使网络更加聚焦于有用特征。

在针对图像的卷积神经网络中,每一个单通道的像素点被称为一个神经元。在脑科学中,激活的神经元对于视觉有着更大的贡献度,而激活的神经元会抑制周围的神经元,因此神经元的重要程度与其最小能量成反比^[19]。度量神经元之

间的线性可分性能够有效辨别该神经元是否被激活。SimAM引入能量函数 e_i 量化神经元的能量,并通过最小化能量函数判断该神经元与该通道内其他神经元是否具有线性可分性,其数学表达式为:

$$e_i(\omega_i, b_i, y, x_i) = \frac{1}{M-1} \sum_{i=1}^{M-1} (-1 - (\omega_i x_i + b_i))^2 + (1 - (\omega_i x_i + b_i))^2 + \lambda \omega_i^2 \quad (1)$$

$$\omega_i = -\frac{2(t - \mu_i)}{(t - \mu_i)^2 + 2\sigma_i^2 + 2\lambda} \quad (2)$$

$$b_i = -\frac{1}{2}(t + \mu_i)\omega_i \quad (3)$$

$$\mu_i = \frac{1}{M-1} \sum_{i=1}^{M-1} x_i \quad (4)$$

$$\sigma_i^2 = \frac{1}{M-1} \sum_{i=1}^{M-1} (x_i - \mu_i)^2 \quad (5)$$

$$e_i^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (6)$$

其中, ω_i 与 b_i 分别是超平面的斜率与偏置, μ_i 与 σ_i^2 分别是第 t 通道中神经元的期望与方差, λ 为正则项系数, M 为神经元的个数。采用二值化标签1与-1表示正负样本,对于当前神经元而言,同一通道内的其他神经元都为负样本,因此激活函数的第一项为超平面与负样本的平均距离;第二项为超平面与正样本的平均距离;第三项为正则项,用于防止模型过拟合。将能量函数最小化可以得到 e_i^* ,重要性系数 $I = 1/e_i^*$,将每个神经元的重要性系数通过sigmoid函数归一化即可生成神经元的权重系数,每一个神经元经过SimAM的输出为 $\tilde{X}_i = \text{sigmoid}(I) * X_i$ 。

2.1.4 Wise-IOU损失函数

由于YOLOv8采用无预设锚框(anchor free)检测方法,因此采用Wise-IOU(WIOU)代替CIOU作为边界框回归(box)的损失函数,动态调整预测框的梯度增益,提高预测的整体性能。与CIOU相比,WIOU在基础的IOU上并没有引入纵横比等其他评价指标,节约了计算成本。

WIOU是一种动态非单调的聚焦机制,能够根据预测框与真实框之间的离群程度动态分配梯度^[20],WIOU的损失函数 L_{WIOU} 的数学表达式为:

$$L_{WIOU} = r R_{WIOU} L_{IOU}, \gamma > 0 \quad (7)$$

$$R_{WIOU} = \exp\left(\frac{(x - x_g)^2 + (y - y_g)^2}{(W_g^2 + H_g^2)^*}\right) \quad (8)$$

$$\beta = \frac{L_{IOU}^*}{L_{IOU}} \quad (9)$$

$$r = \frac{\beta}{\delta \alpha^{\beta - \delta}} \quad (10)$$

其中, L_{WIOU} 是常规IOU损失; R_{IOU} 是 L_{WIOU} 的质量评价系数,用来评价预测框的质量,其大小与预测框和真实框中心点的欧氏距离成正比,与预测框和真实框的最小外接矩形形成反比,当中心距离较大时,给预测框分配较大的梯度增益使其加速收敛; r 是 L_{WIOU} 的聚焦系数,用于聚焦于普通质量的预测框;离散度 β 是当前损失 L_{WIOU} 与平均损失 $\overline{L_{WIOU}}$ 的比值; α 与 δ 是经验系数。如图5所示,当离散度过高和过低时,预测框都会获得较小的梯度增益,引入 r 可以使损失函数聚焦于普通锚框,以提高模型的定位能力。

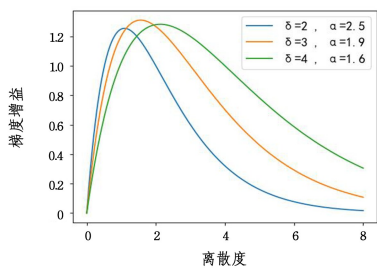


图5 不同参数下离散度与梯度收益之间的关系
Fig. 5 Relationship between discreteness and gradient returns with different parameters

2.2 棋盘局面生成

2.2.1 棋盘分割

为了分割棋盘区域,需要检测棋盘轮廓,由于棋盘为四边形,因此将轮廓检测简化为定位棋盘的4个顶点。将裁切获得的棋盘图像进行灰度化处理,并采用高斯滤波对图像进行降噪。然后对灰度图像进行自适应阈值的二值化操作,阈值由像素邻域的像素值所确定。像素邻域为 $n \times n$ 大小的矩形区域,计算该区域的平均像素值 (\bar{x}, \bar{y}) ,并使用参数 C 对平均像素值进行缩放以获得阈值,阈值 T 的计算式为:

$$T = \begin{cases} (x_i, y_i) \leq (\bar{x}, \bar{y}) + C, & 0 \\ (x_i, y_i) > (\bar{x}, \bar{y}) + C, & 255 \end{cases} \quad (11)$$

对二值化图像使用 Canny 算子进行边缘检测以获得更为清晰的棋盘边缘图像,使用轮廓提取得到多组棋盘边缘候选点集,将每个点集中的候选点按顺序依次连接得到凸包。采用 Douglas-Peucker 算法对凸包进行凸多边形拟合,拟合精度为凸包轮廓周长的 10%,筛选出拟合结果中的凸四边形并依次计算其面积,得到最大凸四边形,由此得到了棋盘位置以及棋盘顶点位置。图 6 给出了棋盘分割的过程。

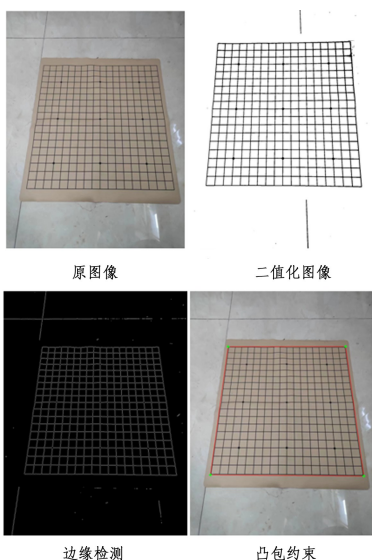


图6 棋盘分割过程示意图

Fig. 6 Schematic diagram of go board segmentation process

2.2.2 棋盘校正

用透视变换对分割后的棋盘图像进行校正,使棋盘图像变换为标准棋盘图像。原始图像中棋盘顶点为 (P_u, P_b, P_n, P_o) ,标准棋盘图像中棋盘顶点为 (P'_u, P'_b, P'_n, P'_o) 。

透视变换指将原始图像投影到目标平面图像的变换

操作,变换关系为 $P' = MP$,其中 P 为原始图像的像素坐标, P' 为目标平面上的像素坐标, M 为变换矩阵,将变换关系展开可得:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} L_{2 \times 2} & T_{2 \times 1} \\ N_{1 \times 2} & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (12)$$

其中, $L_{2 \times 2}$ 为线性变换矩阵, $T_{2 \times 1}$ 为平移变换, (u, v) 是原始图像中的像素坐标, (x, y, z) 是经过透视变换获得的三维坐标,其在目标平面上的投影坐标为 (x', y') ,转换关系为:

$$x' = \frac{x}{z}, y' = \frac{y}{z} \quad (13)$$

将原始图像中的顶点坐标与标准棋盘图像中的预期顶点位置坐标分别依次带入式(13)得到透视变换的变换矩阵,将原始图像中棋盘区域的每个像素进行变换得到校正后的标准棋盘图像。设定标准棋盘图像的大小,将标准棋盘图像大小设置为 620×620 ,以包含完整的棋盘图像,再向边缘延拓 10 个像素,生成大小为 640×640 的包含棋盘与棋盘边缘的标准棋盘图像。图 7 给出了棋盘校正的过程。

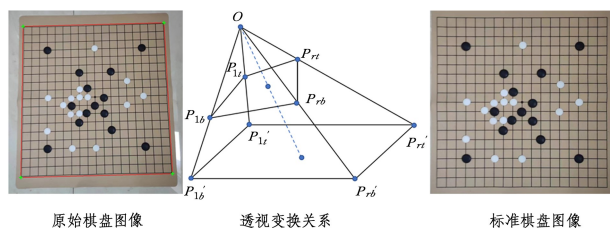


图7 棋盘校正过程示意图

Fig. 7 Schematic diagram of go board correction process

2.2.3 生成棋局

棋局状态可抽象为二维矩阵,每个点存在 3 种状态,即黑子、白子与无子,在棋谱初始化阶段,所有位置均视为无子状态。使用 Shi-Tomasi 算子对空白的标准棋盘图像进行角点检测,并对角点进行从左到右、从上到下的顺序排序,获得棋盘交叉点的坐标,并将棋盘交叉点作为基点。随后,找到各棋子中心点的最邻近基点,认为棋子属于最邻近基点,将棋谱矩阵中的对应位置根据棋子类别进行状态更新,识别结果如图 8 所示。

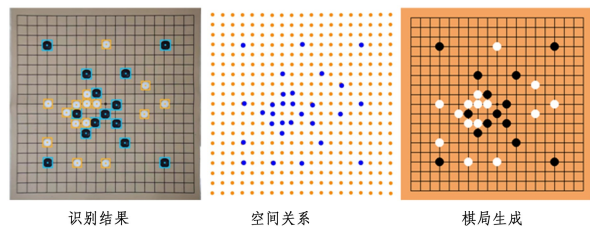


图8 棋子识别与棋局生成示意图

Fig. 8 Schematic diagram of chess piece recognition and chess game generation

用 KD-tree 进行最邻近基点的查找,KD-tree 算法会依次交换维度并对棋盘的交叉点进行均分,以此构建其二叉树结构。通过二叉树查找逐步缩小搜索范围,搜索到没有子树时将棋子坐标与叶子节点坐标依次比较从而获得最邻近点。相比遍历搜索的线性扫描法,KD-tree 的搜索效率提升了 $n/\log n$ 倍,围棋棋盘包含 19 路、13 路与 9 路,KD-tree 相比线性扫描法效率分别提高 14.85,11.67,9.43 倍。

3 实验与结果分析

3.1 数据集制作

采集不同场景棋盘与棋子图片 500 张,使用 Roboflow 数据标注对图像进行标注,标注对象分别为棋盘、黑子与白子;通过旋转、裁切、加噪丰富图像背景;通过遮罩、亮度调整等模拟多样性的应用场景;将多张图像进行裁切与拼接,在提高批大小(batchsize)的同时生成更多小目标特征。

通过上述图像增强技术将数据集扩容,最终生成样本量为 1500 的 YOLO 格式的数据集,通过 7:2:1 的比例将数据集划分为训练集、测试集与验证集。

3.2 模型训练与结果对比

在 64bit Windows11 的硬件设备上搭建基于 Pytorch 框架的模型,使用 GeForce GTX 2050 GPU(4 GB)进行模型训练,开启混合精度训练模式加速网络训练,模型训练的参数如表 1 所列。

表 1 网络训练参数

Table 1 Network training parameters

参数名称	参数意义	数值
Epoch	训练论述	300
Batchsize	批量大小	8
lr0	初始学习率	0.01
lrf	循环学习率	0.01
momentum	学习率动量	0.937
box	iou 损失系数	7.5
cls	分类损失系数	0.5

3.2.1 对比实验

基于 Windows 平台,选择双阶段经典目标检测网络 Fas-

ter RCNN 和 SSD,以及单阶段网络 EfficientNet, YOLOv4, YOLOv5, YOLOv8 和本文提出的模型 Light-YOLOv8 在围棋数据集上进行对比实验。根据平均检测时间(MTTD)、单一类别的平均精确率(AP)和总平均精确率(mAP)这 3 个指标上的表现比较不同模型的性能。

表 2 不同算法在围棋数据集上的性能比较

Table 2 Performance comparison of different algorithms on go chess dataset

算法	AP-50/%			MTTD/ms	mAP-50/%
	黑子	白子	棋盘		
Faster R-CNN	89.31	88.31	90.50	173.0	89.37
SSD	87.51	87.21	89.75	55.0	88.16
EfficientNet	90.11	90.76	92.85	49.0	91.20
YOLOv4	92.82	91.32	94.75	41.0	92.96
YOLOv5	95.05	95.08	98.51	36.0	96.21
YOLOv8	95.83	96.13	98.88	13.0	96.95
Light-YOLOv8	96.70	96.76	96.48	9.6	96.65

由表 2 所列的对比实验结果可知, YOLOv8 在多尺度目标检测任务中对比其他经典目标检测网络均有性能上的提升。Light-YOLOv8 在与其他主流算法进行对比时也有精度与速度上的优势,与 YOLOv5 相比,模型推理速度加快了 73.3%,与 YOLOv8 相比推理速度也有了 26.2% 的增长,能够满足实时推理的需求。同时,改进后的网络提高了对小目标的检测精度,符合预期需求。

3.2.2 消融实验

将改进模块依次加入网络,分别记录改进后网络的参数量、计算量(GFLOPs)与总平均精度(mAP),结果如表 3 所列。

表 3 基于围棋数据集的消融实验

Table 3 Ablation experiment of go chess dataset

PConv+PWConv	SimAM	CARAFE	WIOU	参数量	计算量/GFLOPs	mAP ₅₀ /%	mAP ₅₀₋₉₅ /%
				3.16×10^6	8.9	96.95	85.61
✓				2.61×10^6	7.2	93.31	83.22
✓	✓			2.61×10^6	7.2	94.92	83.96
✓	✓	✓		2.75×10^6	7.5	95.54	84.85
✓	✓	✓	✓	2.75×10^6	7.5	96.65	85.11

消融实验结果表明,使用 PConv+PWConv 代替主干网络中的 C2f 卷积使参数量减少了 17.4%,计算量减少了 19.1%;引入无参注意力机制 SimAM 使得 mAP₅₀ 与 mAP₅₀₋₉₅ 分别提升了 1.61% 与 0.74%;使用 CARAFE 算子代替 FPN 模块中的最邻近上采样算子,在仅增加 0.14×10^6 参数量与 0.3 GFLOPs 计算量的同时使得 mAP₅₀ 与 mAP₅₀₋₉₅ 分别提升了 0.62% 与 0.89%;使用 WIOU 代替 CIOU 作为 box 损失函数使得 mAP₅₀ 与 mAP₅₀₋₉₅ 分别提升了 1.11% 与 0.26%。

对结果进行分析,PConv 缩小了特征提取的范围导致识别精度小幅减小,但减少了参数量与内存开销,加快了推理速度;引入三维无参注意力机制 SimAM,进一步细化特征权重,提高网络对有用特征的提取能力;采用 CARAFE 上采样算子代替最邻近上采样扩大了特征重组的感受野,提升了网络的特征融合能力;采用 WIOU 代替 CIOU 更好地适配于 anchor-free 目标检测网络,提高了网络的检测能力。本文改进的 Light-YOLOv8 成功实现了对原网络的轻量化改进,同时

提高了对小目标的检测能力。

4 移动设备部署验证

通过移动端高性能神经网络计算框架 NCNN 对神经网络进行模型剪枝与参数量化,实现移动端的部署与端侧推理。设置采集图像分辨率为 640×480 ,通过拍照检测实验验证系统检测精度与运行速度,根据实验确定最优图像采集高度与推理所用处理器,并在此条件下进行录制识谱,验证围棋棋谱识别系统的工作速度、精度与稳定性。

4.1 拍照检测

选用不同软硬件配置的安卓设备,将设备分别固定在距离目标 45cm,60cm 与 75cm 的位置,在晴天自然光环境、阴天自然光环境、夜间灯光环境下对布面与木板棋盘进行图像采集,并分别使用 CPU 与 GPU 进行推理,记录棋子的漏检率与误检率,实验结果如表 4 所列。根据表 4 的实验结果可知,系统的端侧运行速度受到硬件设备的影响,识别精度受到检测距离的影响,对棋盘材质、环境光照与拍摄角度不敏感。由

于移动端 GPU 没有 cuda 核心,其运行速度普遍慢于 CPU 推理。此外,系统在检测距离小于 60 cm 时能够稳定工作,检测距离小于 40 cm 时无法完全采集棋盘图像,大于 60 cm 时检测

漏检率迅速上升。综合检测精度与运行速度,将系统的检测距离设置为 60 cm,该设置为最优工作条件,采用 CPU 进行模型推理。

表 4 拍照检测实验
Table 4 Photographic detection experiment

设备	支持 Vulkan	处理器	检测距离/cm	平均亮度/lux	棋盘材质	MTD/ms	漏检率/%	误检率/%	
Redmi K40 Pro	✓	CPU	45	1734	布面	24.93	0.8	0.0	
					木板	25.59	1.6	0.0	
			60	239	布面	25.21	1.7	0.1	
					木板	23.17	1.2	0.2	
				75	63.7	布面	26.14	25.4	2.0
						木板	24.84	21.4	2.0
		GPU	45	1734	布面	35.48	0.0	0.1	
					木板	35.59	1.5	0.0	
			60	239	布面	32.29	1.2	0.0	
					木板	35.33	1.8	0.1	
				75	63.7	布面	33.78	18.9	3.1
						木板	34.88	19.6	2.0
Le X620	✗	CPU	45	1604	布面	165.40	2.0	0.0	
					木板	168.49	2.5	0.0	
			60	227	布面	162.79	1.2	0.1	
					木板	181.61	2.0	0.2	
				75	101.5	布面	179.36	34.6	2.0
						木板	177.02	34.9	2.0
		GPU	45	1537	布面	84.20	1.5	0.0	
					木板	85.20	1.9	0.0	
			60	227	布面	84.80	1.9	0.1	
					木板	85.82	1.3	0.2	
				75	208	布面	88.12	28.4	1.9
						木板	84.30	25.5	2.0
Google Pixel 3a	✓	CPU	45	1537	布面	129.80	0.0	0.0	
					木板	149.55	1.7	0.0	
			60	227	布面	116.81	2.5	0.1	
					木板	148.65	1.3	0.2	
				75	208	布面	112.00	19.5	2.5
						木板	109.42	22.9	2.1

4.2 录制识谱

点击开始记谱按钮,对棋局图像进行实时采集,每间隔 5 帧取一幅有效图像。连续取 3 幅有效图像并依次进行差分,若图像的变化程度低于阈值且没有手部遮挡,则认为棋手完成落子或提子操作,棋局固定。取最后一帧图像作为当前

棋局图像进行识别,将当前识别棋局结果与上一步棋局识别结果进行对比,从而记录每一个落子的序列关系,将落子坐标写入 sgf 文件实现记谱功能,点击结束记谱按钮结束识别并保存 sgf 棋谱。图 9 给出了某一局对弈过程中第 15 手、第 45 手与终局的实时棋局捕获与识别结果。

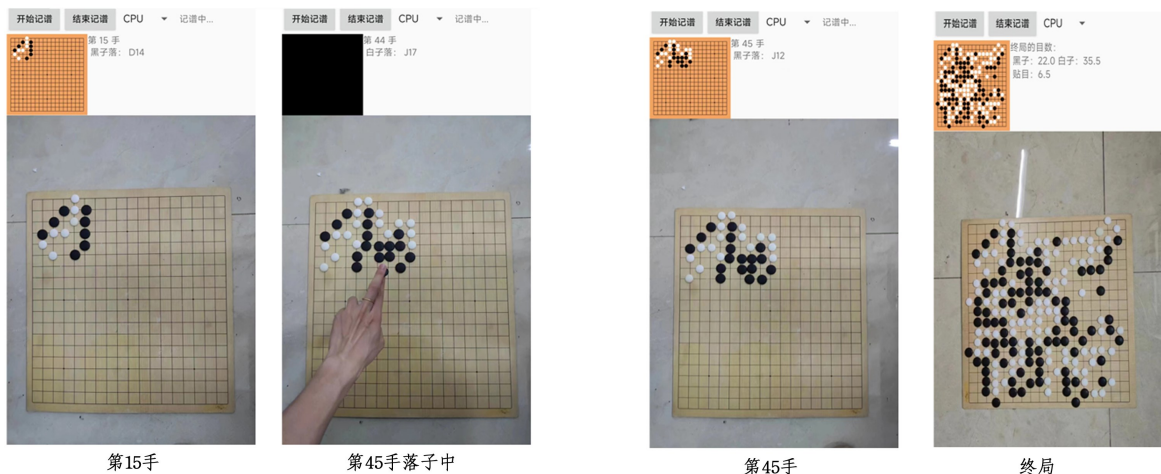


图 9 实时棋局捕获与识别结果

Fig. 9 Real time chess game capture and recognition results

固定检测距离,使用不同安卓设备与不同材质棋盘进行 50 轮记谱实验,记录系统识别的帧率,对比采集的图像

与识谱结果,统计终局胜负判断的准确性,实验结果如表 5 所列。

表 5 录制识谱实验

Table 5 Record go board recognition experiment

设备	平均帧率/fps	棋谱识别 准确率/%	胜负判别 准确率/%
Redmi K40 Pro	60.15	99.2	100
LeX620	22.54	97.1	100
Google Pixel 3a	30.12	98.7	100

结束语 本文使用深度学习方法解决了非固定背景下围棋识别中的棋子定位、棋局识别、棋谱记录与胜负判别问题,该方法可以广泛应用于下棋机器人视觉系统、围棋比赛电子裁判、围棋辅助教学等场景中。

在 YOLOv8 模型中将 3×3 普通卷积替换为 PWConv+PCConv 卷积,将最邻近上采样替换为 CARAFE 上采样算子,加入 SimAM 注意力机制并使用 Wise-IOU 损失函数使得模型的网络参数量减少了 13.0%,计算量减少了 12.7%,推理速度提高了 26.2%。本文方法能够快速、精确地实现棋盘区域的识别与分割、棋子的分类与定位、落子判断与棋局生成,并能够在连续对弈过程中识别棋局的变化与遮挡,实现记谱功能。在拍摄距离 60cm 的条件下对不同安卓设备进行 50 轮记谱实验,平均检测帧率为 37.6fps,记谱准确率高于 97%,胜负判别准确率到达 100%,能够有效解决棋子粘连、光照不均匀、画面畸变等问题。

但相较于 YOLOv8,本文模型 Light-YOLOv8 在棋盘识别率上有所下降,这是由于通过轻量化卷积替换标准卷积会导致部分特征通道信息的丢失,并且轻量化卷积还会限制感受野的大小,从而降低了大目标的识别精度。

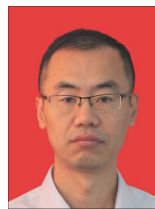
在后续的研究中,将开展在下棋机器人中的应用研究,并通过轻量化主干替换、模型剪枝等轻量化手段对模型体积进行进一步优化,以部署在更多的边缘计算设备中。

参 考 文 献

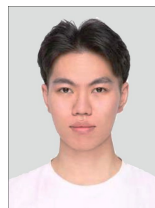
- [1] LIANG C Y, DING Y, YUAN J, et al. Research on Image Recognition Algorithms for Go Tournaments [J]. Modern Computer (Professional Edition), 2016(24): 39-43.
- [2] MAO L M, ZHU P Y, LU Z L, et al. Recognition of Gobang Robot Chessboard Based on LabVIEW [J]. Computer Engineering and Design, 2017, 38(1): 242-246.
- [3] GUI Y, WU Y, WANG Y, et al. Visual Image Processing of Humanoid Go Game Robot Based on OPENCV [C] // 2020 Chinese Control And Decision Conference (CCDC). Hefei, 2020: 3713-3716.
- [4] WANG Y J, ZHANG Y B, WU Y Y, et al. A General Chess Positioning Method under Uneven Illumination [J]. Computer Applications, 2020, 40(12): 3490-3498.
- [5] ZHANG Y, GU F. A Compact and Low-cost gobang Robot [C] // 2022 International Conference on Service Robotics (ICoSR). Chengdu, 2022: 62-66.
- [6] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector [C] // Computer Vision - ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, Part I 14. Springer International Publishing, 2016: 21-37.
- [7] WU B, IANDOLA F, JINP H, et al. Squeezednet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving [C] // Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017: 129-137.

- [8] TAN M, LE Q. Efficientnetv2: Smaller models and faster training [C] // International Conference on Machine Learning. PMLR, 2021: 10096-10106.
- [9] JIANG P, ERGU D, LIU F, et al. A Review of Yolo algorithm developments [J]. Procedia Computer Science, 2022, 199: 1066-1073.
- [10] GIRSHICK R. Fast r-cnn [C] // Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440-1448.
- [11] HE K, GKIOXARI G, DOLLÁR P, et al. Mask r-cnn [C] // Proceedings of the IEEE International Conference on Computer Vision, 2017: 2961-2969.
- [12] PURKAITP, ZHAO C, ZACH C. SPP-Net: Deep absolute pose regression with synthetic views [J]. arXiv: 1712.03452, 2017.
- [13] KONG F G, LI Z Z, LIU Q, et al. Design of the Vision System for Chinese Chess Robots [J]. Mechanical Engineer, 2020, 347(5): 1-3.
- [14] LIU X, JIN J, ZHONG Q, et al. Vision-Based Chess Detection for a Robotic Companion [C] // 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCS). Chengdu, 2021: 248-253.
- [15] HOWARDA G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications [J]. arXiv: 1704.04861, 2017.
- [16] FANG G F, MA X Y, SONG M L, et al. [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023: 16091-16101.
- [17] CHEN J, KAO S, HE H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 12021-12031.
- [18] WANG J, CHEN K, XU R, et al. CARAFE: Content-aware re-assembly of features [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 3007-3016.
- [19] YANG L, ZHANG R Y, LI L, et al. SimAM: A simple, parameter-free attention module for convolutional neural networks [C] // International Conference on Machine Learning. PMLR, 2021: 11863-11874.
- [20] TONG Z, CHEN Y, XU Z, et al. Wise-IOU: bounding box regression loss with dynamic focusing mechanism [J]. arXiv: 2301.10051, 2023.



ZHANG Lei, born in 1972, Ph.D, associate professor. His main research interests include image processing and image compression.



WU Wenzhe, born in 1998, postgraduate. His main research interests include image processing and pattern recognition.