

基于 yolo 的扑克牌识别

李睿涵

国科大杭州高等研究院

Abstract— 本文详细介绍了如何运用 yolo 构建一个可以识别扑克牌的图像识别系统。该系统基于 YOLOv8 模型，展示了不同的性能指标，如 mAP 等。文章解释了 YOLOv8 的原理，提供了相应的训练数据集，并集成了一个基于 PySide6 的界面，展示了最终对于扑克牌的识别结果。

Index terms—yolov8, 图像识别

I. 简介

扑克牌识别作为一项重要的计算机视觉应用，在近年来得到了广泛的研究和应用。其背景不仅仅局限于传统的博彩行业，还扩展到了智能监控、娱乐互动、自动化魔术表演以及教育训练等多个领域。在这些应用中，扑克牌识别技术的意义重大，它可以帮助系统快速准确地识别扑克牌，进而实现自动化决策和处理。本文作者平日在学校里与其他人切磋桥牌时发现了扑克牌识别在桥牌活动中的应用价值，从而开发了一个基于 YOLO 进行扑克牌识别的程序并撰写了本文。

本文介绍了基于 YOLOv8 模型以及公开数据集[1] 的同时集成了一个基于 PySide6 的操作界面的项目，初步实现了识别图像中扑克牌的功能。同时本项目仍然有需要改进的地方，如对于个别牌较低的识别精确率以及对于桥牌叫牌方面识别的缺失等。

本文的主要贡献在于系统地介绍了基于 YOLOv8 的扑克牌识别软件的设计与实现，包括系统的整体架构、关键技术的选择与优化、以及在实际应用中的性能表现。此外，本文还分享了一系列基于实际项目经验的优化建议和最佳实践，为同领域的研究人员和工程师提供了宝贵的参考。

II. 背景

A. YOLOv8

YOLOv8[2] 是 YOLO 系列实时物体检测器的最新迭代产品，在精度和速度方面都具有尖端性能。在之前 YOLO 版本的基础上，YOLOv8 引入了新的功能和优化，使其成为广泛应用中各种物体检测任务的理想选择。

YOLOv8 支持多种计算机视觉任务，包括物体检测、实例分割、姿态/关键点检测、定向物体检测和分类。每个模型变体都针对其特定任务进行了优化，并与推理、验证、训练和导出等各种操作模式兼容。

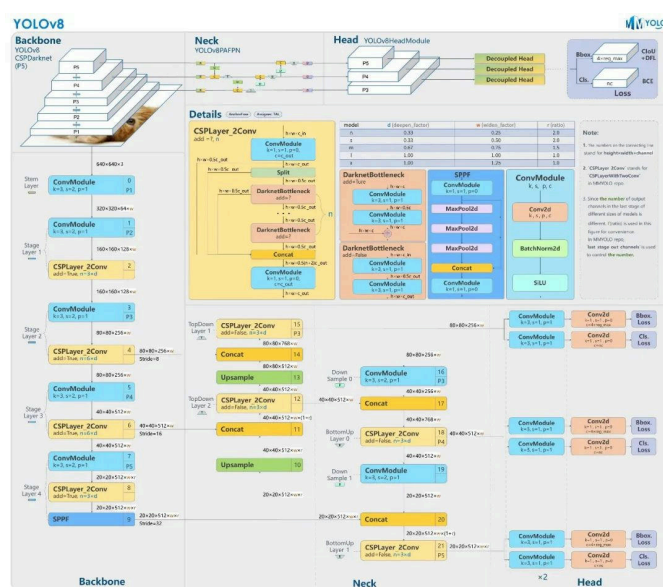


Figure 1: yolov8 架构图

经过若干年的版本迭代，YOLO 已经变的相当复杂，主要包括在 Backbone 中通过卷积和池化网络结构提取特征，在 Neck 部分不断地和之前提取的特征进行融合，Head 部分则是用来进行最终的检测和输出。

YOLO 的 Backbone 部分通过深度和宽度的平衡、残差连接、多尺度特征图、Batch Normalization 和特选的激活函数等技术减少了计算复杂度的同时也提高了模型对特征的识别能力。

Neck 部分，YOLOv8 采用了 SPP。SPP 模块通过使用不同大小的池化窗口，能够有效地捕捉图像中目标的不同尺度信息，这对于处理具有较大尺度变化的目标非常有用。SPP 层可增强模型对于不同尺寸目标的处理能力。

此外，YOLOv8 在模型设计阶段引入了 AutoML 技术。这一策略通过自动化的网络架构搜索来优化模型架构。这一进程利用机器学习算法来进行极其广泛的网络结构搜索，从而找到在特定任务上性能最优的模型架构。

YOLO 的核心思想在于将整张图作为网络的输入，直接在输出层回归 Bounding box 的位置以及 bounding box 所属的类别。

输入图像后，YOLO 的 CNN 网络会将图像划分为 $S \times S$ 的网格。对于每个网格，其会预测 B 个边框（每个边框具有目标的置信度以及每个边框在多个类别上的预测概率），最终可以根据设置的置信度来得出相应的预测结果。

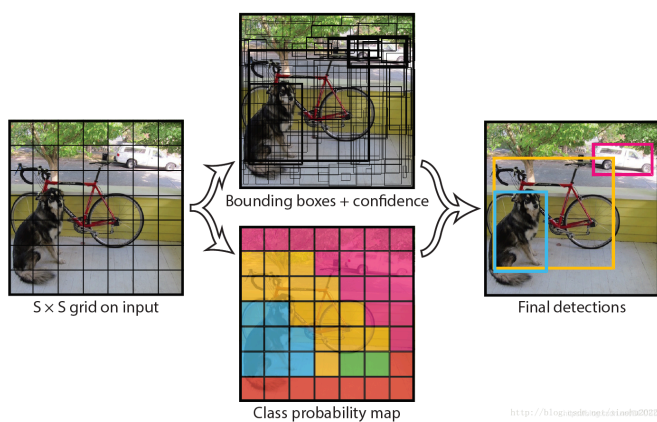


Figure 2: yolo 算法示例

但同时, YOLO 对相互靠的较近的物体以及较小的群体预测表现较差, 这是其的网格预测方式所导致的。

B. 数据集

本项目使用的数据集来自 Playing Cards Computer Vision Project 项目[1]。它包含了 24240 张图片, 分布在训练集、验证集和测试集中, 分别包含 21210 张、2020 张和 1010 张图片。

可以看到在数据集的构成上, 扑克牌的花色及数字的分布是较为均衡的, 这样可以避免模型在训练过程中偏向某个特定花色或数字。

而且, 可以看到图片的背景是多种多样的, 不同的背景图片可以帮助模型适应不同的环境, 从而专注于扑克牌信息的获取。

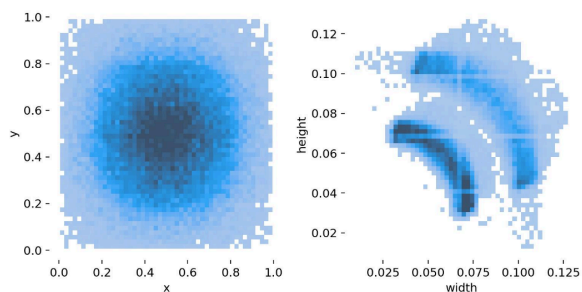


Figure 3: 图像尺寸分布图

如上图所示, 大多数扑克牌在图像中的高宽都是比较小的, 而且扑克牌的位置也是集中于图像中心的。未来可以在这一方向对数据集或是算法进行相应的优化。

C. PySide6

PySide6 是一个 Python 绑定的 Qt 框架, 用于创建跨平台的桌面应用程序。

PySide6 基于 Qt6 库, 允许开发人员使用 Python 语言构建功能丰富的用户界面。它包括了广泛的 UI 控件, 如按钮、文本框、表格、树视图、图表和媒体播放器等, 以及高级功能, 如网络编程、数据库访问、图像处理和多线程支持。

PySide6 与 PyQt 有着类似的 API 和功能集, 但它们是由不同的组织开发和维护的。PySide6 通常被认为是免费和开源的替代品, 而 PyQt 则有商业许可证和 GPL 许可证两种版本。两者事实上没有本质区别。

III. 训练方法

A. 准备工作

首先我们需要使用自定义的函数来统计每个类别 (扑克牌花色及数字) 出现的次数, 帮助模型识别不同类别。

之后我们可以重写构造函数来初始化模型以及其他相关的设置 (使用 yaml 文件配置亦可)。

之后再使用 select_device() 函数来选择模型训练的设备。你也可以在这一步进行一些图像的预处理。在这一步最关键的是要提取识别到的类别名称, 边界框以及置信度。

B. 训练过程

我们需要在程序中定义好数据集的名称以及对应的数据集的路径。之后指定好参数, 使用 train 函数调用加载预训练好的 YOLO 模型即可。

参数	设置	设置
lr0	0.01	学习率
lrf	0.01	学习率的降低速度
batch	8	每次输入模型的样本数
imgsz	640	输入图像的尺寸
worker	1	使用的线程数

Table 1: 重要参数

当然, 你也可以使用默认的参数。

IV. 实验

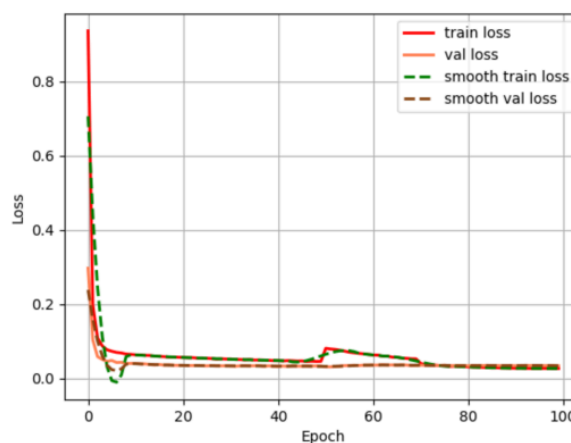


Figure 4: 损失函数图

可以看到损失随着训练论数的增加而显现出显著下降,并且很快便趋于稳定。这表示模型识别物体的能力逐渐增强。

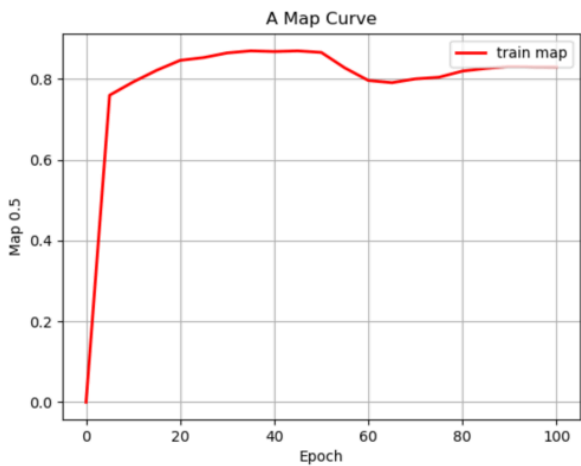


Figure 5: map

同样，map（阈值为 50 时的平均精确度）也相当不错。只是其中的在 50-60epoch 时的降低暂时无法解释。

实验结果图如下：



Figure 6: 实验图 1

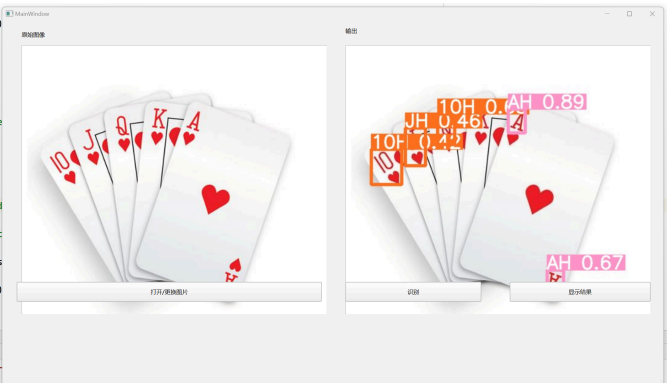


Figure 7: 实验图 2

REFERENCES

[1] "Playing Cards Computer Vision Project : <https://universe.roboflow.com/augmented-startups/playing-cards-ow27d>."

[2] "Ultralytics YOLO Docs : <https://docs.ultralytics.com/>."