基于YOLOv5和U-Net算法的绿植病害识别与检测系统

——概要设计文档

团队设计的绿植病害识别与检测系统概要设计文档部分对该系统从界面概述、功能模块、主要数据结构、主要算法流程和备注进行说明。

一、界面概述

绿植病害识别与检测系统的界面设计部分主要有四部分组成， 系统主界面、图像检测界面、实时监测界面和数据可视化界面，每一部分的功能概述详见使用说明书文档。绿植病害识别与检测系统是基于计算机视觉技术进行开发，旨在自动识别和检测绿植图像中的病害， 并提供可视化展示和结果分析。

二、 功能模块

功能模块主要由图像检测模块、实时监测模块和数据可视化模块组成， 用于对野外拍摄的绿植叶片图像进行检测和摄像头采集的叶片图像实时监测病害。

**2.1** 图像检测模块

图像检测模块由“载入图像 ”“上一页”“下一页”“开始检测 ”

“保存图像”“返回”等功能操作和演示界面组成。

（1）载入图像： 单击“载入图像”按钮用户可以上传绿植图片进行

检测。

1. 载入文件： 载入文件夹的图片进行图像批量检测，再将检测到的数据存储到数据库中，为数据可视化提供数据支持。
2. 上一页、下一页： 单击“上一页”或“下一页”按钮， 用户可

以切换到上一张或下一张图片，并在指定区域进行展示。

1. 开始检测： 单击“开始检测”按钮对当前绿植图像的病害进行 识别和检测，“病害类别”“病害数量”“准确率”在指定区域进行展示。

（5）保存图像： 单击“保存图像”按钮， 用户可以保存标记后的图像到本地电脑指定文件夹内。

**2.2** 实时监测模块

实时监测模块由“开始检测”“停止检测”“拍照”等功能操作和

演示界面组成。

（1）开始检测： 单击“开始检测”按钮， 打开摄像头对采集的叶片 图像进行实时监测， 实现病害位置的标记，缺陷类别和数量的 统计，根据采集图像数量的增加，计算病害监测准确率， 进而 调整模型提高病害检测精度。 打开摄像头，实时获取视频流，

并进行病害识别和检测。

（2）停止检测： 单击“停止检测”按钮， 关闭摄像头，停止实时监

测图像的获取和病害检测。

（3）拍照： 单击“拍照”按钮， 在实时监测图像过程中，用户可以

点击按钮进行拍照，并保存照片。

**2.3** 数据可视化模块

数据可视化界面通过flask框架连接前后端，将图像检测和实时监测时存到数据库中的数据展示在前端界面上，实现数据的可视化显示。

（**1**）后端：在图像检测和实时监测过程中通过python连接数据库，将检测到的病害结果保存到MySQL数据库中。MySQL数据库直接安装到windows镜像里，确保数据库始终处于打开状态，以便可以随时与前端界面连接，完成数据的可视化展示。

（**2**）前端：HTML定义了网页的内容，CSS描述了网页的布局JavaScript控制了网页的行为，三者共同完成一个数据可视化界面。点击界面中央的“开始分析”按钮连接数据库，将数据库信息展示在前端界面，实现数据可视化。在界面的左上角有一个数据切换按钮，目的是控制切换图像检测和实时监测的数据，实现一个界面多个数据的展示，更全面的展示病害的检测结果。

三、 主要数据结构

数据结构部分主要介绍图像的存储路径、当前图像的索引、病害

检测模型、病害标签列表和界面组件部分。

（1）索图像路径列表：保存所有上传的绿植图片的路径。

（2）当前图像索引：记录当前展示的图像在图像路径列表中的索引。 （3）病害检测模型：加载训练好的病害检测模型以进行识别和检测。

（4）病害类型标签列表：保存病害类型的标签，用于结果的展示。

（5）界面组件：包含上传图像按钮、展示图像标签、结果展示标签、

控制按钮等界面组件。

四、 主要算法流程

图像训练模块：

图像训练部分采用 YoLoV5 算法是在 YoLoV4 算法基础上进行改进 得到，YoLoV5可以获得准确、实时、高效的检测结果。相较于 YoLoV4 ， 在保证准确性的前提下，实现了模型轻量化。YoLoV5 网络模型检测 精度高， 推理速度快， 最快检测速度可达 140f/s。YoLoV5 整体结构可

以分为4个模块： 输入端、主干（Backbone）、颈部（Neck）和输

出层。 其中主干网络中， YoLoV5 采用 CSPNet 结构， 从输入图像提取 丰富的图像特征， 解决了其他大型卷积神经网络框架 Backbon 中网络 优化的梯度信息重复问题， 有效减少了模型的参数数量，在保证模型 准确性的同时， 减少了模型尺寸。 YoLoV5 模型的结构设计如图 1 所

示。

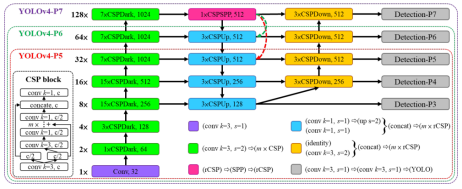


图 1 YoLoV5 模型结构

缺陷区域识别模块：

在此模块中，使用yolov5实现目标检测，对检测物体进行识别；在对yolov5所检测到的缺陷目标后，使用U-net进行图像的分割。U-Net的网络架构独特且简单，具有编码器-解码器结构，能够有效地学习到图像的特征和上下文信息。编码器部分采用了多个卷积层和池化操作，通过逐渐减少图像的尺寸和增加特征通道数，提取图像的高级语义信息。解码器部分则采用了反卷积层和跳跃连接操作，将编码器的特征图与解码器的特征图进行合并，并通过反卷积操作恢复图像尺寸，最终得到分割结果。U-Net的关键特点之一是跳跃连接。跳跃连接将编码器和解码器的特征图进行连接，使得解码器能够利用来自不同层级的低级特征和高级语义信息，从而提高分割的准确性和细节保留能力。总之，U-Net算法有效地结合了编码器和解码器的结构，通过跳跃连接和逐级特征提取，能够在进行语义分割任务时获得较好的性能。U-Net 模型的结构设计如图 2 所示。

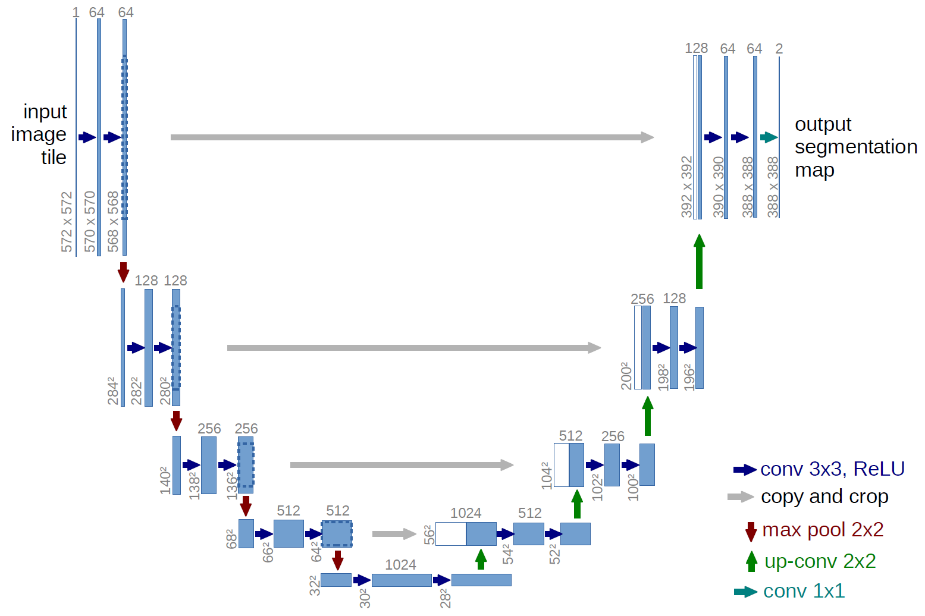


图 2 U-Net 模型结构

图像检测模块：

用户点击上传图像按钮，选择绿植图片进行上传。

用户可以通过上一页和下一页按钮切换图像。

用户点击开始检测按钮，系统加载图像并进行病害识别和检

测，得到结果。

系统展示标记后的图像， 以及病害类型、病害数量和准确率等

结果信息。

用户可以点击保存图像按钮，将标记后的图像保存到本地。

用户可以点击返回按钮，返回上一页或关闭窗口。

实时视频检测模块：

用户点击开始检测按钮，系统打开摄像头，实时获取视频流。

系统加载病害检测模型，对视频流中的每一帧进行病害识别和检测。

系统展示标记后的视频流， 以及病害类型、病害数量和准确率

等结果信息。

用户可以点击停止检测按钮，关闭摄像头，停止病害检测。

在实时视频过程中，用户可以点击拍照按钮， 拍摄当前帧并保

存照片。

五、 备注

本系统使用 PyTorch 和 YOLO 作为病害检测模型，用于识别和检 测绿植图像中的病害。界面设计采用 PyQt5 库， 可以创建美观的图形 用户界面，并实现用户与系统的交互操作。在使用程序时， 请确保摄 像头已连接并正常工作， 为了获得最佳的检测效果， 请确保图像质量 良好且清晰可见。为了系统的稳定性和性能， 建议在具备较高计算能

力的设备和良好的网络环境上运行。