3.6日企业直播答疑：

（这一场直播效果巨差，主播那边网络环境太差了，这是根据服创官方QQ群里的表格整理的）

数据集：

1. 对于缺陷有异议的数据，可以去掉；关于给定数据集不好用的问题，可自己寻找如东北大学数据集使用；
2. 谢韦尔的训练集中字段含义：

ImageId：代表图像，

ClassId：代表缺陷分类ID，有4种缺陷

1: 边缘裂纹（Edge Crack）

​2: 横向裂纹（Longitudinal Crack）

​3: 表面杂质（Surface Inclusion）

​4: 斑块缺陷（Patches）

EncodedPixels：缺陷的位置像素标注

Width 和 Height：钢板的宽度和高度，像素

评价：

1. mAP50 要达到90%以上
2. 企业会从东北大学和谢韦尔的测试集抽取一些典型样本测试，这个的精确度达到90%即可
3. 不强制要求使用国产操作系统，源代码也不强制要求上传，只需要提供说明文档、可执行程序即可；软件的形式没有限制，桌面、web端等都可以
4. 需要标注缺陷位置，多边形还是矩形不限制
5. 性能优先，参数量作为次要考虑，如果算法对参数量优化有优势，也是创新项，要在提交的文档中说明

组内讨论：

1. 数据预处理依然用杨万宏学长说的方式：训练一段时间后来预测图片，预测出来的图片又与源数据集合并，再一起去训练；发现新类别，无监督聚类算法（实现不了也可写进文档当作创新点）；
2. 可以分割和检测多任务进行；
3. Web：系统设计架构要模块化，要求的功能模块有：数据采集模块、数据处理模块、用户界面模块、报表生成模块，API接口模块，便于支持与其他生产管理系统的集成；我们觉得在数据采集方面，还可以增加上传标签的功能，方便系统日后微调模型

3.6~3.10 的log文字分析

1.分割数据处理部分：

原始mask转换为yolo格式标签，转换后数据平均Iou：0.9890。

重新按照四类类别均匀分割了一下数据集，空标签的我暂时剔除出来：

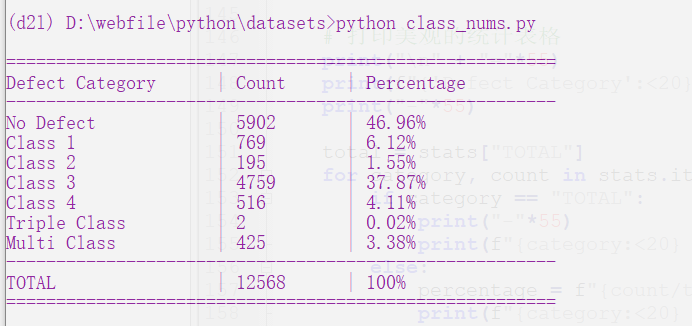
总有效样本（有标签的）: 6666

训练集: 5332 (80.0%)

验证集: 1334 (20.0%)

空标签样本: 5902

所有数据中，各类别的比例如下：

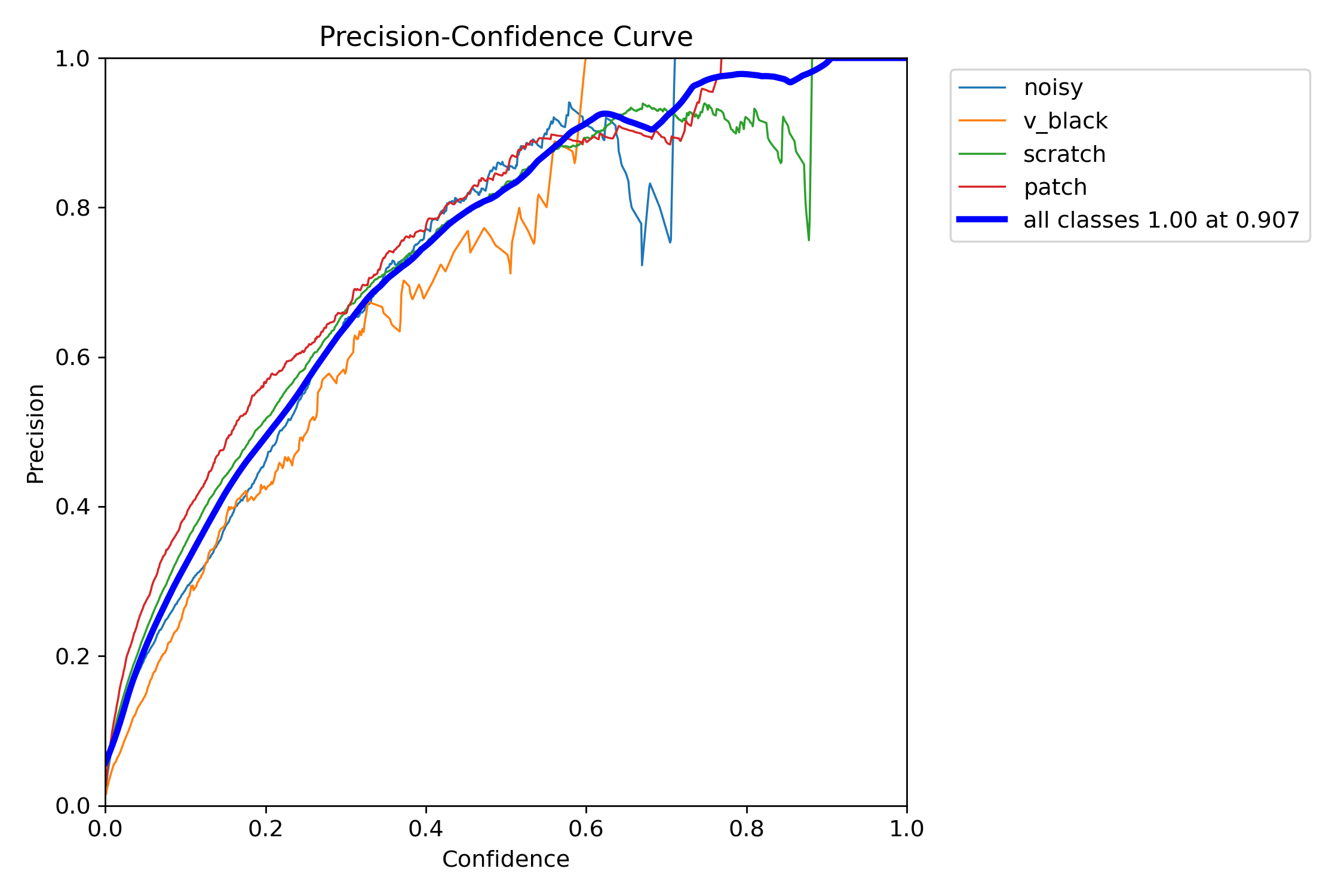
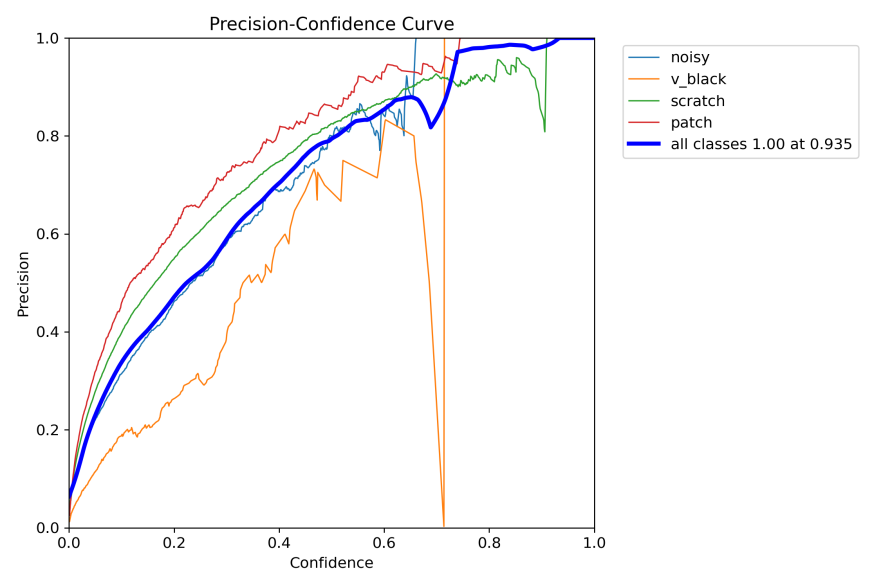


发现第二类的数据类别明显少于其他样本，在训练的过程中该类别的检测有可能会出现波动

2.分割与检测结果分析

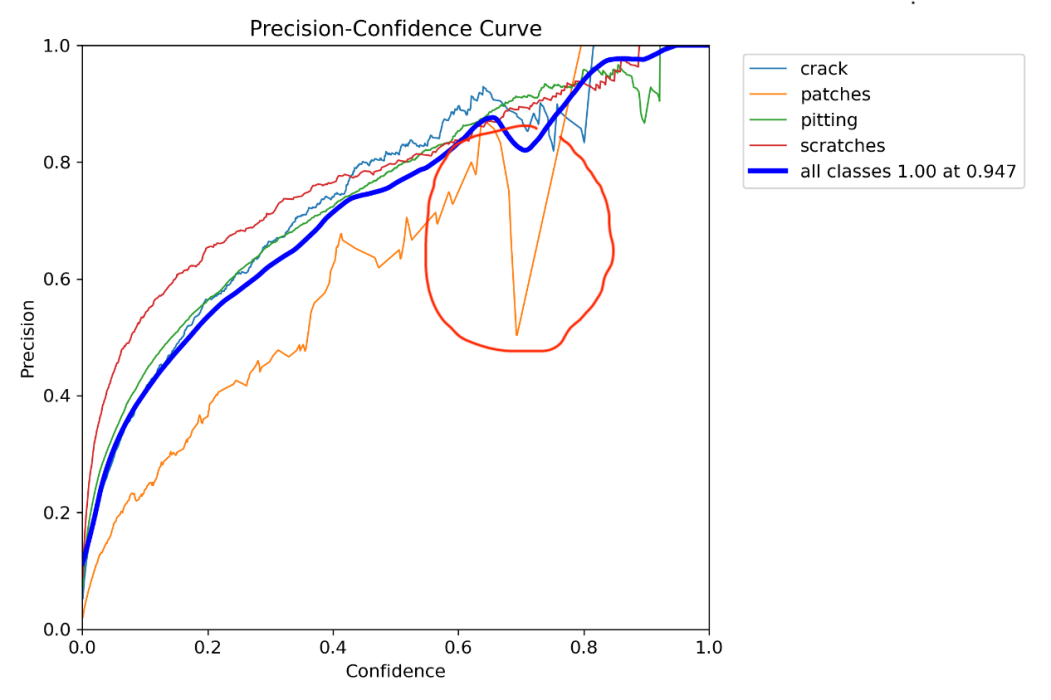
这里将分割模型yolov11n-seg（参数2.9M）与检测模型yolov12n（参数2.6M）进行对比，两者都是yolo系列中参数最小的模型，由于分割模型多了一个分割头，因此参数要略大一点。

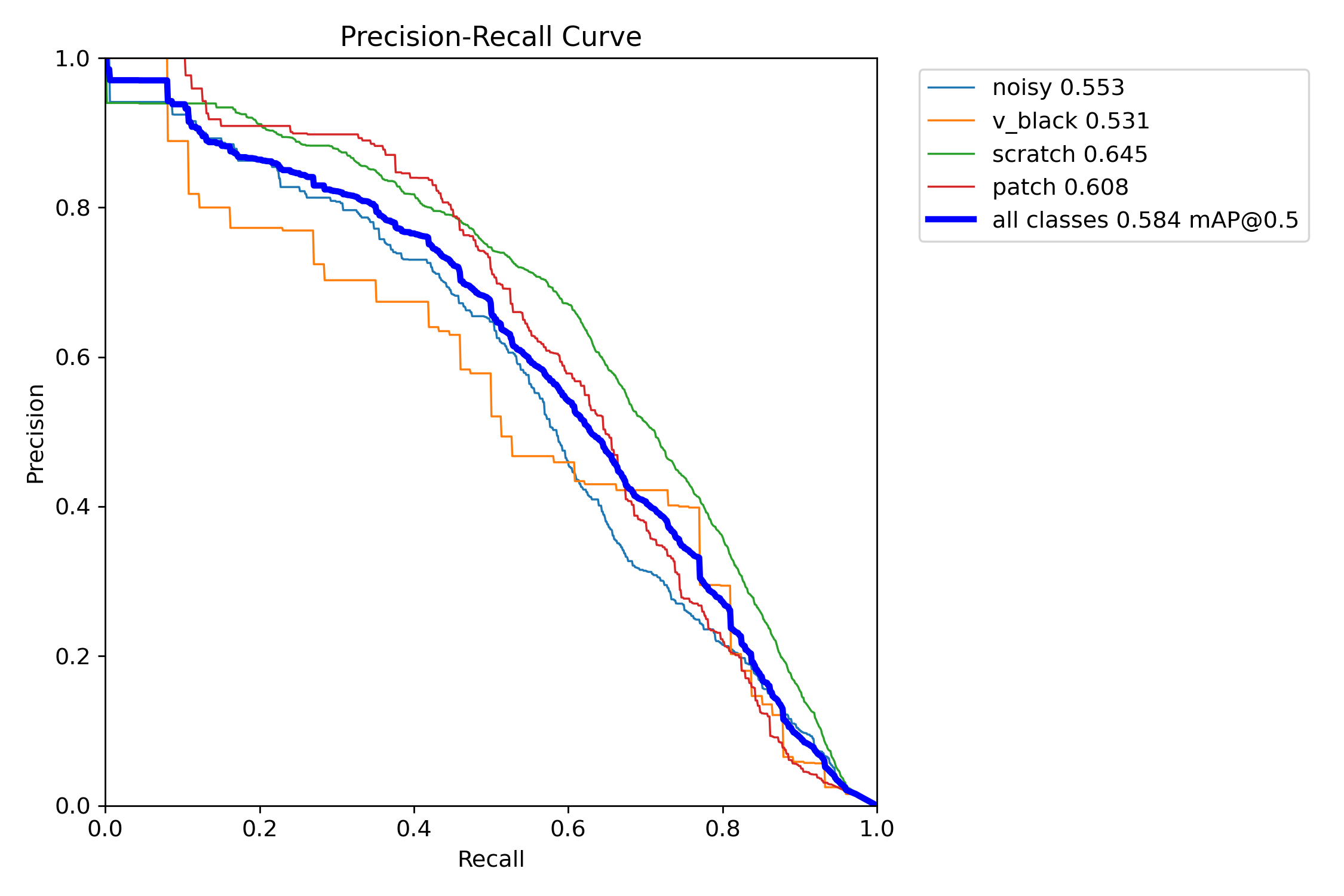
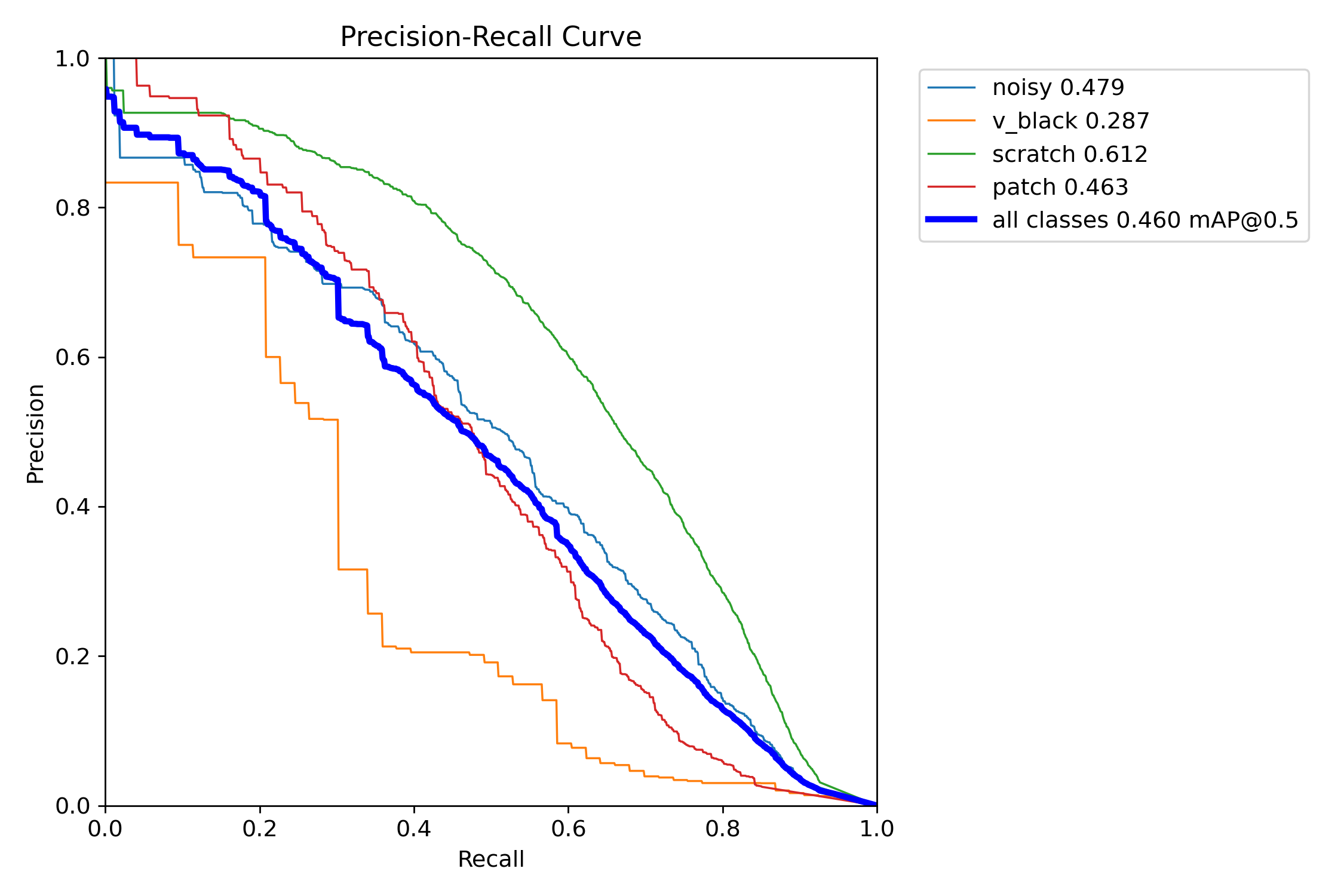
1. BoxP曲线对比，左检测右分割



由于第二类样本数量较少，只占到四类中3.12%，导致在训练过程中波动非常大，但同样的情况在分割模型中并未出现波动如此大的情况。猜测可能是分割掩码的精确轮廓间接提升了检测框的位置回归精度。

同样在其他组员训练的检测模型中也出现了以上情况：





在PR曲线的比较中也容易发现第二类的检测效果在分割模型中表现更好（PR曲线更靠近右上角）

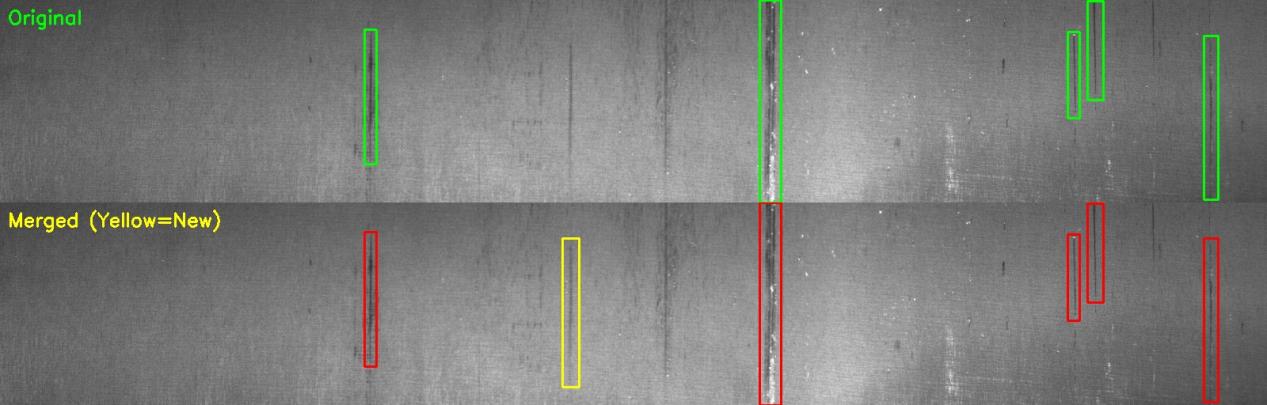
以上原因总结：

YOLO的实例分割模型同时执行检测（检测框）和分割（掩码）两个任务，因此在面对在样本少或标注噪声大的场景下尤为有效，通过任务协同提升了模型的鲁棒性。

1. 解决数据标签没有打全的问题

使用训练100轮后的模型对原始的训练数据进行预测，并将标签合并





若干图片经过预测整理后的标签似乎比原始的标签更好，但仍保持原来的标签不变，这里默认原始已经标注的框更准确，也是为了防止模型噪声传播



1. 目前的问题：

map50达到90还有一段距离；

模型在新合并的数据集上进行训练后如何评估？因为原始的验证集中也存在着标签未打全的；

3.web

1. 1.讨论出功能需求

（1）用户注册与登录：用户可以通过注册账号并登录系统，管理员可以为公司内部员工创建账号，用户输入用户名和密码登录系统。

（2）数据采集：用户可以通过系统导入高质量的钢材表面图片，或直接拍摄材料照片作为检测数据。支持多种数据格式（如JPG、PNG等）的导入。提供拍照功能，确保检测数据的实时性和准确性。

（3）图像预处理：系统对上传的图像进行预处理，以提高检测准确性。（考虑以下可能性：去噪：减少图像中的噪声干扰。增强：提高图像的对比度和清晰度。裁剪：截取图像中的关键区域，减少不必要的计算。）

（4）缺陷检测：系统利用深度学习算法对钢材表面图片进行缺陷检测。自动识别和定位钢材表面的缺陷。

（5）缺陷分类：系统对检测到的缺陷进行分类。根据缺陷的类型（夹杂物、补丁、划痕、 其他缺陷）自动分类并标记。

（6）数据存储与管理：系统支持检测数据的查询、存储、备份与恢复。存储检测结果，包括缺陷类型、位置、数量等信息。支持历史检测记录的查询和导出。提供数据备份与恢复功能，确保数据安全。

（7）报告生成：系统自动生成检测报告。生成详细的检测报告，包括缺陷类型、位置、数量等信息。支持报告导出为多种格式（如PDF、Excel），便于进一步分析和存档。

（8）角色权限管理：管理员可以创建、修改和删除用户角色（如管理员、检测员、质检主管等），并为每个角色分配相应的权限。

1. 2.数据库、数据流图

一、数据库设计说明

1.用户表 (users)：存储用户的基本信息，包括用户名、密码、邮箱等。通过 role\_id 关联到 roles 表，实现角色权限管理。

1. 用户角色权限表 (roles)：存储角色名称和权限信息。权限信息以 JSON 格式存储，便于灵活扩展。
2. 检测数据表 (inspection\_data)：存储用户上传的检测数据，包括图片路径和任务状态。通过 user\_id 关联到 users 表，记录上传数据的用户。
3. 检测结果表 (inspection\_results)：存储检测任务的详细结果，包括缺陷类型、位置、严重程度和报告路径。通过 data\_id 关联到 inspection\_data 表，记录检测结果对应的任务。

二、数据库表设计（可能还要改一改）

1. 用户表 (users)

存储系统用户的基本信息和角色权限。

字段名 数据类型 说明

user\_id INT (主键) 用户ID，自增

username VARCHAR(50) 用户名

password\_hash VARCHAR(255) 密码哈希值

email VARCHAR(100) 用户邮箱

role\_id INT (外键) 关联的用户角色ID

created\_at DATETIME 用户创建时间

updated\_at DATETIME 用户信息更新时间

1. 用户角色权限表 (roles)

存储用户角色及其权限信息。

字段名 数据类型 说明

role\_id INT (主键) 角色ID，自增

role\_name VARCHAR(50) 角色名称（如管理员、检测员、质检主管）

permissions TEXT 角色权限（JSON格式存储）

created\_at DATETIME 角色创建时间

updated\_at DATETIME 角色信息更新时间

3. 检测数据表 (inspection\_data)

存储用户上传的检测数据（如图片、任务信息）。

字段名 数据类型 说明

data\_id INT (主键) 检测数据ID，自增

user\_id INT (外键) 上传数据的用户ID

image\_path VARCHAR(255) 检测图片存储路径

status ENUM('pending', 'completed', 'failed') 检测任务状态

created\_at DATETIME 数据上传时间

updated\_at DATETIME 数据更新时间

4. 检测结果表 (inspection\_results)

存储检测任务的详细结果（如缺陷信息、报告路径）。

字段名 数据类型 说明

result\_id INT (主键) 检测结果ID，自增

data\_id INT (外键) 关联的检测数据ID

defect\_type VARCHAR(50) 缺陷类型（如划痕、裂纹等）

position\_x FLOAT 缺陷在图片中的X坐标

position\_y FLOAT 缺陷在图片中的Y坐标

severity ENUM('low', 'medium', 'high') 缺陷严重程度

report\_path VARCHAR(255) 检测报告存储路径

created\_at DATETIME 检测结果生成时间

顶层数据流图

1流图

1.1流图

1.2流图

1.3流图

1.4流图

系统流程图和功能结构图还在绘制中.......