**YOLOv8犬种类别检测模型训练与验证配置及过程**

**程序配置运行说明**

**文本

AI 生成的内容可能不正确。**

**1. 环境准备**

**在运行程序之前，请确保以下环境已正确配置：**

* **操作系统：Windows、Linux 或 macOS（推荐使用 Windows，因为代码中包含 PyQt5 的 GUI 界面）。**
* **Python 版本：Python 3.8 或更高版本。**
* **依赖库：需要安装项目所需的 Python 第三方库。**
* **数据库：MySQL 数据库服务已安装并运行。**
* **GPU 支持（可选）：如果需要加速训练或推理，建议安装 CUDA 和 cuDNN，并确保 GPU 驱动程序正常工作。**

**2. 安装依赖库**

**在终端或命令行中运行以下命令，安装项目所需的依赖库：**

**pip install ultralytics pymysql PyQt5 requests torch torchvision**

**如果您使用的是 GPU 环境，可以安装支持 CUDA 的 PyTorch：**

**pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124**

**3. 下载预训练模型**

**运行 download\_models.py 脚本以下载所需的预训练模型文件：**

**python download\_models.py**

**该脚本会将模型文件下载到 ./pre\_trained\_model 目录中。**

**4. 准备数据集**

**1. 标注文件转换**

**将 PASCAL VOC 格式的 XML 文件转换为 YOLO 格式。运行 voc\_to\_yolo.py 脚本完成转换：**

**python voc\_to\_yolo.py**

**确保输入目录 (./VOC\_Dataset) 包含 XML 文件，输出目录 (./YOLO\_Dataset) 将生成对应的 YOLO 格式标签文件。**

**2. 数据集配置**

**编辑 data.yaml 文件，确保路径指向正确的训练集、验证集和测试集目录。例如：**

**train: ./YOLO\_Dataset/Train\_YOLO/images**

**val: ./YOLO\_Dataset/Validation\_YOLO/images**

**test: ./YOLO\_Dataset/Test\_YOLO/images**

**nc: 23 # 类别数量**

**names: ['husky', 'beagle', 'chihuahua', ...] # 类别名称**

**5. 训练模型**

**运行 training\_model.py 脚本以训练模型：**

**python training\_model.py**

**如果使用 GPU，请将 device='cpu' 修改为 device='cuda'。训练完成后，模型权重将保存到 ./trained\_models 目录中。**

**6. 配置 MySQL 数据库**

**1. 创建数据库**

**使用 MySQL 客户端工具（如 Navicat 或 MySQL Workbench），执行 dog\_db.sql 文件以创建数据库和表：**

**SOURCE G:/Couresware/DeepLearning\_project/YOLOv8 \_Dogs\_Classification\_Project/YOLOv8 \_Dogs\_classification/dog\_db.sql;**

**2. 修改数据库连接信息**

**打开 dogs\_windows.py 文件，检查 get\_db\_connection 函数中的数据库连接参数，确保与本地 MySQL 配置一致：**

**pymysql.connect(**

**host='localhost',**

**user='root',**

**password='123456',**

**database='dog\_db',**

**charset='utf8mb4',**

**cursorclass=pymysql.cursors.DictCursor**

**)**

**7. 启动 GUI 应用程序**

**运行 dogs\_windows.py 脚本以启动狗品种识别系统的图形界面：**

**python dogs\_windows.py**

**程序启动后，您可以上传图片进行检测，查看识别结果、历史记录等。**

**8. 测试和验证**

**1. 测试模型**

**在训练完成后，使用 model.val() 和 model.predict() 方法对模型进行验证和测试。示例代码已在 training\_model.py 中提供。**

**2. 导出模型**

**如果需要将模型部署到其他平台，可以导出为 ONNX 或 PyTorch 格式：**

**model.export(format='onnx', file='./exported\_models/dog\_classifier.onnx')**

**model.export(format='pt', file='./exported\_models/dog\_classifier.pt')**