**文档编号：*<项目名称>* – SRS – *<\*\*.\*\*>***

***无人车平台班排级作战指挥系统-SRS-1.0***

**软件需求规格说明书**

**日期：2025年3月**

**文档变更历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 变更日期 | 变更人员 | 变更内容详情描述 | 变更后的版本号 |
| 1 | 2025/3/26 | 全体 | 撰写了软件需求规格说明书初稿 | V1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1. 引言](#_Toc7204)

[1.1 编写目的](#_Toc10363)

[1.2 读者对象](#_Toc22095)

[1.3 软件项目概述](#_Toc24762)

[1.4 文档概述](#_Toc8309)

[1.5 定义](#_Toc13378)

[1.6 参考资料](#_Toc28689)

[2. 软件的一般性描述](#_Toc18036)

[2.1软件产品与其环境之间的关系](#_Toc6023)

[2.2限制与约束](#_Toc4740)

[2.3假设与前提条件](#_Toc22044)

[3. 软件功能需求描述](#_Toc12315)

[3.1 软件功能概述](#_Toc11386)

[3.2 软件需求的用例模型](#_Toc28214)

[3.3 软件需求的分析模型](#_Toc17062)

[4. 其它软件需求描述](#_Toc32221)

[4.1 性能要求](#_Toc21071)

[4.2 设计约束](#_Toc29604)

[4.3 界面要求](#_Toc24793)

[4.4 进度要求](#_Toc24079)

[4.5 交付要求](#_Toc11933)

[4.6 验收要求](#_Toc24664)

[5. 软件原型](#_Toc11559)

### 1. 引言

#### 1.1 编写目的

#### 1）本文档通过明确功能需求（指挥控制、态势感知、自主协同等）、性能指标（实时响应、通信可靠性）及技术规范（通信协议、数据接口标准），为军事用户、系统分析员及开发团队提供统一的语言框架，确保各方对战场需求的理解一致性。

#### 2）用户在需求分析阶段可通过文档中的作战场景用例（如“多车协同突击”“动态目标分配”），初步评估系统是否满足战术目标，并作为后续验收的客观依据。开发团队以文档中的技术规范（如硬件选型标准、软件架构设计约束）为设计起点，确保系统开发符合战场实际需求（如复杂电磁环境适应性）。维护团队通过文档快速定位功能模块逻辑，并为后续功能迭代（如新增“反无人机防御模块”）提供可追溯的技术支持。

#### 1.2 读者对象

​用户，分析人员，软件设计人员，项目管理人员

#### 1.3 软件项目概述

**项**目名称：无人车平台班排级作战指挥系统

用户单位：陆军作战部队

开发单位：国防科大计算机学院22级软件工程郭晋鹏小组

背景与功能：针对复杂地形（如森林、室内）下的班排级作战需求，集成实时视频传输、敌我识别、目标定位及地形建模功能，提升指挥决策效率与战场感知能力。

#### 文档概述

软件的一般性描述：系统运行环境、限制条件及假设。

功能需求：用例模型及详细功能分析。

其他需求：性能、界面、交付等非功能性需求。

#### 1.5 定义

YOLO：目标检测算法（You Only Look Once）

#### 参考资料

[1]刘申鑫. 基于YOLO-v7的夜间人体识别的算法研究及实现[D]. 合肥大学, 2024. DOI:10.27876/d.cnki.ghfxy.2024.000182.

[2]王伟东. 一种野外单兵非固定视频采集的无线传输系统的总体设计[J]. 卷宗, 2015, 5(3): 310. ISSN:1005-4669.

### 2. 软件的一般性描述

#### 2.1 软件产品与其环境之间的关系

| 系统组成部分 | 外部环境 |
| --- | --- |
| 无人车平台 | 搭载高清摄像头 |
| 地面指挥终端 | 手机 |

#### 限制与约束

硬件限制：摄像头需支持无线连接

时间约束：需在2024年6月前完成原型开发。

技术约束：动态目标识别准确率≥90%。

#### 假设与前提条件

战场通信链路稳定性可满足实时视频传输需求。

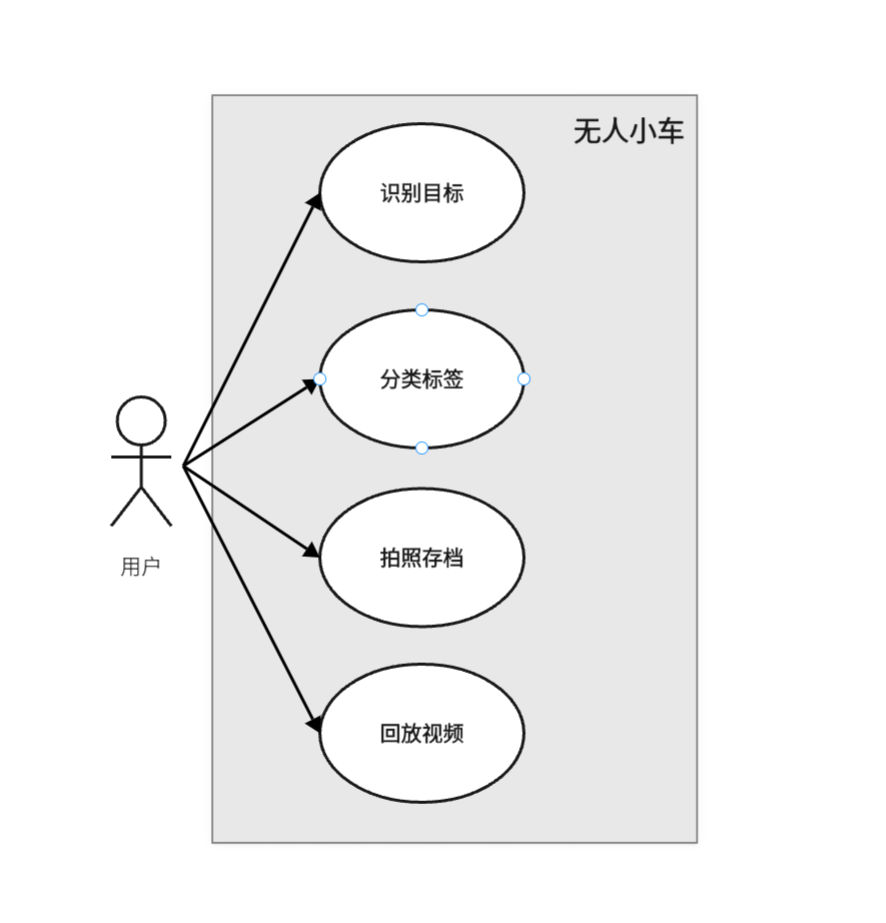
指挥官具备基本战术平板操作能力。

### 3. 软件功能需求描述

#### 3.1 软件功能概述

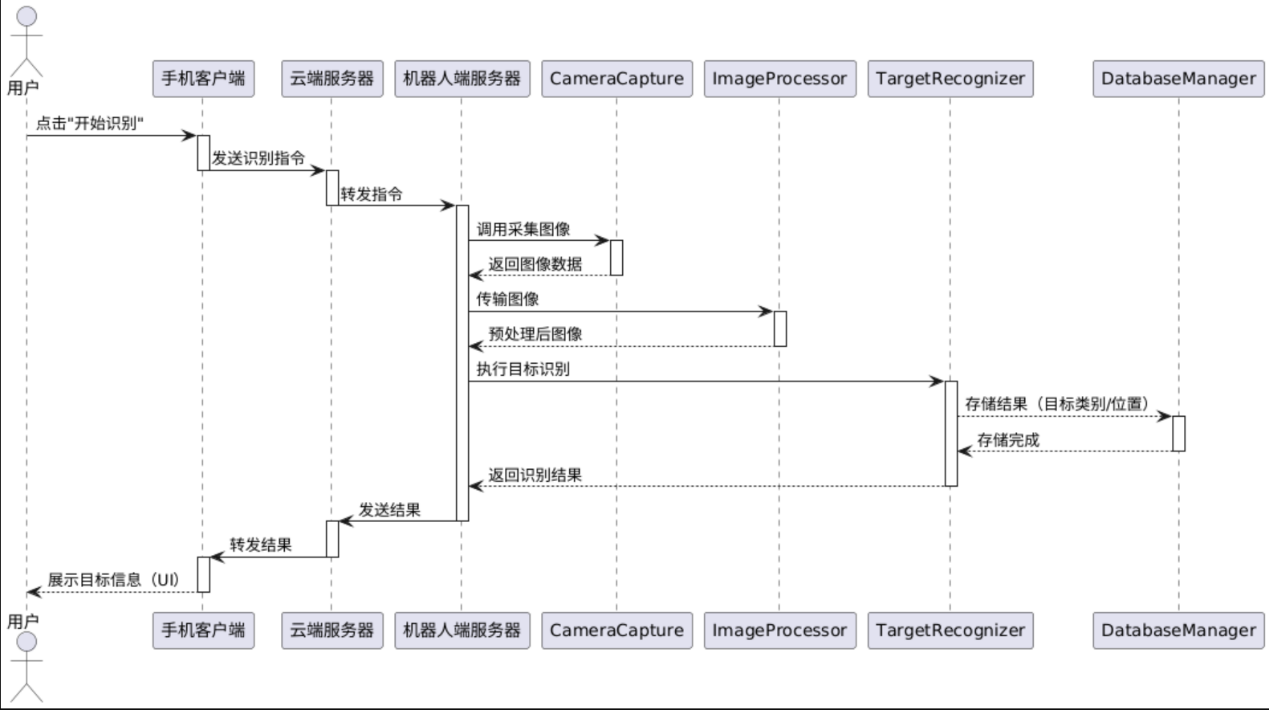
| **功能模块** | **描述** | **优先级** | **使用场景** |
| --- | --- | --- | --- |
| 敌我目标识别 | 自动标注敌我目标 | 高 | 避免误伤 |
| 目标分类 | 为敌方不同目标做不同标记 | 高 | 判别敌方威胁度 |
| 视频回放 | 查看过去的录像 | 中 | 战术规划 |
| 支持拍照存档 | 储存重要时刻图片 | 中 | 战术规划 |

#### 3.2 软件需求的用例模型



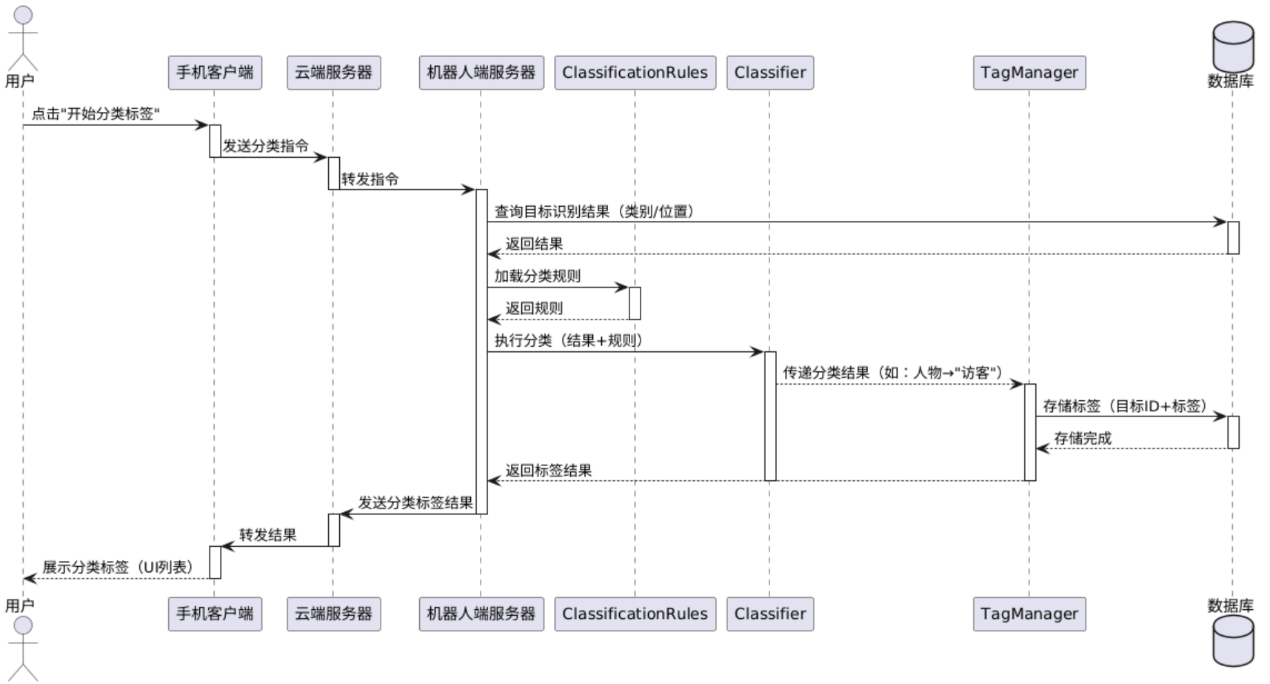
#### 3.3 软件需求的分析模型

### 3.3.1“物体识别”的用例描述



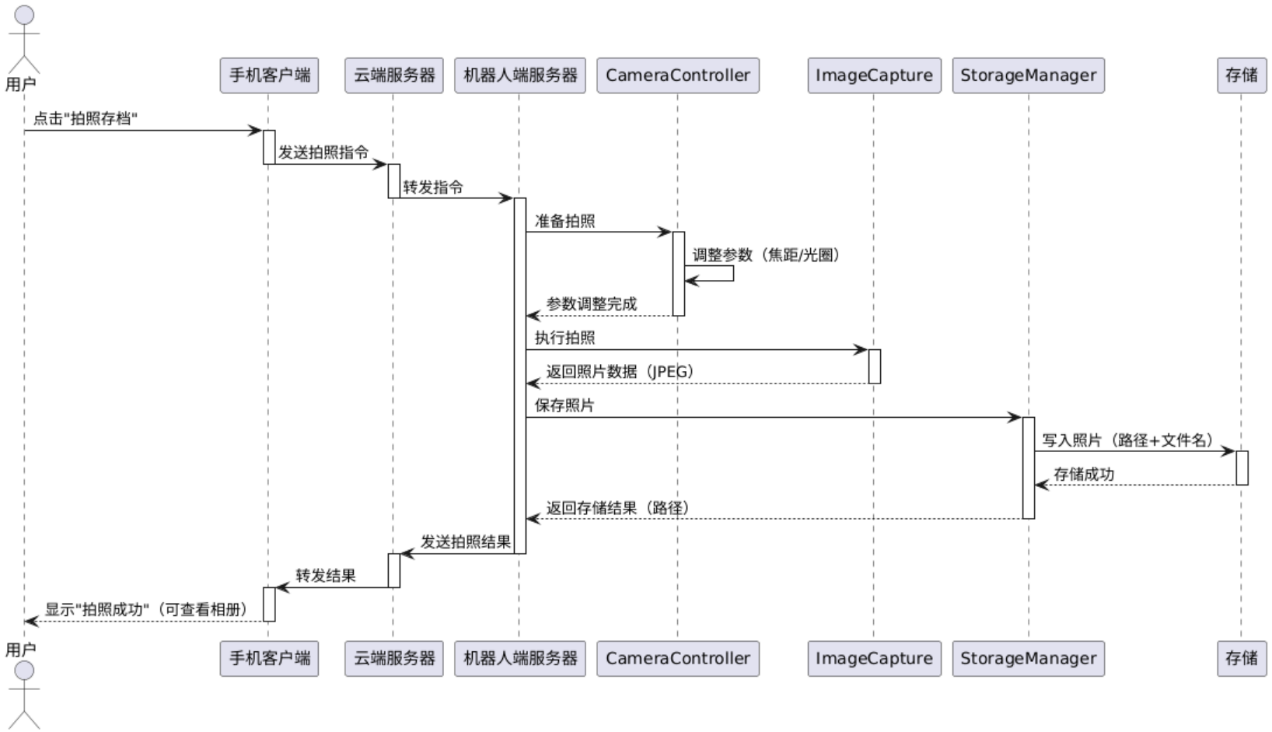
|  |  |
| --- | --- |
| 用例名 | 物体识别 |
| 用例描述 | 识别物体信息 |
| 参与者 | 用户 |
| 过程 | 1. 用户在手机客户端界面点击 “开始识别目标” 按钮，手机客户端向云端服务器发送目标识别指令。 2. 云端服务器接收到指令后，向机器人端服务器转发该指令。 3. 机器人端服务器收到指令后，调用 CameraCapture 类控制摄像头开始采集监控区域的图像。 4. CameraCapture 类获取图像数据后，将其传输给 ImageProcessor 类。 5. ImageProcessor 类对图像进行预处理，如调整图像大小、灰度化、去噪等操作，以提高图像质量，减少干扰因素，然后将处理后的图像数据传递给 TargetRecognizer 类。 6. TargetRecognizer 类运用预训练的目标识别模型，对处理后的图像进行分析和识别。模型会根据图像中的特征，判断是否存在预设的目标类别（如人物、动物、特定物体等），并确定目标在图像中的位置（以坐标和尺寸表示）。 7. 识别完成后， TargetRecognizer 类将识别结果（包括目标类别、位置信息等）发送给 DatabaseManager 类。 8. DatabaseManager 类将识别结果存储到数据库中，以便后续查询和统计分析。 9. 同时， TargetRecognizer 类将识别结果反馈给机器人端服务器。 10. 机器人端服务器将结果发送回云端服务器，云端服务器再将其转发到手机客户端。 11. 手机客户端接收到识别结果后，调用 UserInterface 类在界面上展示识别到的目标信息，如目标的名称、位置标记等，用户可以查看识别结果。 12. 如果用户需要进一步操作，如对特定目标进行跟踪或拍照存档，可以在手机客户端上选择相应的目标，并发送相关指令，系统将根据指令继续执行后续任务。 |

### 3.3.2“标签分类”的用例描述



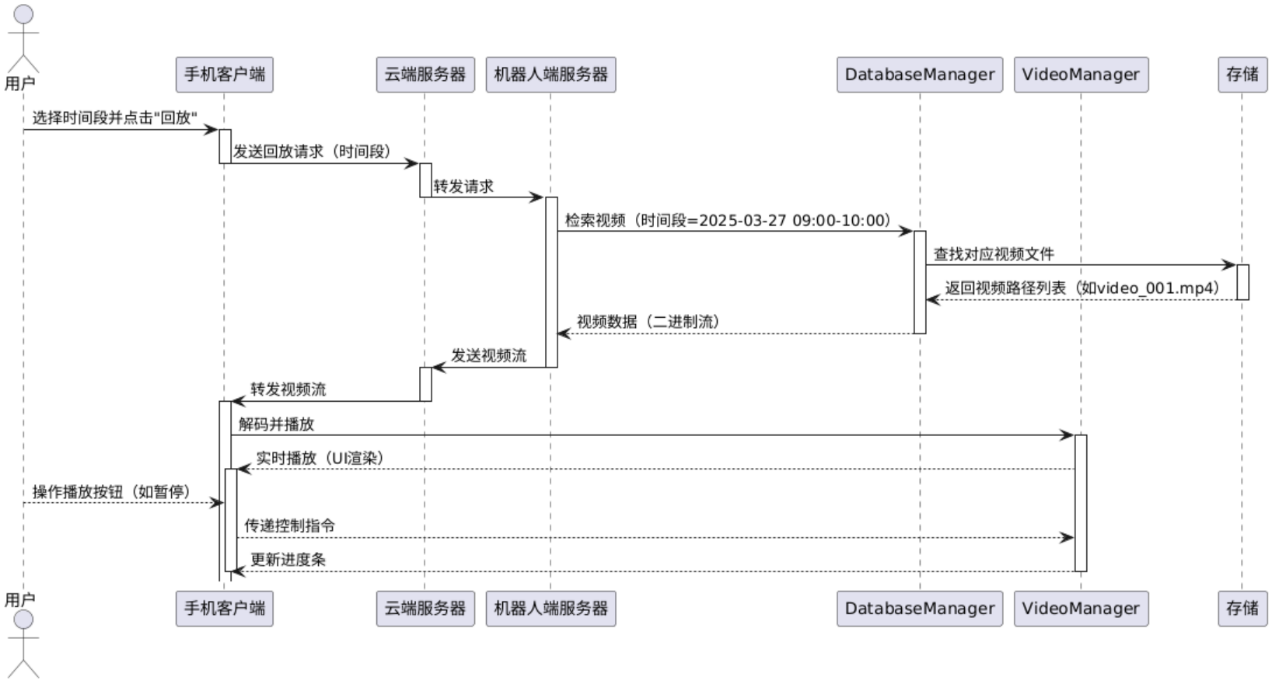
|  |  |
| --- | --- |
| 用例名 | 标签分类 |
| 用例描述 | 识别物体信息后划分标签 |
| 参与者 | 用户、摄像头 |
| 过程 | 1. 用户在手机客户端界面点击 “开始分类标签” 按钮，手机客户端向云端服务器发送分类标签指令。 2. 云端服务器接收到指令后，向机器人端服务器转发该指令。 3. 机器人端服务器收到指令后，从数据库中获取目标识别结果（包括目标类别、位置信息等）。 4. 机器人端服务器调用 ClassificationRules 类，加载预定义的分类规则。 5. 机器人端服务器将目标识别结果和分类规则传递给 Classifier 类。 6. Classifier 类运用分类算法，根据目标的特征和分类规则，对目标进行分类。 7. 分类完成后， Classifier 类将分类结果传递给 TagManager 类。 8. TagManager 类根据分类结果，为每个目标添加相应的标签。 9. TagManager 类将带有标签的目标信息存储到数据库中。 10. TagManager 类将分类和标签结果反馈给机器人端服务器。 11. 机器人端服务器将结果发送回云端服务器，云端服务器再将其转发到手机客户端。 12. 手机客户端接收到分类和标签结果后，调用 UserInterface 类在界面上展示分类后的目标信息和标签，用户可以查看分类和标签结果。 13. 如果用户需要进一步操作，如对特定分类的目标进行拍照存档或回放视频，可以在手机客户端上选择相应的目标，并发送相关指令，系统将根据指令继续执行后续任务。 |

### 3.3.3“拍照存档”的用例描述



|  |  |
| --- | --- |
| 用例名 | 拍照存档 |
| 用例描述 | 对当前场景拍照保存留档 |
| 参与者 | 用户、摄像头、小车 |
| 过程 | 1. 用户在手机客户端界面点击 “拍照存档” 按钮，手机客户端向云端服务器发送拍照指令。 2. 云端服务器接收到指令后，向机器人端服务器转发该指令。 3. 机器人端服务器收到指令后，调用 CameraController 类控制摄像头准备拍照。 4. CameraController 类调整摄像头的参数，如焦距、光圈等，以确保拍摄出清晰的图像。 5. 调整完成后， CameraController 类通知 ImageCapture 类进行拍照操作。 6. ImageCapture 类控制摄像头拍摄照片，并获取图像数据。 7. ImageCapture 类将拍摄的照片数据传递给 StorageManager 类。 8. StorageManager 类根据预设的存储路径和文件命名规则，将照片保存到存储设备中。 9. StorageManager 类将存储结果（如照片的存储路径、文件名等）反馈给机器人端服务器。 10. 机器人端服务器将结果发送回云端服务器，云端服务器再将其转发到手机客户端。 11. 手机客户端接收到存储结果后，调用 UserInterface 类在界面上展示拍照成功的信息，并提供查看已拍摄照片的选项。 12. 如果用户需要查看已拍摄的照片，可以在手机客户端上选择相应的照片查看选项，系统将从存储设备中检索并显示照片。 13. 如果用户需要对照片进行进一步操作，如删除、分享等，可以在手机客户端上选择相应的操作选项，系统将根据指令执行相应的操作。 |

### 3.3.4“回放视频”的用例描述



|  |  |
| --- | --- |
| 用例名 | 回放视频 |
| 用例描述 | 对之前的视频进行回放 |
| 参与者 | 用户、摄像头、小车 |
| 过程 | 1. 用户在手机客户端界面点击 “回放视频” 按钮，并选择要回放的视频时间段，手机客户端向云端服务器发送回放视频指令，包含选定的时间段信息。 2. 云端服务器接收到指令后，向机器人端服务器转发该指令。 3. 机器人端服务器收到指令后，调用 DatabaseManager 类，根据用户选择的时间段从数据库中检索相应的视频文件。 4. DatabaseManager 类在存储设备中查找匹配的视频文件，并将视频文件的数据反馈给机器人端服务器。 5. 机器人端服务器将视频文件数据发送回云端服务器，云端服务器再将其转发到手机客户端。 6. 手机客户端接收到视频文件数据后，调用 VideoManager 类对视频数据进行解码和处理。 7. VideoManager 类根据用户界面的控制指令（如播放、暂停、快进等），控制视频的播放。 8. UserInterface 类在手机屏幕上显示视频播放界面，用户可以通过界面上的控制按钮操作视频回放。 9. 在视频播放过程中， VideoManager 类实时更新视频播放进度，并将播放状态信息反馈给 UserInterface 类，以便在界面上显示。 10. 如果用户需要调整播放位置或速度，可以在手机客户端上操作相应的控制按钮，系统将根据指令调整视频播放。 11. 视频播放结束后， VideoManager 类释放资源，用户可以选择重新播放、保存当前播放位置或其他相关操作。 |

### 4. 其他软件需求描述

#### 4.1 性能要求

​**视**频延迟**：**≤10秒（极端网络条件）。

识别准确率：≥90%（静态目标），≥85%（动态目标）。

#### 设计约束

**硬件要求：摄像头支持无线连接。**

时间约束:必须要在2025年6月初完成这项开发工作的主要任务

技术约束:精确标记分类敌方目标是一项复杂的工作,我们必须高效利用现有的高质量开源代码去合理重组改造,圆满完成我们的工作

#### 界面要求

**跨平台兼容：适配Android、iOS及Windows系统。**

**操作便捷性：一键识别标注视频视图。**

#### 4.4 进度要求

**2025年4月：完成核心算法开发**

**2025年5月：原型系统联调**

#### 交付要求

**交付物：**手机客户端软件

软件设计规格说明书的电子文档

使用说明书的电子文档和纸质文档

#### 4.6 验收要求

**通过实战模拟测试，满足90%功能覆盖率。**

### 5. 软件原型

# *<提供软件原型，并做必要的解释和说明>*