

基于人工智能的飞行试验监控画面 前面板生成算法设计

张英春, 张 为

(中国飞行试验研究院 测试所, 西安 710089)

摘要: 飞行试验监控画面前面板是遥测数据可视化的核心载体, 针对当前采用“示意图绘制—人工解析—手动编码”的传统开发模式, 开发周期长、重复性工作多的问题, 设计了一种基于人工智能的监控画面前面板生成算法, 首先结合跨平台计算机视觉库 (OpenCV) 轮廓提取与光学字符识别 (OCR) 技术, 自动解析监控画面示意图中的控件类型、位置及参数含义, 生成结构化信息; 再通过适配的提示词设计, 引导大语言模型生成 Python 脚本, 用于生成无需修改的 UI 界面程序; 最后构建原型系统并选取代表性画面验证, 结果表明, 该算法能精准提取示意图信息, 生成界面无需修改即可使用, 单画面开发周期从半天缩短至分钟级, 有效提升了监控画面开发效率。

关键词: 监控面板; 画面解析; OpenCV; OCR; 代码生成

Design of Front Panel Generation Algorithm for Flight Test Monitoring Screen Based on AI

ZHANG Yingchun, ZHANG Wei

(Institute of Flight Test, China Flight Test Establishment, Xi'an 710089, China)

Abstract: The front panel of the flight test monitoring screen is the core carrier for the visualization of telemetry data. The current traditional development mode of “diagram drawing—manual analysis—manual coding” has issues such as long development cycles and a lot of repetitive work. Therefore, an AI-based algorithm is designed to generate the front panel of monitoring screen. Firstly, by combining open source computer vision library (OpenCV) contour extraction and optical character recognition (OCR) techniques, it is possible to automatically analyze the types, positions, and parameter meanings of the control in the monitoring screen diagram, thereby generating its structured information. Then, by designing an appropriate prompt, a large language model is guided to achieve Python scripts for generating UI interface programs that do not require modification. Finally, a prototype system is built and representative screens are selected for verification. The results show that the algorithm can accurately extract the information from the diagram, and the generated interface can be used without modification. The development cycle for a single screen is shortened from half a day to minute-level, effectively increasing the development efficiency of monitoring screen.

Keywords: monitoring panel; image analysis; OpenCV; OCR; code generation

0 引言

飞行试验是航空领域验证飞行器性能、保障飞行安全的关键环节^[1]。在试验过程中, 遥测数据处理系统实时监控试验机状态, 快速进行大量数据接收、解析、处理, 并通过可视化监控软件将关键信息呈现给监控人员。这些可视化数据是工程师们进行性能评估、诊断故障的重要依据, 其展示效果直接影响飞行试验的顺利开展及效率^[2-3]。

遥测数据处理系统作为飞行试验地面监控中心的核心软件, 其图形化用户界面 (监控画面前面板) 是监控人员了解试验机状态的唯一途径, 设计质量和开发效率对整个飞行试验的效率具有重要影响。目前, 专业监控画面开发普遍采用“科目驱动”的模式, 具体流程可分为 3 个阶段: 首先, 课题团队根据特定试验科目监控需求, 使用 Visio 等绘图工具画出监控画面的布局示意图; 随后, 监控主管人工解析示意图需求, 逐一识别每个控件的具体类型 (指示灯、数值显示框、时间历程曲

收稿日期:2025-10-31; 修回日期:2025-12-18。

作者简介:张英春(2000-),女,硕士,助理工程师。

引用格式:张英春,张 为. 基于人工智能的飞行试验监控画面前面板生成算法设计[J]. 计算机测量与控制, 2026, 34(5):292-298.

线等)及布局;最后,监控主管基于Qt、C#等开发语言编写代码,完成监控画面前面板开发及数据绑定,实现遥测数据可视化。

随着航空装备研制任务的不断增加,近年来试飞监控任务架次持续增加,监控画面开发需求显著增长,传统的开发模式局限性日益凸显,主要表现为开发周期长、维护成本高。据实际工程统计,每架试验机针对不同科目需开发二十余幅专用监控画面,单个画面控件数少则20余个,多则超过400个,且受科目需求差异的影响,各画面控件布局均不相同。现有模式下,从接收监控画面示意图到前面板开发完成通常需要一周以上时间。更重要的是,这种模式存在大量重复性工作,不仅占用了监控主管大量精力,也很难适应试飞任务快速迭代的需求,因此,探索高效的监控画面自动化开发方法成为当前工程实践中的迫切需求。

近年来,人工智能技术在软件工程领域的应用为解决上述问题提供了新思路,特别是计算机视觉技术在图像识别方面的成熟应用^[4-5],以及大语言模型在代码生成及自然语言处理领域的突破进展^[6-7],使从示意图到前端界面的端到端自动化实现成为可能,通过计算机视觉技术自动提取示意图中的控件信息,结合大语言模型的代码生成能力,有望构建一套完整的监控画面自动化开发流程,大幅缩短开发周期。

然而,现有的工作多集中于通用网页界面或移动应用界面的开发^[8-10],其技术方案难以直接用于飞行试验监控领域。飞行试验监控画面具有控件类型固定,布局逻辑符合监控人员认知等特点,对前面板生成算法提出了更高的领域适配性要求。为此,本文从监控画面前面板开发的实际需求出发,提出了一种基于人工智能的监控画面前面板生成算法,通过融合计算机视觉技术实现监控画面示意图的精准解析,借助提示词工程引导大语言模型生成高适配性的用户界面(UI, user interface)脚本,构建完整的自动化开发流程,为飞行试验监控画面开发提供高效解决方案。

1 需求分析及方法总体框架设计

为了将技术更好应用于工程实践中,切实解决飞行试验监控画面前面板开发的工程痛点,本章首先结合实际开发流程明确算法需实现的核心功能需求,再提出完整的方案总体框架设计。

1.1 需求分析

监控画面示意图来源于《实时/预处理任务书》,由课题团队根据试验科目需求,使用Visio等专业绘图绘制而成,是画面开发流程的输入。示意图中会以简洁的形式标注出各个监控参数需要的控件类型,并清晰规划出各控件的意向布局位置,这种布局方式不仅符合监控

人员的监控习惯,也隐藏了一些遥测参数的内在逻辑。

在实际开发过程中,监控主管通常仅仅会对布局尺寸进行微调处理,以确保最终生成的前面板能完美适配监控终端显示器分辨率,而不会改动控件类型及大致布局方式。这一特点决定了前面板生成算法的核心需求为准确识别并复现示意图中的控件类型、对应参数名称及初始布局,并具备灵活调整画面整体尺寸的能力。结合工程实践经验,具体需求可细化如下。

1) 控件类型识别及生成需求:需准确区分文本框、指示灯、曲线控件等控件类型,其中,文本框用于展示数值型/字符串型参数,需支持参数数值的实时更新显示;指示灯用于展示开关量状态,需通过颜色变化(红色表示异常、绿色表示正常、灰色表示无效等)表示遥测参数状态信息;曲线控件用于时间历程曲线,用于展示参数在飞行试验过程中的动态变化趋势,需支持多参数同框展示及坐标轴自适应调整功能;其他类型控件需预留扩展接口,满足特殊监控需求。

2) 布局调整需求:首先,需支持画面整体尺寸的等比例缩放,确保控件相对位置与示意图保持一致,同时避免控件或参数名称超出显示边界;其次,需对控件位置进行微小调整,满足同种控件水平/竖直对齐的需求。

3) 数据绑定需求:需确保控件与参数名称的精准绑定,每个文本框、指示灯对应唯一参数名,每个曲线控件绑定一个或多个参数名称,为后续数据与控件准确绑定,数据精准呈现提供基础。

4) 风格适配需求:算法生成的UI界面需与传统手工开发的监控画面风格一致,包括背景色、控件颜色、字体等,以符合监控人员的使用习惯。

1.2 总体框架设计

为实现上述功能,本文提出了一种“解析+生成”两步式的飞行试验监控画面前面板生成算法,其总体框架如图1所示。

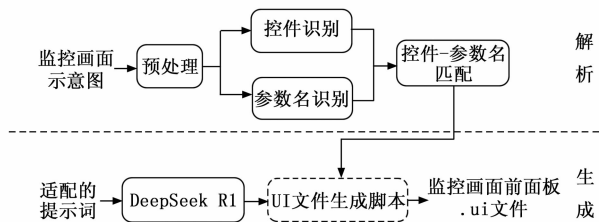


图1 总体框架图

示意图解析模块输入为课题团队绘制的监控画面示意图(JPG或PNG格式),核心功能是通过预处理、控件识别、参数名识别、控件-参数名绑定4个子模块,输出为包含“控件类型-控件ID-参数名称-控件坐标-参数名坐标”的结构化信息(CSV格式)。其中,预处理子模块去除图片背景干扰,提升图像质量;控件识

别子模块从图像中提取有效轮廓并分类,完成控件的准确识别;参数名识别子模块提取控件旁边标注的参数名称文本;控件-参数名绑定子模块通过控件关系关联控件与参数名。

生成模块输入为“解析”阶段输出的结构化信息,输出为 UI 界面程序。核心功能是通过提示词工程引导大语言模型生成 Python 脚本,再由脚本自动将结构化信息转化为可直接运行的 UI 界面程序,同时,该模块采取增量构建策略,首先生成包含基础控件与布局的界面脚本,再增加布局调整与界面美化功能,最终生成尺寸可控、风格统一的监控画面前面板。

两个模块通过结构化 CSV 文件实现信息传递,解析模块输出作为生成模块的输入,生成模块生成的 UI 界面程序在运行过程中出现的问题可通过解析模块进行二次处理,通过大语言模型进行脚本调整,确保生成结果的准确性。

2 监控画面示意图解析方法设计

监控画面解析算法是前面板生成的基础,核心目标为准确识别示意图中控件类型、对应参数名称及初始布局,实现难点在于:①如何准确分割出控件图形与文本标注,避免线条粗细等冗余元素的干扰;②如何关联参数名与对应控件,确保前面板生成后数据与控件绑定的正确性。为了解决以上难点,本文通过预处理、控件识别、参数名识别及绑定 4 个步骤实现。

2.1 示意图预处理

监控画面示意图多为 JPG 或 PNG 格式的彩色图片,受绘图工具默认设置的影响,这些图片往往存在对比度低、线条轮廓不清晰等问题,直接进行轮廓检测容易出现检测错误或者未成功检测的情况。因此,示意图预处理模块通过一系列图像增强与去噪操作,提高图像质量,具体实现流程及逻辑如下:

1) 图片读取。首先使用二进制读取方式获取指定路径下的示意图图片,并解码为跨平台计算机视觉库 (OpenCV, open source computer vision library)^[11] 支持的 BGR 格式,以保留原始彩色图像用于后续处理过程中结果可视化。

2) 灰度转换。由于彩色图像包含 RGB 三个通道的信息,这些信息会增加后续处理的计算量,且对控件轮廓识别没有额外好处。因此,算法将彩色 BGR 图像转化为单通道灰度图,减少数据维度。

3) 对比度增强。示意图中的部分控件颜色在彩色图像中很容易与背景色区分,但两者灰度差异较小,为提高后续处理的准确性,算法采用自适应直方图均衡化算法 (CLAHE, contrast limited adaptive histogram equalization)^[12] 增强图像局部对比度。通过在 5 张示意图中扫

描对比度阈值 (从 2 至 8),比较控件与背景的分度及噪声,该算法将图像划分为 8×8 的网格块,并限制对比度阈值为 2,避免过度增强导致图像噪声放大。

4) 二值化处理。为确保算法能够准确区分出图像中的有效信息 (控件轮廓与参数名称) 和背景,算法将灰度处理后的图片进行二值化处理。经过多次验证,算法采用固定阈值的方式进行图像二值化处理,相较于自适应阈值法,避免了因局部亮度差异导致的轮廓断裂问题。

5) 形态学优化。二值化处理后的图像会存在微小的孔洞或轮廓断裂,算法采用形态学运算^[13] 解决这一问题,保持轮廓边缘的完整性并平滑轮廓。

2.2 基于 OpenCV 轮廓识别的控件识别

控件识别的核心是从二值图像中准确识别并分类出文本框、指示灯、曲线控件 3 种控件类型及坐标信息。该模块通过有效轮廓检测和控件类型分类两个步骤实现,具体流程如下:

1) 有效轮廓检测。首先使用 OpenCV 的 findContours 函数^[14] 从二值图像中检索所有轮廓及坐标信息,并使用 RETR_TREE 模式重新构建轮廓的整个层次结构;其次,由于文字也会被作为轮廓识别出来,但其轮廓围成的面积很小,算法以轮廓面积为依据,去除一部分面积过小的无效轮廓;最后,监控画面示意图中的线条都是有线宽的,其外圈和内圈会被同时识别且代表同一个控件,因此,算法根据所有轮廓的层次架构,仅保留轮廓内边缘作为有效轮廓,在保证准确的同时避免了重复。

2) 控件分类。算法基于空间形状和面积特征进行分类。首先根据轮廓形状,将控件区分为圆形与矩形轮廓,算法认为圆形度 ($4\pi \times \text{面积} / \text{周长}^2$, 理想圆形为 1) 大于 0.8 的轮廓为圆形,对应指示灯控件;认为轮廓宽高比在 0.5~2.0 之间,且轮廓与最小外接矩形的面积比大于 0.8 的轮廓为矩形轮廓,对应文本框与曲线控件。接着,算法利用轮廓面积的统计学规律,将矩形轮廓分为文本框和曲线控件。这是根据一个工程经验:课题团队在绘制监控画面示意图时,通常会使曲线控件的轮廓面积大于文本框。因此,算法先统计所有矩形轮廓的面积,并计算阈值 $[(\text{面积最大值} + \text{面积最小值}) / 8]$,认为面积大于该阈值的控件为曲线控件,展示飞行参数的时间历程曲线,面积小于该阈值的控件为文本框,展示数值型或字符串型飞参结果。剩余无法归类的轮廓为其他类别,后续进行人工处理。

2.3 基于光学字符识别 (OCR, optical character recognition) 的参数名称识别

参数名称文本识别采用 PaddleOCR 引擎^[15] 实现,该引擎对中文文本识别精度高,且支持轻量化部署,适合工程应用场景,需完成 OCR 引擎初始化、文本区域

检测、内容识别及结果解析 4 个步骤, 具体实现如下:

1) OCR 引擎初始化。配置 PaddleOCR 核心参数, 包括启用角度分类以识别旋转文本、设计语言为中英文以识别中文名称及英文 ID、关闭 GPU 加速以适配无 GPU 的工程环境、调整检测阈值以平衡检测精度与召回率。

2) 文本检测。使用 OCR 工具对图像进行文本检测, 得到包含文本区域多边形坐标、识别内容及置信度的结果列表。为保证结果可靠性, 算法对返回结果进行判空操作, 避免因图像中无文本导致的程序异常。

3) 结构解析。算法最后遍历结果列表, 提取每个文本区域的多边形顶点坐标, 并将其左上角坐标、右下角坐标、识别文本内容及置信度组织为字典列表, 为后续控件匹配过程提供结构化文本信息。

2.4 控件一参数名匹配

识别完控件类型及参数名后, 算法需将二者匹配, 以确保解析结果可用, 该部分算法核心逻辑是“参数名与对应控件在示意图中空间位置相近”的工程绘制习惯, 同时参考以下关系特征: ①数量关系。一个文本框只对应一个参数名称, 一个指示灯只对应一个参数名称, 一个曲线控件对应一个或多个参数名称; ②位置关系。指示灯对应参数名称通常在其旁边, 文本框和曲线控件的参数名有可能出现在其内部。具体实现逻辑如下:

1) 文本框匹配。文本框对应的参数名可能位于控件内部或紧邻控件外部。算法首先检查文本框轮廓内部是否包含参数名文本, 若存在, 则直接绑定; 若不存在, 则计算文本框中心与外部各参数名中心的欧氏距离, 筛选距离最近的参数名为该控件对应参数名。

2) 指示灯匹配。指示灯用于展示开关量飞行参数, 其参数名通常标注在控件左边/右边/上边。因此, 算法匹配时计算指示灯中心与各参数名称中心的距离, 匹配距离最近的参数名与控件, 同时排除已匹配的参数名文本。

3) 曲线控件匹配。曲线控件用于展示飞行参数的时间历程曲线, 可能存在一个控件对应多个参数的情况, 且参数名通常标注在控件内部或上方/左方。这一过程中, 算法首先检测控件内部的参数名, 如存在, 直接绑定。对于其他的参数名, 寻找距离最近的控件, 实现绑定。特别注意的是, 文本框的参数名称绑定是以文本框为标准, 每个文本框对应一个参数; 而曲线控件匹配时, 则以参数名为标准, 每个参数名对应一个控件。

4) 结果导出。匹配完成后, 算法将结果整理为两个表格, 一个是匹配结果表, 包含“序号、参数名、控件 ID、控件类型、参数名位置、控件位置、参数名置信度”等字段; 另一个是未匹配控件表, 记录未找到对应参数名的控件信息, 两个表格均以 CSV 格式导出, 方便后续人工修正与代码生成算法调用。

3 监控画面前面板生成

示意图解析算法已将监控画面前面板需求解析为结构化文档, 生成模块则需要将该文档转化为可直接运行的 UI 界面程序。近年来, 大语言模型在代码生成领域展现出了强大的能力, 能够基于文本需求输出语法正确、功能完整的代码片段, 因此本文考虑借助大模型工具实现这一功能。

3.1 生成策略选择

面对监控画面前面板生成这一特定的应用以及结构化文档输入、标准化代码输出的明确应用需求, 本文首先考虑使用模型微调的方式, 使大语言模型适配应用场景, 输出符合需求的 UI 界面程序。

然而, 模型微调面临两大难题^[16-17]: ①数据集构建成本极高。根据大语言模型微调的行业经验, 模型微调首先需要构造大量的数据集作为训练数据, 数据集里至少需 2 000 条数据。在监控画面前面板生成这一应用中, 不同机型、不同课题的监控画面布局需求差异较大, 以每个开发过的监控画面作为一个数据, 整理结构化文档和对应 UI 代码工作量巨大, 以每幅画面 0.5 小时计算, 2 000 条数据需投入 1 000 小时的人工成本, 且难以覆盖所有场景; ②计算资源与时间成本过高。模型微调过程依赖多块高性能 GPU, 单轮微调耗时通常超过 72 小时, 且需要多次调整超参数以优化效果, 这些计算资源和时间成本在实际工程项目中是难以承受的负担。

因此, 本文拟采用提示词工程方法实现这一功能。与模型微调相比, 提示词工程具有独特优势, 无需修改模型内部参数^[18-19], 且灵活性和适应性都比较高, 能够快速根据需求变化调整模型输出。

然而, 在采用提示词工程方法时, 发现了如图 2 所示问题, 大语言模型仅仅给予方法参考, 而非完全可用语句。

这种情况通常出现在控件数比较多的情况下, 这是由于大语言模型在处理长文本时存在 Token 数量限制, 当结构化文档中控件数量较多时, 需要的 Token 资源过多, 导致大语言模型无法生成完整界面描述文件, 只能返回简化的方法说明。

为解决这一问题, 本文提出了基于脚本的间接生成策略, 使用大语言模型生成能够自动创建界面的脚本程序, 而不是直接生成界面代码。这是由于脚本程序的表达效率更高, 相同的功能可以用更少的代码量实现, 从而在 Token 限制内完成包含大量控件的复杂界面的任务。

3.2 提示词工程实现

提示词工程的核心是通过精准描述需求引导大语言模型生成高质量脚本, 本文采用增量构建的策略, 分步



图 2 大语言模型输出示例

引导大语言模型生成脚本，确保每一步输出都清晰可验证。首先，生成界面框架脚本、包括基本控件和基础布局；其次，增加脚本调整监控画面整体尺寸及比例的功能；最后，增加脚本美化界面的功能。

1) 基础需求：请编写一个 Python 脚本，能够将包含 UI 控件信息的 CSV 文件转换为 C++ Qt 的 UI 文件 (XML 格式)。输入文件为 CSV 文件，包含以下列：参数名、控件 ID、控件类型、参数名位置 x_min 、参数名位置 y_min 、参数名位置 x_max 、参数名位置 y_max 、控件位置 x_min 、控件位置 y_min 、控件位置 x_max 、控件位置 y_max ，输出控件名称必须为纯英文 (只包含字母、数字和下划线)。同时，调整曲线对应参数名位置，放置在在曲线控件内部左上角，逐行排列，透明背景。

2) 补充功能需求：在此基础上，请添加尺寸调整功能，将整体界面尺寸固定为 $1\ 920 \times 1\ 080$ ，调整控件尺寸，尽量铺满整个界面。同时，将基本在一行/列的控件对齐。

3) 界面美化需求：请将背景颜色设置为黑色，文本颜色设置为绿色，指示灯为圆形。输出最终完整脚本。

4) 关键需求强化：请确保界面整体尺寸符合要求、控件相对位置不变，所有控件“name”字段均为英文，曲线控件中参数名需为透明背景，不能遮挡曲线。

4 实验结果与分析

本文以不同科目的监控画面为实验对象，验证提出的飞行试验监控画面前面板生成方法的有效性及效率，同时分析该算法在工程应用中的优势与局限性。

4.1 试验环境及数据选取

实验环境为 Intet Core i7-10710U 处理器，16 GB 内存、无 GPU 加速的 Windows 10 系统，软件依赖 Python 3.9，OpenCV 4.5.5，PaddleOCR 2.8.0，大语言模型为腾讯元宝版 DeepSeek^[20] 以及 Qt 5.12.8。

实验数据以 5 个典型飞行试验监控画面为测试样本，所有监控画面均由中国飞行试验研究院相关课题根据实际试飞科目需求绘制而成，从《××飞机××课题实时/预处理任务书》中“监控画面示意图”小节摘取，具体信息如下：

1) 飞行性能操稳画面。如图 3 所示，该示意图由 3 种控件组成：数值型控件 (23 个)、指示灯 (8 个)、时间历程曲线 (20 个控件、58 个参数)，分辨率为 $2\ 749 \times 1\ 587$ 。

2) 起落架画面。数值型控件 (16 个)、指示灯 (15 个)、时间历程曲线 (8 个控件、23 个参数)、特殊控件 (2 个)，分辨率为 $2\ 076 \times 2\ 099$ 。

3) 载荷画面。数值型控件 (18 个)、指示灯 (5 个)、时间历程曲线 (18 个控件、72 个参数)，分辨率为 $1\ 867 \times 2\ 269$ 。

4) 冷却通风画面。数值型控件 (11 个)、时间历程曲线 (7 个控件、25 个参数)，分辨率为 $2\ 223 \times 1\ 457$ 。

5) 动力装置画面。数值型控件 (18 个)、指示灯 (56 个)、时间历程曲线 (20 个控件、20 个参数)、特殊控件 (5 个)，分辨率为 $2\ 076 \times 2\ 099$ 。

4.2 评价指标定义

结合工程需求，从解析、生成两个层面设计如下评价指标。

1) 控件识别准确率：正确识别的控件数量占总控件数量的比例，用于评估示意图解析算法中轮廓识别与类型分类的精度。计算公式为：控件识别准确率 = (正确识别的文本框数 + 正确识别的指示灯 + 正确识别的曲线控件数) / 总控件数 $\times 100\%$ 。

2) 参数名称识别准确率：正确识别的参数名称占总参数名称的比例，计算公式为：参数名称识别准确率 = 正确识别的参数名称数量 / 总参数名称数量 $\times 100\%$ 。

3) 控件-参数名匹配准确率：正确绑定控件与参数名的对数占总匹配对数的比例，计算公式为：匹配准确率 = 正确绑定对数 / 总参数名称数量 $\times 100\%$ 。

4) 界面可用性评价：直接运行生成的 UI 界面程序，而不进行任何更改的，并主观比较其生成结果与监控画面示意图要求效果。

5) 开发效率评价：统计从输入示意图到生成可用 UI 界面的总耗时，与传统手工开发进行对比。

4.3 示意图解析算法实验结果

表 1 展示了算法识别及匹配结果统计。

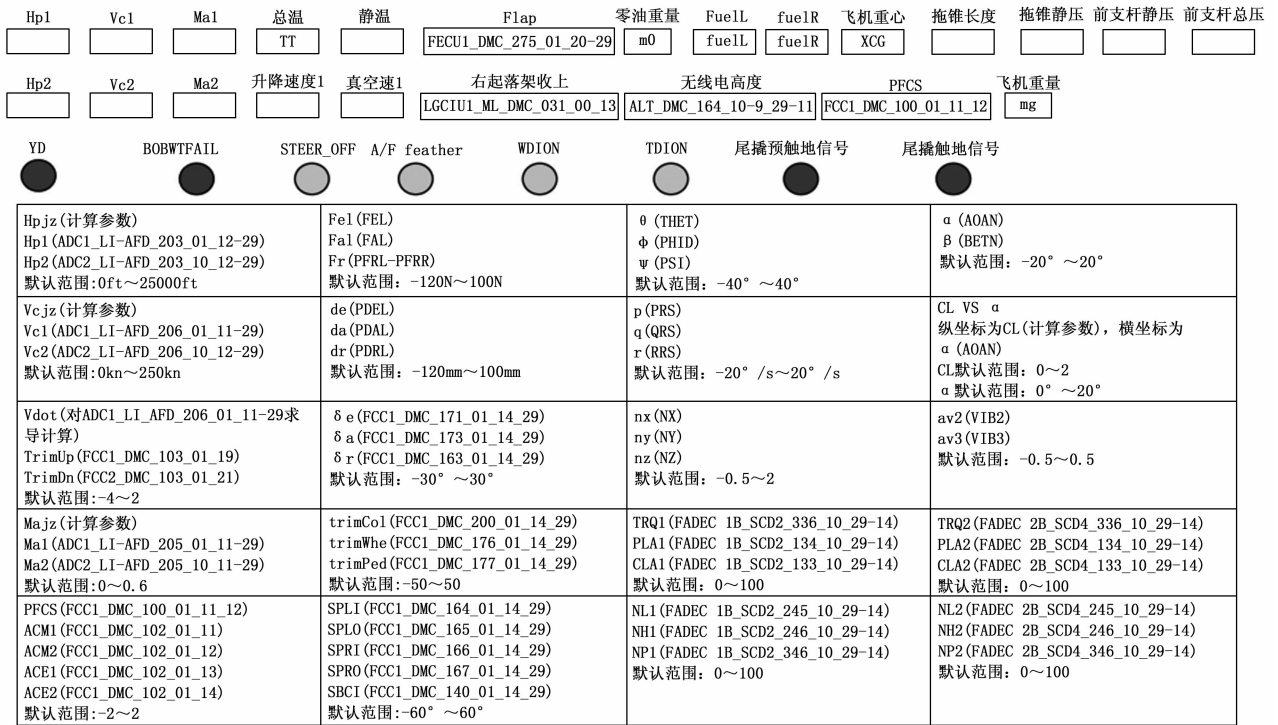


图 3 飞行性能操稳画面示意图

表 1 算法识别及匹配结果

实验样本	控件数	正确识别控件数	控件识别准确率/%	参数名称数	正确识别参数名称数	参数名称识别准确率/%	控件-参数名匹配准确率/%
操稳画面	51	51	100	89	88	98	98
起落架画面	41	39	95	56	54	96	96
载荷画面	41	41	100	95	90	94	94
冷却通风画面	18	18	100	36	36	100	96
动力装置画面	99	98	98	99	96	96	96

从表 1 中可以看出, 算法的总体解析精度较高, 识别并匹配成功的准确率在 95% 以上。未能成功识别的控件均为特殊控件, 由于形状特殊, 超出预设的分类范围; 未能成功识别的参数名均为单个字符 (例如 α), 尺寸过小, 导致识别偏差; 匹配错误多发生在参数名称距离另外控件更近的情况, 欧氏距离计算时存在近邻干扰。

4.4 前面板生成实验结果

1) 界面一致性: 图 4 展示了图 3 所示的监控画面最终生成的前面板。与其余样本一致, 生成界面与示意图均达到“基本一致”及以上等级。

2) 功能可用性: 算法生成的 UI 文件, 直接放置在 Qt 项目中, 替换默认的 mainwindow.ui, 均可无需更改直接运行, 证明了生成结果可用。

3) 开发效率对比:

各样本开发时间统计如表 2 所示, 其中传统手工开

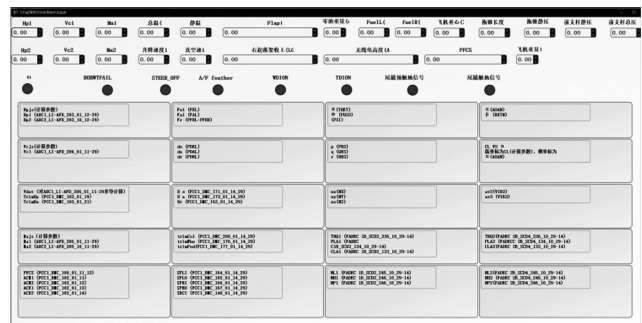


图 4 飞行性能操稳画面生成结果

发时间为各级别监控主管预估时间平均值, 本文算法开发时间包括人工修正时间:

表 2 开发效率对比

实验样本	传统手工开发时间	本文算法开发时间	效率提升倍数
操稳画面	2 小时	3 分钟	40 倍
起落架画面	2.5 小时	10 分钟	15 倍
载荷画面	2 小时	4 分钟	30 倍
冷却通风画面	20 分钟	4 分钟	5 倍
动力装置画面	4 小时	15 分钟	16 倍

可以看出, 本文算法将单画面开发时间从传统的小时级缩短至分钟级, 显著降低了开发时间成本。

5 结束语

本文针对当前飞行试验监控画面开发采用的“示意图绘制—人工解析—手动编码”的传统模式, 存在开发

周期长、重复性工作多的问题,设计了一种基于人工智能的监控画面前面板生成算法,首先结合 OpenCV 轮廓提取与 OCR 技术,自动解析监控画面示意图中的控件类型、位置及参数含义,生成结构化信息;再通过适配的提示词设计,引导大语言模型生成 Python 脚本,用于生成无需修改的 UI 界面程序;最后构建原型系统并选取多幅典型监控画面进行验证,结果表明,该方法能精准提取示意图信息,识别文本框、指示灯和曲线控件及其坐标信息,生成界面无需修改即可使用,单画面开发周期从半天缩短至分钟级,有效提升了监控画面开发效率。

后续可以从如下方向深化技术应用:

1) 算法层面。需突破特殊控件识别瓶颈。当前算法仅支持 3 类基础控件,针对前轮转弯角度仪表盘、滑动条等特殊控件,可继续研究其形态学特征或引入深度学习语义分割模型。

2) 试验样本层面。目前呈现的实验样本均为民机数据,验证了方法的可行性,未来需在专用网络中使用大量多样化、更复杂的监控画面示意图进行测试与优化,进一步提高算法的鲁棒性及泛化能力。

3) 工程落地层面。后续可根据算法原型构建开发一体化平台。将示意图解析、脚本生成、UI 预览等功能集成至同一平台,并提供示意图一键导入和导出功能,同时规定接口规范,以适配多型号试验机多科目需求。

参考文献:

- [1] 刘选民. 实践出真知——飞行试验的重要作用和发展变革 [J]. 国际航空, 2009 (3): 25-28.
- [2] 虞一帆. 试飞遥测数据实时监控系统的研制 [D]. 哈尔滨: 哈尔滨工业大学, 2022.
- [3] 何林杰, 孙毅. 民用飞机飞行试验测试参数管理方法 [J]. 飞机设计, 2021, 41 (3): 63-67.
- [4] 罗明明, 诸峰, 王东升. 基于计算机视觉的船舶图像自动识别方法研究综述 [J]. 软件导刊, 2018, 17 (7): 5-8.
- [5] 谢永杰, 智贺宁. 基于机器视觉的图像识别技术研究综述 [J]. 计算机应用, 2025, 45 (3): 685-696.
- [6] 汪莹, 李骏涛, 娄一翎, 等. 大语言模型辅助软件开发的经验研究 [J]. 计算机应用与软件, 2025, (10): 1-11.
- [7] WANG Y, LE H, GOTMARE A, et al. CodeT5+: open code large language models for code understanding and generation [C] //Singapore: Association for Computational Linguistics, 2023.
- [8] THAKUR S, AHMAD B, PEARCE H, et al. VeriGen: a large language model for verilog code generation [R]. ArXiv, 2023: 2308.00708.
- [9] 韩春燕. 基于 Web 应用界面的代码自动生成软件设计 [J]. 现代电子技术, 2017, 40 (14): 55-57.
- [10] 江一舟, 李斌, 高新闻. 智能仪表交互界面的软件自动生成技术 [J]. 仪表技术, 2009, (5): 15-17.
- [11] 罗一中. 基于深度学习的 OpenCV 图像处理软件设计 [J]. 软件, 2025, 46 (1): 50-52.
- [12] 沈洁, 梅劲松, 沈均贤. 一种改进直方图均衡化的图像增强方法 [J]. 计算机与数字工程, 2024, 52 (10): 3112-3115.
- [13] 宋锐, 杨腾云浩, 李依倚, 等. 一种融合形态学闭运算的双 SVM 卫生陶瓷缺陷检测方法 [J]. 自动化应用, 2021, (6): 14-17.
- [14] 李振杰, 臧春华, 苏宝玉, 等. 基于 OpenCV 图像处理软件的设计 [J]. 物联网技术, 2024, 14 (6): 47-49.
- [15] CUI C, SUN T, LIN M H, et al. PaddleOCR 3.0 技术报告 [R]. ArXiv, 2025: 2507.05595.
- [16] 秦小林, 古徐, 李弟诚, 等. 大语言模型综述与展望 [J]. 计算机应用, 2025, 45 (3): 685-696.
- [17] 吴春志, 赵玉龙, 刘鑫, 等. 大语言模型微调方法研究综述 [J]. 中文信息学报, 2025, 39 (2): 1-26.
- [18] 陈晓阳, 高飞, 韩翔宇, 等. 大模型代码生成技术及航天领域潜在应用 [J]. 航天控制, 2025, 43 (1): 8-16.
- [19] 王东清, 芦飞, 张炳会, 等. 大语言模型中提示词工程综述 [J]. 计算机系统应用, 2025, 34 (1): 1-10.
- [20] AI 赋能智创未来——你的智能秘书助手腾讯元宝 [J]. 秘书之友, 2025, (8): 2.
- [11] 陈勇, 范平清, 袁涛. 多因素改进势场蚁群算法的路径规划 [J]. 云南大学学报: 自然科学版, 2022, 44 (4): 718-728.
- [12] 袁佳乐, 刘蓉, 王闯. 改进人工势场的无人机避障机动策略 [J]. 电光与控制, 2022, 29 (7): 74-80.
- [13] 黄国良, 周毅, 郑坤, 等. 基于改进蚁群算法的全局船舶路径规划方法 [J]. 船海工程, 2023, 52 (2): 97-101.
- [14] 孙宇, 唐炜, 谭啸, 等. 改进势场蚁群算法下的

- [J]. 科学技术创新, 2018 (7): 74-75.
- [6] 汪莹, 李骏涛, 娄一翎, 等. 大语言模型辅助软件开发的经验研究 [J]. 计算机应用与软件, 2025, (10): 1-11.
- [7] WANG Y, LE H, GOTMARE A, et al. CodeT5+: open code large language models for code understanding and generation [C] //Singapore: Association for Computational Linguistics, 2023.
- [8] THAKUR S, AHMAD B, PEARCE H, et al. VeriGen: a large language model for verilog code generation [R]. ArXiv, 2023: 2308.00708.
- [9] 韩春燕. 基于 Web 应用界面的代码自动生成软件设计 [J]. 现代电子技术, 2017, 40 (14): 55-57.
- [10] 江一舟, 李斌, 高新闻. 智能仪表交互界面的软件自动生成技术 [J]. 仪表技术, 2009, (5): 15-17.
- [11] 罗一中. 基于深度学习的 OpenCV 图像处理软件设计 [J]. 软件, 2025, 46 (1): 50-52.
- [12] 沈洁, 梅劲松, 沈均贤. 一种改进直方图均衡化的图像增强方法 [J]. 计算机与数字工程, 2024, 52 (10): 3112-3115.
- [13] 宋锐, 杨腾云浩, 李依倚, 等. 一种融合形态学闭运算的双 SVM 卫生陶瓷缺陷检测方法 [J]. 自动化应用, 2021, (6): 14-17.
- [14] 李振杰, 臧春华, 苏宝玉, 等. 基于 OpenCV 图像处理软件的设计 [J]. 物联网技术, 2024, 14 (6): 47-49.
- [15] CUI C, SUN T, LIN M H, et al. PaddleOCR 3.0 技术报告 [R]. ArXiv, 2025: 2507.05595.
- [16] 秦小林, 古徐, 李弟诚, 等. 大语言模型综述与展望 [J]. 计算机应用, 2025, 45 (3): 685-696.
- [17] 吴春志, 赵玉龙, 刘鑫, 等. 大语言模型微调方法研究综述 [J]. 中文信息学报, 2025, 39 (2): 1-26.
- [18] 陈晓阳, 高飞, 韩翔宇, 等. 大模型代码生成技术及航天领域潜在应用 [J]. 航天控制, 2025, 43 (1): 8-16.
- [19] 王东清, 芦飞, 张炳会, 等. 大语言模型中提示词工程综述 [J]. 计算机系统应用, 2025, 34 (1): 1-10.
- [20] AI 赋能智创未来——你的智能秘书助手腾讯元宝 [J]. 秘书之友, 2025, (8): 2.
- [15] 费思远, 鲜斌, 王岭. 基于群集行为的分布式多无人机编队动态避障控制 [J]. 控制理论与应用, 2022, 39 (1): 1-11.
- [16] 周振, 耿晨晨, 崔若庚, 等. 基于改进融合蚁群算法的 AGV 路径规划 [J]. 计算机仿真, 2024, 41 (4): 441-445.
- [17] 魏瑶, 刘志成, 蔡彬, 等. 基于深度循环双 Q 网络的无人机避障算法研究 [J]. 西北工业大学学报, 2022, 40 (5): 970-979.