

Jetson Nano B01 表情识别系统开发前期

1. 开发板的组装与硬件连接（2024-09-01~2024-09-19）

任务描述：

- 1) 完成 Jetson Nano B01 开发板的组装工作，包括设备线路连接、机箱组装。
- 2) 安装主要硬件设备，特别是 USB 摄像头，并确保所有硬件设备的连接稳定。
- 3) 对 SD 卡进行格式化、分区，并烧录操作系统镜像文件。

详细步骤：

- 1) 检查所有硬件组件，包括开发板、USB 摄像头、数据线、电源适配器等。
- 2) 按照官方指南连接线路，确保电源和数据传输线路无误。
- 3) 组装机箱，确保所有组件固定，避免运行中因震动造成接触不良。
- 4) 安装 USB 摄像头，检查连接稳定性，确保数据传输无延迟。
- 5) 使用专业工具对 SD 卡进行格式化和分区，选择适合的文件系统。
- 6) 烧录操作系统镜像文件到 SD 卡，验证烧录过程无错误。

完成情况：所有步骤均已完成，开发板组装和硬件连接稳定，SD 卡烧录成功。

2. 系统安装与硬件设备验证（2024-09-20~2024-10-07）

任务描述：

- 1) 进行基础系统安装，确保操作系统顺利启动并运行。
- 2) 通过命令行检查硬件设备信息，验证处理器、内存、摄像头等设备的正确识别。
- 3) 验证 Jetson Nano B01 提供的计算机视觉库是否成功导入。

详细步骤：

- 1) 插入烧录好的 SD 卡，启动 Jetson Nano B01，观察启动过程。
- 2) 登录系统后，使用 `lscpu`、`free -m`、`lsusb` 等命令检查硬件信息。
- 3) 安装并运行 OpenCV 测试程序，调整摄像头的分辨率和帧率，确保满足后续视频流处理的需求。
- 4) 导入 Jetson Nano B01 的计算机视觉库，检查是否成功安装并可用。

完成情况：系统安装顺利完成，所有硬件设备均被正确识别，计算机视觉库导入成功。

3. 配置开发环境与工具（2024-10-08~2024-10-25）

任务描述：

- 1) 配置集成开发环境（IDE），选择 PyCharm 作为开发工具。
- 2) 安装 JDK 11，支持 PyCharm 的运行。

详细步骤：

- 1) 下载并安装 JDK 11，确保环境变量配置正确。
- 2) 下载 PyCharm 2020 版本，按照官方指南进行安装。
- 3) 配置 PyCharm，包括设置解释器、插件安装等。
- 4) 测试 PyCharm 是否能够顺利运行，包括创建测试项目和运行简单程序。

完成情况：JDK 11 安装成功，PyCharm 配置完成，开发环境搭建顺利，可以正常运行。

4. 确定开发任务与功能配置（2024-10-26~2024-10-31）

任务描述：

1) 明确项目开发目标，核心任务是实现基于 Jetson Nano B01 开发板的表情识别嵌入式开发。

- 2) 安装额外的库和工具，包括 PyQt5 和 Facenet_PyTorch 库。

详细步骤：

- 1) 根据项目需求，列出所需库和工具，并进行版本检查。
- 2) 安装 PyQt5，用于前端界面设计。
- 3) 安装 Facenet_PyTorch 库，用于人脸识别功能。
- 4) 配置库文件，确保与项目环境兼容。

完成情况：项目目标明确，所有必要的库和工具均已安装配置完成。

Jetson Nano B01 表情识别系统开发中期

1. 人脸识别算法的选定与优化（2024-11-01~2024-11-10）

任务描述：

- 1) 选择适合的人脸识别算法，并进行初步测试。
- 2) 针对性能问题，优化算法以提高用户体验。

详细步骤：

- 1) 评估并选择了 `facenet_pytorch` 库作为初始的人脸识别算法工具。
- 2) 通过测试代码调用预训练模型，验证了基本的人脸识别功能。
- 3) 识别并记录了程序启动时间过长和界面卡顿的性能问题。
- 4) 决定替换算法，选择了 Haar 特征级联人脸检测算法以减少计算资源消耗。
- 5) 调整摄像头配置参数，包括分辨率和帧率（很重要），以适应 Jetson Nano B01 的硬件环境。
- 6) 重新测试并验证了优化后的摄像头捕捉效果和程序响应速度。

完成情况：成功优化人脸识别算法，提升了系统性能，改善了用户体验。

2. 表情识别模型的选定与训练（2024-11-11~2024-11-19）

任务描述：

- 1) 选择并实现适合的表情识别模型。
- 2) 训练模型并验证其准确率。

详细步骤：

- 1) 选择了基于卷积神经网络（CNN）的表情识别模型。
- 2) 设计了卷积神经网络结构，包括卷积层、池化层和全连接层。
- 3) 使用包含 55338 张图片的数据集进行模型训练和验证。测试模型对 8 种情绪表情（生气、厌恶、沮丧、害怕、开心、正常、惊喜、疲惫）的识别能力。
- 4) 验证了模型在数据集上的准确度，接近 90%。

完成情况：模型训练完成，准确度满足项目需求可以用于检测人脸表情。

3. 疲惫提醒算法的实现（2024-11-20~2024-11-21）

任务描述：

实现基于滑动窗口方法的疲惫提醒算法。

详细步骤：

1) 设计了算法逻辑，通过统计一段时间内窗口中的疲惫状态个数来判断用户是否处于疲惫状态，而不是仅仅根据识别当前帧。

2) 实现了算法代码，包括状态统计、队列管理和阈值判断。

3) 测试了算法的有效性，并调整了阈值（自己进行测试调出一个合适的阈值）以确保提醒的合理性。

完成情况：疲惫提醒算法实现并测试通过，能够合理地提醒用户。

4. PyQt 界面开发（2024-11-22~2024-12-07）

任务描述：

使用 PyQt5 开发直观、易用的用户界面。

详细步骤：

1) 设计了界面布局，包括视频显示区域、情绪识别结果显示、系统控制面板和报告生成按钮。

2) 选择了现代化的界面布局和颜色搭配，确保界面美观、易用。

3) 对各个功能模块进行了多次测试，确保功能的稳定性与流畅性。

完成情况：用户界面开发完成，实现了一个直观又实用的界面，通过图形化界面方便和用户之间的沟通。

Jetson Nano B01 表情识别系统开发后期

1.提醒功能测试（2024-12-08~2024-12-09）

任务描述：

验证系统是否能够识别用户的表情状态。

详细步骤：

- 1) 打开摄像头运行系统，用户表现出 8 种表情状态，观察右侧的概率条以及框选区域上方的表情的显示
- 2) 观察系统识别出表情状态是否和自己表现出的表情状态一致。

测试结果：

系统识别出的表情和用户表现出的表情基本一致，可以完成表情识别的任务，符合预期。

2.提醒功能测试（2024-12-08~2024-12-09）

任务描述：

验证休息提醒和疲惫提醒功能的正常运行。

详细步骤：

- （1）休息提醒功能测试：
 - 1) 用户在系统设置中开启休息提醒功能，并设置提醒时长。
 - 2) 系统根据用户设置记录工作时长，并在设定时间到达时提醒用户休息。
- （2）疲惫提醒功能测试：
 - 1) 启用疲惫状态检测功能，系统自动检测并记录疲惫状态的频率。
 - 2) 当疲惫状态超过设定阈值时，系统提醒用户休息。

测试结果：

- 1) 休息提醒功能能够准确执行用户设定的提醒时长。
- 2) 疲惫提醒功能能够在实际使用中准确执行，及时提醒用户休息。

3. 情绪报告生成功能测试（2024-12-10~2024-12-10）

任务描述：

测试情绪报告生成功能，确保能够准确记录和展示用户情绪变化。

详细步骤：

- 1) 开启摄像头并运行系统，记录用户在一定时间内的情绪状态。
- 2) 系统自动记录各情绪的出现时间和持续时长，点击生成报告观察生成的情绪报告。

测试结果：

情绪报告功能能够准确统计用户情绪状态的持续时间，并生成详细的情绪报告。