Objectives

1. Complete the transformation of all remaining vector-based functions into the double 2D pointer method.
2. Update `adjustments.h` and its source files to ensure compatibility with the new data handling method.
3. Enhance UI and control panel for seamless integration with updated processing functions.
4. Streamline code by removing obsolete vector-based functions.
5. Debug and ensure the stability of the transformed dark line and interlace functions.

Activities

1. **Function Transformations:**
    - Updated **`adjustments.h`** (Gamma, Sharpen, Contrast, and region adjustments) to use double 2D pointers:
        - Replaced `std::vector<std::vector<uint16_t>>` with `double**`.
        - Added height and width parameters.
        - Utilised `malloc2D` for memory allocation and `free` for cleanup.
        - Replaced `uint16_t` with `double` for precision.
        - Updated threading and clamp operations for compatibility with raw pointers.
    - Removed `vectorToMat` and `MatToVector` conversion functions as they are no longer needed.
    - Updated key functions such as `constructor`, `destructor`, `loadImage`, `updateAndSaveFinalImage`, and `processYXAxis` to align with the double 2D pointer method.
    - Converted all interlace-related functions to the double 2D pointer approach.
2. **Dark Line Function Overhaul:**
    - Integrated the dark line detection and removal into `darkline_pointer.h` and its source code using double 2D pointers.
    - Updated the UI to handle dark line processing exclusively through the pointer method.

- Debugging the dark line function after integration (issue persists, pending further investigation).

3. **Control Panel Integration:**
   - Updated the control panel for compatibility with the newly transformed double 2D pointer functions.
   - Ensured that UI elements and buttons are properly linked to the updated functions.
4. **Clean Code Practices:**
   - Removed vector-based implementations from `image_processor.h` and other source files.
   - Ensured CLAHE processing now directly uses the pointer-based implementation, bypassing vector-to-matrix conversions.

Achievements

1. All vector-based functions in `adjustments.h` and related modules have been successfully transformed to double 2D pointer methods.
2. Control panel now supports and applies all updated pointer-based functions effectively.
3. Cleaned up obsolete vector-based functions, streamlining the codebase for improved maintainability.
4. Integrated interlace-related functionality into the pointer-based approach without affecting performance.

Problems & Solutions

1. **Dark Line Function Integration Issues:**
   - **Problem:** The dark line function encounters issues post-integration.
   - **Solution:** Pending investigation to determine the root cause and resolve stability issues.
2. **Interlace Function Adjustments:**

- **Problem:** Initial complications with the interlace transformation into double 2D pointers.
- **Solution:** Debugged and completed the transformation successfully.

3. **UI Updates:**
    - **Problem:** Ensuring seamless application of pointer-based functions via the control panel.
    - **Solution:** Thorough testing and integration to verify that UI operations correspond correctly to the updated methods.