Objectives

- Integrate the current project into the debug version of OpenCV in a new project (`Image_Processor`).
- Enhance the project with better calibration and line detection functionalities.
- Resolve performance issues and implement an improved line detection system.
- Begin tasks for a new line detection function with vertical, horizontal, and combined detection.
- Submit the updated line detection function to the team lead.

Activities

- **Integration into `Image_Processor` Project:**
  - Updated the `.pro` file to link OpenCV and QCustomPlot for histogram plotting.
  - Linked the `CGProcessImage` library provided by the team lead into the project.
  - Replaced PowerShell commands with `copy` commands in the `.pro` file for library copying.
- **UI and Functional Updates:**
  - Added a "Data Calibration" button, utilising a double 2D pointer method to calibrate images.
  - Added interlace and merge buttons but identified issues with merging, leading to program crashes when using folding mode.
  - Integrated debugging messages to identify issues in the interlace and merge functionalities.
- **Performance Optimisations:**
  - Resolved GPU memory leaks by ensuring proper CUDA resource cleanup.
  - Reduced lag by optimising histogram calculations and limiting frequent calls to `replot()`.
  - Improved histogram performance but acknowledged persistent slowdowns during loading due to simultaneous plotting.
- **Line Detection Function Development:**
  - Added functions for specific line detection:
    - `detectVerticalLines`: Detects only vertical lines.
    - `detectHorizontalLines`: Detects only horizontal lines.

        - `checkforVertical`, `checkforHorizontal`, and `checkforBoth` for handling detection modes.
    - Updated UI to test the new functions, allowing users to select detection modes via checkboxes.
- **Code Refactoring and Optimisation:**
    - Modified detection functions to return a boolean instead of `DarkLineArray*`.
    - Simplified function parameters and added error handling with an `outLines` parameter.
    - Enhanced label drawing for better visual clarity of detected lines.
- **Team Collaboration:**
    - Briefed by the team lead on new tasks, including:
        - Horizontal/vertical line detection.
        - Transforming the CLAHE function to a double 2D pointer method.
        - Delaying the 3D histogram integration until line detection and CLAHE updates are complete.
    - Submitted the updated line detection function for review.

Achievements

- Successfully integrated the debug version of OpenCV into the `Image_Processor` project.
- Implemented a new calibration method using the double 2D pointer structure.
- Resolved major performance issues in histogram plotting and GPU memory usage.
- Developed and tested advanced line detection functions with improved UI integration.
- Submitted a refined line detection function for team lead review.

Problem & Solution

- **Problem:** Performance issues due to GPU memory leaks and frequent replotting. **Solution:** Optimised CUDA resource cleanup and restructured histogram plotting logic.
- **Problem:** Crashes during merging in interlace functionality. **Solution:** Identified issues with the double 2D pointer structure and introduced debugging logs for detailed investigation.
- **Problem:** Redundant parameters in line detection functions. **Solution:** Refactored functions to streamline parameters and improve error handling.

- **Problem:** Overlapping labels in line detection visualisation. **Solution:** Enhanced label positioning logic and added boundary checks.
- **Problem:** Library path issues in `.pro` file. **Solution:** Corrected paths and updated configurations for proper linking.