1. **Objectives:**
    - Ensure OpenCV build runs in GPU mode and outputs time.
    - Debug and fix Pixel Info display in the control panel and mouse movement updates.
    - Correct the rectangle drawing function and ensure proper event handling.
    - Debug image effects and filters, ensuring they are applied properly.
    - Improve the Revert and Last Action functionalities.
    - Solve various UI and image display issues, including incorrect grey filters and noise after calibration.
    - Ensure CLAHE and other functions apply correctly, handling user input properly.
    - Fully integrate OpenCV with GPU support using Microsoft Visual Studio (MSVC).

---

**2. Activities:**

- **OpenCV Build:**
    - Tried to run the OpenCV build configured via Visual Studio 17 2020, using CMake, OpenCV Repo, and OpenCV-Contrib repo from GitHub.
        - **Successful:** The output is running in GPU mode, displaying images and processing time with GPU acceleration (using other sample files, not the project).
- **Pixel's Info Not Displayed in Control Panel:**
    - The pixel info was previously debugged in the console instead of being shown at the designated bar inside the control panel.
        - **Solved:** The pixel info is now correctly displayed in the control panel.
- **Pixel Info Not Updating on Mouse Movement:**
    - The pixel information was not updating as the mouse moved across the image.
        - **Solved:** Updated the mouse signal to correctly trigger the Pixel's Info update function, ensuring real-time info display as the mouse moves.
- **Rectangle Drawing Not Displayed:**
    - The rectangle for selecting regions did not appear on the screen during the drawing.
        - **Solved:** Mouse events (press, move, release) now correctly track the selection process using the isSelecting flag and update the selectedRegion coordinates.

        - The paintEvent method draws a red rectangle overlay based on the current selection state and region coordinates.
        - The update() method is called after changes to ensure the rectangle is displayed correctly.

- **Quick Functionality Checks:**
    - Performed a quick check on each function to ensure that the applied operations reflected correctly on the image output.
- **Crop and Region Adjustment Issues:**
    - Found that cropping and region adjustments were facing issues when the cursor selection allowed for negative x values (selecting in the reverse direction).
        - **Solved:** Adjusted the rectangle algorithm to compute the minimum and maximum values for the left/right and top/bottom coordinates. This normalizes the width to avoid negative values and ensures correct cropping and region adjustments.
- **Padding, Rotation, and Distortion Issues:**
    - The padding, rotation in both directions and distortion effects were not being applied as expected.
        - **Solved:** Assigned the output to finalImage, which allowed the effects to appear correctly.
- **Last Action Bar:**
    - Added functionality to display the last action performed by the user. If reverted, it should show the action before the reverted step.
        - **Status 1:** UpdateLastAction () initially had a bug, causing incorrect updates to the Last Action bar. It was removed temporarily to resolve other bugs.
        - **Status 2:** Reimplemented updateLastAction() after resolving other bugs. The function now updates the last action for every step taken, though the action before the reverted step has not yet been handled properly.
            - **Functionality added:** All actions are stored in the action history, and the correct action is displayed when "Revert" is applied, though showing “Revert” in the Last Action bar is still under work.
- **Revert Function Issues:**
    - The Revert function was not always reverting to the last step, and in some cases, it reverted all the way to the initial image loaded.

        - **Solved:** Added saveCurrentState() to all functions before they were called (except loadTxtImage()), ensuring that the correct state is saved and Revert works as expected.

- **Region Sharpen & Region Contrast Not Applied:**
    - These functions were not updating the image output after adjustments.
        - **Solved:** Added saveImageState() to ensure the effects were updated in the output after adjustments.
- **Control Panel Separation:**
    - The control panel was separated into its own header file and class file for better organization and maintainability.
- **Pinned Pixel's Info Bar:**
    - The Pixel's Info bar was pinned to the top of all button layouts to make it more visible and easily accessible during image processing.
- **Unclear Output and Grey Filter Issue:**
    - The loaded output appeared unclear, with a grey filter and improper noise-cleaning after calibration.
        - Investigated loadTxtImage() – no issues were found.
        - Investigated ProcessSplitandMerged() – no issues were found.
        - Investigated updateImageDisplay() – **Problem detected:**
            - **Solved:** Updated the initializer for the loop counter to use an integer and converted 16-bit pixel values to 8-bit when setting the pixel colour (using >> 8).
- **CLAHE Function Not Applied to Output:**
    - The CLAHE function detected bugs where the effect was not being applied to the image output.
        - **Solved:** The result of matToVector() was not being used to update the final image. This was fixed by properly linking matToVector in the header file.
- **Updated Last Action Bar Parameters:**
    - The Last Action bar was updated to show the parameters applied to functions via the pop-up input box.
- **Input Box Issues for Functions Requiring Parameters:**

    - The effect was still being applied even when the user cancelled the operation in the input box.
        - **Solved:** Added logic to check if "OK" was pressed and only then apply the function.

- **OpenCV Integration with MSVC:**
    - Successfully integrated the OpenCV project with the MinGW kit into OpenCV_MSVC, using Microsoft Visual Studio 17 2022 for debugging and release. The system now runs OpenCV with GPU functionality.

---

**3. Achievements:**

- Successfully built and ran the OpenCV build with GPU support, including displaying GPU processing time.
- Fixed Pixel's Info display and ensured real-time updates on mouse movement.
- Resolved issues with the rectangle drawing function, ensuring it appears properly during region selection.
- Debugged and fixed various issues related to effects not applying, including padding, rotation, and distortion.
- Improved the Last Action and Revert functionalities, with partial success in showing action histories.
- Solved the grey filter issue on output and ensured noise-cleaning works after calibration.
- Ensured CLAHE and other effects applied correctly after adding proper function calls.
- Completed integration of OpenCV with MSVC for GPU functionality, including CUDA support.

---

**4. Problems & Solutions:**

- **Problem:** OpenCV build with GPU using MingW failed due to lack of CuDNN support.
    - **Solution:** Switched to MSVC and CMake to build OpenCV with GPU, which was successful.
- **Problem:** The rectangle drawing function was not displaying after file separation.
    - **Solution:** Updated mouse event handling and repainting logic to ensure the rectangle appears properly during selection.
- **Problem:** Image output appeared with a grey filter, and noise was not cleaned after calibration.

    - **Solution:** Investigated and resolved issues in updateImageDisplay(), including updating the loop counter and converting pixel values from 16-bit to 8-bit.

- **Problem:** CLAHE function was not being applied to the final image output.
    - **Solution:** Linked the result of matToVector() to the header file and ensured it updated the final image correctly.
- **Problem:** Input box issues caused functions to apply even when the operation was cancelled.
    - **Solution:** Added logic to check for the "OK" confirmation before applying the function.