Plate Analysis Script Instructions

Version 1.0

Enhanced Instructions & Cleaned Reasoning

1. Core Fixes & Enhancements

- Correct the Red/Blue swap: assign color channels properly.
- Add three numeric columns: `Red`, `Green`, `Blue`.
- After printing the well-level table, display:

  Mean ± 2·SD for R: . . . , G: . . . , B: . . .

2. Command-Line Modes

`--mode all` Analyze all 96 wells; output full plate table and overall RGB statistics.

`--mode left` Analyze wells A1–H6 (48 wells); output left-side table and its RGB statistics.

`--mode right` Use left side as control, compute its Mean±2·SD for each channel. On the right side (wells A7–H12), add a `WithinRange` column (TRUE/FALSE) based on whether each well's RGB triplet lies within control ±2·SD. Then output the right-side table, control stats, and "Number within range: {`count`}".

3. Additional Features

- `--avoid_edges`: exclude outer-border wells from analysis.
- `--sd_range K`: use $K \times \sigma$ instead of fixed 2·SD.
- `--stats [mean,median,p25,p75,IQR]`: select summary statistics.

4. Image Handling

- Standard naming: `img_001.png`, `img_002.png`, . . .
- `auto_crop.py`: crops a center circle (radius = 20 px) around each well.
- See inline comments and `docs/tutorial.mp4` for capture standards.

5. Output & Export

- `--output results.tsv`: save the well-level table as TSV.
- `--summary results_summary.txt`: save the statistics report.

6. Advanced & Future Extensions

- Batch-mode: process an entire directory of plate images.
- Classifier training: build color-vs-dilution models from known data.
- Dataset builder: generate CSV of dilution factors vs. RGB metrics.
- GitHub integration: set up CI tests, release tagging, and documentation site.

7. Usage Examples

- `python plate_analyze.py --mode all --stats mean,median --avoid_edges`
- `python plate_analyze.py --mode left --sd_range 1.5`
- `python plate_analyze.py --mode right --output test_results.tsv`

8. Notes

- Images must reside in `./images/` following the naming convention.
- Adjust crop radius in `auto_crop.py` for precise centering.
- Refer to `docs/tutorial.mp4` for a complete screen-recorded walkthrough.