Infinite-ISP Tuning Tool

User Guide

v0.3

Sr.	Description	Version	Date
1.	Initial draft	v0.1	26-05-2023
2.	Figures and description updated	v0.2	29-05-2023
3.	Formatting and description improvements	v0.3	29-05-2023

Table of Contents

List of Figures

List of Tables

Introduction

Overview

The Infinite-ISP Tuning Tool is specifically designed to analyze and fine-tune the performance of the modules within the Infinite-ISP pipeline. Infinite-ISP is a collection of camera pipeline modules implemented at the application level to convert an input RAW image from a sensor to an output RGB image. The overall workflow for this tuning tool is visible in Figure 1.

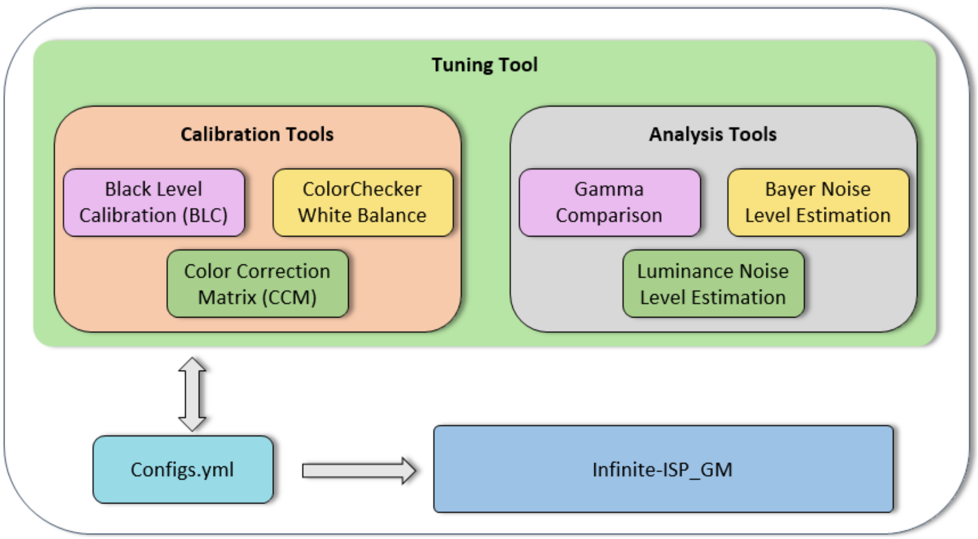

Figure 1: An overview of the Infinite-ISP Tuning Tool

This tool can be used separately as a stand-alone application for image quality analysis. This comprehensive toolset not only provides the user with precise control over the calibration modules, but also allows the user to achieve outstanding image quality through the analysis modules. With respect to its usage, it is divided into two categories:

1. Calibration Tools: Used to generate algorithm parameters for modules like black levels, white balance gains, and color correction matrix.
2. Analysis Tools: Used to provide common data and graphs to support analysis like gamma curves, bayer & luma noise levels.

To make sure the tuning tool and the Infinite-ISP pipeline work together seamlessly, they use a special file called configs.yml. This file plays a crucial role by specifying the specific input parameters for each module within the pipeline. It grants users the flexibility to adjust these parameters according to their unique needs and requirements. For this, the tuning tool will help to fine tune the modules by providing calibration and analysis assistance. By doing so, users can fine-tune the performance of the pipeline, ensuring optimal results.

Features

Infinite-ISP provides a set of camera pipeline modules that needs calibration and analysis for providing quality results. For this, tuning tool provides user with the following features present in Table 1:

Table 1: Features of the Infinite-ISP Tuning Tool

	Modules	Description
Calibration Tools	Black Level Calibration (BLC)	Calculates the black levels of a raw image for each channel (R, Gr, Gb, and B).
	ColorChecker White Balance (WB)	Calculates the white balance gains (R gain and B gains) on a ColorChecker RAW or RGB image.
	Color Correction Matrix (CCM)	Calculates a 3x3 color correction matrix using a ColorChecker RAW or RGB image.
Analysis Tools	Gamma Curves (GC)	Compares the user-defined gamma curve with the sRGB color space gamma ≈ 2.2.
	Bayer Noise Level Estimation (BNE)	Estimates the noise levels of the six grayscale patches on a ColorChecker RAW image.
	Luminance Noise Level Estimation (LNE)	Estimates the luminance noise level of the six grayscale patches on a ColorChecker RAW or RGB image.

Directory Description

Within the base directory, *InfiniteISP_TuningTool*, the user will find following folders and files that are essential for the functioning of the tuning tool as shown in Figure 2.

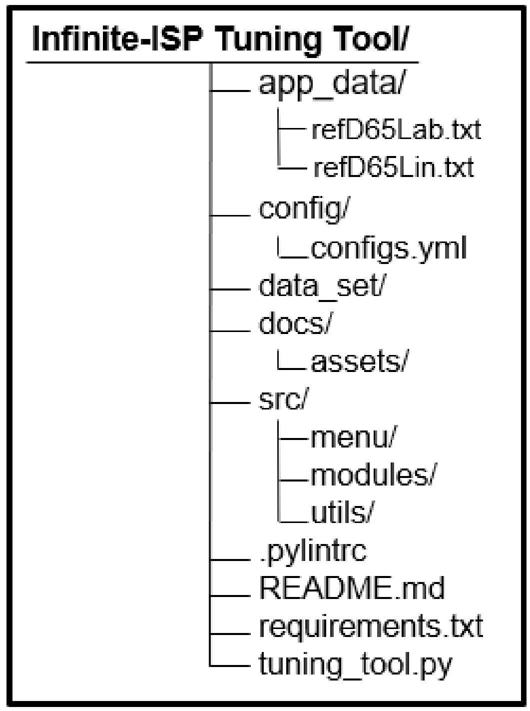

Figure 2: Directory description

Here's a description of each folder and file:

1. The folder ‘app_data’ contains essential reference files: refD65Lab and refD65Lin. These files are necessary for the CCM algorithm to determine the correct color correction matrix.
2. The 'config' folder serves as the default location for configs.yml file similar to the configuration file present in the infinite-ISP. The configs.yml file specifies the input parameters and output settings for each module, ensuring smooth integration between the Tuning Tool and the Infinite-ISP.
3. This folder contains sample images to test the tuning tool. The sample raw images must follow the following naming format.
   File name format: Name_WxH_Nbits_Bayer.raw
   For example: ColorChecker_2592x1536_12bits_RGGB.raw
4. tuning_tool.py: This file serves as the main entry point to run the tuning tool.

Tuning Tool Workflow & Description

The whole workflow of the tuning tool can be seen in Figure 3. The Tuning Tool follows a workflow to load a config file, execute each module, implement its algorithm and save the result. Here's a step-by-step guide on how to use the tool effectively. The key modules of this tuning tool are,

- Calibration Tools
    - Black Level Calculation (BLC)
    - White Balance (WB)
    - Color Correction Matrix (CCM)
- Analysis Tools
    - Gamma Curve (GC)
    - Bayer Noise Estimation (BNE)
    - Luminance Noise Estimation (LNE)

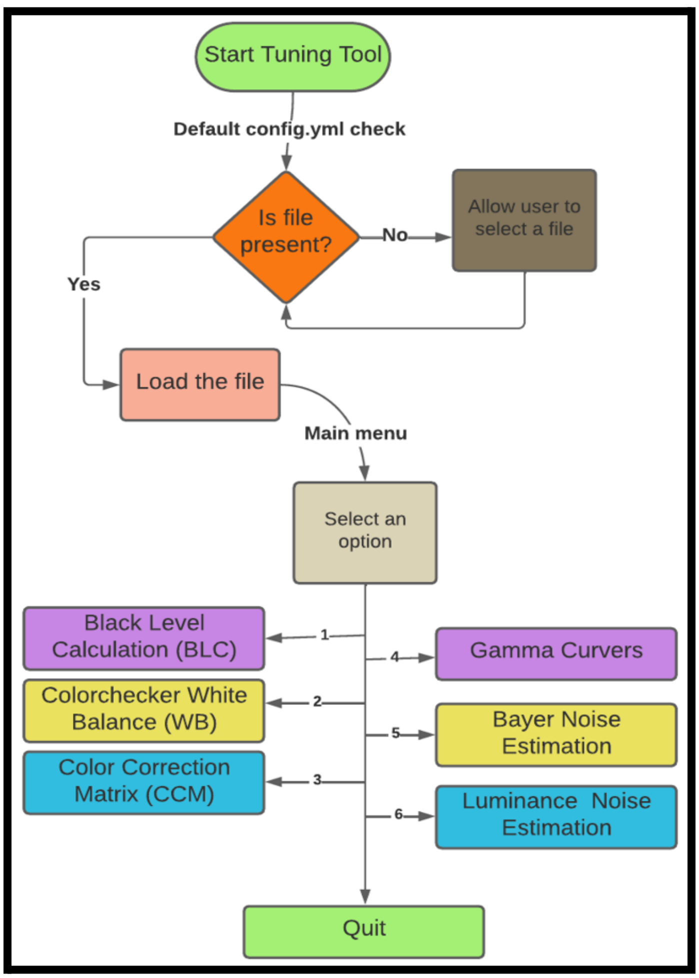

Figure 3: Main menu workflow

Getting Started

Load the Config File

To make use of the configs.yml file, it needs to be loaded into the tuning tool associated with the Infinite-ISP. Loading the configs.yml file to the tuning tool is necessary as this file serves as a central file for configuration parameters and settings that govern the behavior of different modules in the infinite-ISP pipeline.

The contents of the configs.yml file may vary depending on the specific requirements and preferences of the user. However, certain modules within the infinite-ISP pipeline typically require tuning and analysis to achieve desired image processing outcomes. The configs.yml file provides a structured format to define and manage these tuning parameters. It grants users the flexibility to manually adjust these parameters according to their unique needs and requirements

- By default, the config folder in the tool's directory is considered the default path for the config file. Ensure that the necessary config file is present in this location. If the file is found, the following menu in Figure 4 will appear on starting the tuning tool.

```
                         Welcome to Tuning Tool

--------------------------------Main Menu--------------------------------

Select a command:
1. Calculate Black Levels
2. Calculate Colorchecker White Balance
3. Calculate Color Correction Matrix
4. Generate Gamma Curves
5. Estimate Bayer Noise Level
6. Estimate Luminance Noise Level
7. Quit

Enter a number to choose an option:
```

Figure 4: Main menu if the configs.yml file is found

- If the config file is not found in the default path, the tuning tool will prompt the user to load the config file with specific parameters.
- Select 1 to load the configs.yml file. The selected file name will be displayed as shown in Figure 5.

Figure 5: Menu flow after loading the user's choice config file

Module Workflow & Description

Common Functionalities

In this section, three essential functions are explained that are common across all modules. These functions provide fundamental capabilities that are frequently utilized in various modules of the tool. Let's take a closer look at each of them.

Loading an Image

For BLC, WB, CCM, BNE & LNE, loading an image is a must task to run the module shown in Figure 6.

- For BLC, a raw black image
- For WB, CCM, and LNE, a raw or RGB image with ColorChecker
- For BNE, a raw image with ColorChecker.

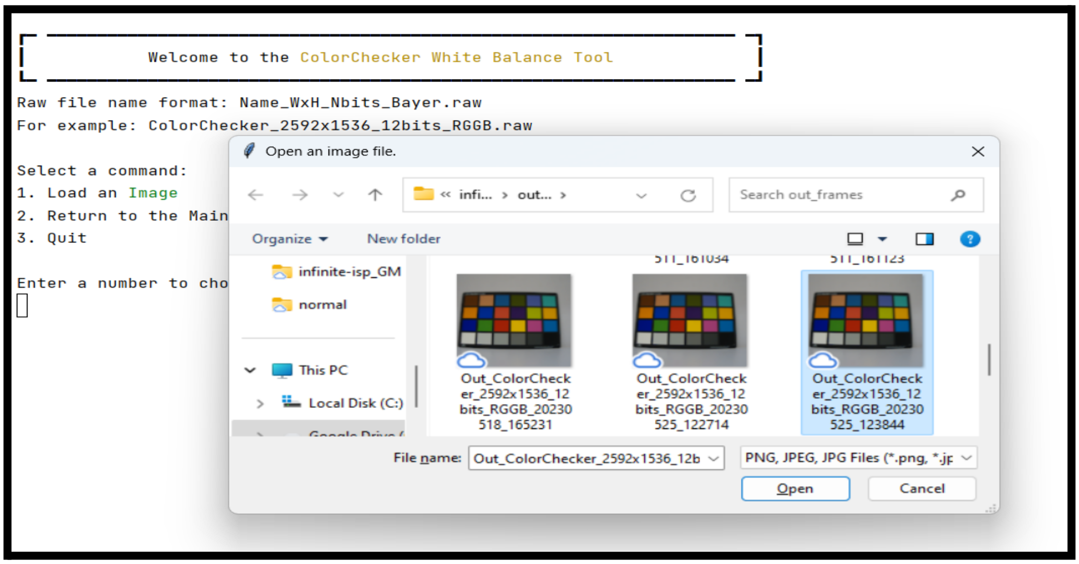

Figure 6: Example to load an image

If the image is not loaded properly, the tool displays an error and will prompt the user to load the file again.

ColorChecker Area Selection

Upon selecting an image that contains a ColorChecker, the image will be displayed to the user as shown in Figure 7.

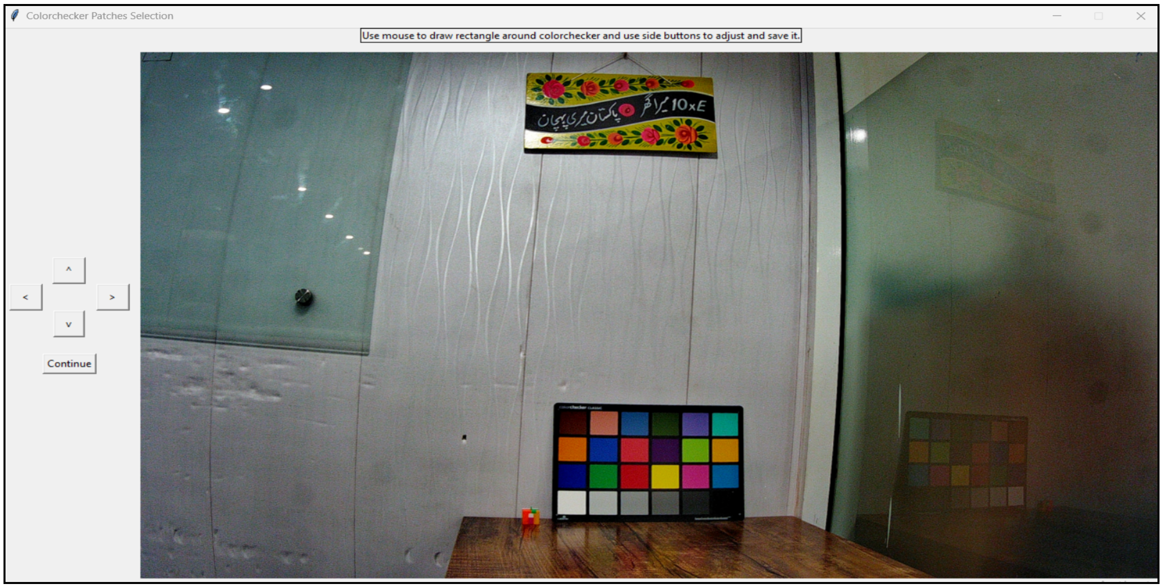

Figure 7: ColorChecker area selection frame to draw rectangle around it

Follow the steps below to locate and select ColorChecker area,

- Use the mouse to draw a rectangle around the ColorChecker as shown in Figure 8.
- Position the cursor at the top-left point of the ColorChecker region.

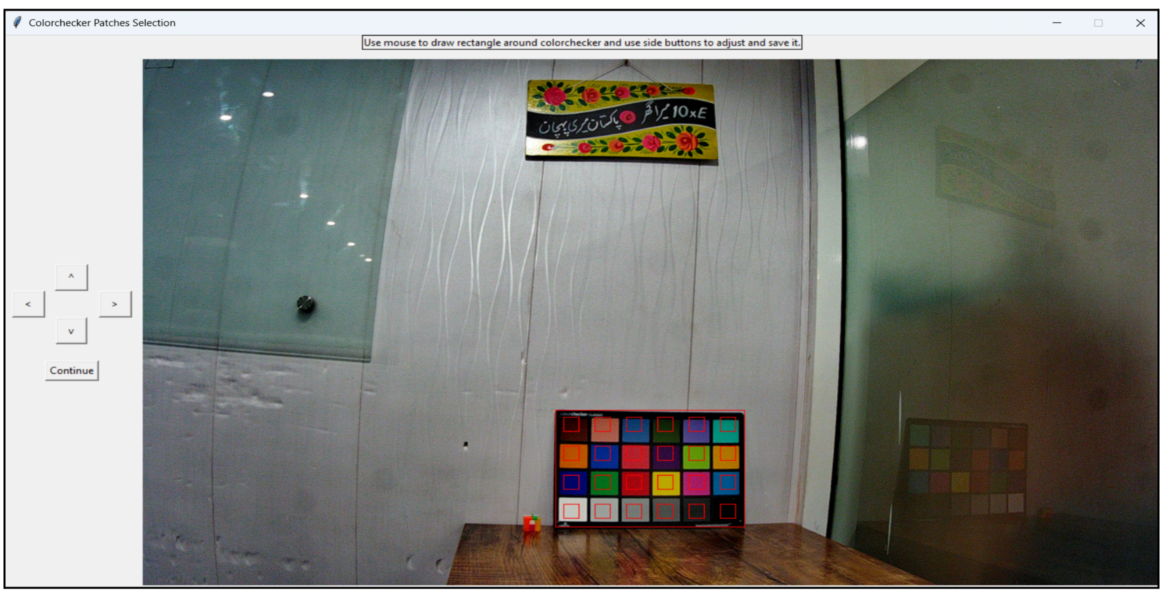

Figure 8: Drawing rectangle around ColorChecker by dragging mouse from top left to bottom right corners of the ColorChecker

- Click and hold the mouse button, then drag the cursor to the bottom-right point of the ColorChecker, forming a rectangle.
- Within the drawn rectangle, small rectangles will be automatically placed around each patch of the ColorChecker image, serving as boundaries for each patch.

- To adjust the size of these small rectangles, use the scroll-up and scroll-down functionalities. Scroll-up increases the size, while scroll-down decreases it.
- If needed, user can fine-tune the position of the rectangle using the arrow buttons provided on the left pane of the panel as shown 1 in Figure 9. These buttons allow adjustments for right, left, top, and bottom movements.

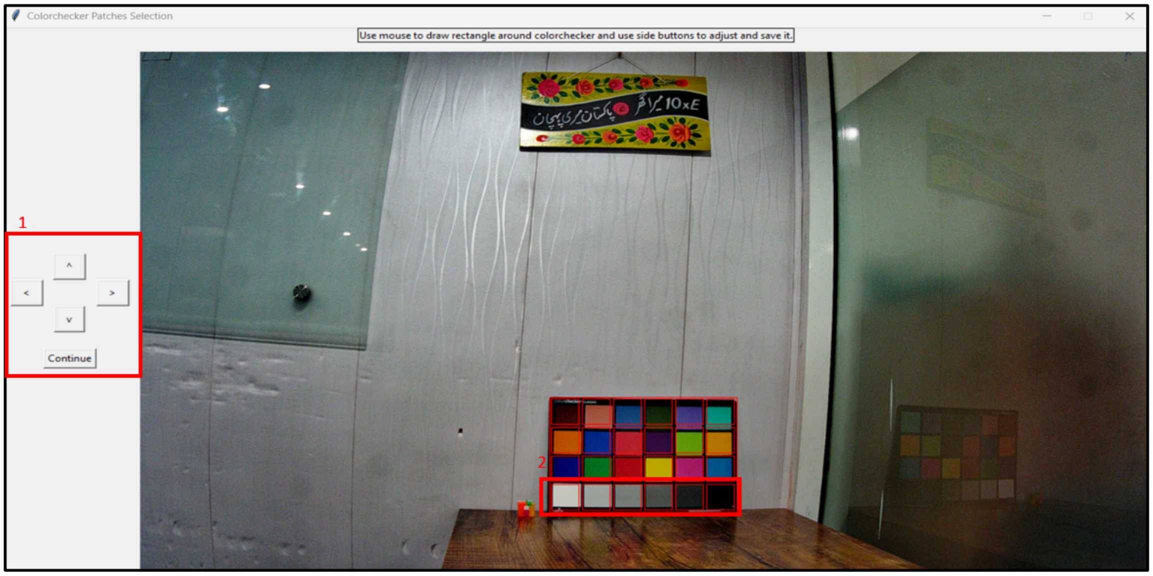

Figure 9: Scrolling up feature has increased the size of inner rectangles. 1. The side pane has arrow buttons to adjust the drawing and a continue button to save and proceed further. 2. Gray patches row (last six patches in the ColorChecker)

- Once user have successfully drawn the rectangle and positioned it appropriately, locate the "Continue" button placed below the arrow buttons.
- Press the "Continue" button to proceed with executing the algorithm flow on the image, applying the necessary operations and calculations.
- Note that, the last six patches on the ColorChecker are knows as grayscale patches shown as 2 in Figure 9, are used in the white balance and noise estimation modules.

Saving the Data

Users have the option to save three types of data based on their preference:

1. After completing the module, user have the option to save the data to a configuration file. For saving data to config file a dialogue box opens up as shown in Figure 10.

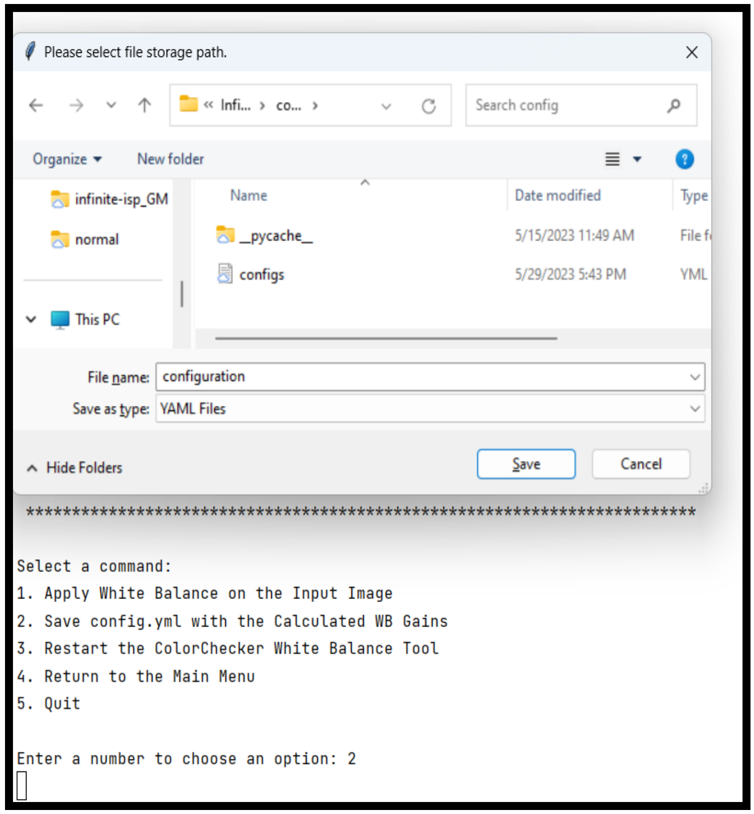

Figure 10: Example of saving data to the config file

2. Saving the data to a CSV file by clicking on the save button provided on the graphical user interface (GUI) that can be seen in Figure 11.

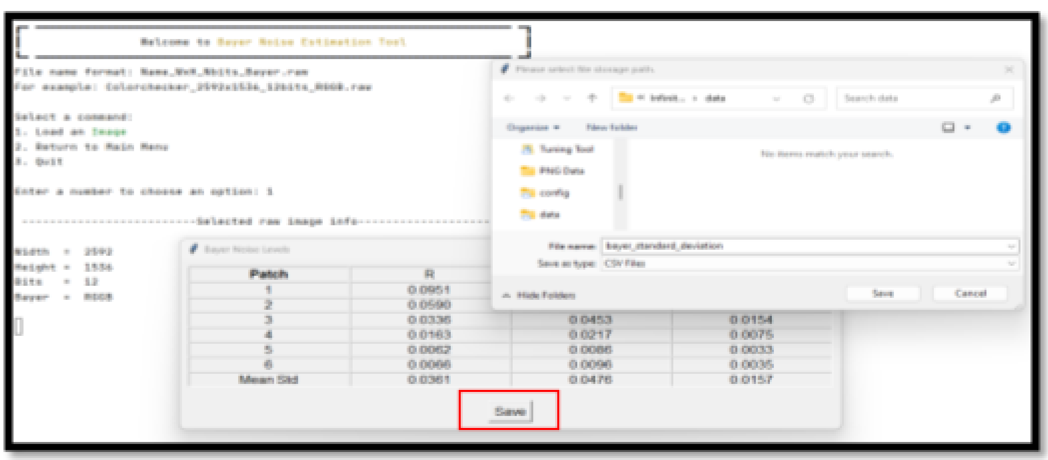

Figure 11: Example of saving csv file from the GUI using save button

3. Saving the data as a PNG image also through the save images button present on the respective panel. Example is shown Figure 12.

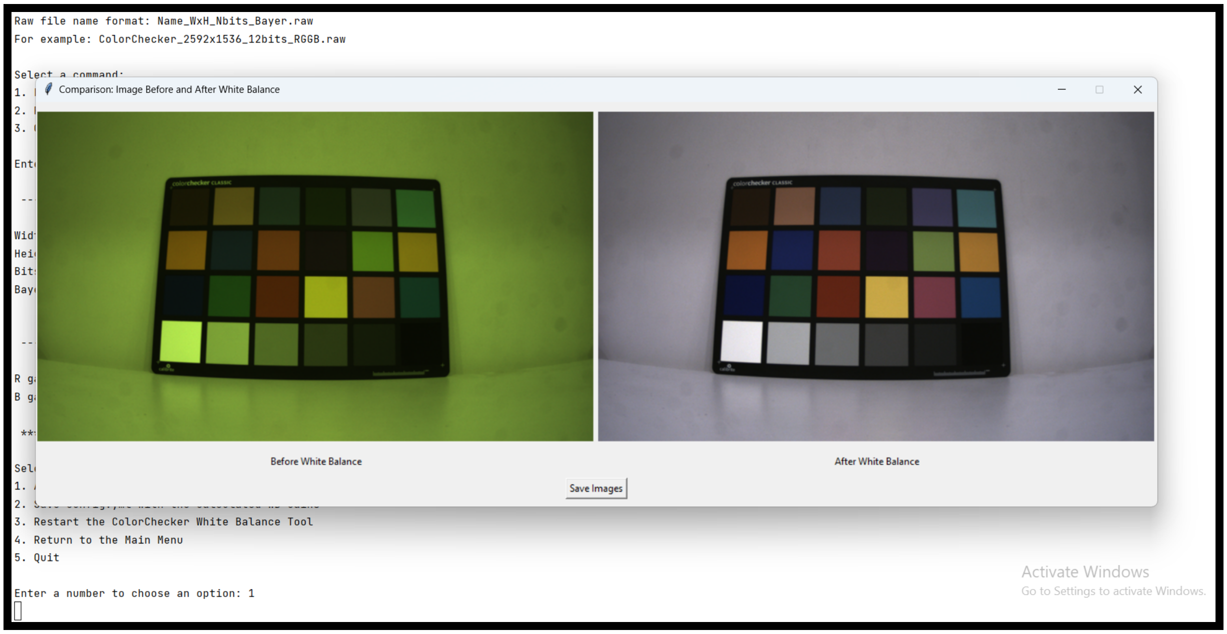

Figure 12: Example of saving PNG data

When the user chooses to save the data, the tool will display a dialogue box, enabling them to select the desired location where the data should be saved. After saving the file, the path of the saved file is shown on the console and end menu reappear as shown below in Figure 13.

Figure 13: Display the path of the saved file on the console

Black Level Calculation

Black level correction (BLC) is a technique used to compensate for the inherent offset present in sensors, where the 0 value of a pixel does not accurately represent a complete absence of light. This offset can lead to non-zero values being outputted even in the absence of light. BLC corrects this sensor limitation by adjusting the output to align the 0 value with true black. By applying BLC, accurate representation of black or dark areas in an image is achieved, ensuring reliable rendering in various lighting conditions.

Module's Input

To start the Black Level Calculation module, follow these steps:

- Open the tuning tool. Access the main menu and choose 1 to select the BLC module as seen in Figure 14.

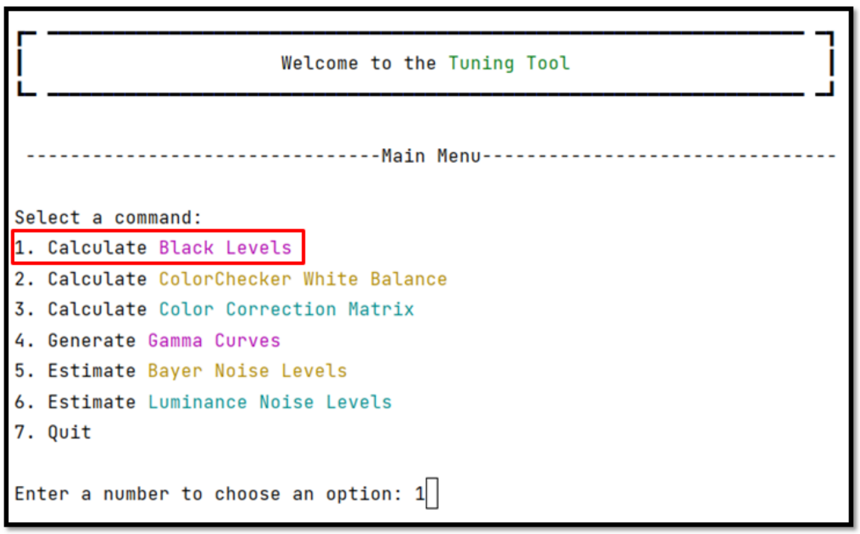

Figure 14: BLC selection from the main menu

- Loading a Black Image: To calibrate the black levels of a sensor, it is necessary to obtain a uniform black image directly from the sensor. The ideal scenario is capturing this image in complete absence of light. To do this, an effective approach involves capturing the image with the lens left uncovered. This approach allows for establishing the minimum or "black level" of brightness that should be present in any image captured by the sensor or processed by an ISP.
- Select option 1 from BLC menu to load such a reference image Figure 15 as described already in load image section in Figure 6.

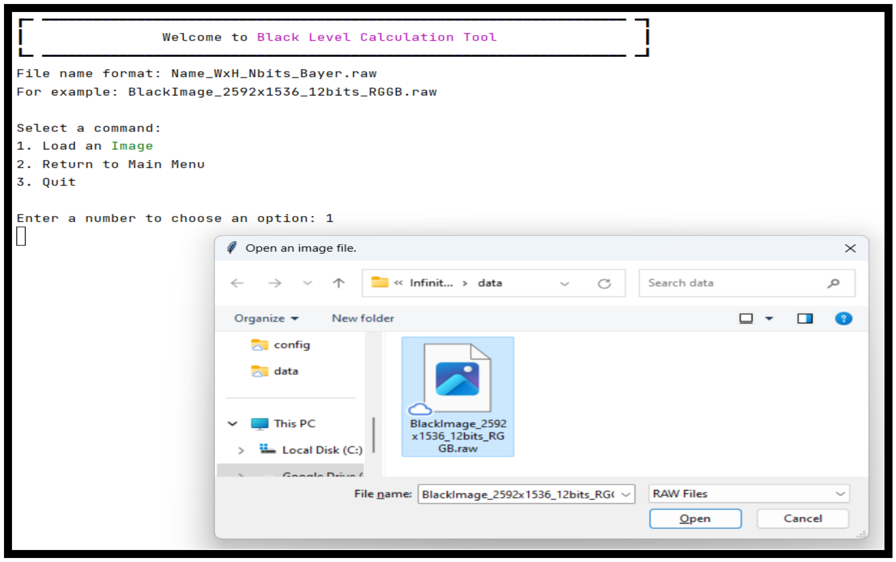

Figure 15: Loading black image for BLC

Results Display/Saving

Upon selecting an image, the BLC algorithm executes and calculates black for each bayer channel separately and displays the calculated black levels as R, Gr, Gb, and B on the console as shown in Figure 16.

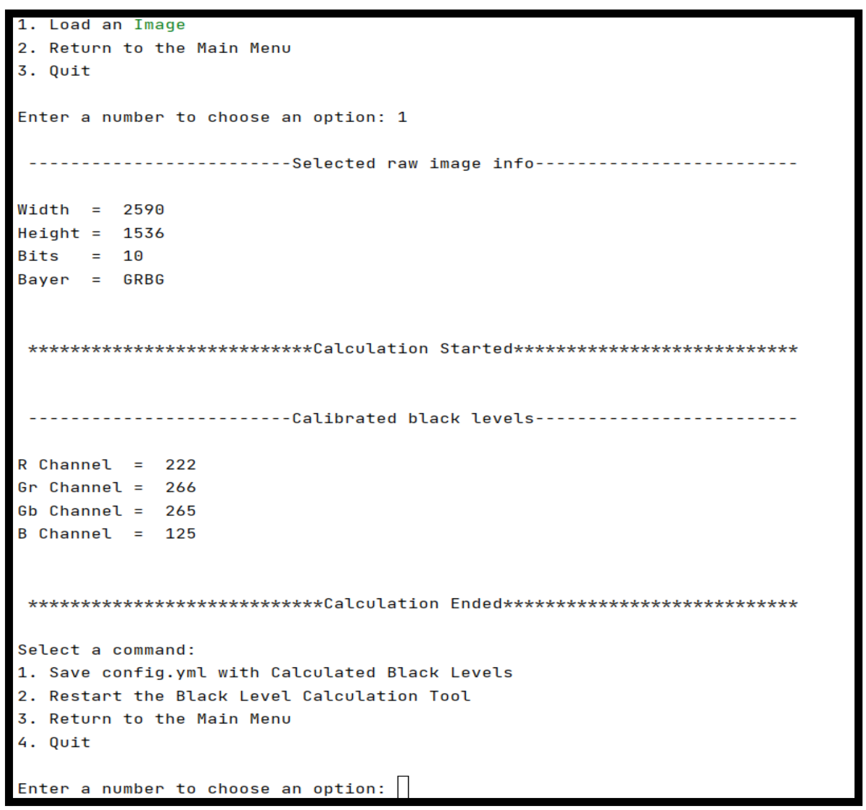

Figure 16: Results display for the BLC module

Subsequently, an end menu with the following options becomes available,

- Save to Config File: Pressing 1 allows user to store the calculated black levels in a configuration file at his desired location as explained in save data section.
- Restart the Module: Opt for 2 to restart the black level calculation tool, enabling user to re-evaluate and adjust the calibration if needed.
- Return to the Main Menu: Enter 3 to conveniently navigate back to the main menu of the tuning tool, granting access to other modules and functionalities.
- Quit: Press 4 to exit the program gracefully and conclude the tuning tool session.

ColorChecker White Balance Calculation

White balance gains (WB) are adjustments that can be made to the red, green, and blue channels in an image. These adjustments are used to ensure that colors appear natural and accurate. By adjusting the gains, we can make sure that white objects in the image look truly white, without any unwanted color tints. The goal is to achieve color fidelity and reproduce colors as faithfully as possible.

Module's Input

To start the White Balance Calculation module, follow these steps:

- From the tuning tool main menu, choose 2 to select the WB module as shown in Figure 17.

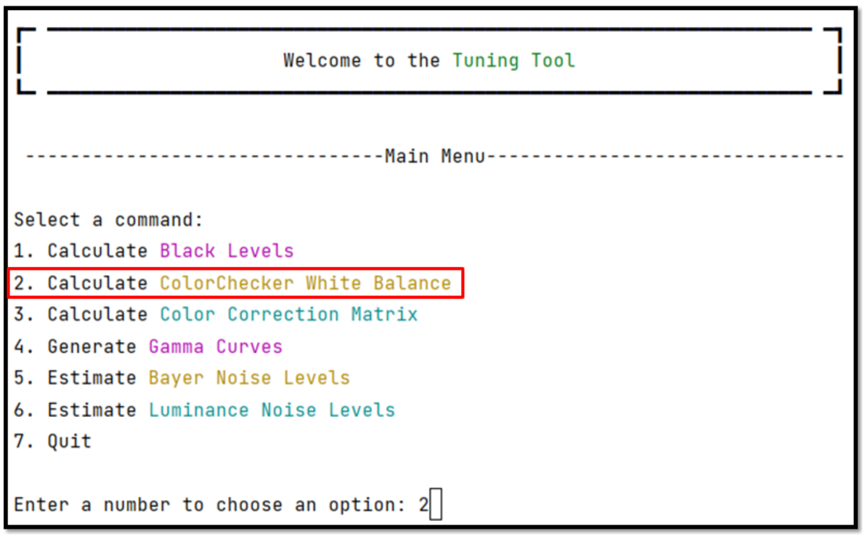

Figure 17: WB selection from the main menu

- Load the image as depicted in Figure 6. For white balance module, user need to ensure that the selected image contains a color checker, as it is required for the accurate calibration. Starting this module will, opens up a dialogue box, allowing user to select an image (either raw or RGB) for white balance calibration.

Calibration Procedure

- Locating the ColorChecker: Once the image is selected, a panel will open, displaying the image as already shown in Figure 7. Use his mouse to draw a rectangle around the color checker region as explained in ColorChecker area selection section above. Fine-tune the position and size of the color checker using the arrow buttons present on the left pane and press continue to run WB algorithm.
- WB Calculations: The gray patches present in the last row of the ColorChecker are used to calibrate the wb gains. The algorithm executes and the white balance gains will be displayed on the console as shown below in Figure 18.

- If user needs to apply white balance, a graphical user interface (GUI) will open, displaying a comparison of the image with and without white balance seen in Figure 19. After closing the result without saving it, the end menu will reappear, providing the aforementioned five.

```
                 Welcome to Colorchecker White Balance Tool

Raw file name format: Name_WxH_Nbits_Bayer.raw
For example: Colorchecker_2592x1536_12bits_RGGB.raw

Select a command:
1. Load an Image
2. Return to Main Menu
3. Quit

Enter a number to choose an option: 1

 ---------------------------White Balance Gains---------------------------

R gain =  1.3463
B gain =  1.0

 *************************************************************************
```

Figure 18: Results display for WB module on the console

Figure 19: Results display after applying wb gains to the input image

Results Display/Saving

After displaying wb gains, an end menu will appear, offering the following options:

- Apply White Balance: Pressing 1 will apply the calculated white balance gains to the image.
- Save Gains to Config File: Selecting 2 will save the white balance gains to a configuration file.

- Restart the WB Tool: Choose option 3 to restart the White Balance Calibration module.
- Return to Main Menu: Pressing 4 will take user back to the main menu of the tuning tool.
- Quit the Program: Select option 5 to exit the program gracefully.

If a user wants to save the results press ‘save images’ button on the panel to save both the images as .png format at his desired location.

.

Color Correction Matrix Calculation

Calculating the Color Correction Matrix (CCM) is an important module in the tuning tool for an Image Signal Processor (ISP). Color correction matrix (CCM) enables precise color adjustments by manipulating the transformation matrix that maps the camera's captured colors to the desired output colors.

The Color Correction Matrix is typically a 3x3 matrix that defines the transformation from the camera's captured colors to the target colors. Each element in the matrix represents the contribution of the corresponding color channel (Red, Green, and Blue) to the final output color. The CCM is used to correct color imbalances, adjust saturation levels, and achieve accurate color representation.

Module's Input

To start the White Balance Calculation module, follow these steps:

- To start CCM module, choose 3 from the tuning tool main menu shown in Figure 20.

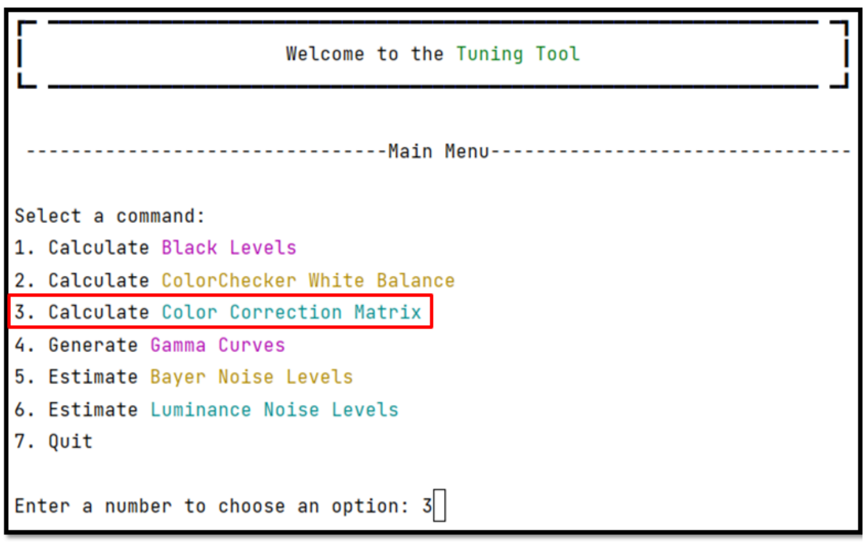

Figure 20: Selection of CCM from the main menu

- Then load a ColorChecker image. For CCM module, user need to ensure that the selected image (either raw or RGB) contains a color checker, as it is required for the accurate calibration.

Calibration Procedure

- Locating the ColorChecker: Once the image is selected, a panel will open, displaying the image. Use his mouse to draw a rectangle around the color checker region as explained in ColorChecker area selection in Figure 7. Fine-tune the position and size of the color checker using the arrow buttons present on the left pane and press continue to proceed further.

- Error Matrix Selection: After ColorChecker area selection, the user will be asked to choose between $\Delta E_{ab}00$ and $\Delta C_{ab}00$ (Figure 21). The reference files for both the algorithms are placed in the AppData folder.
- White Balance Choice: After selecting the error matrix, the user will be prompted to decide whether to maintain white balance in the image or not.

```
-------------------------Selected raw image info-------------------------

Width  =  2590
Height =  1536
Bits   =  10
Bayer  =  GRBG

Parameters required for ccm algorithm:

Select Error Matrix:
1. ΔC 00
2. ΔE 00

Enter a number to choose an option: 1
ΔC 00 is selected.

Do you want to maintain white balance?:
1. Yes
2. No

Enter a number to choose an option: 1
White balance is maintained.
```

Figure 21: Error matrix & white balance choice for the CCM

- Depending on the user's choice, the corresponding algorithm will be executed. The color correction matrix is designed to minimize the ΔE_{ab} or ΔC_{ab} values between the original and corrected colors. It is common for color correction matrices to follow the constraint of the sum of each row equaling one. This constraint ensures that the color correction matrix maintains color balance and doesn't introduce any unintended color shifts during the correction process.
- After algorithm executes, the corrected red, green, and blue rows will be displayed on the console. Two type of matrices will be displayed as depicted in Figure 22, one is in floating-point notation and the other one is the integer type for make it hardware friendly.

```
White balance is maintained.

 -------------------------Algorithm is running--------------------------


 ********************************Result*********************************

Calculated Color Correction Matrix with following parameters:
1. Error matrix: Delta Eab
2. White balance: Maintained

 ------------------------------Integer CCM------------------------------

Corrected red   =  [1948, -762, -162]
Corrected green =  [-549, 1717, -144]
Corrected blue  =  [119, -907, 1812]

 -------------------------Floating-point CCM----------------------------

Corrected red   =  [ 1.9027 -0.7441 -0.1585]
Corrected green =  [-0.5366  1.6767 -0.1402]
Corrected blue  =  [ 0.1158 -0.8853  1.7695]

 ---------------------------Process completed---------------------------


Select a command:
1. Save config.yml with Calculated Color Correction Matrix
2. Restart the Color Correction Matrix Tool
3. Return to the Main Menu
4. Quit
```

Figure 22: Results display for the CCM module

Results Display/Saving

After displaying corrected CCM on the console, an end menu will appear, offering the following options:

- Save CCM to Config File: Selecting 1 will save the corrected CCM to a configuration file.
- Restart the CCM Tool: Choose option 2 to restart the color correction matrix calculation tool.
- Return to Main Menu: Pressing 3 will take user back to the main menu of the tuning tool.
- Quit the Program: Select option 4 to exit the program gracefully.

Gamma Analysis

The Gamma module in the Tuning Tool allows user to visualize the gamma curves.

Module's Input

For gamma module, the user-defined gamma curve is loaded from the configs.yml file and is compared with the usual gamma created at $\gamma = 2.2$. If the configs.yml file is missing, the user will be asked to load the file in order to proceed further.

When the Tuning Tool's main menu is displayed, press the number '3' to select the Gamma module. This will take user to the Gamma module menu as in Figure 23.

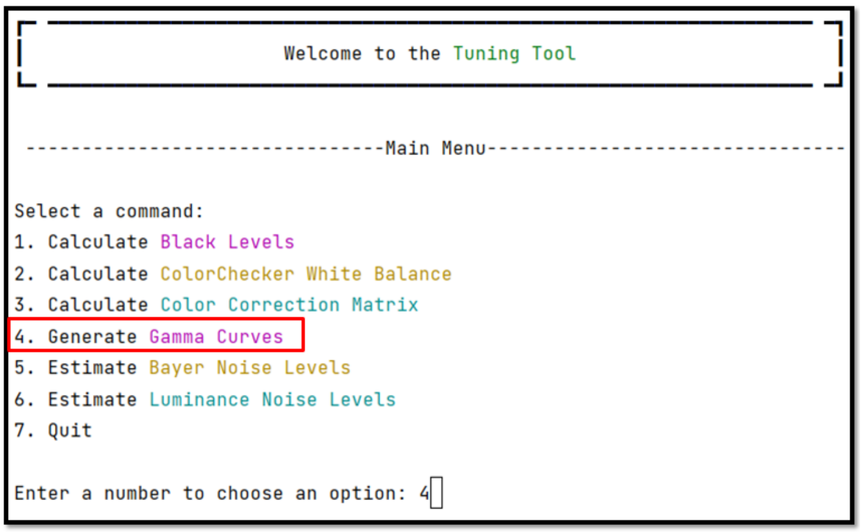

Figure 23: Selection of gamma curve module from the main menu

The Gamma module menu presents user with three options as:

- Press '1' to show gamma plots.
- Press '2' to return to the Tuning Tool main menu.
- Press '3' to quit the program.

Results Display/Saving

To visualize the gamma curves, press '1' in the Gamma module menu. This will display the plots for two types of gamma curves: user-defined and the common gamma curve generated $\gamma = 2.2$ that can be seen here in Figure 24.

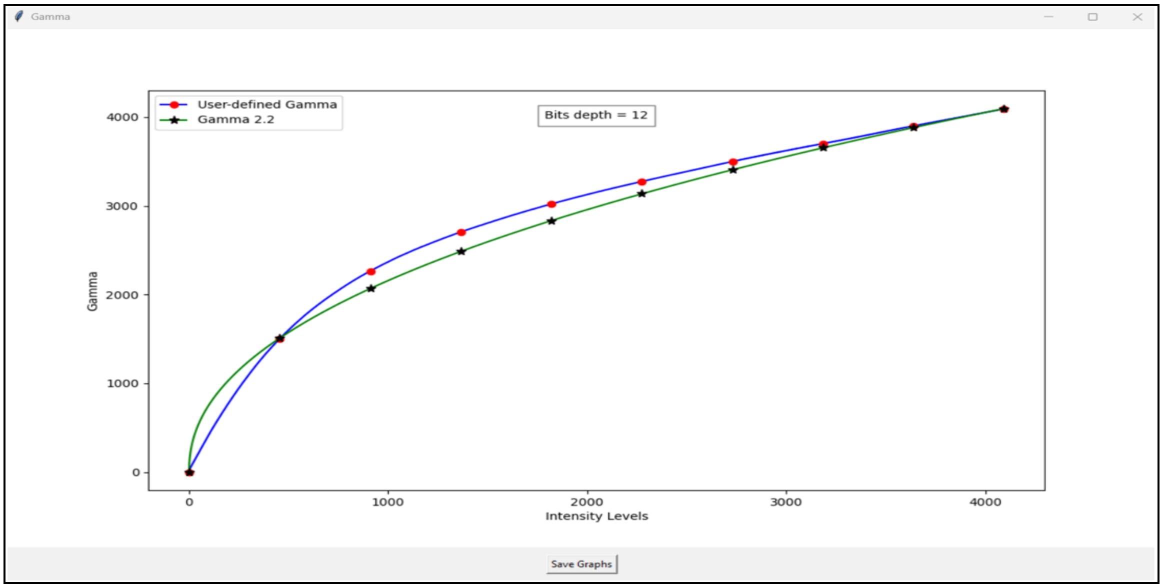

Figure 24: Results display for the gamma curves

On the panel displaying the gamma plots, user will find a save button. If user decide not to save the gamma plots, user have the option to cancel the save operation shown in Figure 25 and Figure 26.

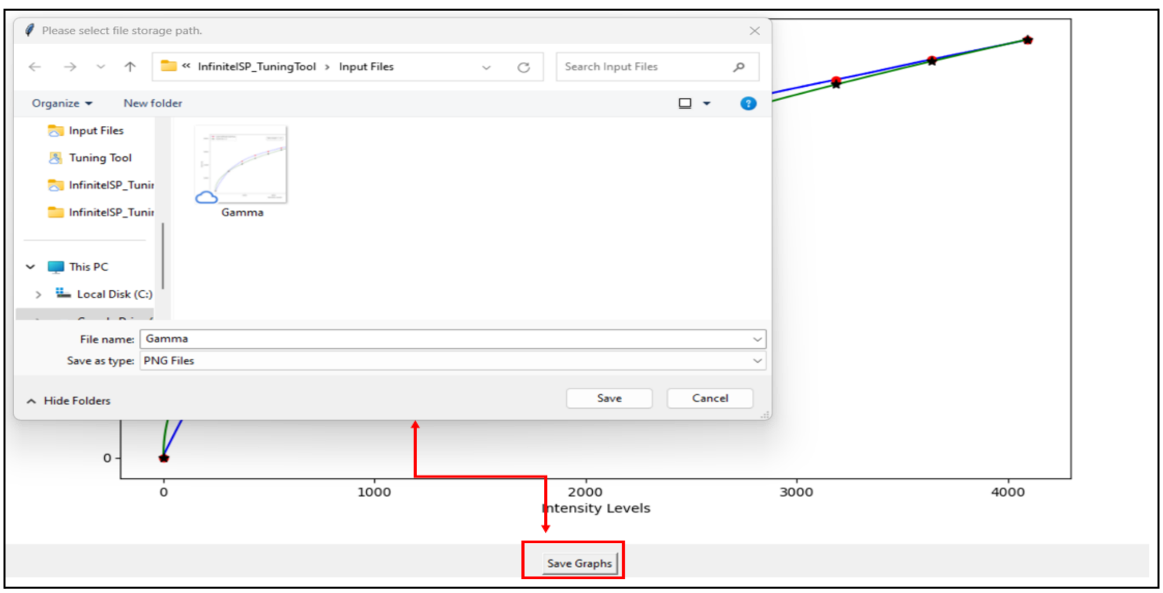

Figure 25: Saving gamma plots using save button on GUI

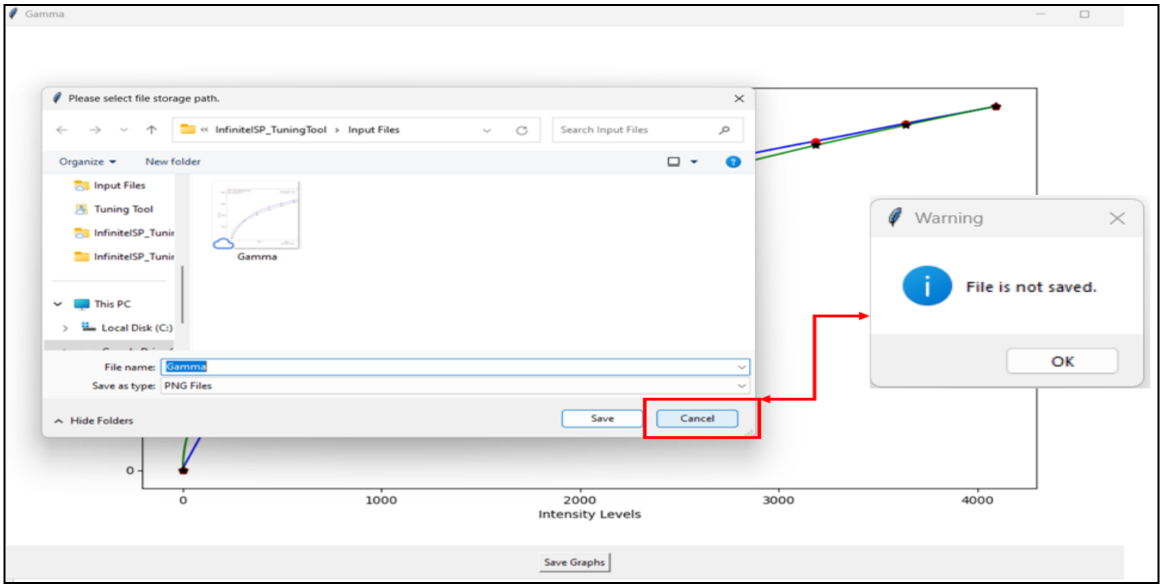

Figure 26: File saving or cancelling option for the user

Bayer Noise Estimation

The Bayer Noise Estimation module is designed to assist users in estimating noise levels in a Bayer pattern by analyzing raw images and locating a ColorChecker. The Bayer image, composed of the red (R), green (G), and blue (B) channels, is then split into these individual channels. By leveraging gray patches within each channel, the algorithm performs noise estimation. Since the gray patches ideally contain uniform color values, any deviations from uniformity can be attributed to noise.

By following the steps outlined in this guide, user can accurately measure the Bayer noise in the image and obtain valuable information for further tuning and processing.

Module's Input

To start the Bayer Noise Estimation module, follow these steps:

- Access the tuning tool main menu and choose 5 to select the Bayer Noise Estimation module as seen in Figure 27.

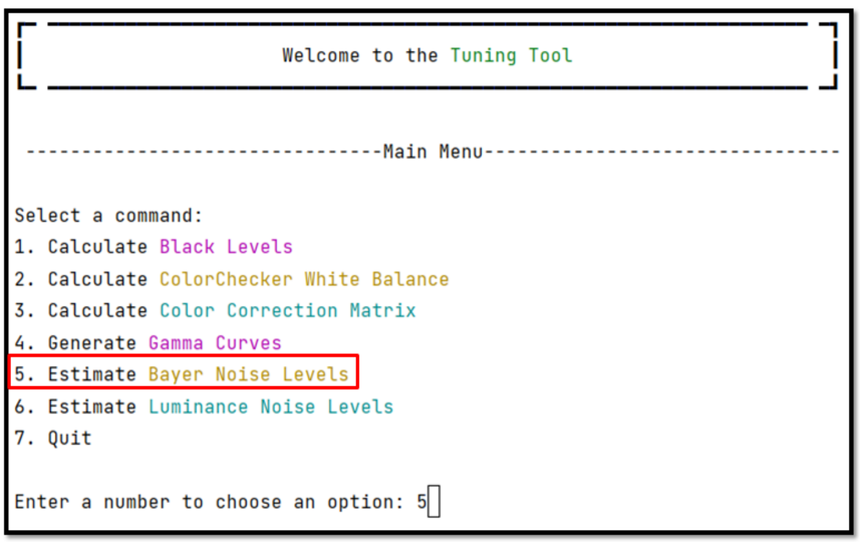

Figure 27: Selection of BNE from the main menu

- Load the raw image. Once the file is selected, another panel will open, that displays the input image.
- ColorChecker & Patches Selection: In the panel showing the RGB image, locate the ColorChecker and adjust it as described in previous sections.

Calibration Procedure

The algorithm is initiated when the patches from the ColorChecker image are chosen, triggering the estimation of noise levels in the bayer domain. The bayer image is then divided into three bayer channels: R, G (Gr and Gb), and B. By utilizing gray patches within these channels, the algorithm proceeds to calculate the standard deviation of each of the gray patch present in the ColorChecker.

Results Display/Saving

A GUI will appear, showing a table with the standard deviations of last 6 grey patches (as mentioned as 2 in figure 4) from each R, G (Gr & Gb), and B raw channels. This is clearly shown in Figure 28 below.

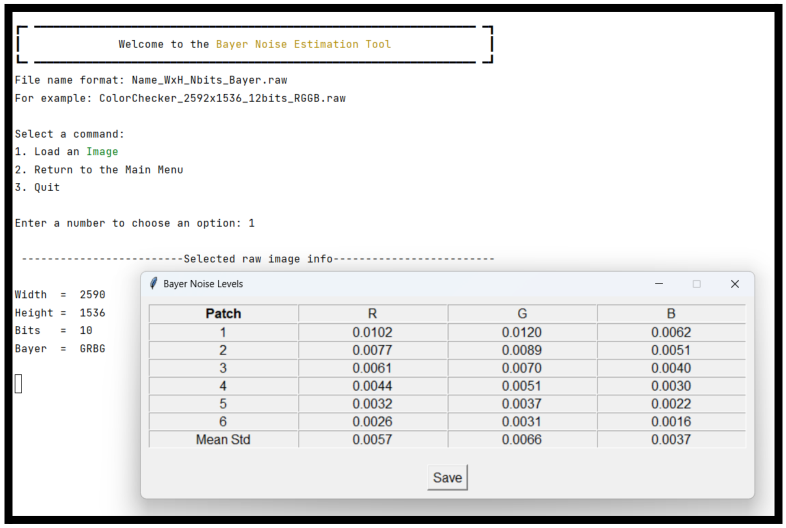

Figure 28: Results display for BNE module

If desired, click the "Save" button on the GUI to save the table data to a CSV file at his preferred location. Cancel to proceed further.

Luminance Noise Estimation

The Luminance Noise Estimation module is designed to assist users in estimating noise levels in an RGB image by locating a ColorChecker. The analysis involves evaluating the statistical characteristics of the pixel values within the gray patches. By examining the variances or standard deviations of these pixel values, the algorithm gains insights into the degree of noise affecting the luminance channel. By following the steps outlined in this guide, user can accurately measure the noise in the image and obtain valuable information for further tuning and processing.

Module's Input

To start the Luminance Noise Estimation module, follow these steps:

- Choose 6 to select the Luminance Noise Estimation module from the main menu as in Figure 29.

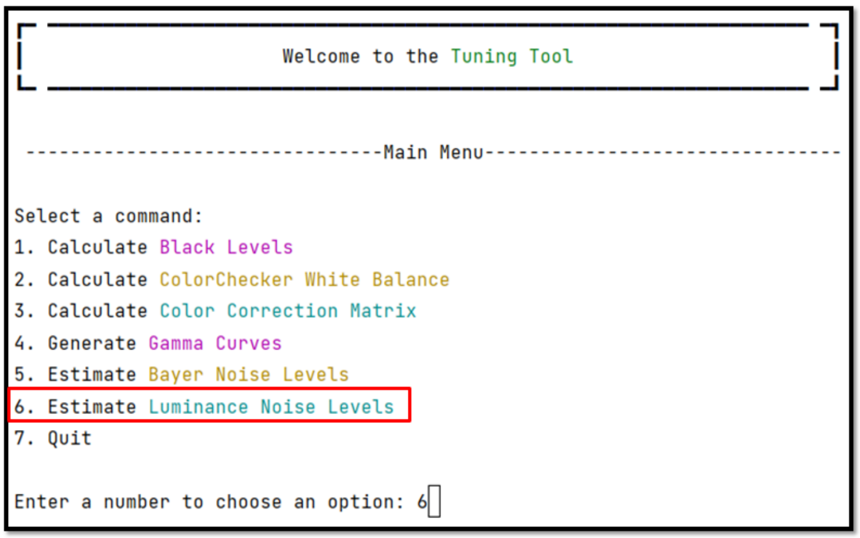

Figure 29: Selection of LNE module from the main menu

- Load the image, another panel will open, displaying the RGB. Note that if a raw image is selected it will be converted to RGB. This module gives user a friendly choice to rather select a raw image or an RGB image (.png, .jpg or .jpeg). The user will be asked if he wants to apply white balance on the selected image or not. Note that if the raw image is selected white balance is automatically applied.

Calibration Procedure

- ColorChecker & Patches Selection: In the panel showing the RGB image, locate the ColorChecker and adjust it as described in previous sections.
- Apply White Balance: After selection of patches, user will be asked if user want to apply white balance on the selected image as shown in Figure 30. Press either 1 or 2 to choose between Yes and No. If raw image is selected, white balance is automatically applied.

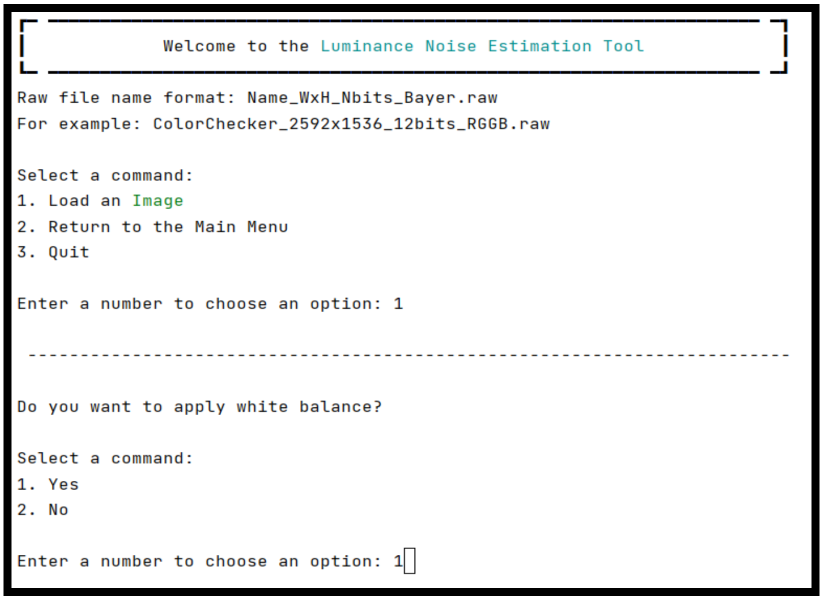

Figure 30: White balance choice for LNE module

Results Display/Saving

A GUI will appear, showing a table with the standard deviations of last 6 grey patches from the luminance channel of the image along with the patches images as shown in Figure 31.

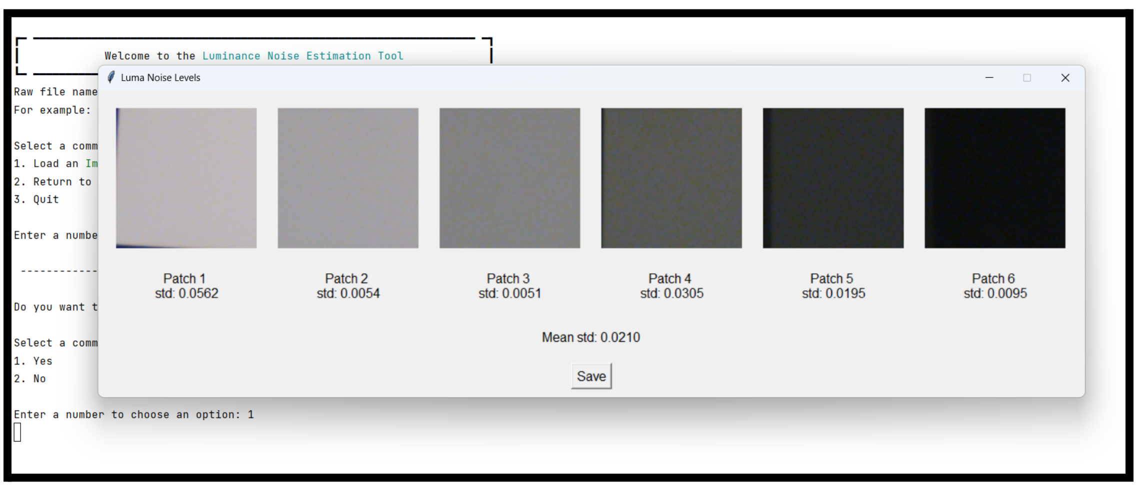

Figure 31: Results display for the LNE module

If desired, click the "Save" button on the GUI to save the table data to a CSV file at his preferred location. Cancel to proceed further.