SYNCELL

AUTOSCOOP™

Software for MICROSCOOP® / MINT

User Manual

November 2025 | Version C

LB02.C.0001

Content

Made in Taiwan.

1. INSTALLATION AND SETUP

Microscoop is pre-installed with Autoscoop software; therefore, no initial installation is required. However, when a new version of the software becomes available, it is recommended to immediately install the update.

This chapter provides instructions for updating the Autoscoop software.

1.1 Autoscoop Launcher

Autoscoop Launcher allows you to manage and upgrade the Autoscoop software. Follow the steps below to perform a software upgrade.

Step 1: Open Autoscoop Launcher

- Right click “Autoscoop Launcher” icon on windows taskbar.
- Click open to launch “Autoscoop Launcher” user interface.
- If “Autoscoop Launcher” is not running, you can find it in Start Menu and start it manually.

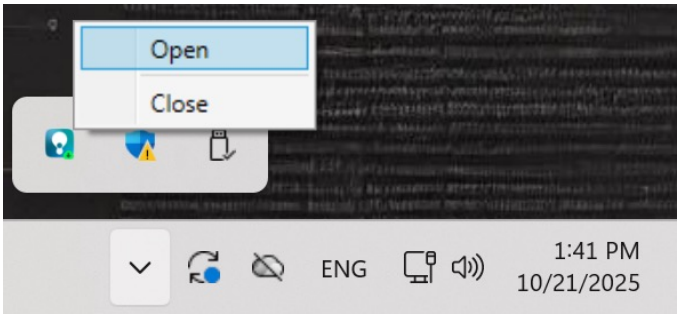

Fig. 1- 1 Open Autoscoop Launcher

Step 2: Select software and install

- Click the checkbox next to the software you want to upgrade.
- Click Install. The new version of software will be downloaded and installed automatically. The upgrade process may take a few minutes to complete.
- To view the release notes, please click the file icon to the right of the version number.

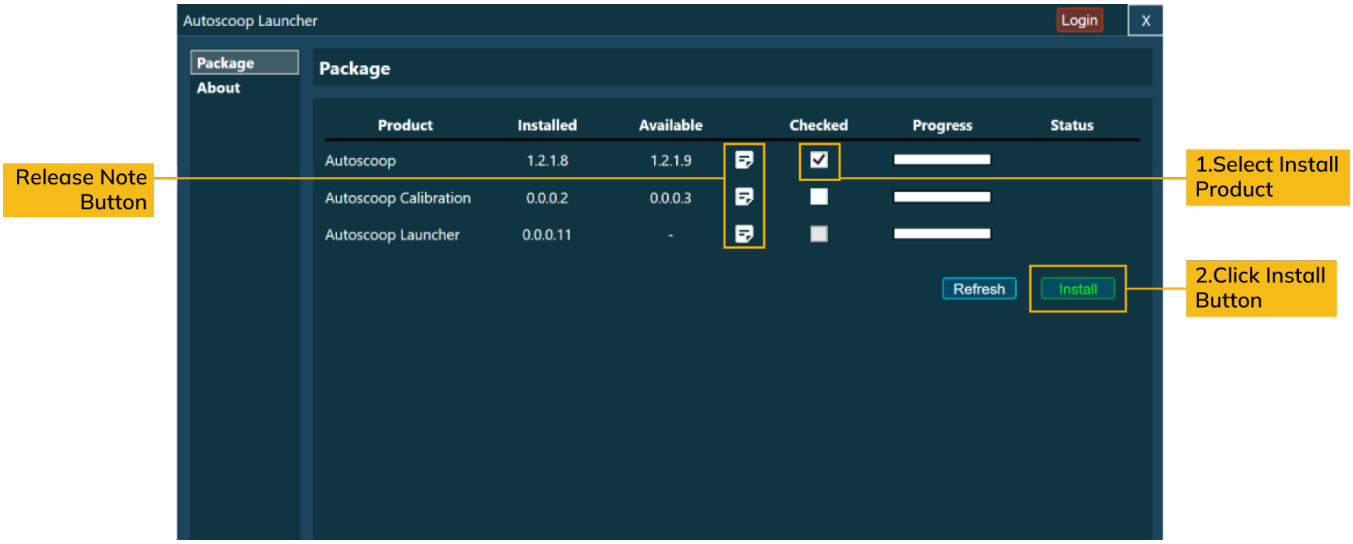

Fig. 1- 2 Installation window

Step 3: Installation Complete

- When the progress bar reaches 100% and displays “success”, the installation is complete.
- You can now start using the updated version of the Autoscoop software.

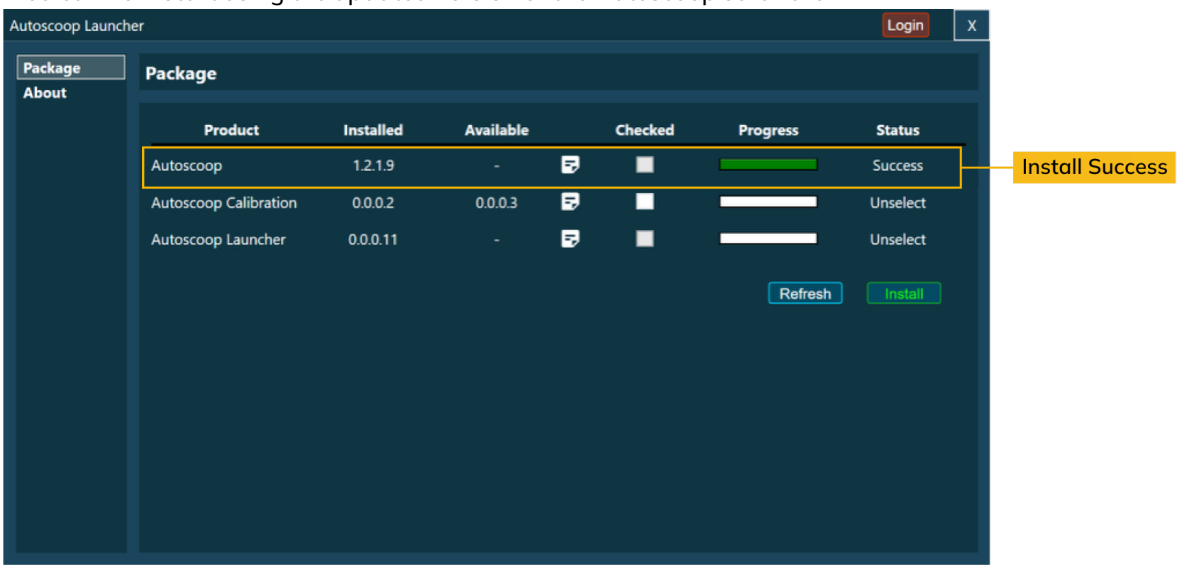

Fig. 1- 3 Installation Success

1.2 Installation Path

Fig. 1- 4 show the installation path when the software is installed correctly. Desktop shortcuts and start menu shortcuts will be generated automatically.

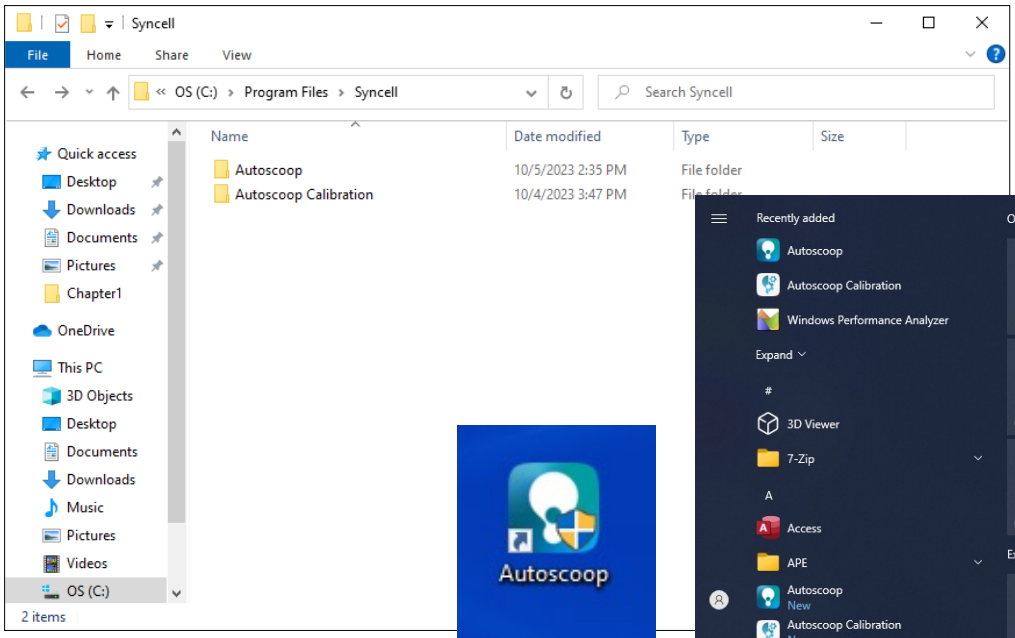

Fig. 1- 4 Installation path

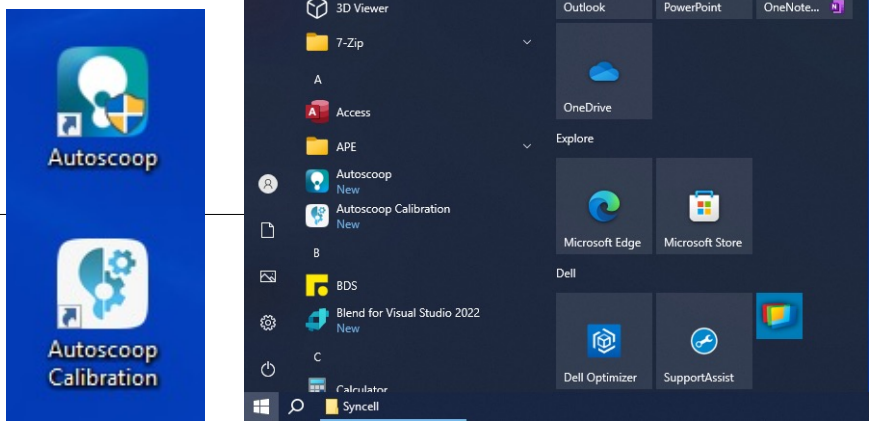

Fig. 1- 5 Desktop shortcuts and start menu shortcuts

1.3 Version Check

The version number will be shown on the initialization window when starting Autoscoop and Autoscoop Calibration software.

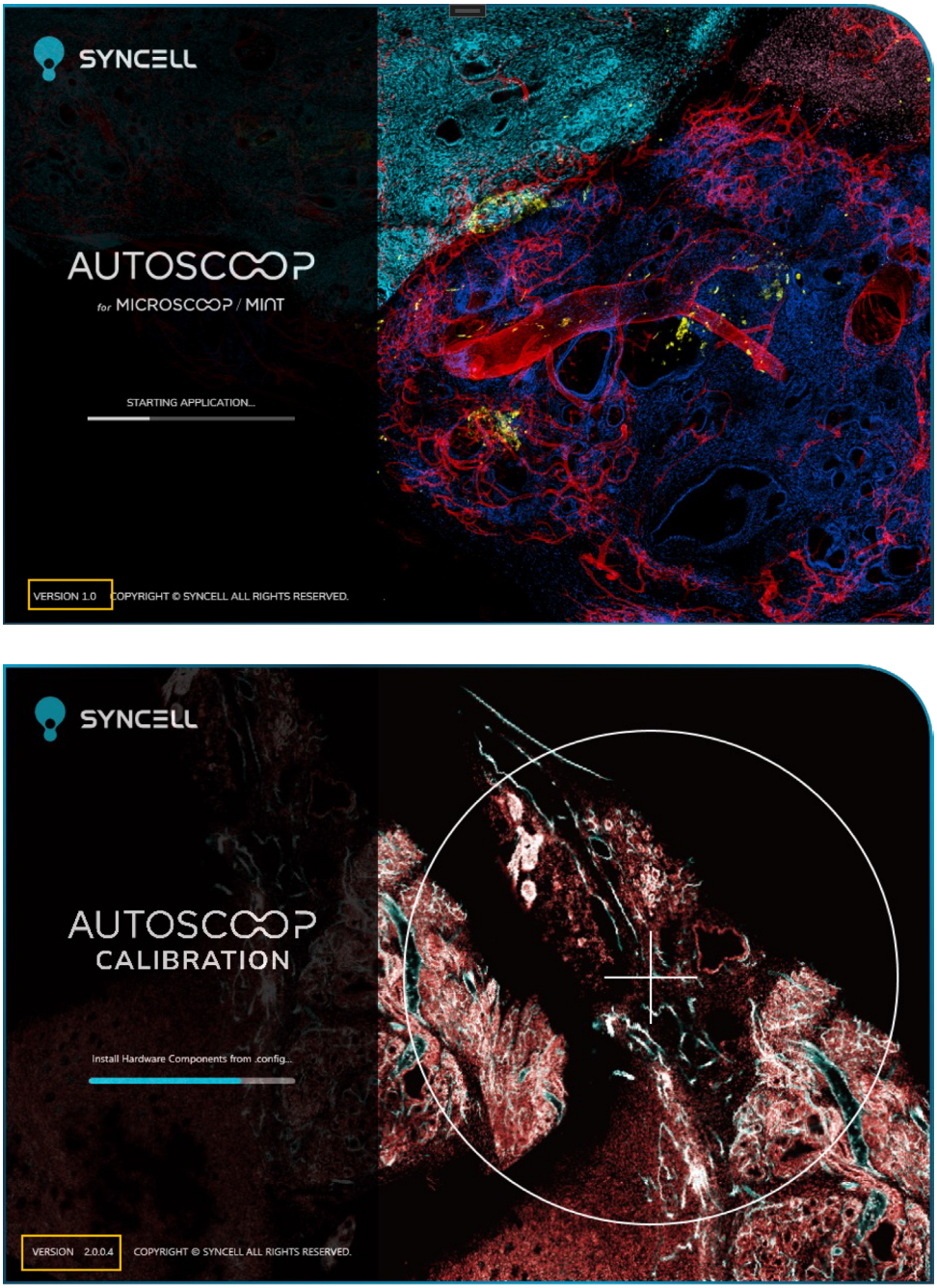

Fig. 1- 6 Initialization window

2. USER INTERFACE

2.1 Imaging Page

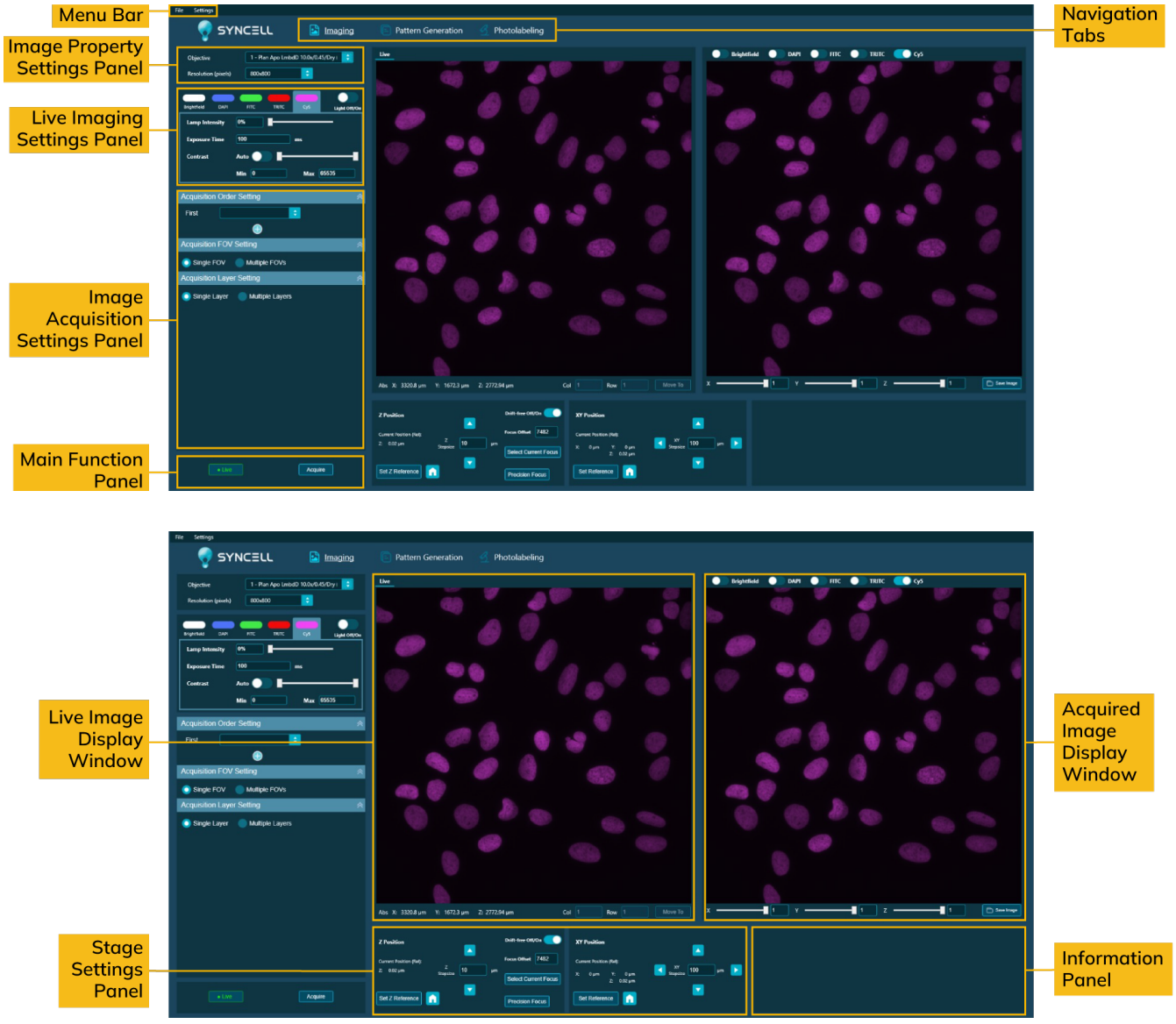

Fig. 2- 1 Imaging page

- Menu Bar: The menu includes image save and load, program exit, and some settings.
- Image Property Settings Panel: Settings for objectives and image resolution.
- Live Image Settings Panel: Settings for real time live-image visual quality enhancement.
- Image Acquisition Settings Panel: Settings for image acquisition. Support for multiple FOVs acquisition and multi-layer acquisition.
- Main Function Panel: Controls for real-time imaging and image acquisition.
- Navigation Tabs: Switch between pages by selecting a tab.
- Live Image Display Window: Displays the image captured from the camera in real time.
- Stage Settings Panel: Panel for controlling the stage of the microscope.
- Acquisition Image Display Window: Display the images which are acquired from settings.
- Information Panel: (reserve).

2.2 Pattern Generation Page

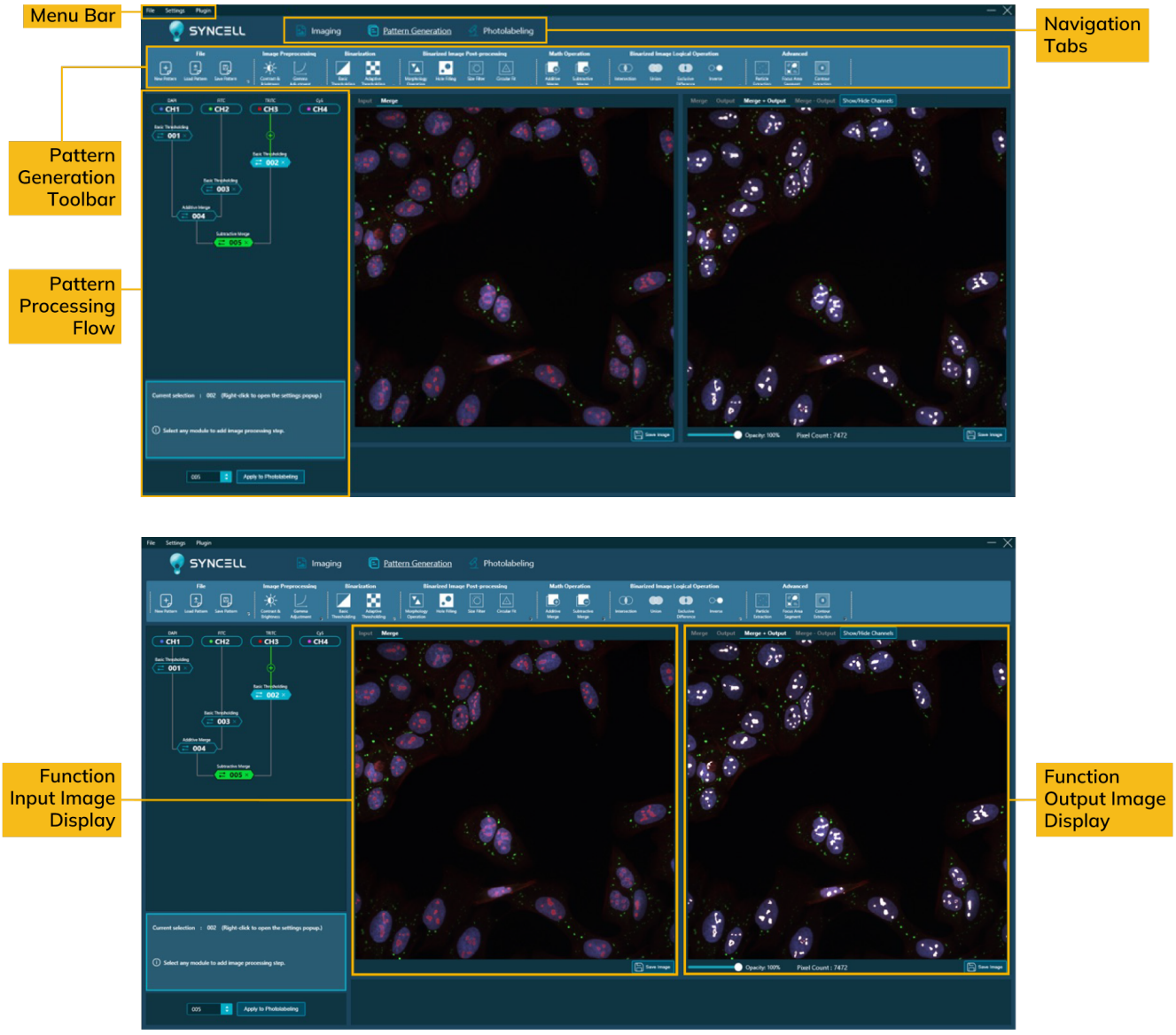

Fig. 2- 2 Pattern generation page

- Menu Bar: The menu includes image save and load, program exit, and some settings.
- Pattern Generation Toolbar: Selection toolbar for image processing functions.
- Pattern Processing Flow: Flow chart for showing the image processing function flow and adjusting the parameters for each function.
- Navigation Tabs: Switch between pages by selecting a tab.
- Function Input Image Display: Image processing input display (Dependent on step selection)
- Function Output Image Display: Image display for the output image of the image processing function. It will according to which function is focused. Image processing output display (Dependent on step selection)

2.3 Photolabeling Page

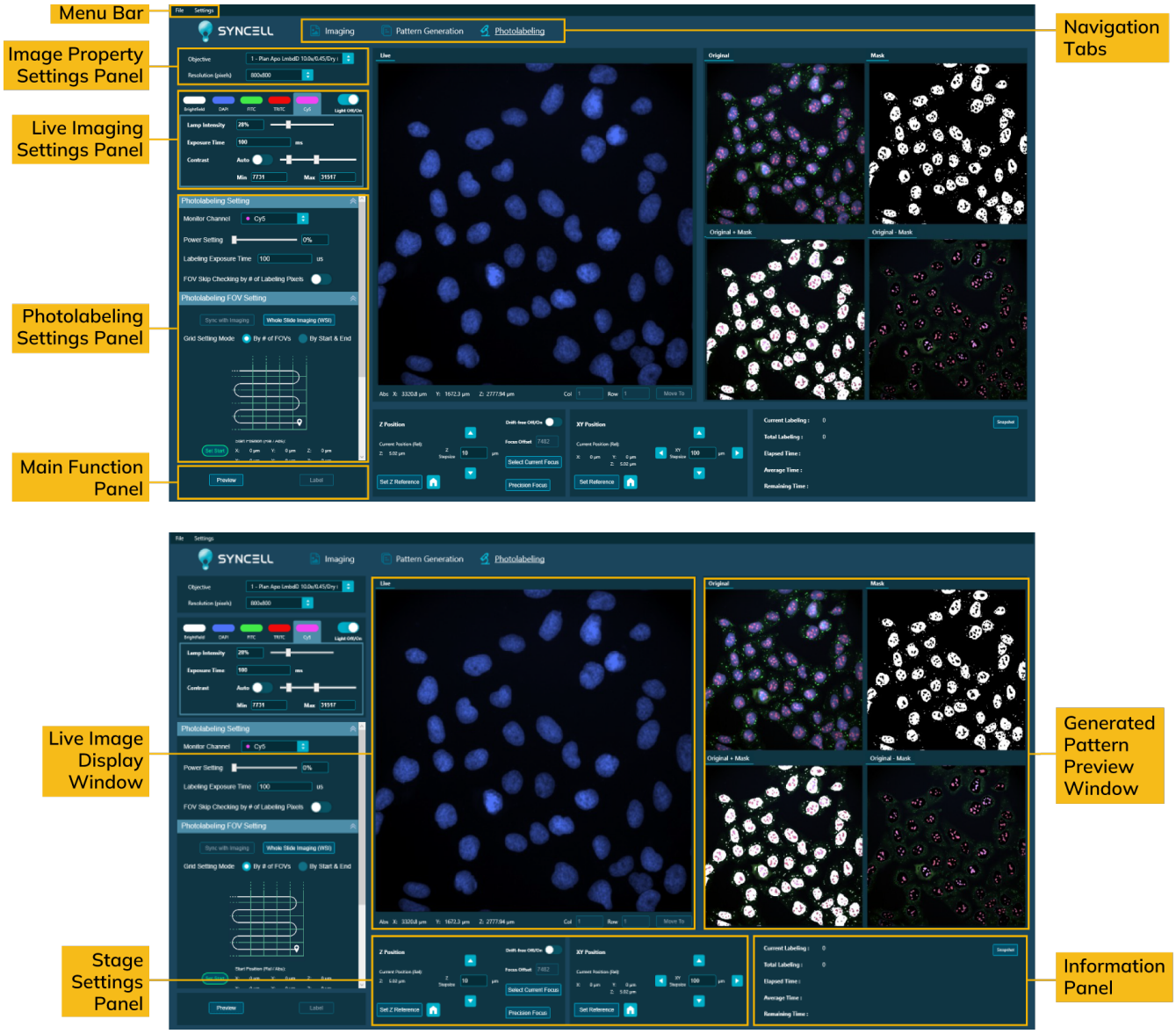

Fig. 2- 3 Photolabeling page

- Menu Bar: The menu includes image save and load, program exit, and some settings.
- Image Property Settings Panel: Settings for objectives and image resolution.
- Live Image Settings Panel: Settings for enhancing the visual quality of live images in real-time.
- Photolabeling Settings Panel: Settings for photolabeling including laser settings, focus settings, and FOV settings.
- Main Function Panel: Pattern generation preview and start photolabeling controls.
- Navigation Tabs: Switch between pages by selecting a tab.
- Live Image Display Window: Display the image captured from the camera in real time.
- Stage Settings Panel: Panel for controlling the stage of the microscope.
- Generated Pattern Preview Window: Display the images acquired from the camera and the pattern process by pattern generation.
- Information Panel: some information for photolabeling, including how many pixels would be photolabeling, estimated time for photolabeling... etc.

3. SYSTEM SETTING

3.1 System Configuration

Before starting the photolabeling task, the Autoscoop software requires parameters check for system configuration. In the current version, the user may need to do the following tasks:

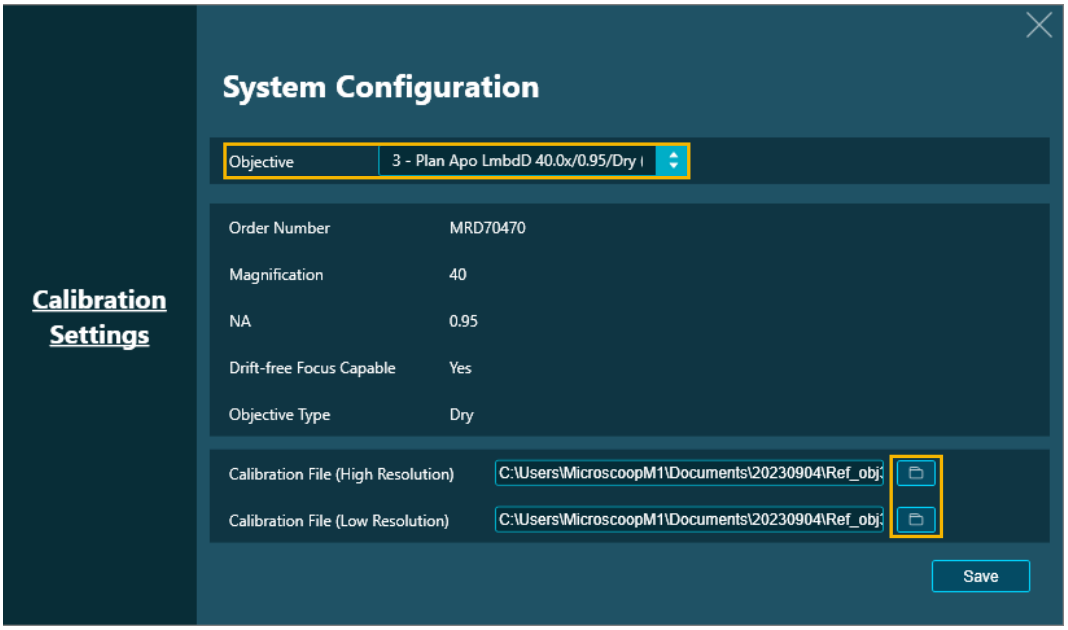

Fig. 3- 1 System Configuration popup

Step 1: Select “Setting” > “System configuration”

Step 2: Choose objective

- Please select the objective which will be used for photolabeling.

Step 3: Load calibration file

- Click the “file” button to choose the calibration file.
- Each resolution needs to load a specific calibration file. Autoscoop software will check after choosing.

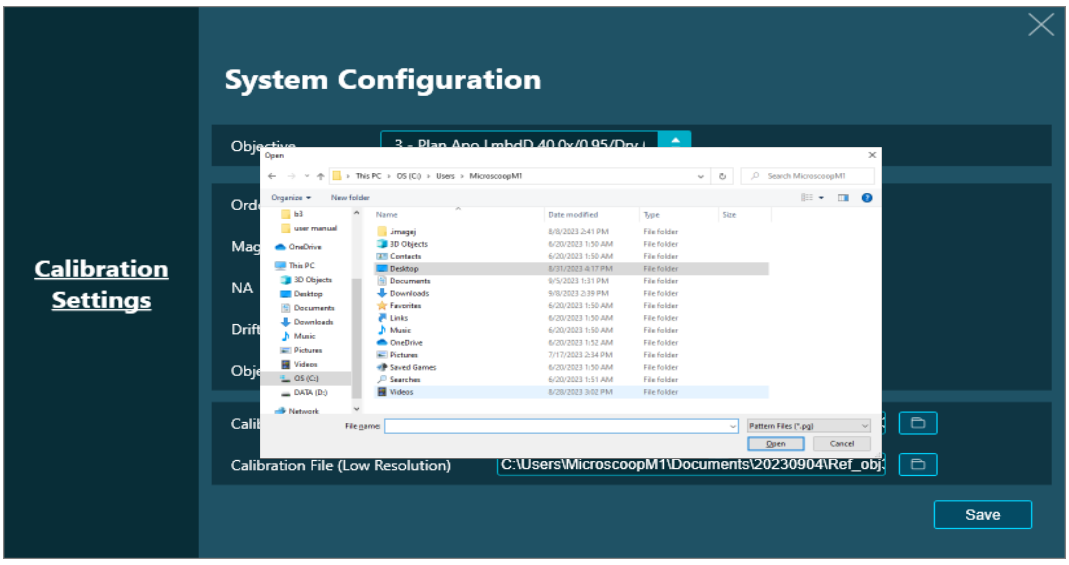

Fig. 3- 2 Calibration file loading popup

3.2 Advanced Setting

Autoscoop automatically loads recommended Advanced Settings for photolabeling during installation. These parameters were optimized and verified during factory testing and product delivery to ensure stable performance. Changing them may affect labeling accuracy, image quality, or system stability. Do not modify these settings unless directed by Syncell Support or an authorized service engineer.

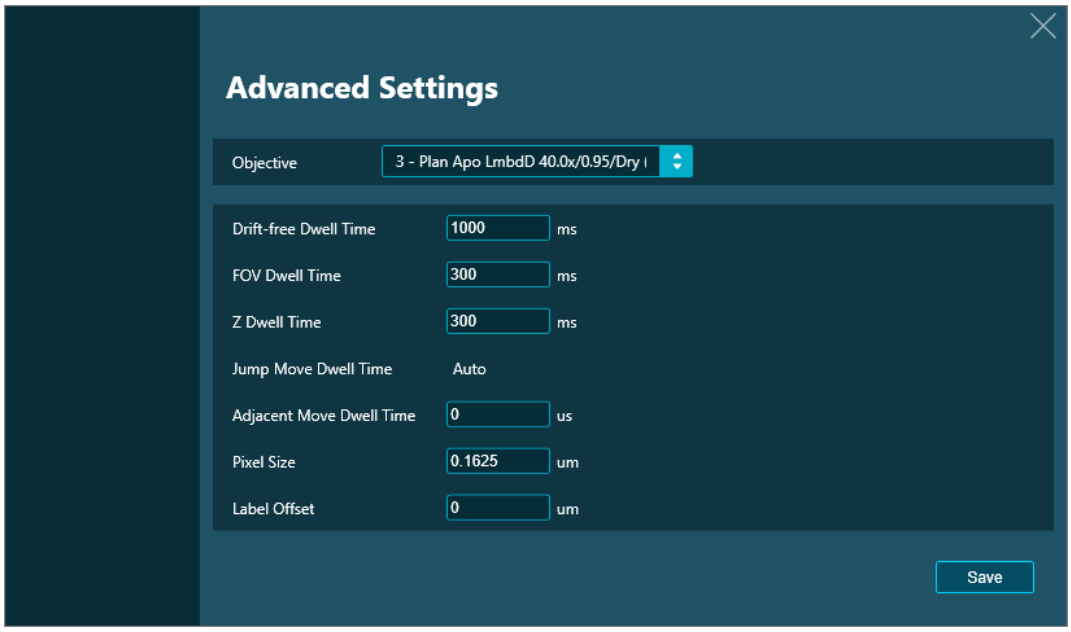

Fig. 3- 3 Advanced Setting popup

4. WORKFLOW

Autoscoop is a user-friendly software solution for the control of the Microscoop platform for your spatial proteomic research. It enables precise image-guided photolabeling at your region of interest (ROI) using the Synlight-Rich™ kit. This software consists of three sections: Imaging, Pattern Generation, and Photolabeling. In the first step, the imaging parameters are adjusted on the [Imaging] page to acquire a clear and sharp image of your region of interest (ROI). Next, a mask pattern is defined in the [Pattern Generation] page for the image of your ROI. Finally, in the [Photolabeling] page, the photolabeling parameters to define the region (i.e., the number of fields of view, FOV) to be labeled are set. The entire process enables the automatic and precise photolabeling of sufficient ROIs in thousands of FOVs. The resultant samples are ready for protein enrichment with the Synpull™ kit, followed by LC-MS/MS analysis to elucidate the target proteome.

5. IMAGING

The "Imaging" page serves as a central hub for a range of essential functions in microscopy. It provides a real-time display of the live image captured by the microscope camera, thereby allowing users to observe their specimens with precision and detail. Additionally, this page allows the users to control the LED light intensity and exposure time, thus ensuring optimal image quality and clarity for various samples and conditions.

5.1 Choose Objective

- Choose an objective for imaging: 10x, 20x, 40x.

The FOV side length will automatically switch according to the pixel size of different objective lenses.

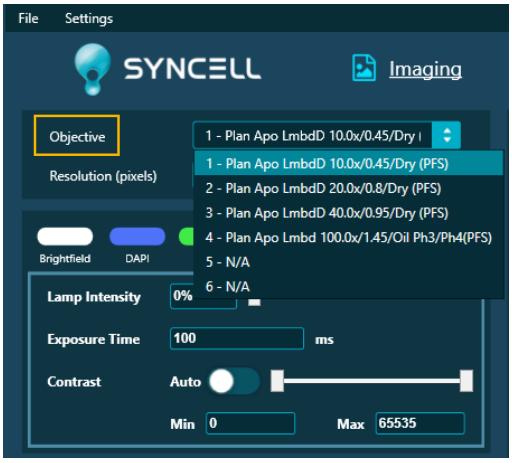

Fig. 5- 1 Objective selection

5.2 Choose Resolution

Choose an image resolution: 800x800 (Low resolution) or 1600x1600 (High resolution). In general, the resolution selection depends on the size of your ROI. High-resolution mode is recommended for ROIs smaller than 750nm in diameter.

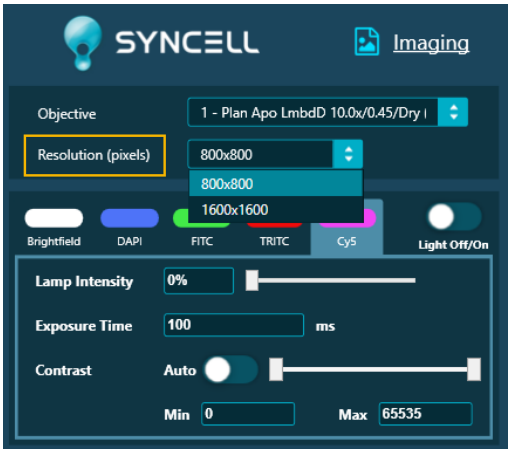

Fig. 5- 2 Resolution selection

5.3 Set Light Channel

- Switch off/on the light and select a channel to activate.

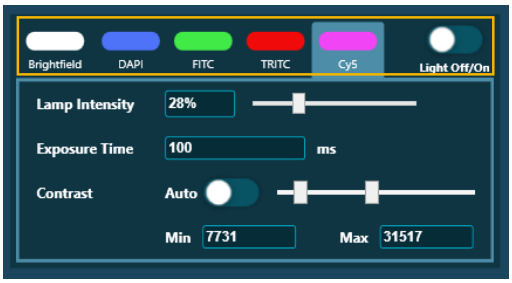

Fig. 5- 3 Channel selection and light switch

- Set lamp intensity (%) and camera exposure time (ms) for each channel.

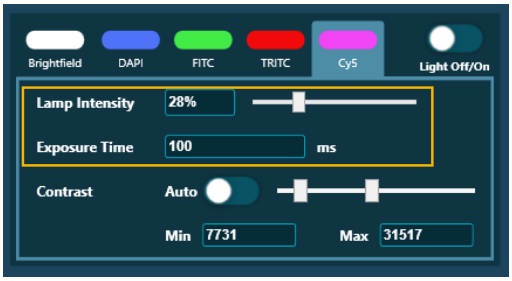

Fig. 5- 4 Light intensity and camera exposure time

- Set contrast (0-65535)
  - o Auto: simply click the “Auto” function to check your image with auto contrast.

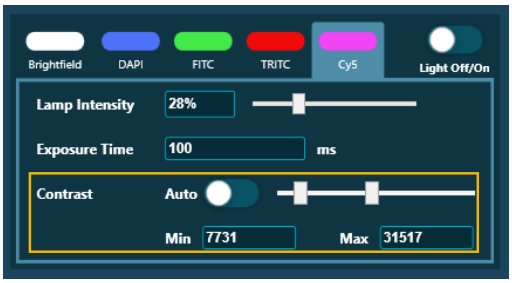

Fig. 5- 5 Contrast (LUT) setting

5.4 Acquisition Channel Setting

Set the channel for image acquisition. If no selection is made, the default setting of the current Live channel (single color) will be active. If multiple channels are selected, image acquisition will be performed according to the setting order (multi-color).

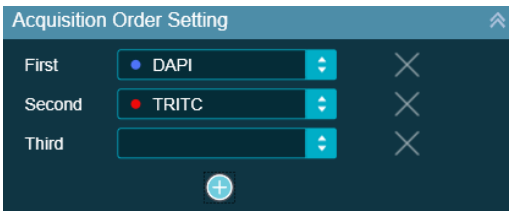

Fig. 5- 6 Acquisition channel setting

5.5 Acquisition FOV (Field of View) Setting

- To acquire a single FOV image on the live display window, choose the “Single FOV” function.

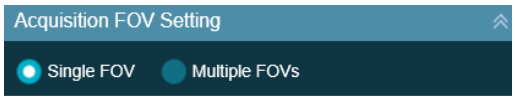

Fig. 5- 7 Acquisition FOV setting – single FOV

- To acquire a set of images, choose the “Multiple FOVs” function.
  - o Multiple FOVs by Columns & Rows: set the number of FOVs to be acquired by specifying the number of columns and rows. As shown in Fig. 5-8 Acquisition FOV setting – multiple FOVs set by Columns & Rows, 3 columns and 3 rows are selected, resulting in 3×3 FOVs to be acquired.
  - o Set Start: click this button to define a start point (location) for the acquisition.
    The start location refers to the absolute coordinates encompassing x, y, and z axes. It is highly recommended to find the focal plane before you click 'Set Start' button. If you are using Drift-free or Autofocus + Drift-free (see Chapter 5.7) to acquire images. Please set drift-free settings before you click 'Set Start' button.

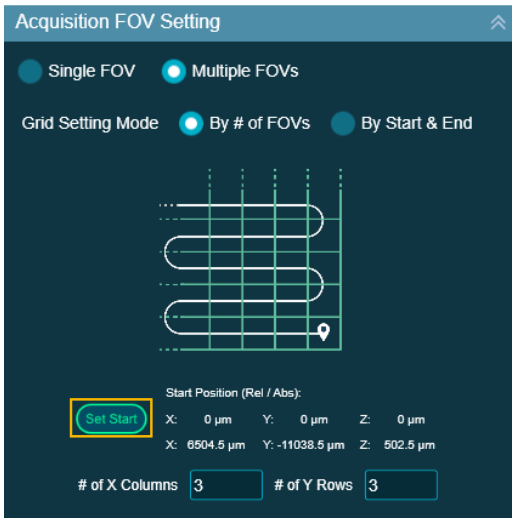

Fig. 5- 11 Acquisition Layer setting – multiple layers

  - o Set Start & End FOV: set a start and end FOV for the acquisition. This setting automatically determines the column and row numbers of FOV. The side length of the FOV varies depending on the objective lens. For example, the side length under the 40X objective lens is approximately 256 µm.

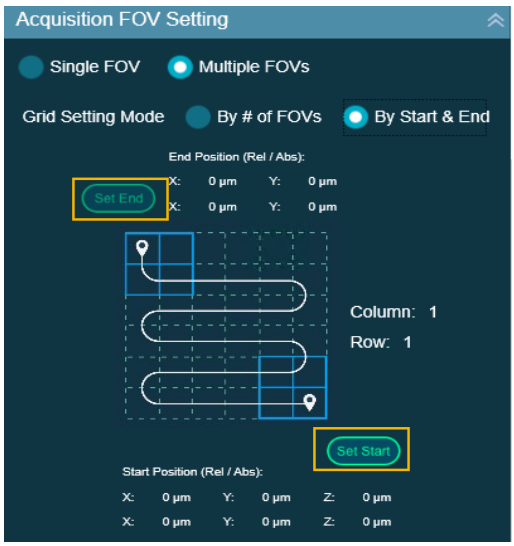

Fig. 5- 9 Acquisition FOV setting – multiple FOVs set by start and end

5.6 Acquisition Layer Setting

- Select the “Single Layer” function to acquire a single image at the specified focal plane on the live display window.

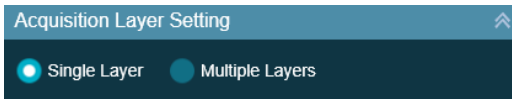

Fig. 5- 10 Acquisition Layer setting – single layer

- Select “Multiple Layers” to acquire a set of images with different z positions.
  - Layers: set the number of layers for multilayer acquisition.
  - Z Pitch: set the interval of z-stack image. In this case, 21 layers of the z-stack images with 0.5 µm interval (Z Pitch) will be imaged, resulting in a 10-µm-thick image stack, starting from -5 µm and ending at +5 µm offset (compared to the relative origin axis).

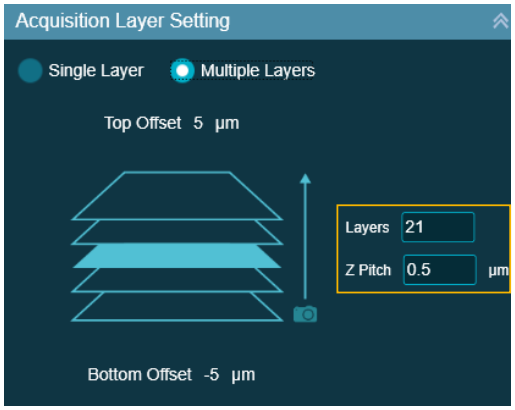

Fig. 5- 11 Acquisition Layer setting – multiple layers

5.7 Focus Setting

Focus Setting is only displayed while using the multiple layers acquisition setting. It has 3 options:

- None: image acquisition without any focus mode
- Drift-free: employs the microscope’s optical focus stabilization system to continuously maintain the correct focal plane during acquisition, ensuring stable and drift-free imaging across all FOVs.

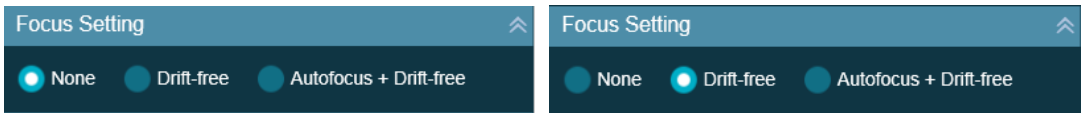

Fig. 5- 12 Focus setting “None” and “Drift-free”

- Autofocus + Drift-free: utilizes Drift-free to establish the initial Z position for each FOV, then capture multilayer images and determine the optimal focus position.
  - Layers and Z Pitch: set parameters for multilayer acquisition.
  - Channel: choose a channel to determine the image focus.
  - 16-bit Raw Data: Focus using 16-bit raw images or use contrast-adjusted 8-bit images for focusing (please refer to section 5.3 Set contrast).

  The Autofocus function determines the focal plane by capturing a Z-stack of images and applying an algorithm to locate the optimal focus. If the computed focal plane falls near or beyond the upper or lower boundary of the captured range, it may be excluded by the system to prevent unreliable focus detection. This exclusion is designed to avoid incorrect focusing or photolabeling on abnormal or out-of-focus FOVs.

  To reduce the risk of such misjudgment, we recommend extending the Z-stack range by adding 4 to 6 extra layers beyond the estimated sample thickness. For example, with a sample thickness of approximately 5 micrometers and a Z pitch of 0.5 micrometer, one might initially calculate 11 layers.

However, we suggest setting it to 15–17 layers to ensure the actual focal plane is fully captured within the range and not mistaken as being outside of it.

Failing to extend the layer range may lead to FOVs being skipped or misclassified as out of focus, even if they contain usable sample areas. This can impact both the success rate of autofocus and the accuracy of photolabeling.

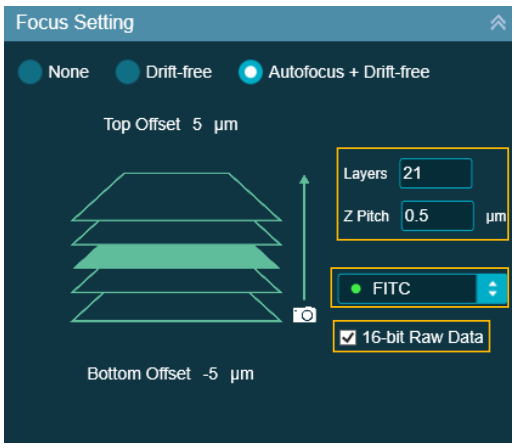

Fig. 5- 13 Focus setting “Autofocus + Drift-free”

5.8 Live and Acquire

- Live: turn on “Live” to view the current image
- Acquire: click “acquire” to capture the image using the settings mentioned above

Fig. 5- 14 Live and acquire button

5.9 Display Window

- Live Image Display Window (Left)
  - o Abs XYZ (Absolute XYZ position): The numbers represent the absolute XYZ position of the current FOV.
  - o Move To: move from the current FOV to another FOV by specifying the column and row number. It is only enabled if the user sets the starting point within the multiple FOVs setting.

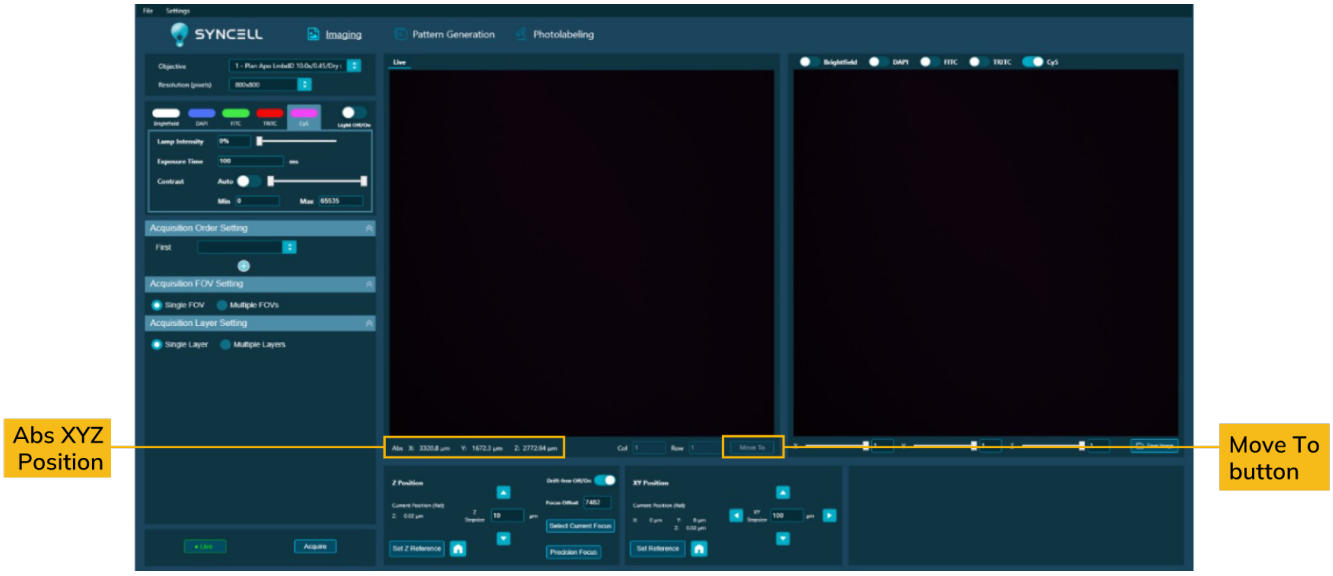

Fig. 5- 15 Live Image Display window (left)

- Acquired Image Display Window (Right)
  - o Channel display switch: view the image using current channel selection
  - o Display controller: view different FOVs or layers in an image with XYZ directional sliders.
  - o Save image: click to save the image set into the specified folder.

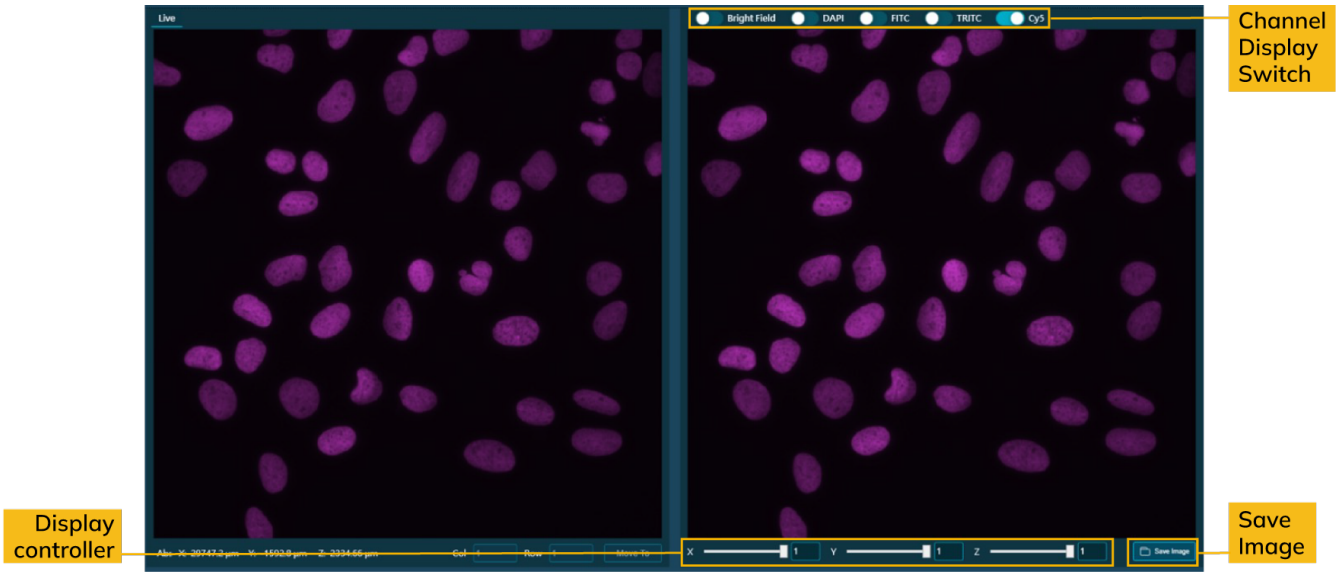

Fig. 5- 16 Acquired Image Display window (right)

5.10 Z Position Control

- Sets the Z position on the image. The Drift-free optical focus stabilization function can be enabled through precision focus mode. It can also be directly enabled using the switch and adjusting the focus offset.
- Drift-free Lock: apply current focus offset on the image by opening the Drift-free lock. Focus offset can only be changed by the Z controller when the Drift-free lock is open.
- Select Current Focus: set the focus offset to the current position on the image.
- Set Z Reference: set the current Z-position as the original position and click to go back to the origin Z position.

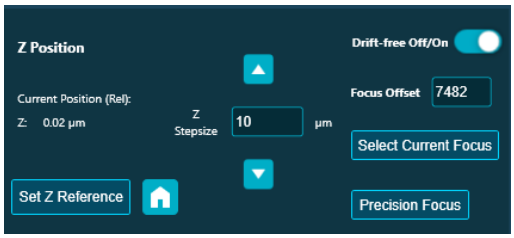

Fig. 5- 17 Stage control - Z Position control

5.10.1 Precision Focus

Precision Focus determines a Z-focal plane by acquiring a Z-stack image with input numbers of layers and a Z-pitch setting.

Step 1: Setting channel, layers, and z pitch for multilayer acquisition.

- It is recommended to select “16-bit Raw Data” to use raw images for focusing. Then select ”Next” button to acquire images Wait for all images to finish acquisition.

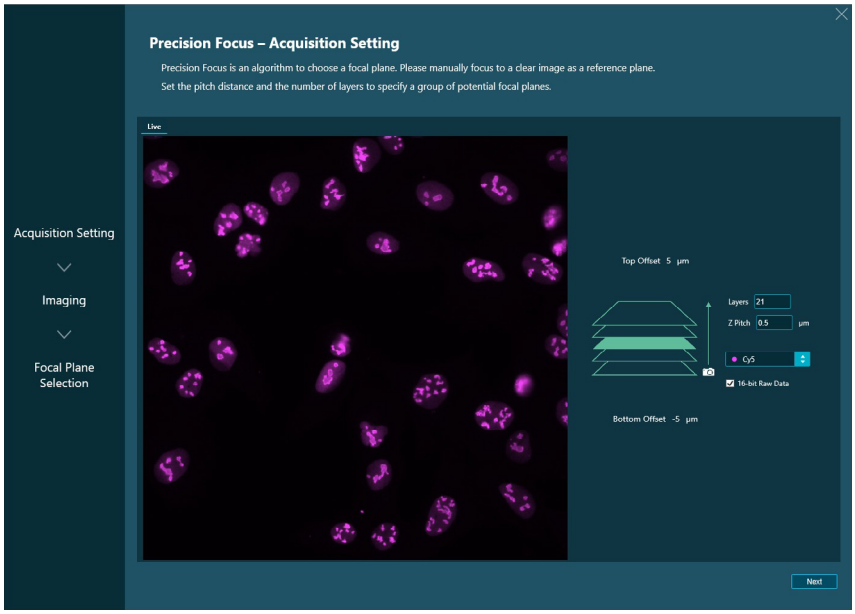

Fig. 5- 18 Precision Focus –Acquisition setting

Step2: Choosing the focal plane to apply Live Image Display Window.

- Auto Focus: automatically chooses a focus plane within the Z-stack image.
- Z-layer controller: manually choose a focal plane by observing the Z-stack image using up/down buttons.
- Apply: set a focus plane on Live Image Display Window after auto-focus or manual focus.

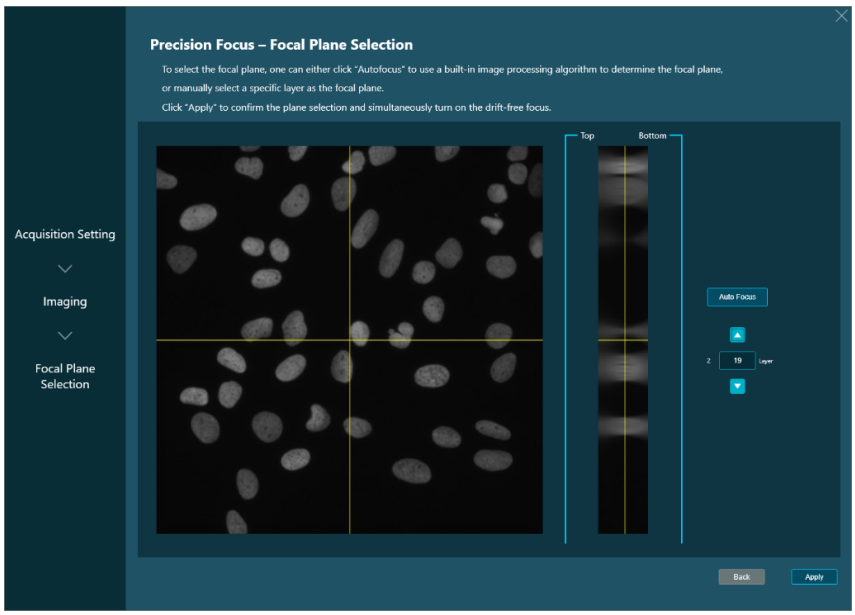

Fig. 5- 19 Precision Focus – Focal plane selection

5.11 XY Position

XY Position is to set up XY position on the image using XY controller.

- Set Reference: set the current XYZ position as the original position and click to go back to the original XYZ position.

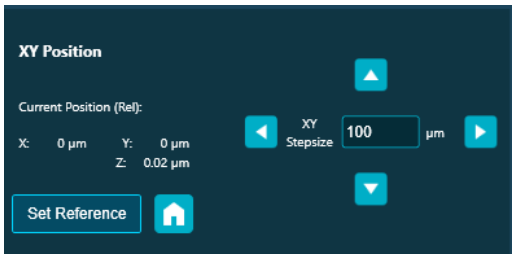

Fig. 5- 20 Stage control - XY Position Control

6. PATTERN GENERATION

The "Pattern Generation" page offers a list of functions for target extraction. Users can harness a variety of image processing functions, including image preprocessing, binarization, and post-processing of binary images. User will also have access to advanced functions and integration with AI modules. This flexibility empowers users to craft tailored combinations of these functions to create pattern masks suitable for photolabeling.

6.1 Pattern File

- New Pattern: create a new pattern file with the module combination for your region of interest.
- Load Pattern: load a saved pattern file to generate a pattern for your current image source.
- Save Pattern: save the current module combination as a pattern file to an assigned file path.

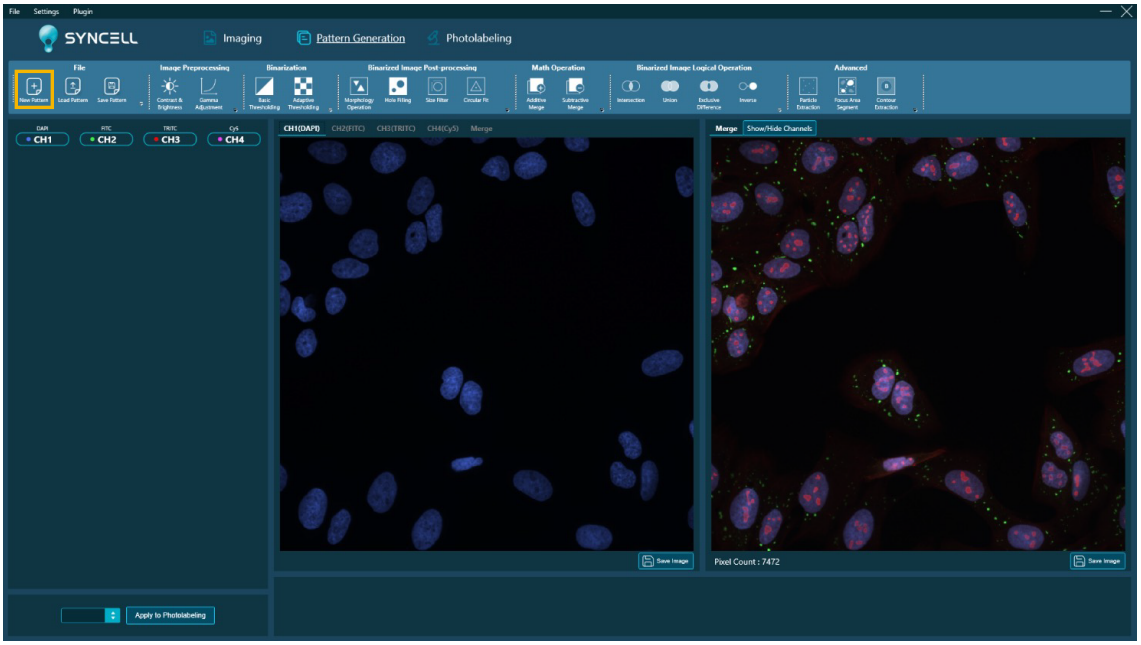

Fig. 6- 1 Pattern file – new pattern

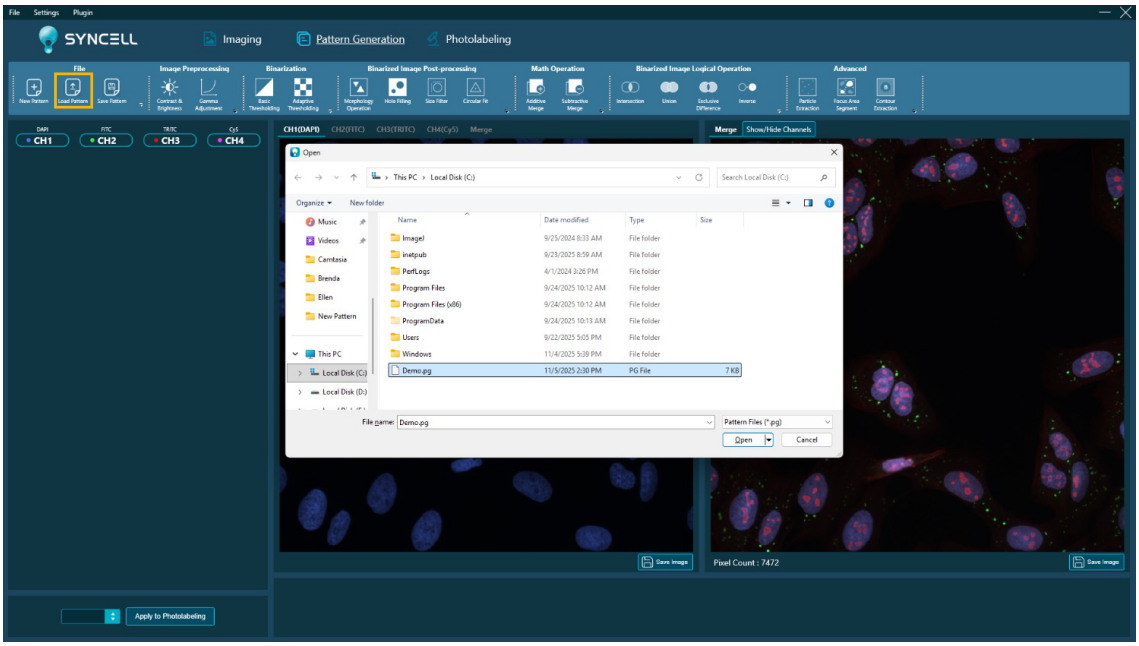

Fig. 6- 2 Pattern file – load pattern

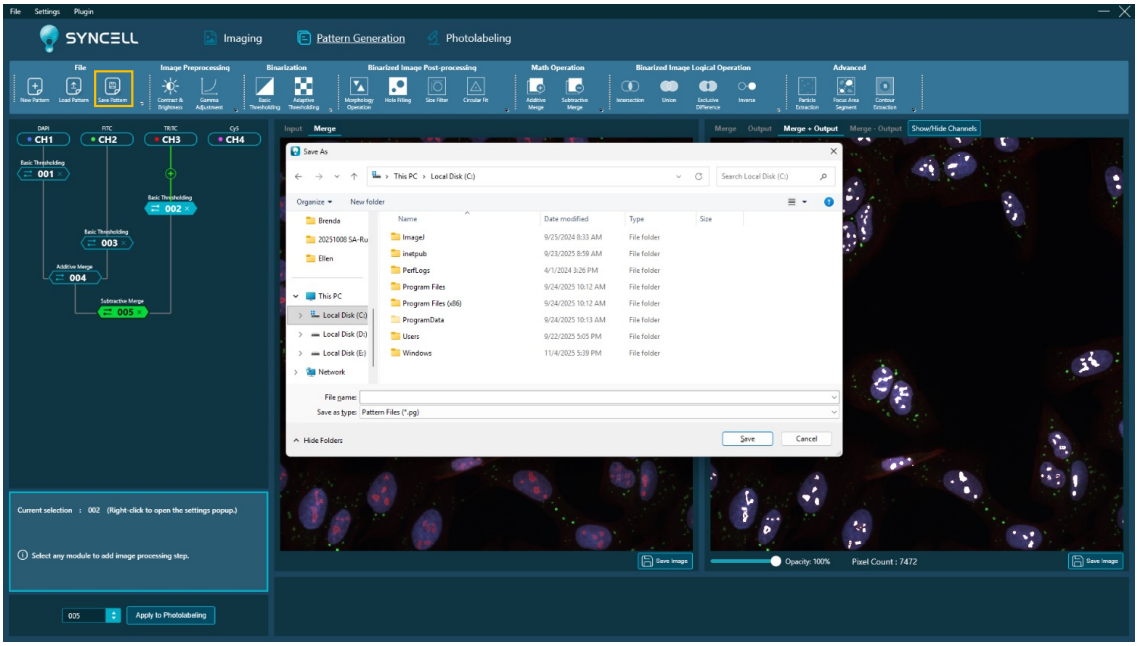

Fig. 6- 3 Pattern file – save pattern

6.2 Pattern Generation Setting

Toolbar: There are image preprocessing, binarization, binary image postprocessing, math operation, advanced, and AI modules (only shown in additional upgrades) provided for mask generation.

- Please refer to Appendix 1, Appendix 2, and the instructions in the pattern generation tutorial, as well as instructions on how to use each module in a pattern generation application to generate the correct mask.

Fig. 6- 4 Pattern generation toolbar

- Workflow Panel: A tree structure containing channels (source images) and steps (image processing modules) as tree nodes. You can add, delete, modify, or insert steps here.

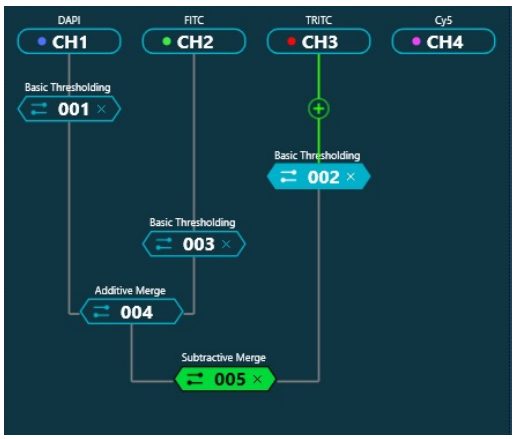

Fig. 6- 5 Pattern generation workflow panel

- Channels: The image source used as the root of the pattern generation. After acquisition or image loading, the captured channels will be automatically loaded into this panel.

Fig. 6- 6 Channels (Source Images)

- Add Step: Select any node (channel or step), then select the image processing module. After confirmation, a new image processing step will be added. Alternatively, you can select the image processing module first, then choose the node. After adding a new step, a parameter setting and application window will appear.
- Selection Step: Pattern generation offers two different selection methods. Left-clicking connects the input sources of the current step with a green line, allowing you to quickly identify the input sources, which is particularly useful when the tree diagram is complex. Right-clicking reopens the parameter setting and application window, allowing you to adjust and apply parameters, while highlighting all process steps related to the current step. This is especially useful in multi-color image processing, as it allows you to quickly identify which channels are included in the source of the current step.
- Insertion Step: With any step selected, click the green plus sign above the step, then click the image processing module you want to insert before this step. If the insertion conditions are met, you can click "OK" to confirm; otherwise, an insertion conflict message will be displayed.
- Deletion Step: Click the "X" button to the right of any step to delete it. If there are dependencies between steps, you need to delete the dependent steps first before you can delete the original step.

Fig. 6- 7 Selection, Insertion, and Deletion

- Apply to Photolabeling: Select the step you want to apply to photolabeling. You can only select binarized image output. After clicking the button, the calculation will be recalculated and the results updated. The selected step will be marked in bright green.

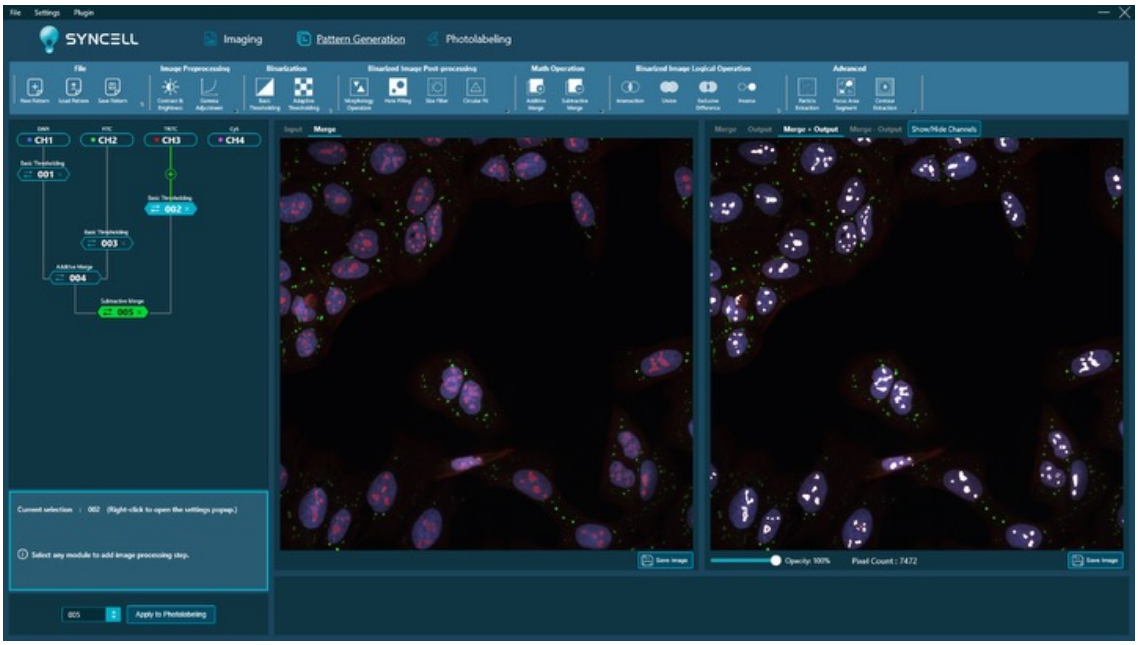

Fig. 6- 8 Apply to Photolabeling

7. PHOTOLABELING

The Photolabeling page serves as a central hub for laser-related parameter adjustments before the photolabeling process. Here, users can fine-tune the laser parameters for their photolabeling tasks. Additionally, this page allows the users to evaluate the effectiveness of the pattern generation across different Field of Views (FOVs). With the seamless adjustment of all parameters, users can streamline and automate the entire process.

7.1 Photolabeling Setting

- Monitor Channel: choose a channel as the live monitoring channel during photolabeling.
- Power Setting: set the laser power percentage for photolabeling.
- Labeling Exposure Time: set the exposure time (μs) per pixel for photolabeling.
- FOV skip checking by number of photolabeling pixels:
  - Skip FOV to do the photolabeling when the number of photolabeling pixels is not within the range specified in the settings.
  - Maximum photolabeling pixels will always be less than the image resolution.

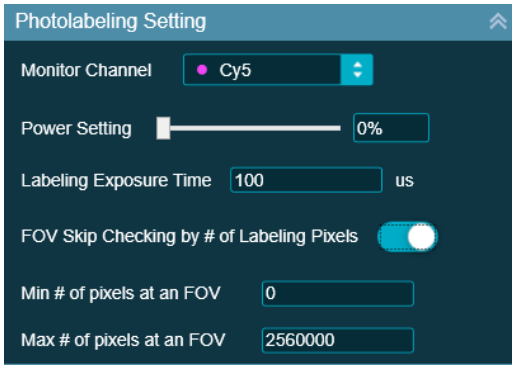

Fig. 7- 1 Photolabeling setting

7.2 Photolabeling FOV Setting

- Photolabeling FOVs by Columns & Rows: set a start FOV by clicking “Set Start,” then specify the number of columns and rows for photolabeling. As shown in Fig. 7-2 Photolabeling FOV setting – by Columns & Rows, 5 columns and 3 rows are selected, resulting in 15 FOVs to be labeled starting from the designated point. The house icon can be used to return to the last “Set Start” position.

  The start location refers to the absolute coordinates encompassing x, y, and z axes. It is highly recommended to find the focal plane before you click 'Set Start' button. If you are using Drift-free or Autofocus + Drift-free (see Chapter 7.3) to do Photolabeling. Please set drift-free settings before you click 'Set Start' button.

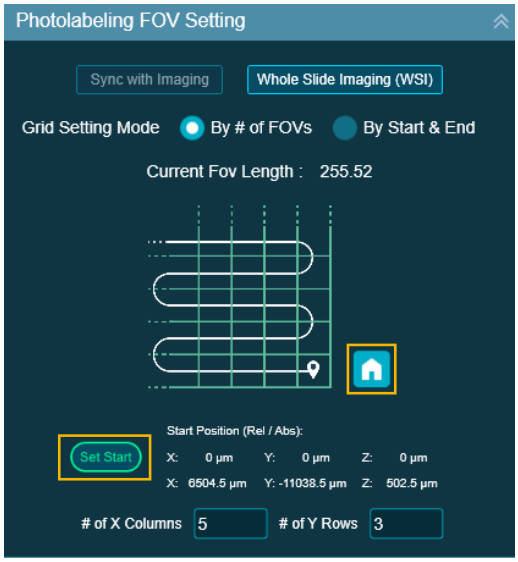

Fig. 7- 2 Photolabeling FOV setting – by Columns & Rows

- By Start and End: set a start and an end FOV to define the FOV range for photolabeling. The end FOV will be fitted as the nearest one to the center of the indicated FOV. The house icon can be returned to last "Set Start" position.

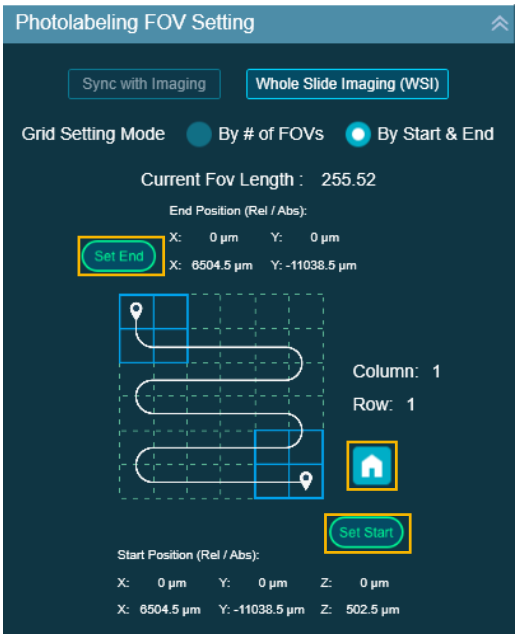

Fig. 7- 3 Photolabeling FOV setting – by start and end

7.2.1 Whole Sliding Image (WSI)

The Whole Slide Imaging (WSI) function allows users to manually select the Fields of View (FOVs) for photolabeling. The FOVs are chosen after a quick preview using selected imaging objective, and the system automatically calculates the path function for the selected FOVs to perform photolabeling.

Step 1: Magnification and Channel Settings.

- The WSI function button is available in Photolabeling FOV Settings. Click the button to open WSI settings page.

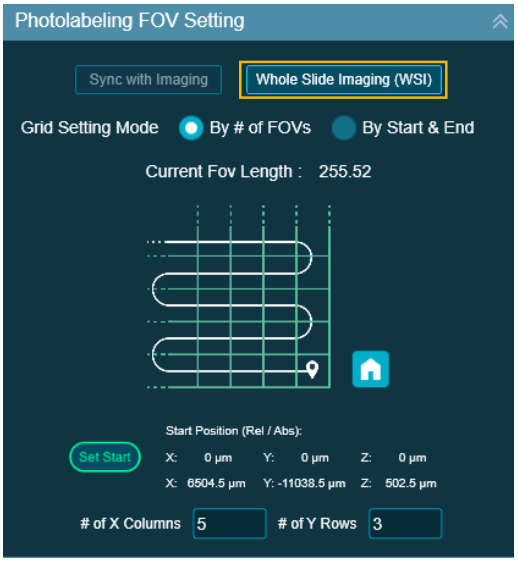

Fig. 7- 4 Photolabeling FOV setting – WSI function button

- Select the objective magnification for WSI imaging (WSI Imaging Magnification) and the magnification for photolabeling (Photolabeling Magnification).
- Select one or multiple acquisition channels for WSI imaging (WSI Acquisition Channel).
- Click Next to proceed to the next step.

Fig. 7- 5 Magnification and Channel Settings

Step 2: Imaging Acquisition Settings.

- The system will automatically switch to the selected objective for WSI imaging. After switching, you may need to readjust the fluorescence intensity, exposure time, and contrast for each channel, as well as refocus to ensure optimal WSI image quality.
- After focusing, it is recommended to enable the Drift-free function. You can determine the appropriate Focus Offset using the Precision Focus function or by Selecting Current Focus after manual adjustment. Also make sure the Focus Setting is set to Drift-free.
- In Photolabeling FOV Setting, define the WSI imaging area. Make sure to perform Set Start while the system is focused, and Drift-free mode is On (recommended) to store the correct Z-axis information.
- Click Next to proceed.

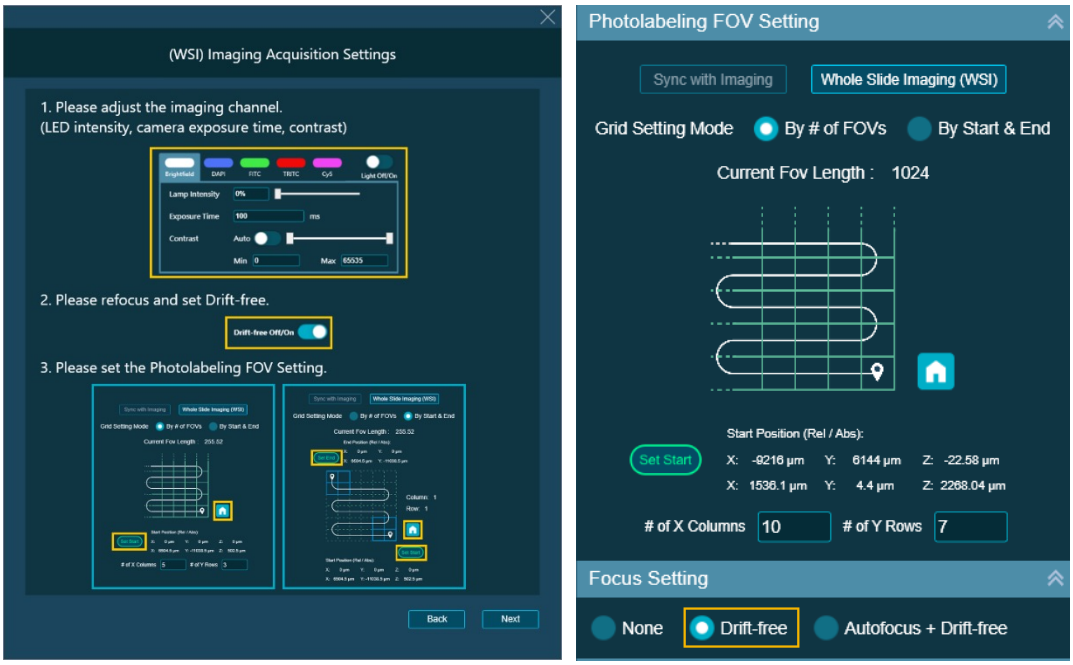

Fig. 7- 6 WSI Imaging Acquisition Settings & Focus Setting

Step 3: Wait for Image Acquisition.

Step 4: Select FOVs for Photolabeling.

- After image acquisition is complete, the WSI image will be displayed along with FOV segmentation based on the selected photolabeling magnification.
- By default, the View mode is enabled. Use the mouse scroll wheel to zoom in and out of the image and drag with the left mouse button to pan the image.
- Switch to Select mode to manually select and click on the specific FOVs you wish to use for photolabeling.
- Click Apply and proceed to step 5.

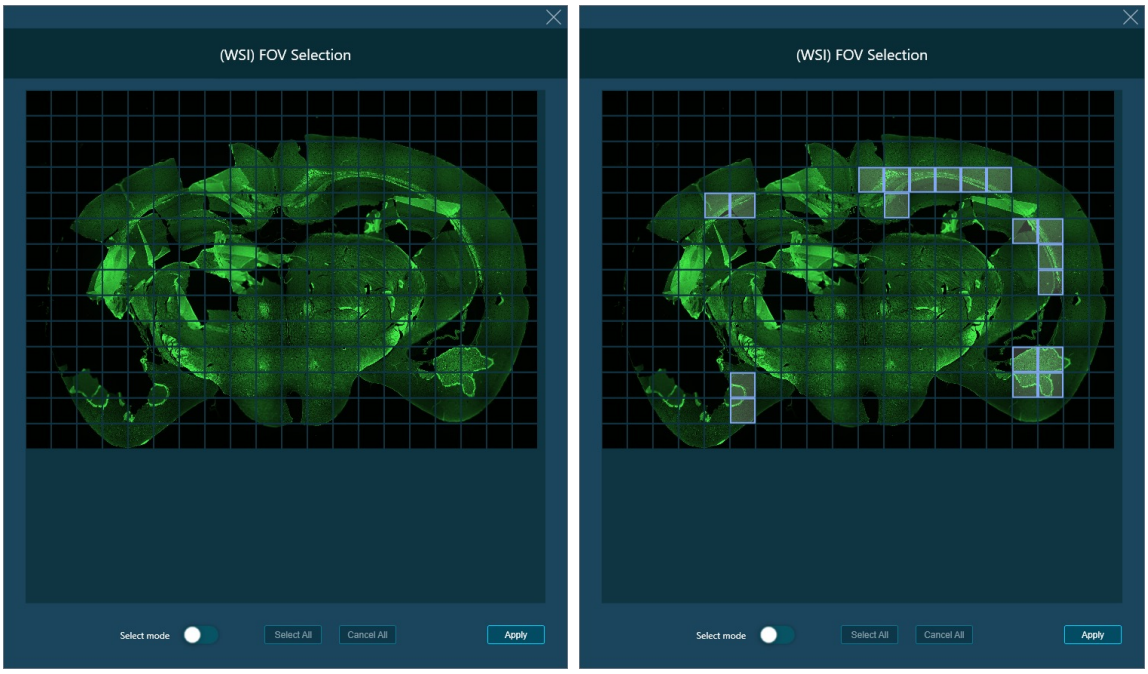

Fig. 7- 7 FOV Selection

Step 5: Apply WSI Method.

- The system will automatically switch to the photolabeling objective and display a confirmation dialog. You may need to adjust fluorescence intensity, exposure time, and contrast for each channel, as well as refocus or reconfigure Drift-free mode.
- Note: When WSI mode is enabled, the system will use the current Z-axis focus or Focus Offset as the initial focus plane for the selected FOVs.
- Once the WSI method is configured, the WSI button will turn green, indicating that WSI mode is active and ready for photolabeling.
- While WSI mode is active, the Set Start button will be disabled. If you need to redefine the start position, you must first release the WSI lock before adjusting.

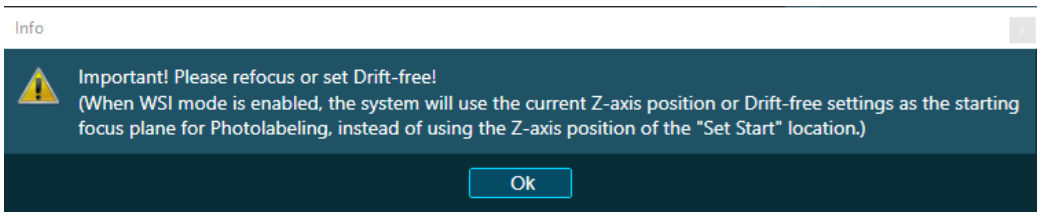

Fig. 7- 8 Important information for WSI

Step 6: Exit WSI Mode or Re-select the Photolabeling FOVs.

- When WSI mode is active, clicking the green WSI button will reopen the WSI settings page.
- Click Exit WSI Mode to disable WSI mode.
- Click FOV Selection to reselect the photolabeling area.
- WSI mode remains active after photolabeling is completed. If you want to perform photolabeling again on the same selected FOVs, reopen the WSI settings page and click Apply to return to Step 5. This allows you to refocus and perform photolabeling at different focal planes.

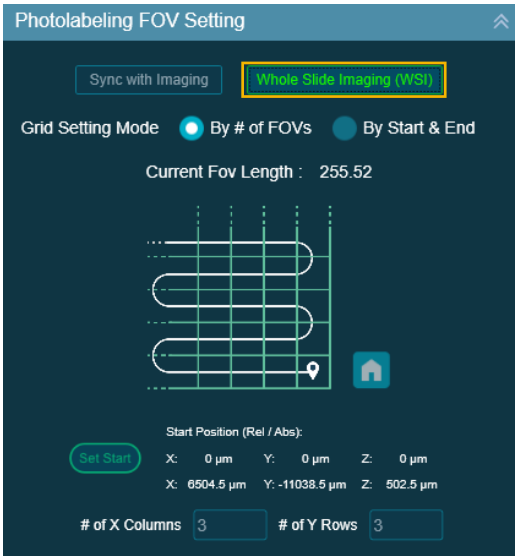

Fig. 7- 9 Photolabeling FOV setting – WSI function button

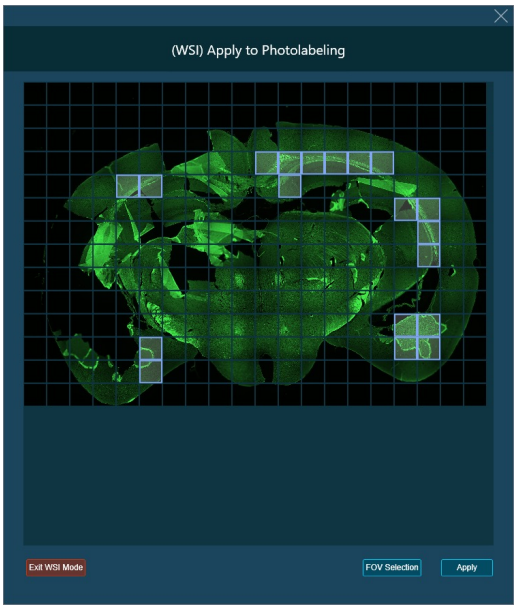

Fig. 7- 10 WSI settings page

7.3 Focus Settings

- None: image acquisition without any focus setting.
- Drift-free: automatically determines focus on each FOV using the optical property.

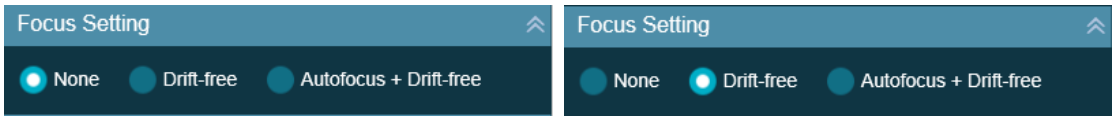

Fig. 7- 11 Focus setting “None” and “Drift-free

- Autofocus: utilizes Drift-free to establish the initial Z position for each FOV, then capture multilayer images and determine the optimal focus position.
  - o Layers and Z Pitch: set these parameters for multilayer acquisition.
  - o Channel: choose a channel to determine the image focus.
  - o 16-bit Raw Data: Focus using 16-bit raw images or use contrast-adjusted 8-bit images for focusing (please refer to section 5.3 Set contrast).

The Autofocus function determines the focal plane by capturing a Z-stack of images and applying an algorithm to locate the optimal focus. If the computed focal plane falls near or beyond the upper or lower boundary of the captured range, it may be excluded by the system to prevent unreliable focus detection. This exclusion is designed to avoid incorrect focusing or photolabeling on abnormal or out-of-focus FOVs.

To reduce the risk of such misjudgment, we recommend extending the Z-stack range by adding 4 to 6 extra layers beyond the estimated sample thickness. For example, with a sample thickness of approximately 5 micrometers and a Z pitch of 0.5 micrometer, one might initially calculate 11 layers. However, we suggest setting it to 15–17 layers to ensure the actual focal plane is fully captured within the range and not mistaken as being outside of it.

Failing to extend the layer range may lead to FOVs being skipped or misclassified as out of focus, even if they contain usable sample areas. This can impact both the success rate of autofocus and the accuracy of photolabeling.

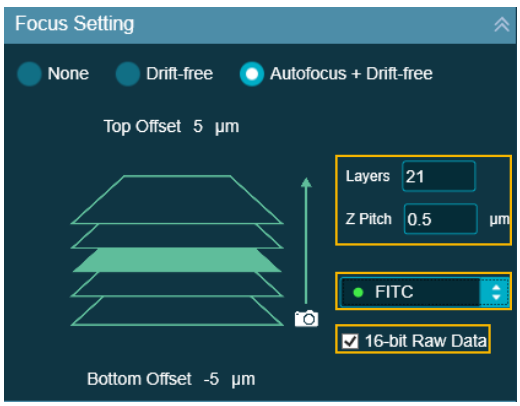

Fig. 7- 12 Focus setting “Autofocus + Drift-free”

7.4 Archive Setting

- Archive Path: choose a document path to save all the files generated during photolabeling. These include the original images and their associated masks.

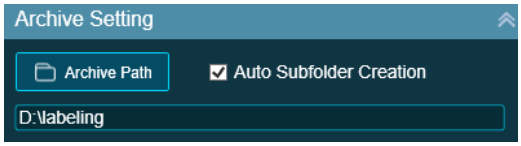

Fig. 7- 13 Archive Setting

7.5 Preview and Label

- Preview: click to create a mask for the current image (Live) with pattern generation workflow. The image/mask result will be shown in the display window to the right.
- Label: click to start photolabeling with the current settings.

Fig. 7- 14 Preview and Label

7.6 Display Windows

- Live Image Display Window (Left)
  - o Abs XYZ (Absolute XYZ position): The numbers represent the absolute XYZ position of the current FOV.
  - o Move To: move from the current FOV to another FOV by specifying the column and row number. (Only enabled if the user sets the starting point of Photolabeling FOV setting).
- Original Image/Mask Display Windows: there are four image/mask windows including original image, mask, original image plus mask, and original image minus mask on the right side of the display windows.

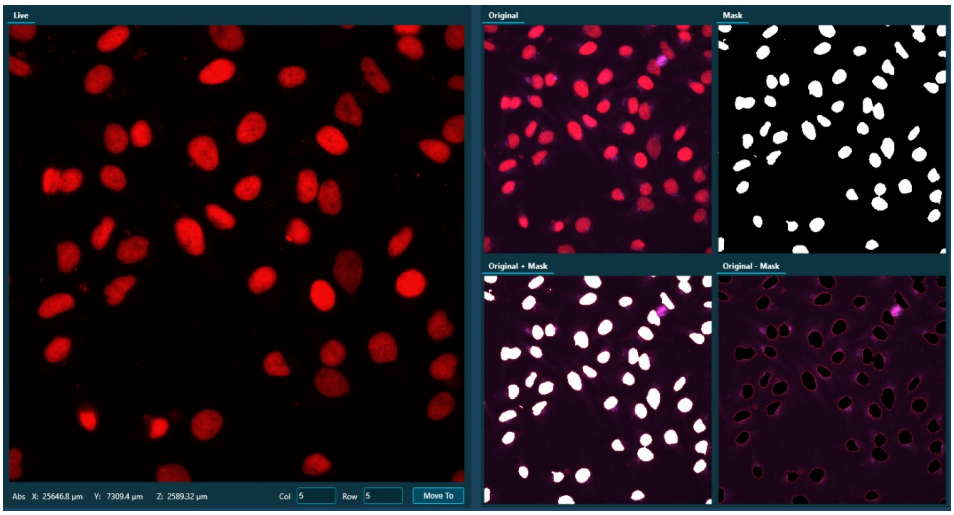

Fig. 7- 15 Display windows

  - o Window Enlargement: Double-left-click the tab (Original/Mask/Original+Mask/Original-Mask) to enlarge the indicated display window.
  - o Channel Display: choose display channels on original/mask/original+mask/original-mask image.

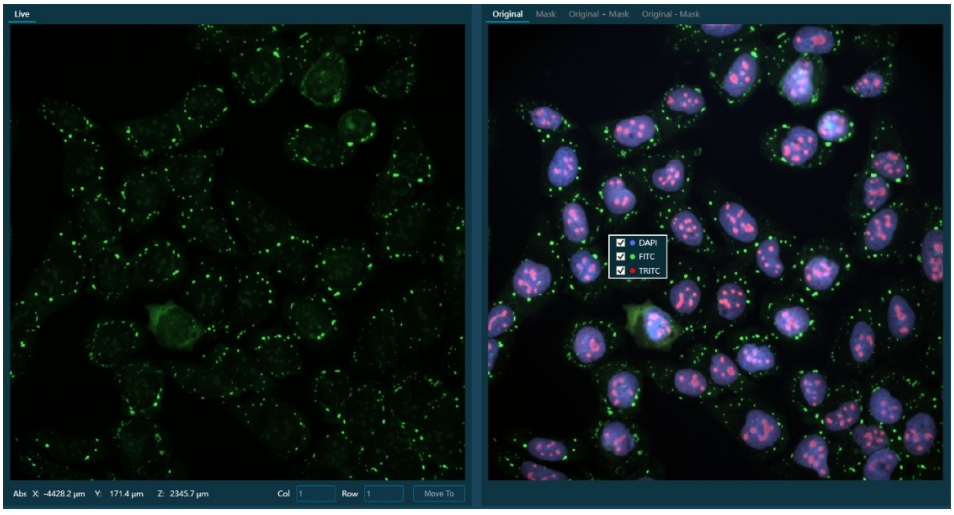

Fig. 7- 16 Function in the display window

8. TROUBLESHOOTING

This chapter lists common problems and how to solve them. The content includes hardware settings, system environment and software settings. Please troubleshoot according to the instructions.

8.1 Camera Initialization Failed

Autoscoop will try to connect to the camera when it starts. If the hardware device is missing or other applications are occupying camera resources, the initialization may fail.

When you see this error message, please check whether the camera is powered on and connected to the computer correctly. And confirm that no other software that may occupy camera resources is running.

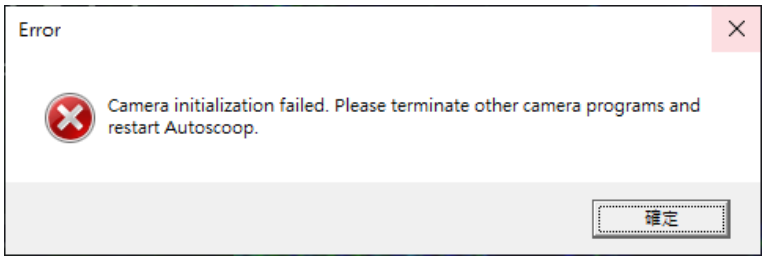

Fig. 8- 1 Camera initialization failed

8.2 Can Not Do Photolabeling

8.2.1 Failed to Load Calibration File

When you see this message, it means that the calibration file is damaged or incomplete. Please use the correct settings to recalibrate and generate the calibration file.

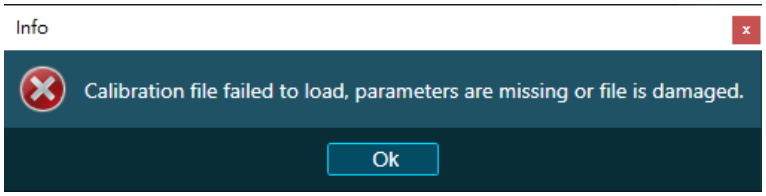

Fig. 8- 2 Calibration file failed to load

8.2.2 Calibration File Cannot Be Found

When you see this message, it means that the calibration file is not loaded or the file on the path no longer exists. Please reload the calibration file.

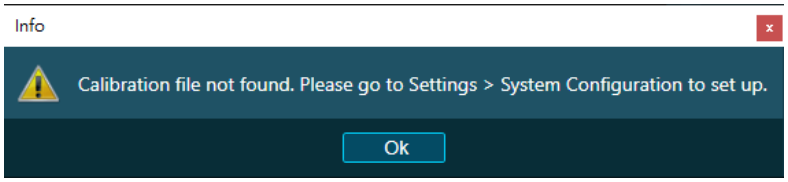

Fig. 8- 3 Calibration file not found

8.2.3 Error Message with Error Code

If you see this type of error message and the same message still appears after trying again. Please contact Syncell technical support and we will assist you in troubleshooting.

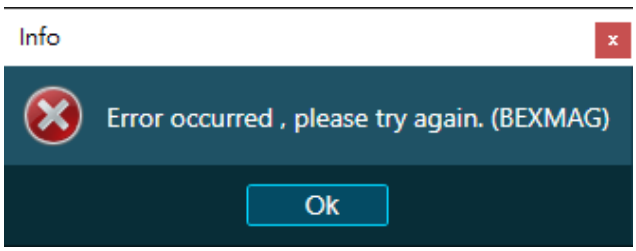

Fig. 8- 4 Error message with error code

8.3 Optical Engine Initialization Failure

When initializing Autoscoop, encountering the error message "Optical engine initialization failed" (Fig. 8- 5) indicates that the COM port may be abnormal. To resolve this:

- Manually reset the corresponding COM port:
  a. Note the COM port number from the error message.
  b. Access the Device Manager (Fig. 8- 6).
  c. Disable and then re-enable the identified COM port (Fig. 8- 7, Fig. 8- 8). If prompted, follow the system's reboot instructions.
- If problems persist, attempt a system restart.
- For unresolved issues, please contact Syncell technical support.

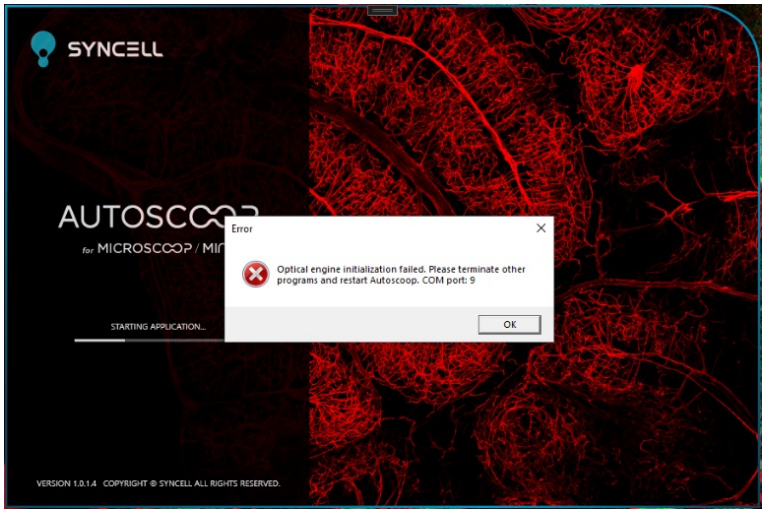

Fig. 8- 5 Optical engine initialization failed error message

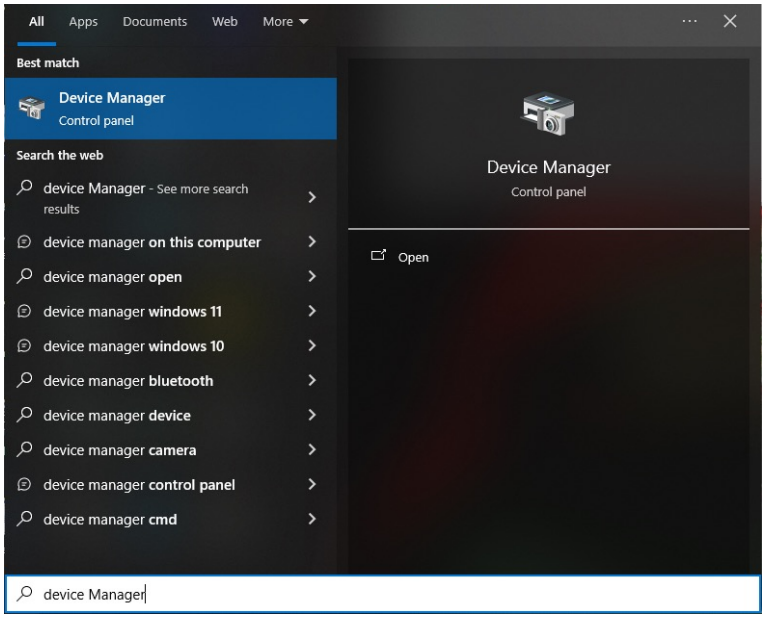

Fig. 8- 6 How to open device manager

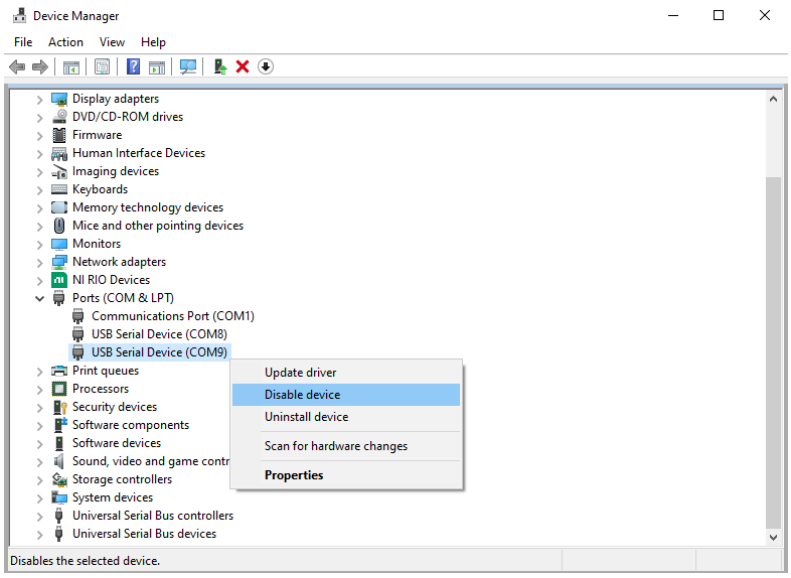

Fig. 8- 7 Disable COM port device

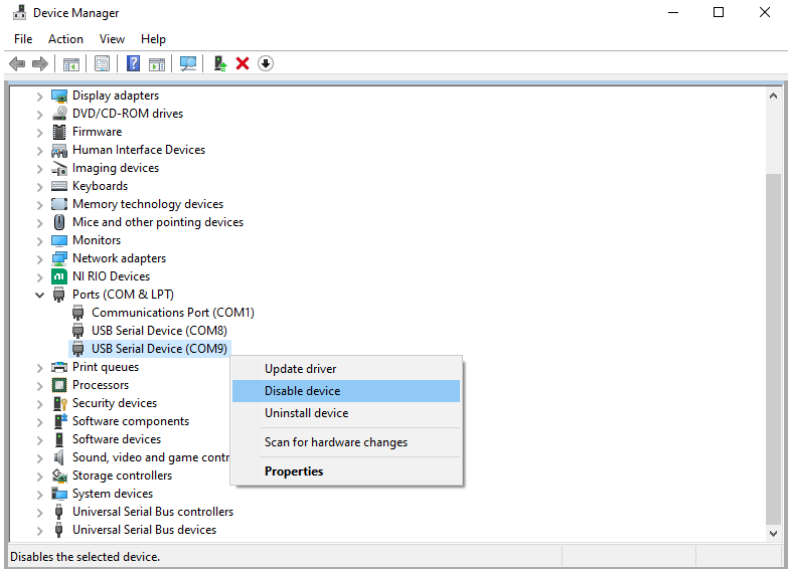

Fig. 8- 8 Enable COM port device

8.4 Do Not Use Joystick of Microscope During Automation Process

Utilizing the microscope's joystick during Autoscoop's automated processes may lead to unpredictable outcomes:

- Exercise caution as joystick operation during Precision Focus image acquisition may cause system hang-ups. In such instances, forcibly terminate Autoscoop and restart it (Fig. 8- 9).
- A protective mechanism, disabling the joystick during automation, is being developed for implementation in the upcoming version.

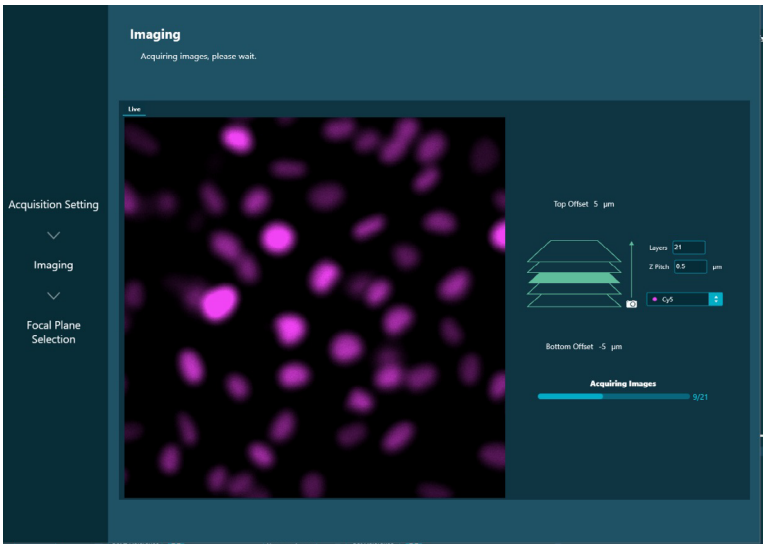

Fig. 8- 9 Multi-layer image acquisition process in Precision Focus

8.5 H0007 Key Not Found

Autoscoop utilizes hardware keys for protection. If encountering the "H0007 Key not found" error message (Fig. 8-10), follow these steps:

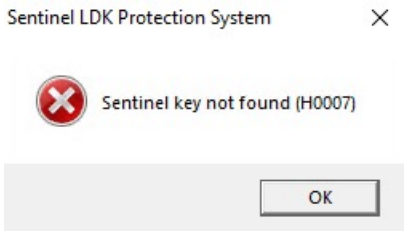

Fig. 8- 10 H0007 Key not found

- Ensure the hardware key is correctly inserted into the USB port.
- If the error persists after clicking OK, terminate Autoscoop via the task manager.
- Restart the Autoscoop program.
- For unresolved issues, contact Syncell technical support.

8.6 Unexpected Autoscoop Shutdown

Under normal circumstances, users can properly close Autoscoop through "Menu bar > File > Exit" to ensure that resources are safely released. This includes verifying whether the laser and LED are properly turned off.

- However, if Autoscoop is closed under abnormal conditions, such as insufficient system resources or crashes for unknown reasons, or if it is directly closed from the taskbar or task manager, it is possible that the shutdown operations of Autoscoop may not be completed. In such cases, you may find that the LED has not been turned off. We recommend reopening Autoscoop, waiting for the welcome screen to complete system initialization, and then properly closing Autoscoop.

Regarding laser safety: The laser is only activated during the execution of photolabeling tasks and is turned off after completing labeling for each FOV. It also immediately turns off when the user presses "Hold" or "Stop". If Autoscoop unexpectedly terminates during a photolabeling task, the laser will also be turned off after completing the labeling for the current FOV. This usually takes a few seconds (depending on the number of pixels to be labeled and parameter settings), so you don't need to worry about the laser remaining on in this situation.

- Nevertheless, we still recommend reopening the Autoscoop program to perform initialization operations and advise against staring directly at the laser emission position.

Please follow these troubleshooting steps to address common issues encountered while using Autoscoop. For further assistance or persistent problems, do not hesitate to contact Syncell technical support.

9. CALIBRATION

9.1 What is Calibration?

The Autoscoop Calibration program is employed for the fine adjustment of the photolabeling laser point's XY positions. This program specifies the laser point coordinates within the field of view at both high resolution (1600 x 1600 pixels) and low resolution (800 x 800 pixels). It provides the XY position references and defines the optical scanning range of the Microscoop platform to enable precise photolabeling at the specified region of interest.

9.2 User Interface

The main layout of the Autoscoop Calibration software and a description of each of the functions are listed below:

9.2.1 Control Layout

- Main menu: includes functions of closing and shrinking programs, and Log collection tool.
- Image settings: including objective lens selection, resolution adjustment, camera exposure time setting and image contrast setting.
- Laser Control: panel used to adjust laser position and laser intensity.
- Camera Control: turn on and off camera live.
- Image panel: displays images captured by the camera.
- Calibration button: start calibration.
- Calibration steps: display calibration steps and current progress.
- Cancel button: cancel calibration.

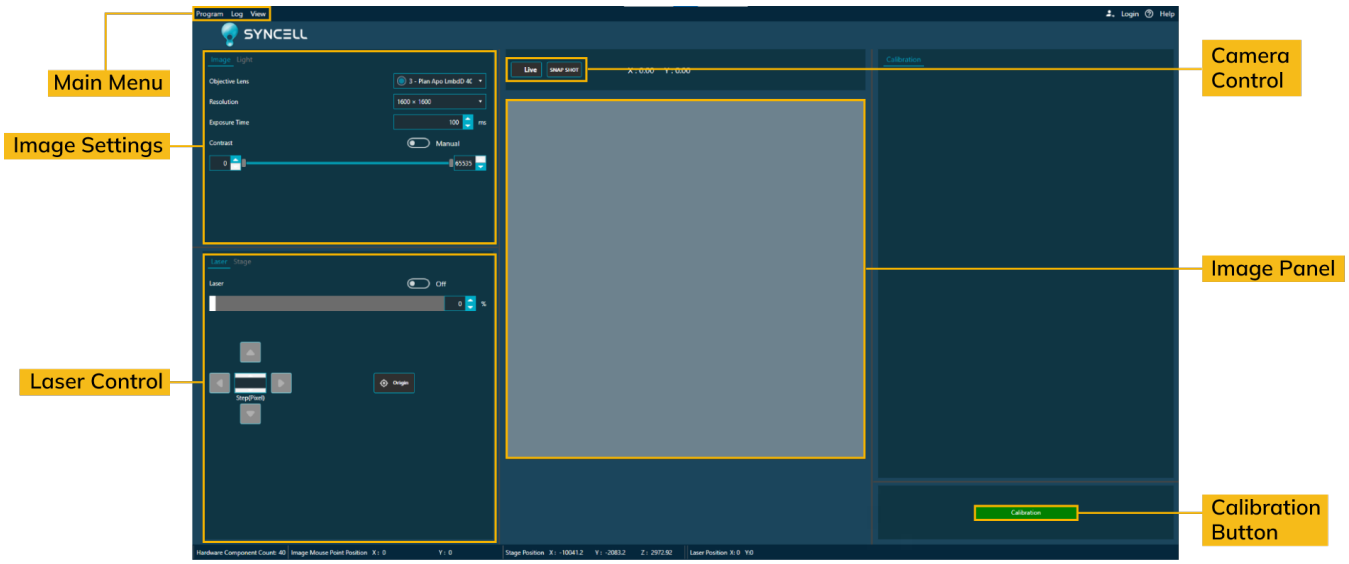

Fig. 9- 1 User Interface of Autoscoop Calibration software

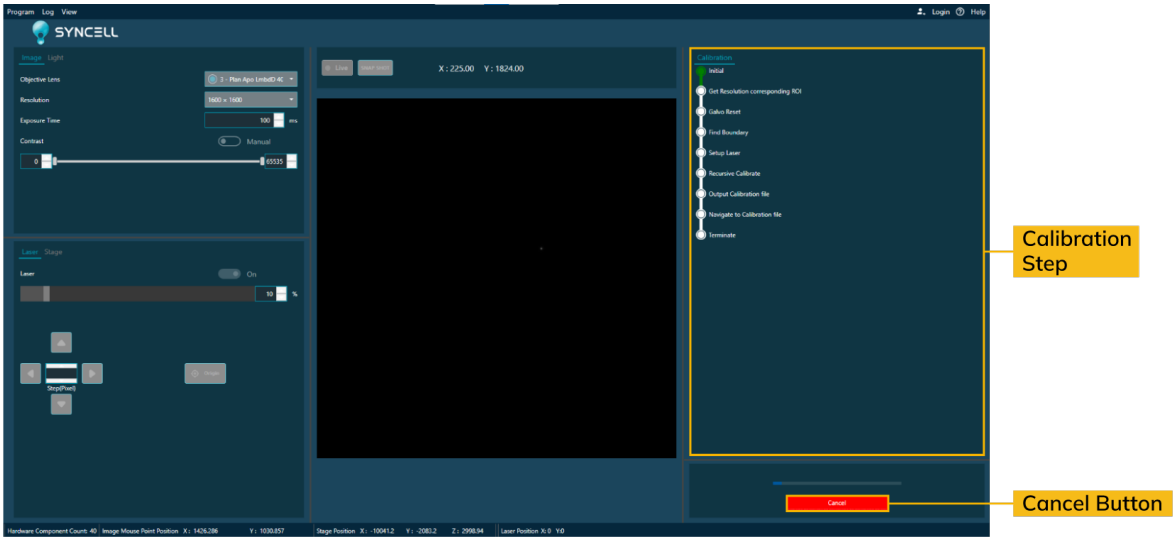

Fig. 9- 2 User Interface of Autoscoop Calibration software

9.2.2 Calibration Reference File Saving and Structure

Calibration parameters are stored in a reference file. Once the calibration process is complete, a pop-up window will appear, guiding you to the location where the calibration file is saved.

For instructions on how to load the calibration file using Autoscoop software, please refer to Chapter 3.1.

9.3 Workflow

9.3.1 Calibration Preparation

- Ensure that the laser switch on the Laser Control Panel is set to "Off." Place the Calibration Reagent flat on the microscope stage and cover it with the dedicated light shield to prevent interference from external light sources.
- For information on Calibration Reagent, please refer to the document "Spatial Proteomics with MICROSCOOP - Instructions for Use of Calibration Reagent."

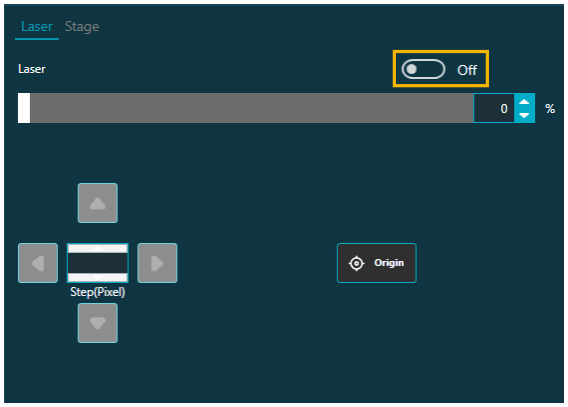

Fig. 9- 3 Laser Switch Off

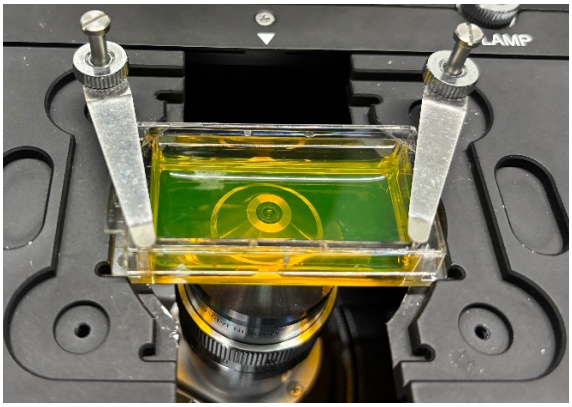

Fig. 9- 4 Calibration Reagent

9.3.2 Select the Objective Lens and Resolution

- Objective Lens: select an objective lens for calibration.
- Resolution: select a resolution (800x800 or 1600x1600) for calibration.

After setting the objective lens and resolution, click the "Live" button to display the live image. Changing the resolution requires stopping camera live first.

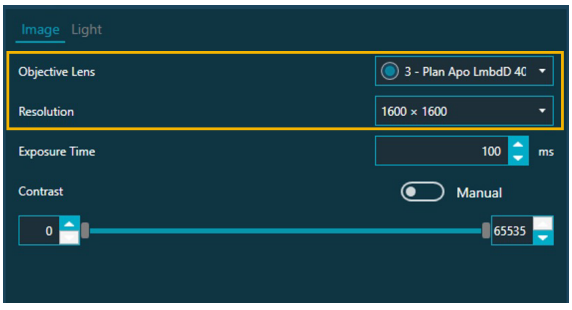

Fig. 9- 5 Objective Lens and Resolution

9.3.3 Image and Laser Settings

- Exposure Time: set an exposure time for image capture. Usually, 100 ms is enough for a good laser spot image.
- Contrast: set the LUT (0-65535) range of the image. Clicking "Auto" can help you quickly find the correct z-position of the laser point. After finding the laser point, it is recommended to cancel "Auto" and adjust the focus manually to make the laser point clear and not exposed.
- Laser Power: adjust laser power by sliding the bar to increase the clarity of the laser spot image. The recommended value is between 0% and 18%. An overly bright laser spot may degrade the calibration quality.

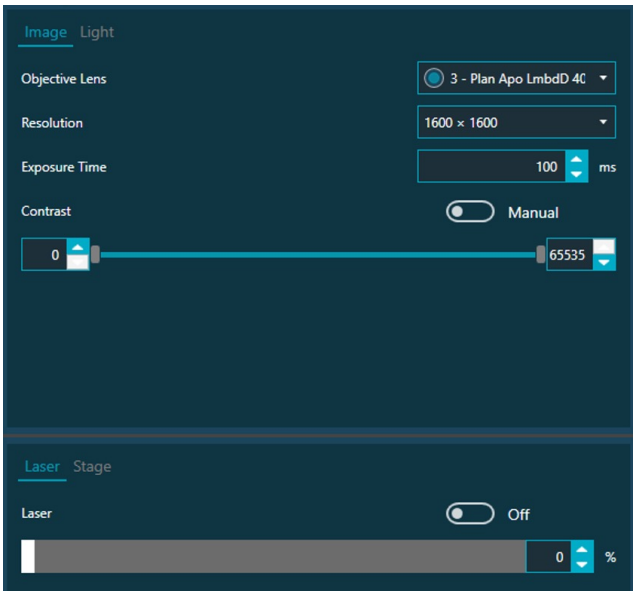

Fig. 9- 6 Image and Laser Settings

9.3.4 Photolabeling Laser Position Adjustment (Can be skipped)

- Origin: click origin to return to the original laser position.
- Direction buttons and Step size: set a step size for moving the laser point with the direction buttons.

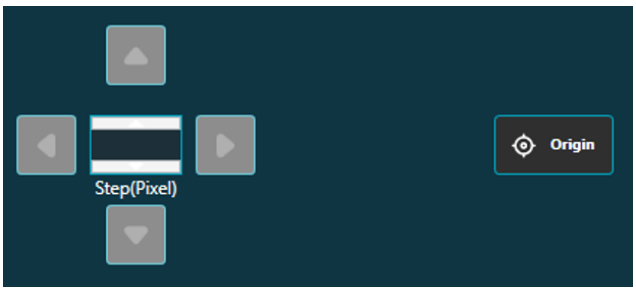

Fig. 9- 7 Laser Position Adjustment

9.3.5 Example of Good Laser Spot for Calibration

- Without adjusting the Contrast setting (0~65535), you can directly see a clear laser spot on the screen. Once calibration starts, the program will automatically adjust the focus.
- In principle, the smaller and brighter the laser spot, the better.
- Under the same laser intensity, the laser spot is darker at high resolution (1600x1600) and brighter at low resolution (800x800). Therefore, it is recommended to use lower laser intensity for calibration at low resolution.

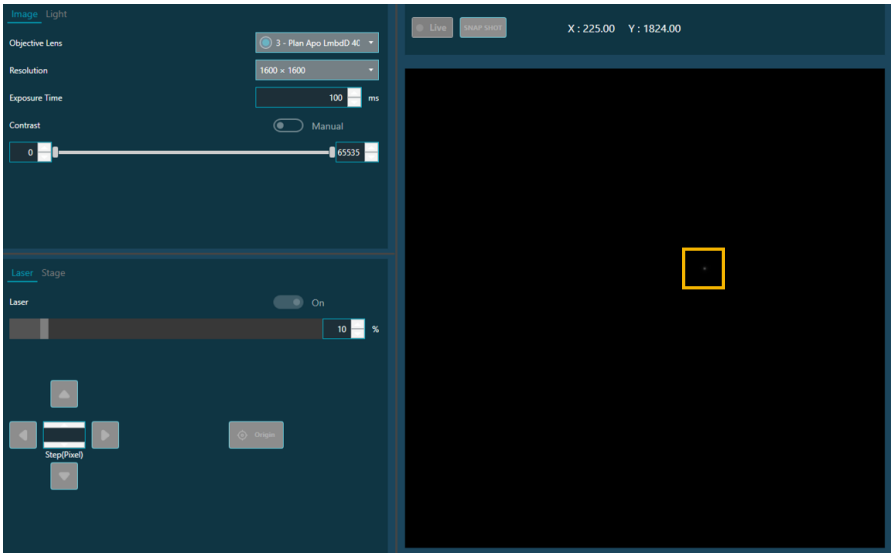

Fig. 9- 8 Example of Good Laser Spot

9.3.6 Start Calibration

- After obtaining laser spot, click "Calibration" button to start the calibration process.

9.3.7 Camera ROI Alignment

- The first step in calibration is to align the motion range of the laser galvanometer with the boundary of the camera ROI.
- Camera ROI Alignment does not need to be executed every time.
- System will pop up a prompt window asking whether to re-find the ROI boundary.

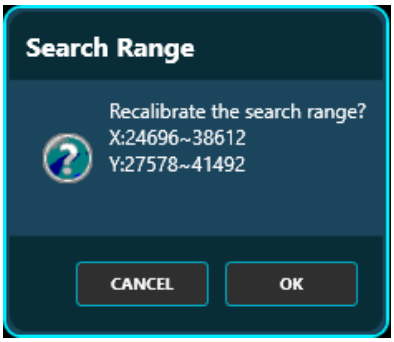

Fig. 9- 9 Recalibrate Search Range Prompt

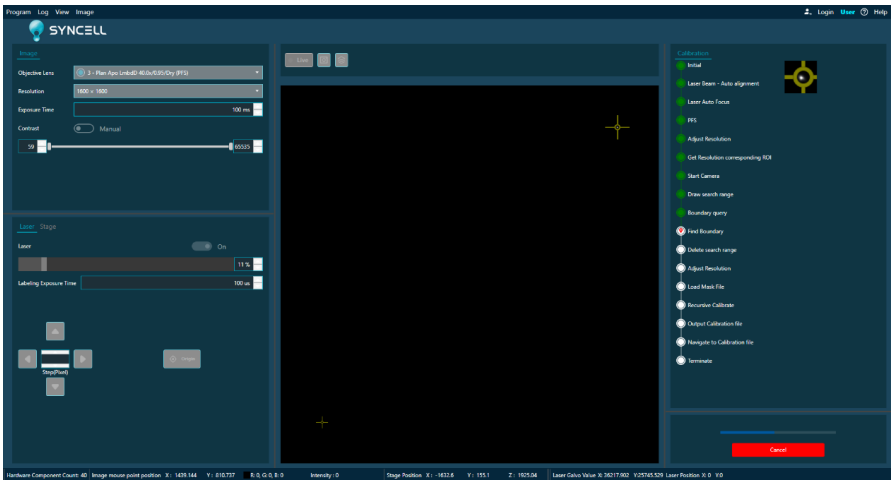

Fig. 9- 10 Auto Camera ROI Alignment

9.3.8 Calibration Processing

When you see the following screen, it means that the calibration is in progress. Please wait patiently for the calibration to be completed.

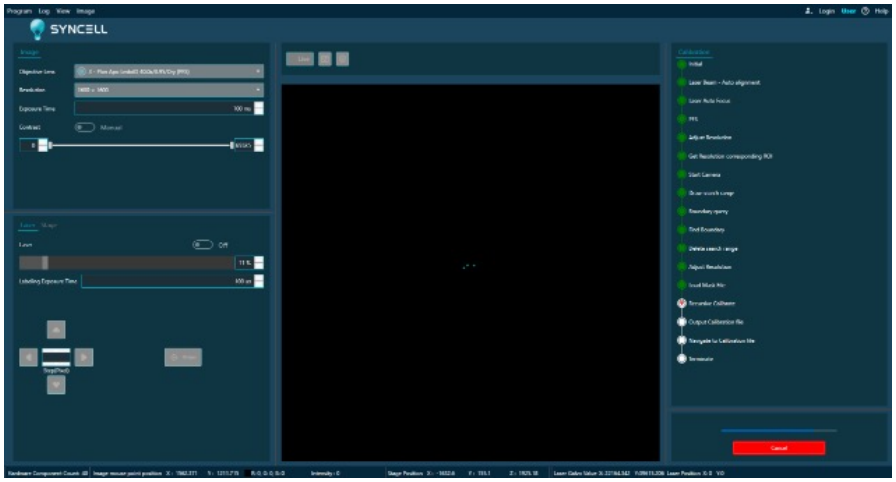

Fig. 9- 11 Calibration in Progress

9.3.9 Calibration File

When the calibration is completed, the system will generate the calibration file and open the file location.

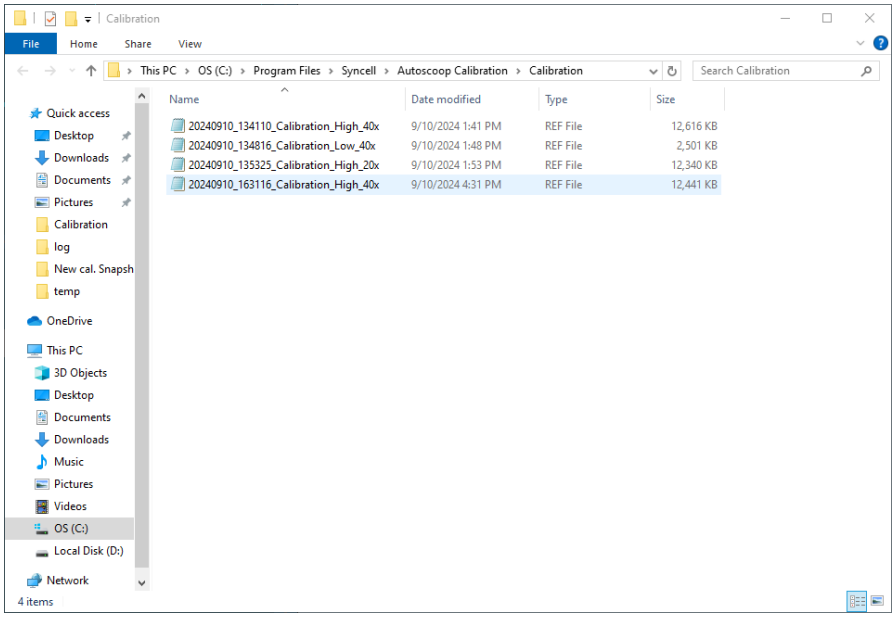

Fig. 9- 12 Calibration File

9.4 Troubleshooting

9.4.1 Calibration Failed

When calibration fails, a red dot will appear in the calibration steps on the right side of the screen, and an error message will pop up for certain errors.

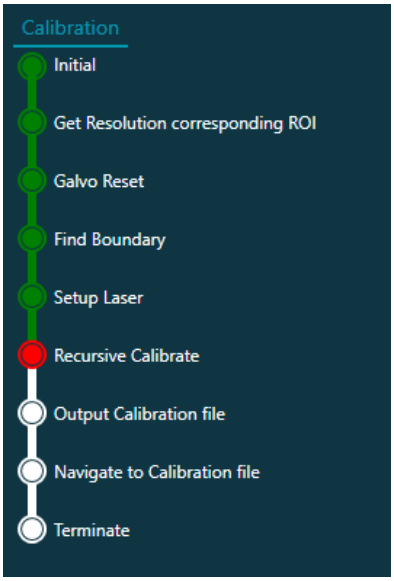

Fig. 9- 13 Calibration Failed

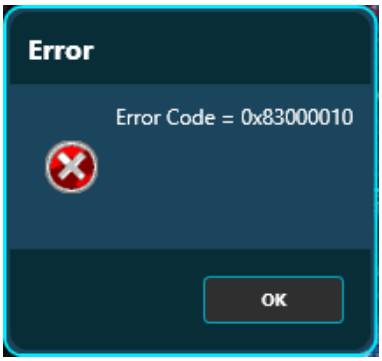

Fig. 9- 14 Calibration Error Message

9.4.2 Log Packaging and Error Reporting

If the problem still cannot be solved after retrying the calibration, please follow the steps below to package the logs and contact Syncell technical support.

a. Click the “Log” option on the “Main Menu”, the log packaging tool will pop up.

b. Select the date problem occurred on the log packaging tool, fill in the subject and problem description (a more detailed problem description will help us solve your problem), and click the “Apply” button.

c. In the pop-up window of windows file explorer, select or create a folder to store log data for packaging, and click “Select Folder”.

d. The system will package the log data into the specified folder and compress it into a zip file. Please send this compressed file to Syncell technical support and we will do our best to solve your problem.

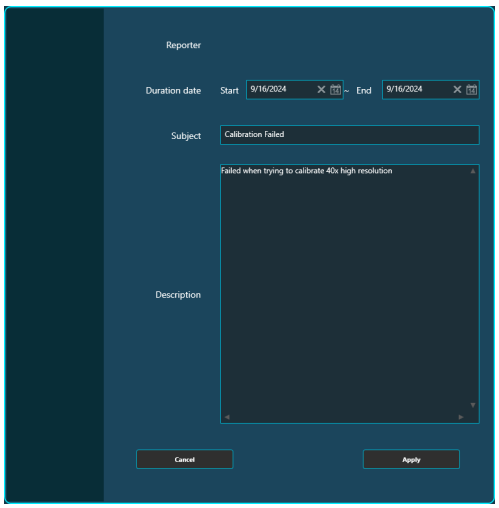

Fig. 9- 15 Log Packaging Tool

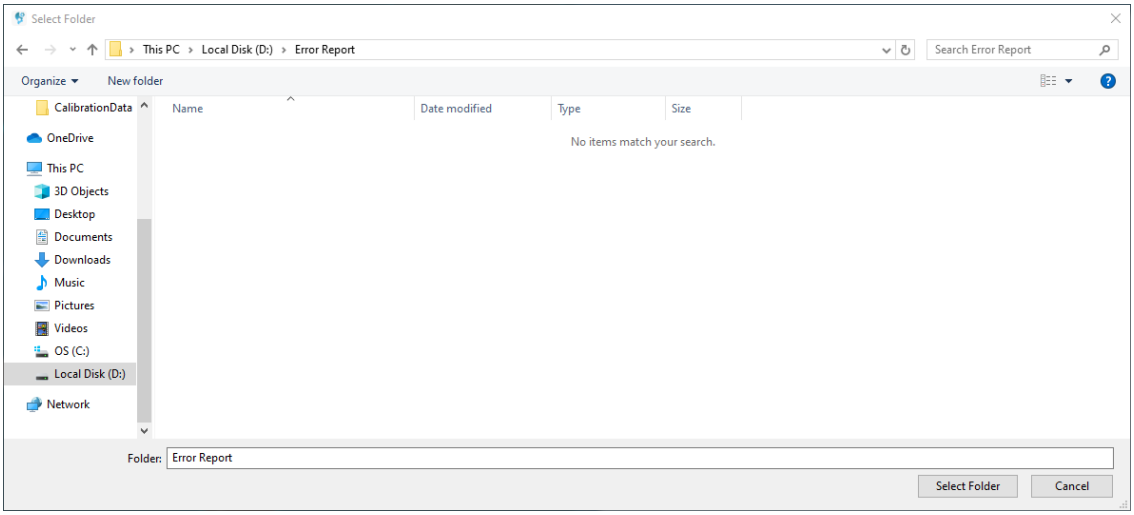

Fig. 9- 16 Select or Create Packaging Folder

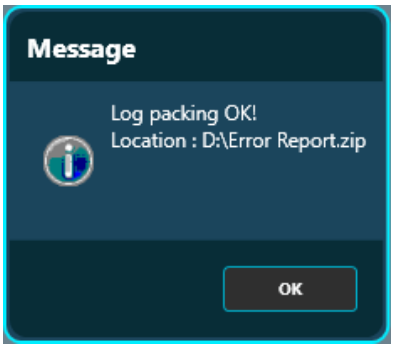

Fig. 9- 17 Packaged Log Message Prompt

10. PLUGIN

The Plugin section provides access to additional analytical tools and modules integrated into the software. These plugins extend the system's core capabilities by offering specialized functions for advanced image processing, quantitative analysis, and data visualization.

10.1 S/N Ratio

The S/N Ratio (Signal-to-Noise Ratio) module allows users to calculate the signal-to-noise ratio of an image through a guided multi-step process. This function helps evaluate image quality by quantifying how much the signal stands out from the background noise.

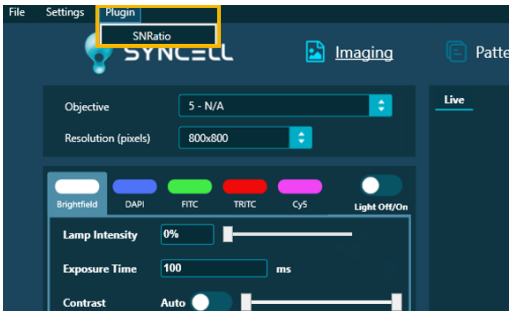

Fig. 10- 1 S/N Ratio plugin in Menu bar

Before starting the procedure, click the Load Image button to import an image. Proceed through the following steps sequentially. After completing each step, click Next to move to the next stage.

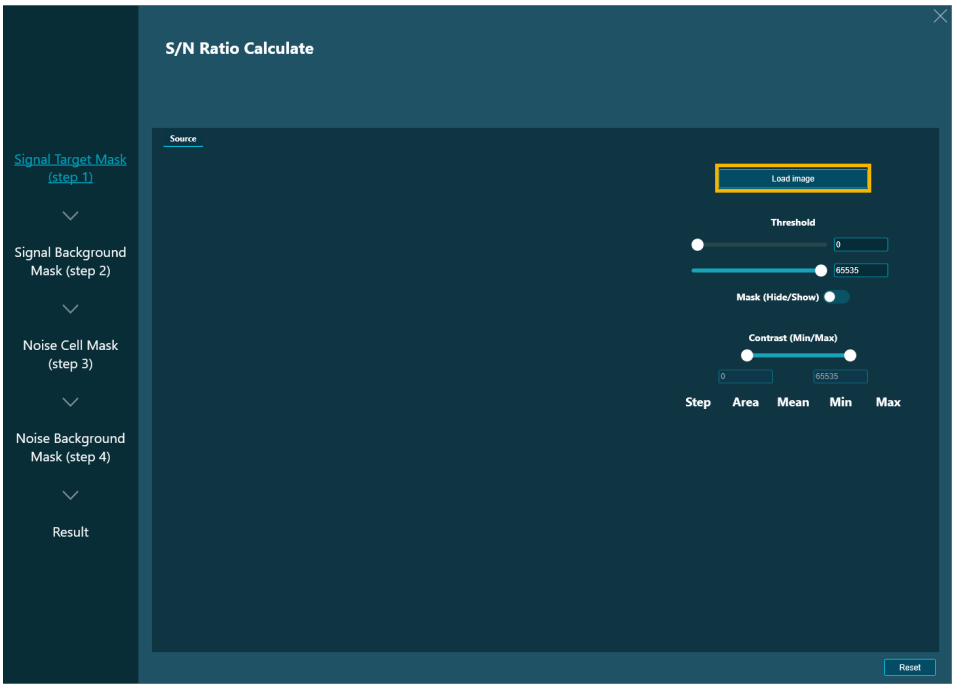

Fig. 10- 2 S/N Ratio popup with load image button

Step 1: Use the Upper Threshold Bar and Lower Threshold Bar to select the Signal Target Mask from the image.

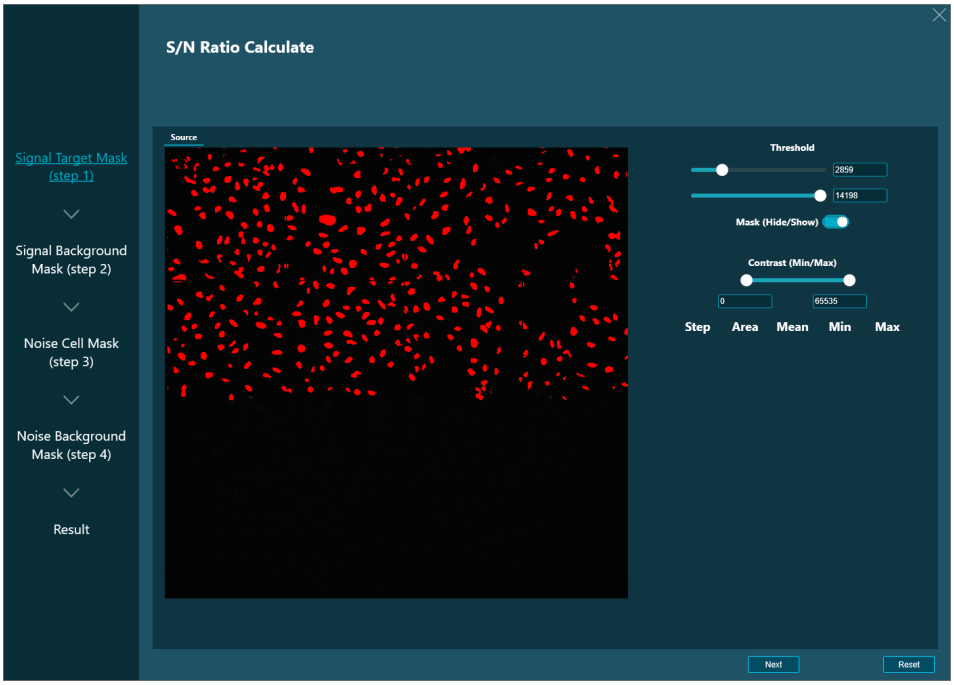

Fig. 10- 3 Signal Target Mask

Step 2: Similarly, use the threshold bars to define the Signal Background Mask.

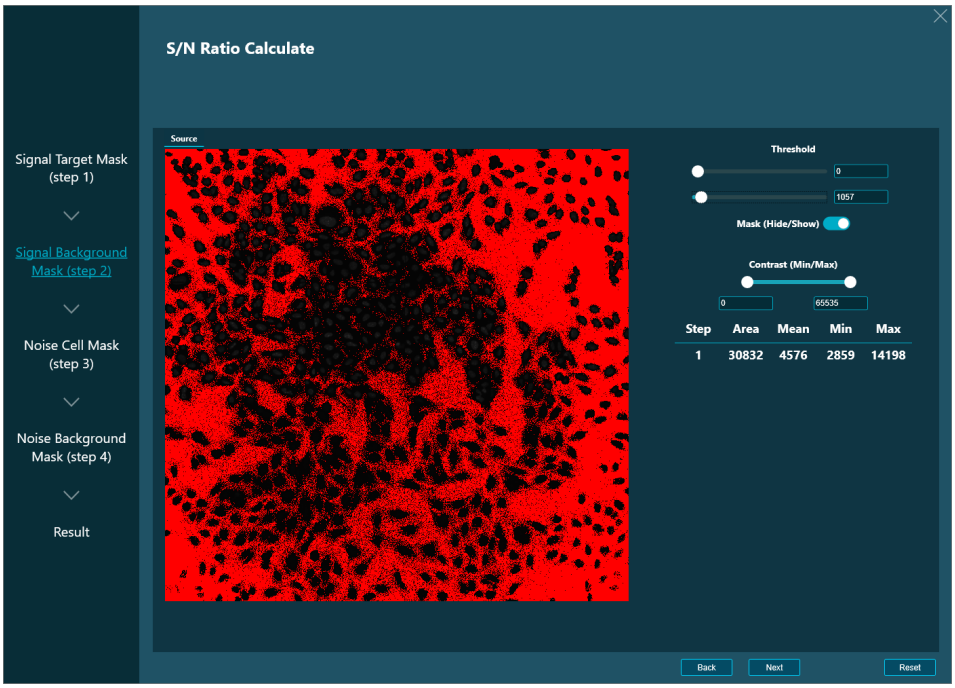

Fig. 10- 4 Signal Background Mask

Step 3: Apply the same method to extract the Noise Target Mask.

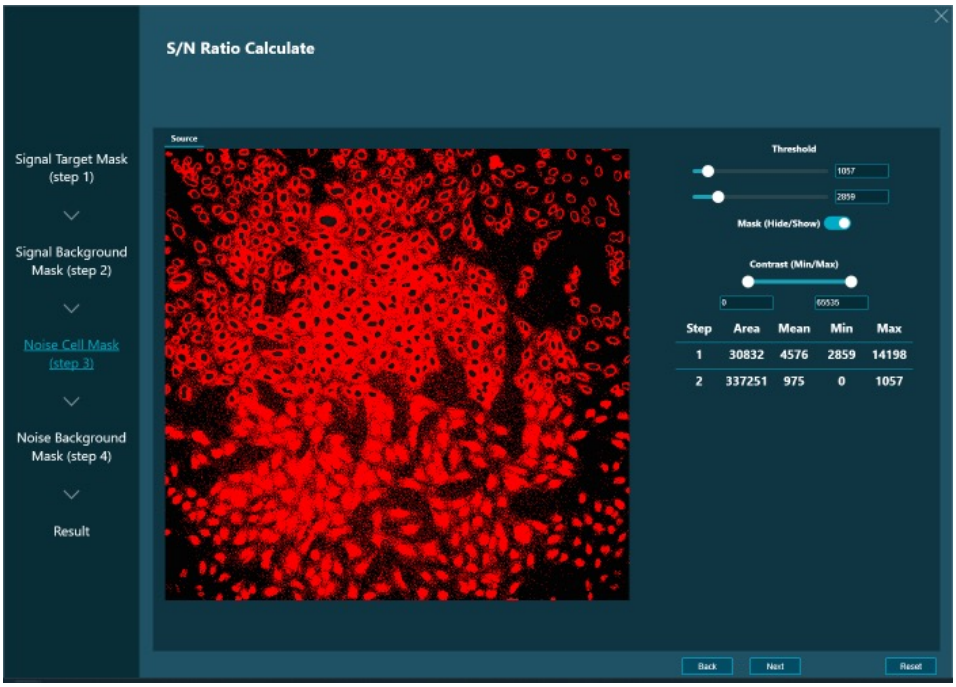

Fig. 10- 5 Noise Target Mask

Step 4: Define the Noise Background Mask. After setting all masks, click Next again to proceed with the S/N Ratio calculation.

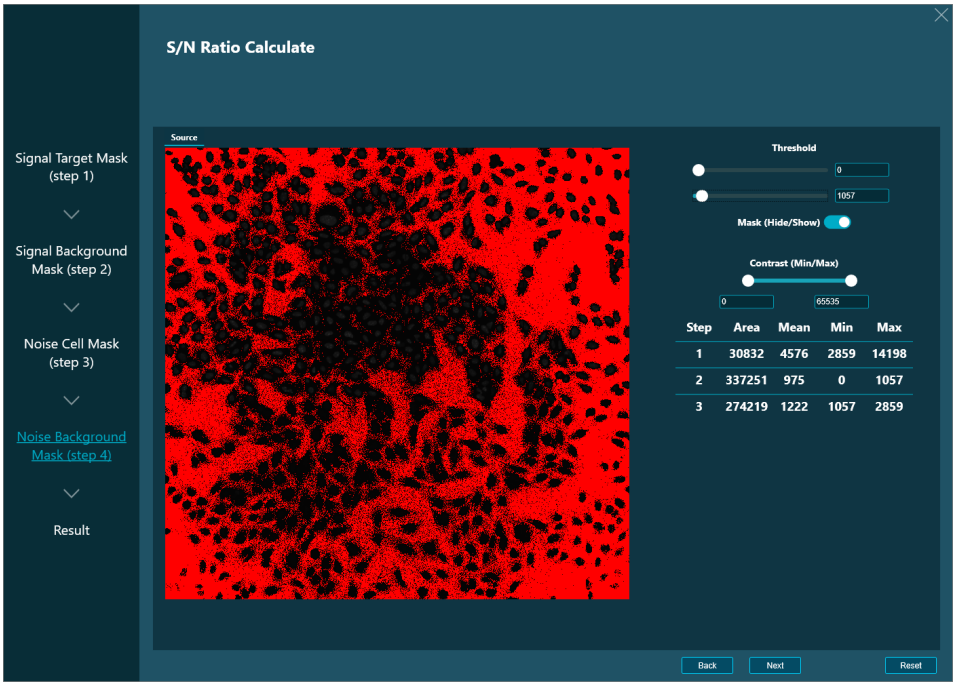

Fig. 10- 6 Noise Background Mask

Upon completing Step 4, click Next once more to enter the result display, where the calculation process and the final S/N Ratio value are shown.

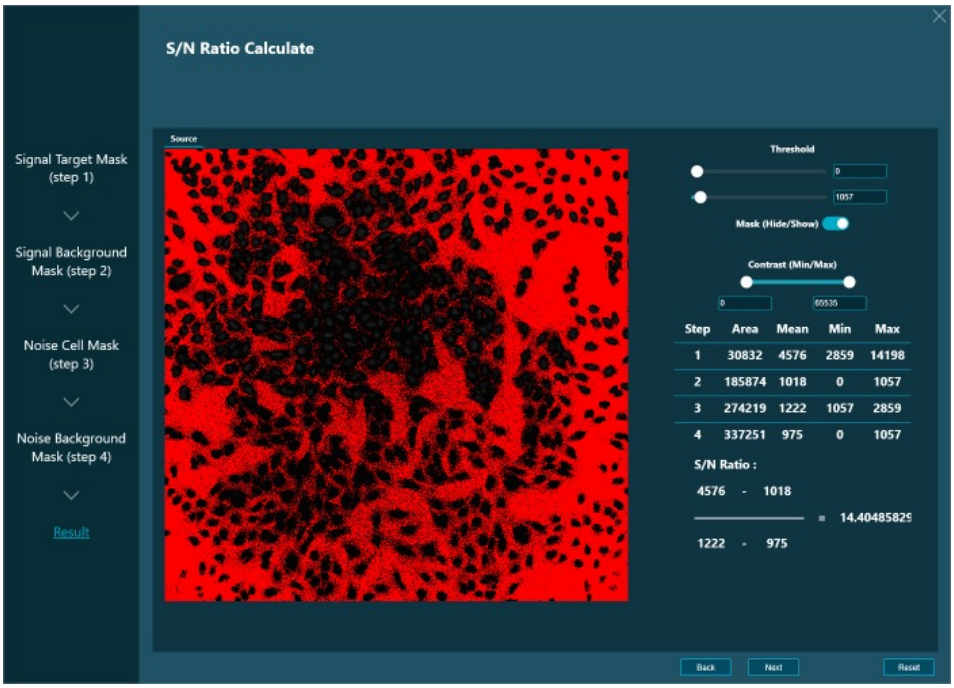

Fig. 10- 7 S/N Ratio result

11. CONTACT

If you have any further questions, please contact Syncell Inc. directly or via your local contact. Appropriate contacts can be found online at https://www.syncell.com.

Company information

Email: support@syncell.com

Tel: +886-2-2785-6780

Address: 14F, No. 508, Sec. 7, Zhongxiao E. Rd., Nangang Dist., Taipei City 115, Taiwan

Manufactured by Syncell Inc.

APPENDIX 1 - PATTERN GENERATION USER MANUAL

1.1 Terms and Definitions

1.1.1 Pixel

A pixel is the smallest unit of an image. It represents a single point or dot that contains information of its color or grayscale value. The grayscale value of a pixel is called the pixel value, intensity, or pixel intensity.

The number of pixels along the height and width of an image is called image resolution (e.g., a resolution of 1920*1080 means that the image width is 1920 pixels while the image height is 1080 pixels.)

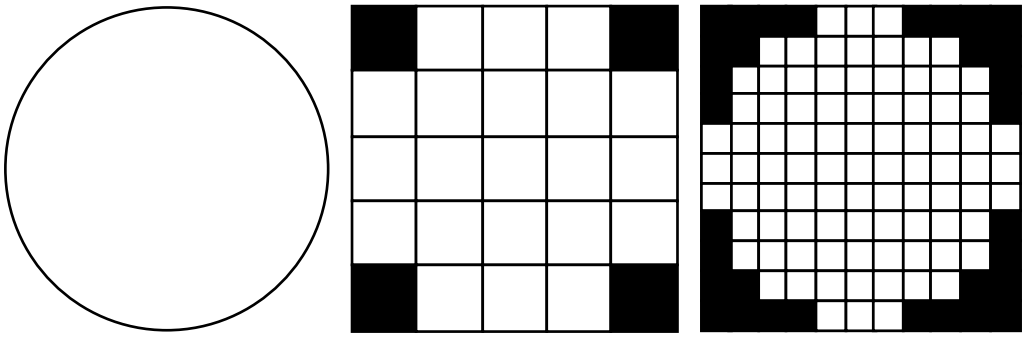

Fig 1. A representative drawing of a white circle in a low/high-resolution image

1.1.2 Image Input

In the current software version, an image captured by a camera is in grayscale and is composed of pixels ranging from 0 to 65535, where 0 is presented as black and 65535 is presented as white. The pixel values between 0 and 65535 are shades of gray, with values closer to 0 being darker while values closer to 65535 are lighter.

For the live imaging settings panel in the imaging page, adjusting the contrast transforms the pixel value ranging from 0~65535 to 0~255. Therefore, all the images in the pattern generation page would use the range 0~255 as the image pixel value range.

Scaling the pixel value range of the image only clips the pixel values to a valid range for display purposes. It does not affect the sample, such as a change in the LED intensity or exposure time.

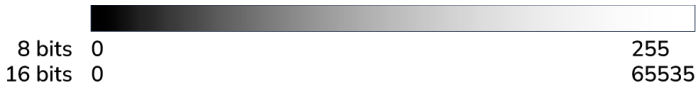

Fig 2. Pixel value range

1.1.3 Binary Image / Pattern Mask

A binary image or a pattern mask is an image composed solely of pixels with values of either 0 or 255, represented by black or white color, respectively. The mask is marked by white colored pixels and represents the region slated for photolabeling in the subsequent step.

1.1.4 Binarization (Thresholding)

Binarization is a simple image processing technique used to convert a grayscale or color image into a binary image.

1.1.5 Kernel

A kernel, when applied to an image, acts as a template or mask to enable operations like filtering, convolution, and morphological transformations. The values within the kernel determine the specific characteristics of the operation being performed on the image. The size of the kernel determines the extent of the surrounding pixels taken into consideration while processing a particular pixel. In most cases, the size of the kernel must be an odd number larger than 3.

1.1.6 Equalization

Image equalization adjusts the brightness levels of an image to enhance the balance between the dark and light areas. It works by stretching the range of pixel values in the image to cover the entire intensity spectrum.

1.1.7 Normalization

Normalization rescales the pixel values of an image to a common range in order to make them more manageable and consistent.

1.1.8 Morphology

Morphology is a branch of image processing that focuses on the shape, structure, and spatial arrangement of objects in an image. It involves the use of mathematical operations known as morphological operations to manipulate and analyze the shape and size of image components, including regions, edges, and boundaries.

1.2 Functions and Parameters

1.2.1 Image Preprocessing

Image preprocessing involves applying various techniques to prepare an image for further analysis or improvement. The image may be optimized for specific purposes by brightness/contrast adjustment, noise removal, and feature enhancement. These functions can be applied to both grayscale image and binary image.

- **Contrast & Brightness**

  Modifies the contrast and brightness of the image. This modification is similar to LUT adjustment.

  - The unit of adjustment is percentage. The adjustment ranges from -100 to 100.

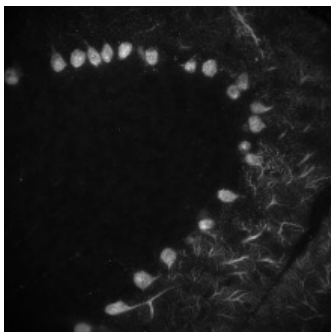

Fig. 3-1 Original image

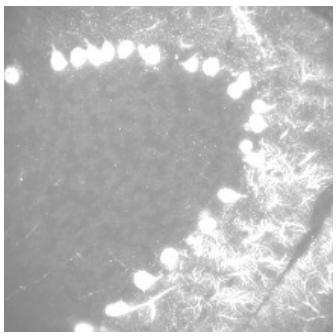

Fig. 3-2 High contrast and brightness

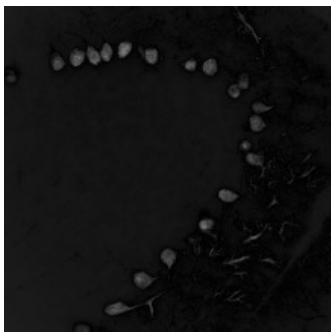

Fig. 3-3 Low contrast and brightness

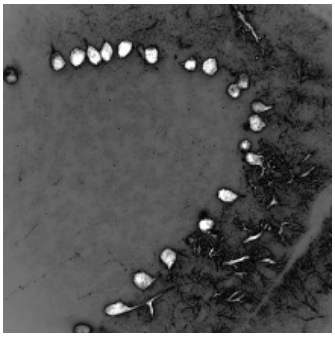

Fig. 3-4 High contrast with low brightness

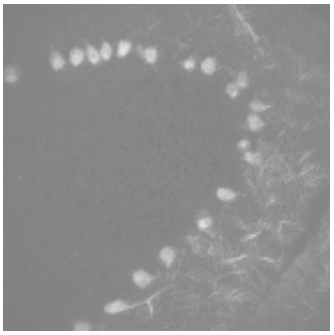

Fig. 3-5 Low contrast with high brightness

- **Gamma Adjustment**

  Gamma adjustment is a non-linear luminance mapping method used to modify the distribution of brightness values in an image. It applies a power function to redefine the relationship between input and output intensity levels, expressed as:

  $$y = x^{y}$$

  where x is the normalized input intensity (typically in the range 0–1), and y is the gamma value. Unlike linear operations such as brightness or contrast adjustment, gamma correction primarily affects the mid-tone luminance distribution, improving tonal balance and making the image appearance closer to human visual perception.

  - When 0<y<1, the mapping brightens the image and enhances shadow details by expanding darker intensity levels.
  - When y>1, the image becomes darker, compressing mid-tone and shadow regions to emphasize highlights and contrast.

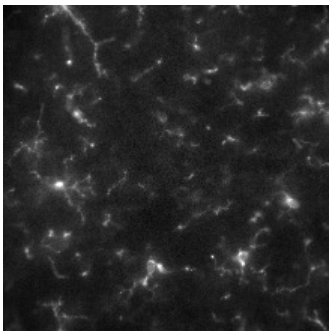

Fig. 4-1 Original image

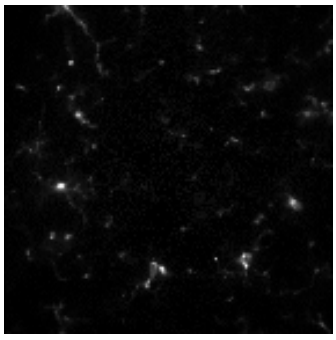

Fig. 4-2 High gamma

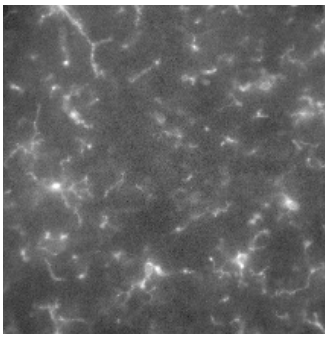

Fig. 4-3 Low gamma

- **Morphological Operation I**

  Morphological operation I is a set of operations in image processing that utilize the mathematical morphological operators to alter the shape, size, and connectivity of objects within an image.

  - Morphological processing relies on a structuring element, often referred to as a kernel, which defines how the operation interacts with an image. The shape of this kernel influences the transformation results, as different structures prioritize specific directions or patterns.
    - "Rectangle", "Cross" and "Ellipse" are the three types of shapes that can be selected. (Fig 5 shows examples of the three shapes at 5x5 pixel size)

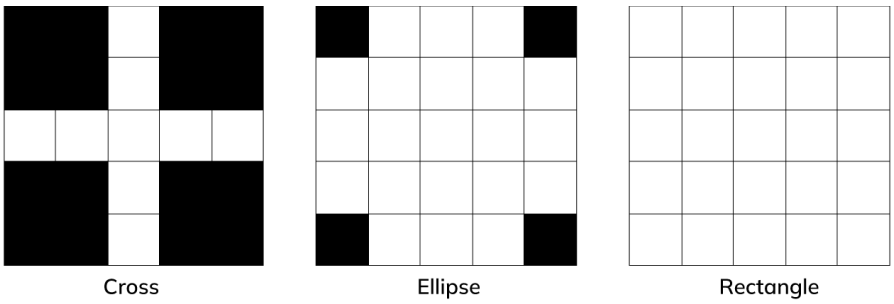

Fig 5. Example of the kernel shape

  - Parameter "Morph Method" are methods for image morphology processing and include dilation, erosion, opening, closing, top-hat and black-hat.
  - Parameter "kernel size" represents the number of pixels that would be considered for the function calculation. It should be a positive odd number >= 3.
  - Parameter "iteration" indicates the number of times the method is repeated.
  - Parameter suggestions:
    - Since morphology method would change the shape and property of the image, it is advisable to start with a "Cross" shape, small kernel size and less iterations for initial trials.
    - Dilation operation expands the high intensity area.

- Erosion operation shrinks the high intensity area.
- Opening operation performs the erosion operation first, followed by the dilation operation.
- Closing operation performs the dilation operation first, followed by the erosion operation.
- Top-hat operation is performed by subtracting the result of opening operation from the original image.
- Black-hat operation is performed by subtracting the result of closing operation from the original image.
- Top-hat and black-hat are common methods used for handling non-binary image processing.

  - Read more: https://en.wikipedia.org/wiki/Mathematical_morphology

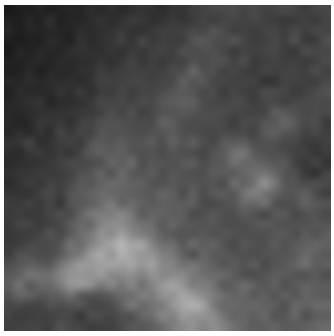

Fig. 6-1 Original image

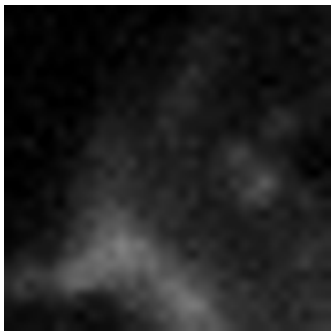

Fig. 6-2 Top-hat with kernel size = 7

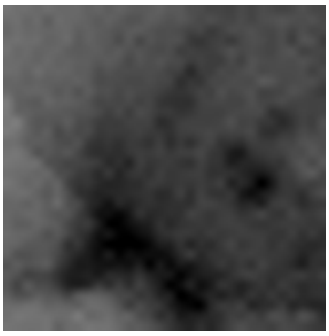

Fig. 6-3 Black-hat with kernel size = 7

- **Histogram Equalization**

  Histogram equalization is a contrast enhancement technique that redistributes the intensity values of an image across the full dynamic range. By adjusting the pixel intensities, this method enhances the visibility of details in both bright and dark regions, making the overall image clearer and more balanced. It is particularly useful for images with poor contrast, where important features may be obscured due to uneven lighting or limited intensity variations. By spreading out the most frequently occurring intensity values, histogram equalization improves the distinction between different areas of the image, aiding in tasks such as object detection, feature extraction, and image preprocessing.

  - Read more: https://en.wikipedia.org/wiki/Histogram_equalization

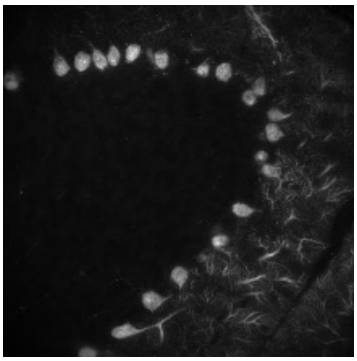

Fig. 7-1 Original image

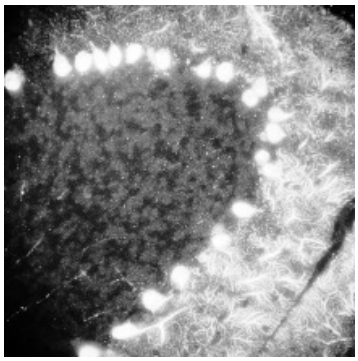

Fig. 7-2 Equalization

- **Contrast limit Equalization (CLAHE)**

  Contrast Limited Adaptive Histogram Equalization (CLAHE) is an advanced histogram equalization technique that enhances local contrast while preventing over-amplification of noise. Unlike traditional histogram equalization, which applies the transformation uniformly across the entire image, CLAHE operates on small regions (tiles) and redistributes pixel intensities within each tile independently. To avoid excessive contrast enhancement, a contrast limit is applied, ensuring that noise and artifacts are not exaggerated. The resulting image is then smoothly blended to eliminate artificial boundaries between tiles. CLAHE is particularly useful for improving details in low-contrast images while maintaining natural transitions, making it ideal for medical imaging, remote sensing, and other applications requiring controlled contrast enhancement.

  - Default parameter value: limit = 1
  - Read more: https://en.wikipedia.org/wiki/Adaptive_histogram_equalization#Contrast_Limited_AHE

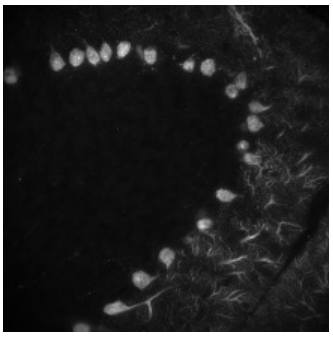

Fig. 8-1 Original image

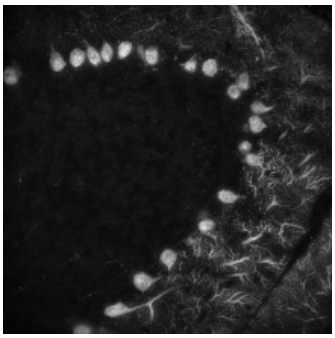

Fig. 8-2 Clip limit = 1.0

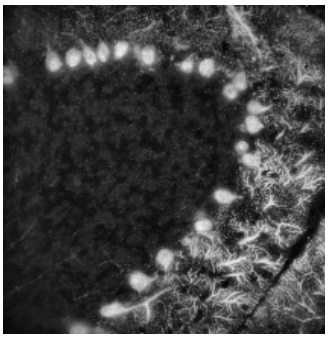

Fig. 8-3 Clip limit = 5.0

- **Median Filter**

  A Median Filter is a nonlinear filtering technique used in image processing to reduce noise while preserving edges. It works by replacing each pixel's value with the median of the intensity values within a defined neighborhood (kernel size). Unlike linear filters, which can blur edges, the median filter effectively removes salt-and-pepper noise by maintaining sharp transitions between different regions. This makes it particularly useful for applications where preserving structural details is important, such as medical imaging, object detection, and image restoration. The choice of kernel size determines the level of smoothing, with larger sizes removing more noise but potentially affecting finer details.

  - Parameter "kernel size" represents the number of pixels that would be considered for the function calculation. It should be a positive odd number more than 3.
  - Parameter suggestion:
    - A bigger kernel size results in a smoother output image.
    - A smaller kernel size is recommended when the ROI of the processing image is small.
  - Read more: https://en.wikipedia.org/wiki/Median_filter

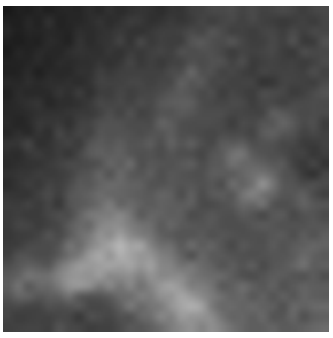

Fig. 9-1 Original image

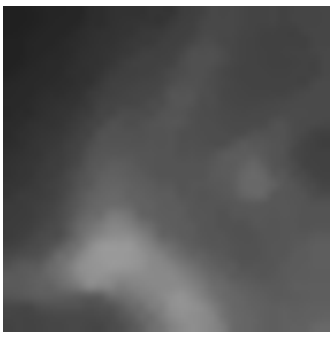

Fig. 9-2 kernel size = 5

Fig. Fig. 9-3 kernel size = 11

- **Gaussian Filter**

  A Gaussian Filter is a linear smoothing filter used in image processing to reduce noise and blur an image while preserving overall structure. It applies a Gaussian function to assign weights to neighboring pixels, giving higher importance to pixels closer to the center of the kernel and gradually decreasing the influence of those further away. This results in a smooth, natural-looking blur that effectively reduces high-frequency noise while maintaining gradual transitions between intensity variations. The degree of blurring is controlled by the standard deviation (σ) of the Gaussian function, with larger values producing stronger blurring effects. Gaussian filtering is commonly used as a preprocessing step in edge detection, feature extraction, and image analysis to enhance image quality and reduce unwanted details.

  - Parameter “kernel size” represents the number of pixels that would be considered for the function calculation. It should be a positive odd number more than 3.
  - Parameter suggestion:
    - A bigger kernel size would result in a smoother output image.
    - A small kernel size is suggested when the ROI in the processing image is small.
  - Read more: https://en.wikipedia.org/wiki/Gaussian_filter

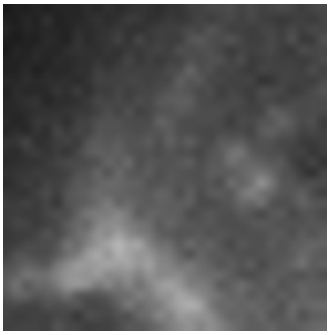

Fig. 10-1 Original image

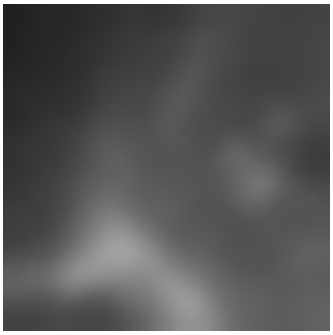

Fig. 10-2 kernel size = 5

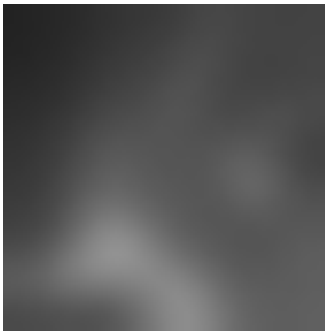

Fig. 10-3 kernel size = 11

- **Sharpen image**

  Image sharpening enhances the edges and details in an image, making it appear clearer and more defined. This is achieved by increasing the contrast along the edges, which helps to highlight fine details.

  - Parameter suggestion:
    - Higher values (closer to 10.0) increase the intensity of sharpening, but excessive sharpening can lead to unnatural-looking images with halos or noise. Adjust the parameter based on the desired level of detail and the quality of the original image.

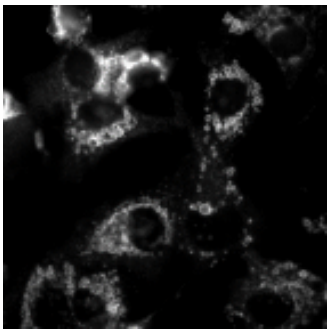

Fig. 11-1 Original image

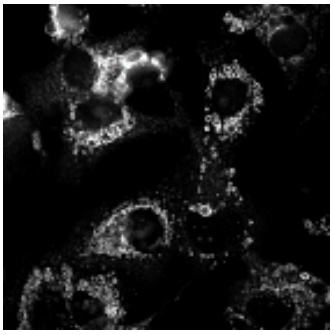

Fig. 11-2 sharpness = 3.0

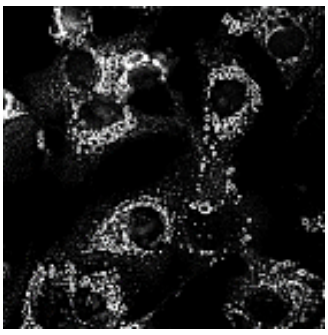

Fig. 11-3 sharpness = 10.0

- **Sobel Transform**

  The Sobel Transform is an edge detection operator used in image processing to highlight the edges within an image by detecting changes in intensity. It works by applying two convolution kernels — one for detecting horizontal edges and one for vertical edges. These kernels are used to compute the gradient of the image intensity at each pixel, which represents the rate of change in pixel values in both the horizontal and vertical directions. The result is two gradient images, one for the horizontal edges (Gx) and one for the vertical edges (Gy).

  The magnitude of the gradient is then computed by combining the horizontal and vertical gradients using the formula: $Magnitude = \sqrt{(G_x^2 + G_y^2)}$

  This combined gradient magnitude highlights the edges, where high values correspond to strong edges in the image. The Sobel transform is particularly effective in detecting edges at different orientations and is widely used in tasks like object detection, image segmentation, and feature extraction.

  - Read more: https://en.wikipedia.org/wiki/Sobel_operator

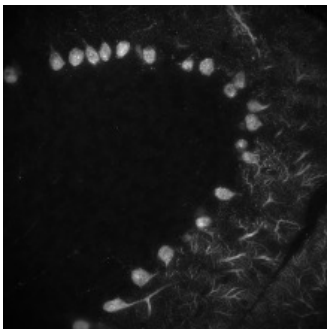

Fig. 12-1 Original image

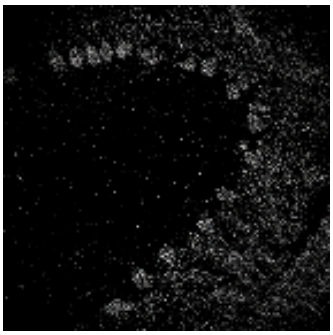

Fig. 12-2 Sobel transformation

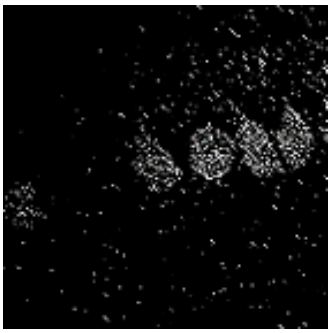

Fig. 12-3 zoom-in

- **Laplacian Transform**

  Laplacian Transform is a second-order derivative operator used in image processing to detect edges and highlight regions of rapid intensity change. Unlike the Sobel operator, which calculates the gradient of an image, the Laplacian measures the rate of change in the gradient itself, providing information about the curvature of the intensity function.

  The Laplacian operator works by applying a convolution kernel that approximates the second derivative of the image intensity. This highlights areas where the intensity changes sharply, typically corresponding to edges or boundaries between objects. The result is a map of regions with high curvature, where edges and transitions between light and dark areas are accentuated.

  The Laplacian transform is particularly useful for detecting edges in images, but it is more sensitive to noise than first-order derivatives like the Sobel operator. To mitigate this, the Laplacian is often combined with a Gaussian filter (as in the Laplacian of Gaussian, or LoG) to smooth the image before applying the transformation. The Laplacian is commonly used in edge detection, object recognition, and feature extraction.

  - Read more: https://en.wikipedia.org/wiki/Laplace_transform

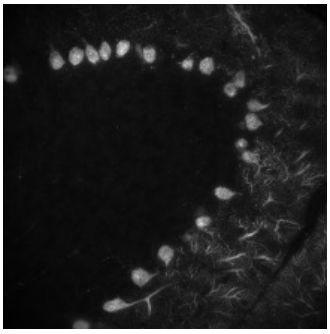

Fig. 13-1 Original image

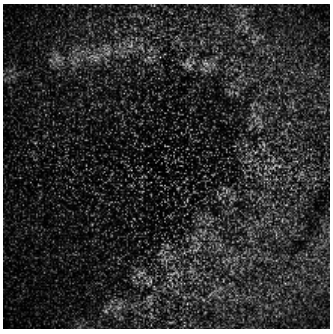

Fig. 13-2 Laplacian transformation

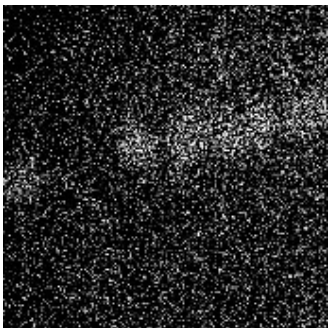

Fig. 13-3 zoom-in

- **Hough Transform**

  The Hough Transform is used to detect and extract circles and ellipses from digital images by representing them as parameterized equations and transforming them into a parameter space.

  - It should be noted that this function will only draw the fitted circle for the detected target, not the edge of the target.
  - The parameter "min dist" represents the minimum distance between the centers of detected circles.
  - Canny Edge Threshold: The resolution of the parameter rho in pixels, which determines the distance resolution of the accumulator space.
  - Accumulator Threshold: The resolution of the parameter theta in radians, which determines the angular resolution of the accumulator space.
  - The "min radius" and "max radius" are optional parameters used in the multi-scale Hough transform for detecting circles and ellipses.
  - Parameter suggestions:
    - para1 = 100 to 200 and para2 = 10 to 50 are generally suitable for high-intensity targets.
  - The function may require more processing time when both para1 and para2 are smaller than 50.
  - Read more:
    - https://en.wikipedia.org/wiki/Circle_Hough_Transform
    - https://youtu.be/4zHbI-fFIlI

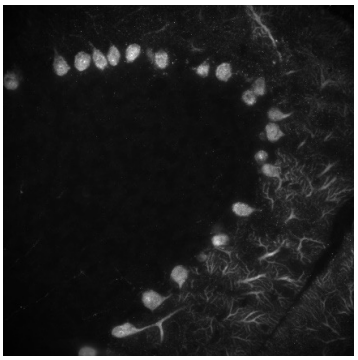

Fig. 14-1 Original image

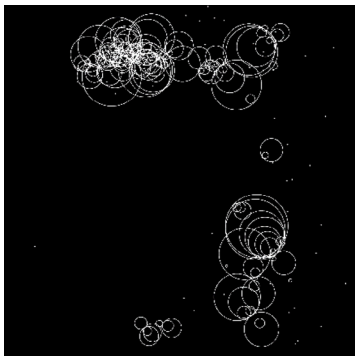

Fig. 14-2 Hough transform

1.2.2 Binarization (Thresholding)

Binarization is a simple image processing technique used to convert a grayscale or color image into a binary image.

- **Basic Thresholding**

  Simple thresholding is a technique used in image processing to convert a grayscale image into a binary image by assigning a threshold value that separates pixels into black and white.

  - Parameter “threshold type” can used to choose the method to be used.
    - BINARY: turn the pixel white if pixel value higher than threshold and black if pixel lower than threshold
    - BINARY_INV: turn the pixel black if pixel value higher than threshold and white if pixel lower than threshold
    - BINARY_OTSU: the image will be binarized by the threshold value which is determined by Otsu algorithm
  - The range of Parameter “threshold value” is 0~255.
  - Except the Otsu’s method, other methods allow manual setting of the Parameter “threshold value”.
  - Parameter suggestion:
    - Except the Otsu’s method, a threshold value of 127 can be used as the starting value for mask generation for checking which value is suitable for used.
  - Read more: https://en.wikipedia.org/wiki/Thresholding_(image_processing)

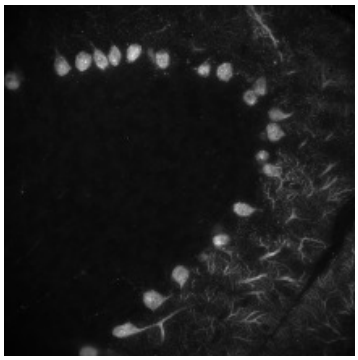

Fig. 15-1 Original image

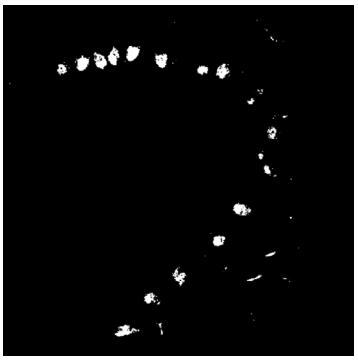

Fig. 15-2 Binary threshold = 127

Fig. 15-3 Binary inverse threshold = 127

Fig. 15-4 Otsu method (auto threshold)

- **Adaptive Thresholding**

  Adaptive thresholding is an image processing technique that determines threshold values dynamically based on local image characteristics. Unlike global thresholding, which applies a single threshold across the entire image, adaptive thresholding adjusts to variations in lighting and contrast, making it effective for images with uneven illumination or complex backgrounds. This method is commonly used in applications like document binarization, medical imaging, and object segmentation, where consistent feature extraction is required despite varying image conditions.

  - The parameter "kernel size" defines the number of neighboring pixels considered during the calculation. It must be a positive odd number, with a minimum value of 3.
  - Parameter suggestions:
    - The kernel size should be chosen based on the expected size of the region of interest (ROI).
    - The "offset" parameter is recommended to be set to 0 initially. It is used for fine-tuning the thresholding process, and negative values are allowed. A suggested range for the offset is between -2 to -15.
  - Read more: https://en.wikipedia.org/wiki/Thresholding_(image_processing)#Global_vs_local_thresholding

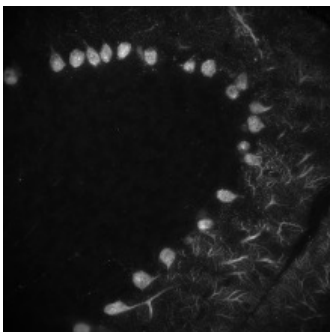

Fig. 16-1 Original image

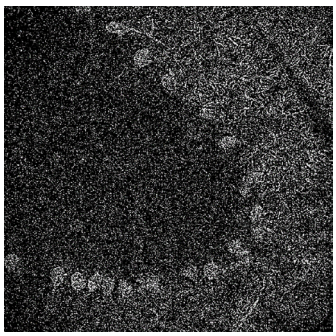

Fig. 16-2 Kernel size = 5, offset = 0

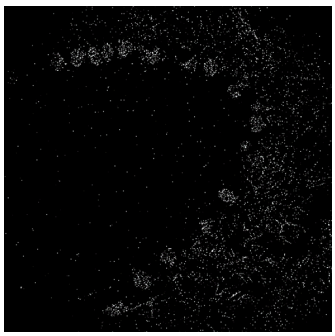

Fig. 16-3 Kernel soze = 5, offset = -2

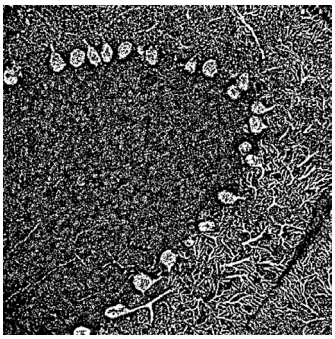

Fig. 16-4 Kernel size = 51, offset = 0

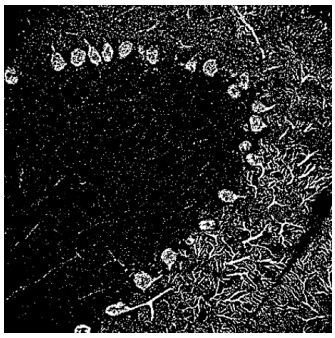

Fig. 16-5 Kernel size = 51, offset = -2

1.2.3 Binary Image Processing

Binary image processing involves the manipulation and analysis of images that consist of only two colors, typically black and white. It focuses on the application of operations and algorithms specifically designed for binary images.

- **Morphological Operation II**

  Morphological operation II is a set of operations in image processing that modify the shape, size, and connectivity of objects in an image using mathematical morphological operators.

  - Morphological processing relies on a structuring element, often referred to as a kernel, which defines how the operation interacts with an image. The shape of this kernel influences the transformation results, as different structures prioritize specific directions or patterns.
    - Rectangle, cross and ellipse may be chosen. (Fig 5 shows examples of the three shapes at 5x5 pixel size)
  - Parameter “Morph Method” are methods for image morphology processing, which include:
    - Dilation, erosion, opening, closing, top-hat and black-hat.
    - Dilation, erosion, opening, and closing are common methods used to handle binary image processing.
  - Parameter “kernel size” refers to the number of pixels that would be considered for the function calculation. It should be a positive odd number more than 3.
  - Parameter “iteration” refers to the number of times the method is repeated.
  - Parameter suggestions:
    - Since the morphology method changes the shape and property of the image, it is better to use “Cross”, small kernel size and less iterations to begin with.
    - Dilation operation expands the high intensity area.
    - Erosion operation shrinks the high intensity area.
    - Opening operation performs the erosion operation first, followed by the dilation operation.
    - Closing operation performs the dilation operation first, followed by the erosion operation.
    - Top-hat operation is performed by subtracting the result of the opening operation from the original image.
    - Black-hat operation is performed by subtracting the result of the closing operation from the original image.
    - It is suggested to use dilation, erosion, opening, closing methods for processing binary images.
  - Read more: https://en.wikipedia.org/wiki/Mathematical_morphology

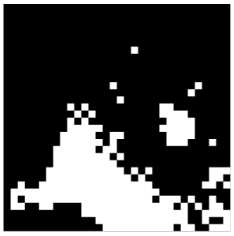

Fig. 17-1 Original binary image

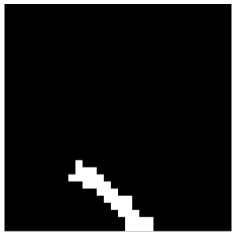

Fig.17-2 Erosion with kernel size = 7

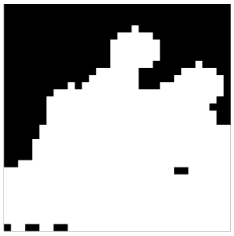

Fig. 17-3 Dilation with kernel size = 7

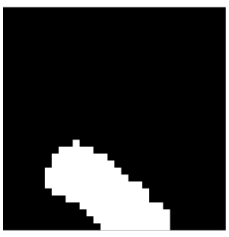

Fig. 17-4 Opening with kernel size = 7

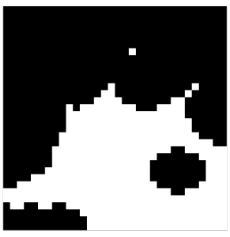

Fig.17-5 Closing with kernel size = 7

- **Hole Filling**

  Fill the black hole in the white area for the mask image.

Fig. 18-1 Original binary image

Fig. 18-2 Fill hole

- **Size Filter**

  Filter out or keep the white area according to the size of the area.

  - The unit of size threshold is the target area pixel size.
  - Parameter “size upper bound” is the maximum pixel size threshold for the extraction of each ROI in mask pattern.
  - Parameter “size lower bound” is the minimum pixel size threshold for the extraction of each ROI in mask pattern.

Fig. 19-1 Original binary image

Fig. 19-2 Size filter

- **Circular Fit**

  Mark the center of each white area in the image

Fig. 20-1 Original binary image

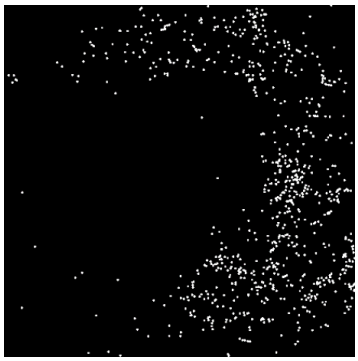

Fig. 20-2 Mark center

- **Direction Filter**

  A Direction Filter for bounding box direction in image processing is used to filter objects based on the orientation of their bounding boxes. This filter evaluates the angle of the bounding box relative to x-axis and retains only those objects that fall within a specified directional range.

  - Parameter “target angle” refers to the direction in which the mask needs to be extracting out. 0 represents a horizontal direction of extraction.
  - Parameter “angle div” denotes the acceptable amount of angular displacement.
  - Parameter “eliminate or keep” refers to the choice to filter out or keep the extracted mask.

Fig. 21-1 Original binary image

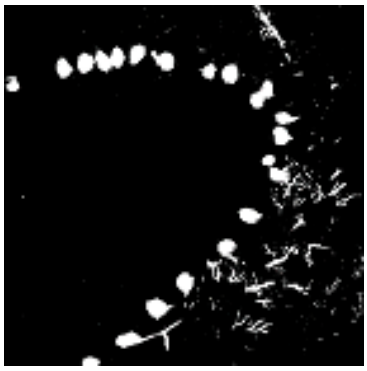

Fig. 21-2 Direction filter with target angle=0, angle div=15

- **Aspect ratio Filter**

  An Aspect Ratio Filter in image processing is used to filter objects or regions based on their bounding box aspect ratio, which is the ratio of width to height. This filter helps in selecting or excluding objects that meet a specific shape criterion, ensuring that only objects with the desired proportions are retained.

  - Each isolated white area can find a minimum bounding rectangle. The aspect ratio can be calculated by the rectangle.
  - The checkbox can switch if the object which fit the condition should be removed or remain.

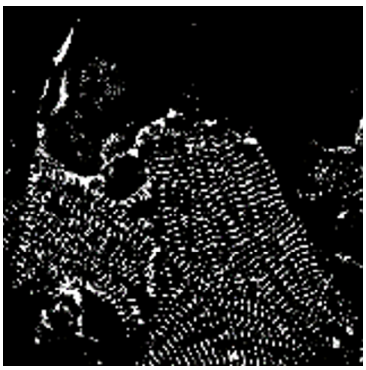

Fig. 22-1 Original image

Fig. 22-2 low boundary=3, high boundary=100 with no filter inverse

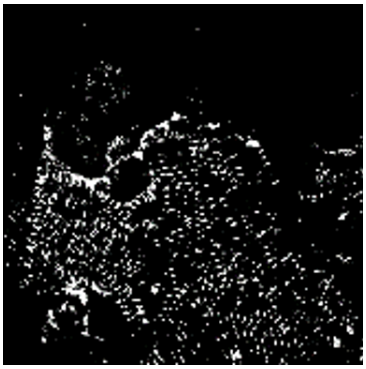

Fig. 22-3 low boundary=3, high boundary=100 with filter inverse

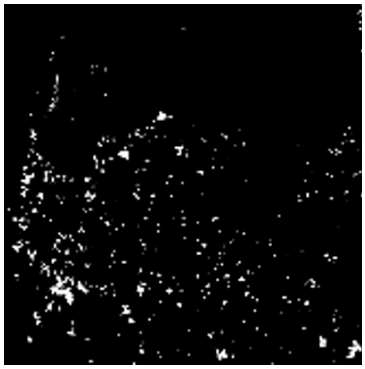

Fig. 22-4 low boundary=1, high boundary=100 with no filter inverse

1.2.4 Math Operation

Arithmetic operations (addition, subtraction) are pixel calculations performed on individual pixels. These operations allow the manipulation and combination of pixel values to achieve the desired effects in image processing and analysis.

- **Additive Merge**

  Add the value of each pixel with that of another image at the same localization or a constant value, as shown in the example below.

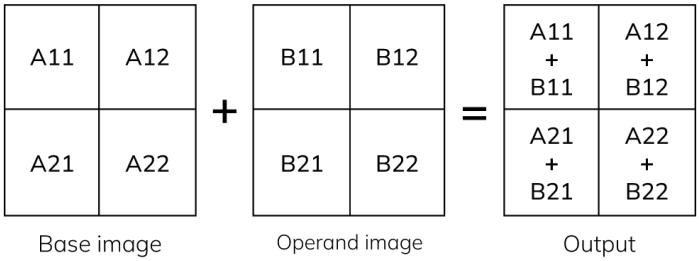

Fig. 23 Add operation by pixel to merge images

- **Subtractive Merge**

  Subtract the value of each pixel with that of another image at the same localization or a constant value, as shown in the example below.

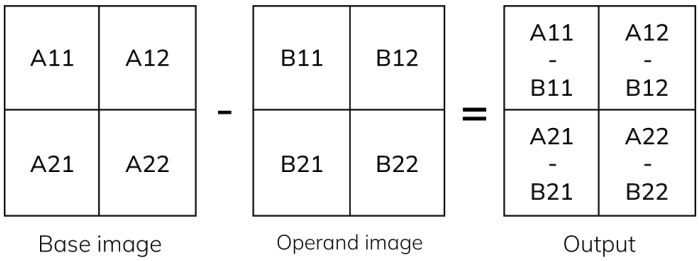

Fig. 24 Subtract operation by pixel to merge images

- **Multiplicative Merge**

  Multiply the value of each pixel with that of another image at the same localization or a constant value, as shown in the example below.

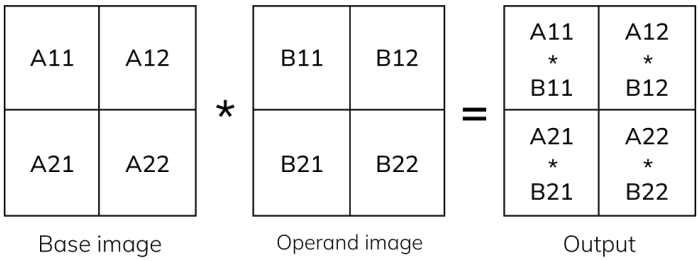

Fig. 25 Multiply operation by pixel to merge images

- **Divisive Merge**

  Divide the value of each pixel with that of another image at the same localization or a constant value, as shown in the example below.

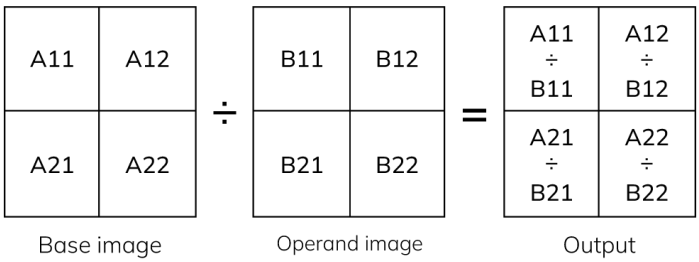

Fig. 26 Divide operation by pixel to merge images

- **Image Constant Addition**

  Add the value of each pixel with a constant value, as shown in the example below.

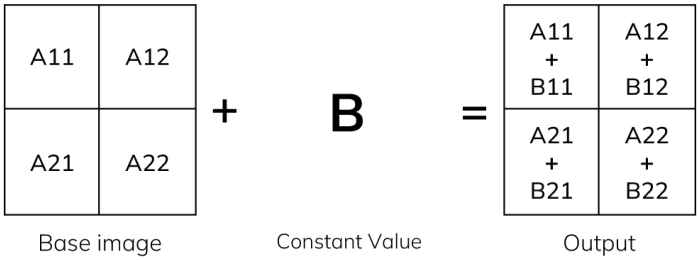

Fig. 27 Add operation by pixel

- **Image Constant Subtraction**

  Subtract the value of each pixel with a constant value, as shown in the example below.

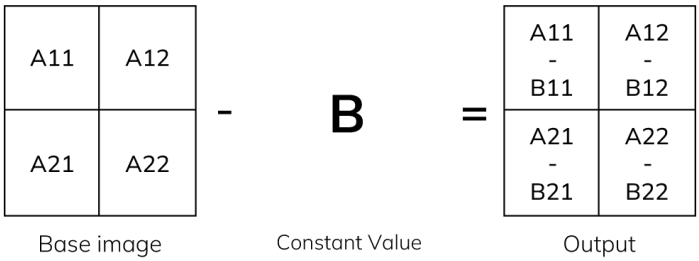

Fig. 28 Subtract operation by pixel

- **Image Scalar Multiplication**

  Multiply the value of each pixel with a constant value, as shown in the example below.

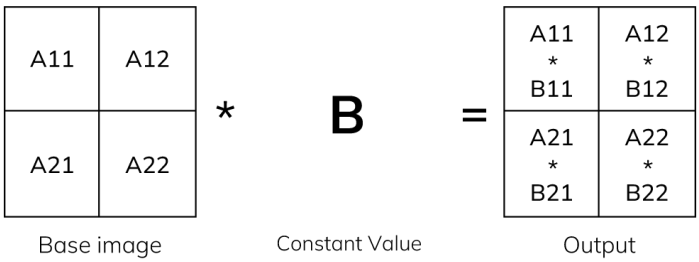

Fig. 29 Multiply operation by pixel

- **Image Scalar Division**

  Divide the value of each pixel with a constant value, as shown in the example below.

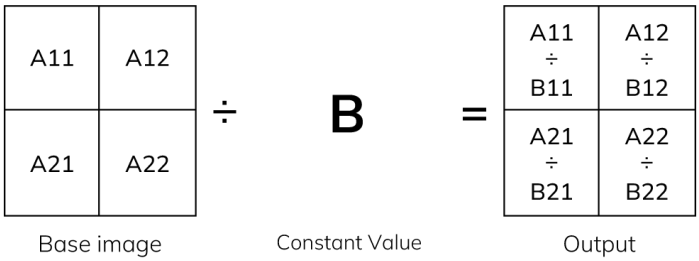

Fig. 30 Divide operation by pixel

1.2.5 Binarized Image Logical Operation

Performs logical operations between two binarized images to combine or compare their masks.

Depending on the selected mode (e.g., AND, OR, XOR, NOT), the resulting image highlights overlapping, unique, or inverted regions, allowing flexible mask manipulation and comparison.

- **Intersection**

  Merge two masks with "and" operation by pixel. The result will keep the overlapping pixels in the two masks.

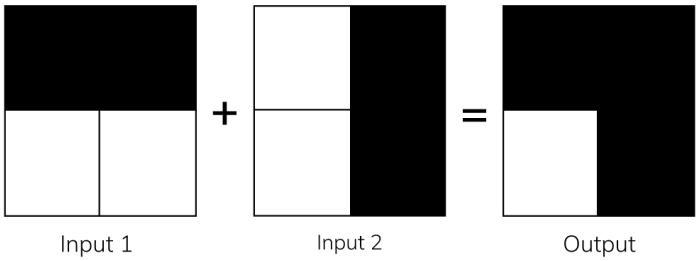

Fig. 31 “AND” logic operation by pixel

- **Union**

  Merge two masks with "or" operation by pixel. The result will be the union area of the two masks.

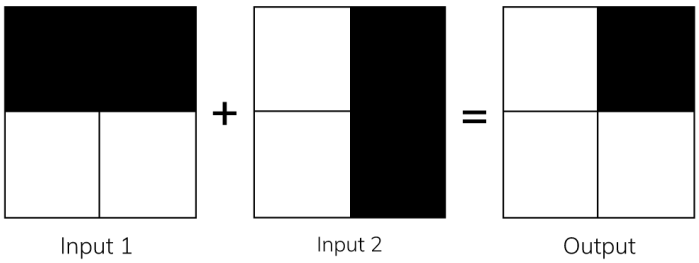

Fig. 32 “OR” logic operation by pixel

- **Inverse**

  The result of the "not" operation of the mask by pixel will be the reversing of the black and white areas of the mask.

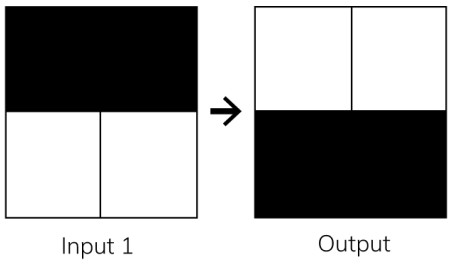

Fig. 33 “NOT” logic operation by pixel

- **Exclusive Difference**

  Merge two masks with "xor" operation by pixel. The result will highlight the differing regions between the two masks, while suppressing their overlapping areas.

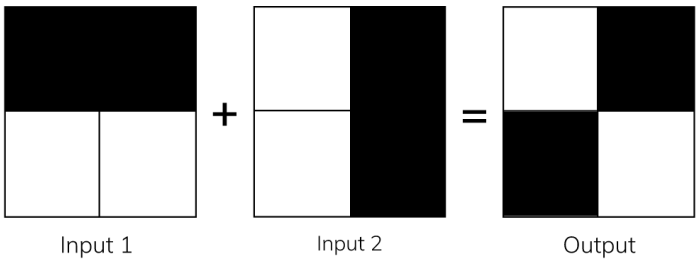

Fig. 34 “XOR” logic operation by pixel

1.2.6 Advanced Processing Function

Many capabilities within the advanced function class are composed of multiple intricate functions, necessitating careful consideration during utilization. Users should exercise diligence in configuring and confirming the intricate combinations of functions to achieve effective outcomes in the desired image processing.

- **Particle Extraction**

  Particle extraction is an algorithm to extract the small particles in the input image with relatively high intensity.

  - Parameter “kernel size” refers to the number of pixels that would be considered for this function calculation. It should be a positive odd number and >=3.
  - Parameter “threshold” controls the intensity of the extraction.
  - Parameter suggestions:
    - The larger the particle for extraction, the larger should be the “kernel size”.
    - It is suggested that the “threshold” should begin with 0.

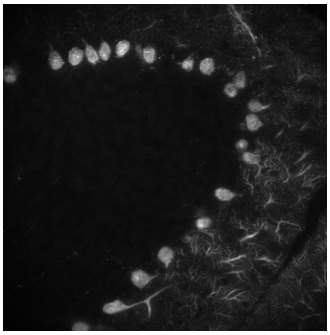

Fig. 35-1 Original image

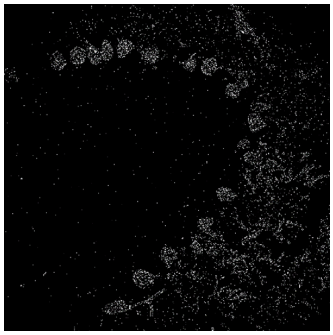

Fig. 35-2 Canny edge detection

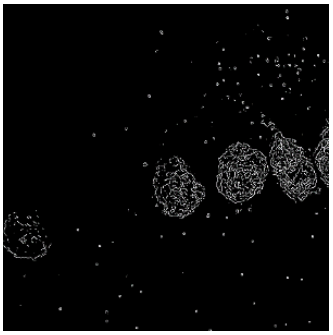

Fig. 35 Zoom-in

- **Focus Area Segmentation**

  Focus Area Segment isolates regions of the image that are considered in focus based on local intensity transitions. This process highlights the most visually sharp regions while maintaining flexibility in how tightly or broadly the focus mask is defined.

  - The Focus Level (0–1) controls the sensitivity of focus detection — higher values make the segmentation more selective, emphasizing sharper and more distinct regions.
  - The Dilation Kernel defines the extent of expansion applied to the detected focus areas, helping to connect nearby regions or smooth the segmented boundaries.

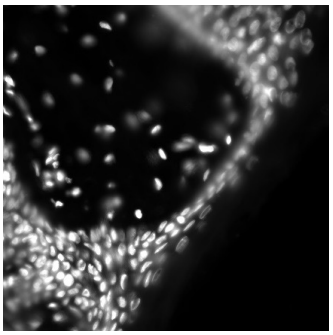

Fig. 36-1 Original image

Fig. 36-2 low focus level

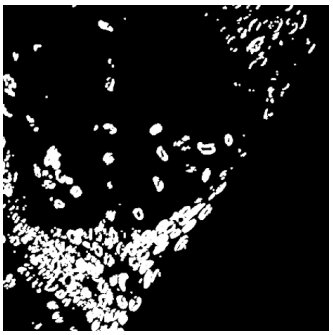

Fig. 36-3 High focus level

- **Contour Extraction**

  Contour extraction is an algorithm which draws the contours for all the white areas in the input binary image.

  - Note: this function will only draw the contour for the white area of a binary image, hence the input image should be a binary image.

- **Canny Edge Detection**

  Canny Edge Detection is a widely used edge detection technique in image processing that identifies the boundaries or edges of objects within an image by detecting areas of rapid intensity change. It was developed by John F. Canny in 1986 and is known for its effectiveness in detecting sharp, clean edges while minimizing noise.

  The process involves several key steps:

  a. Noise Reduction: The image is first smoothed using a Gaussian filter to reduce noise, as noise can lead to false edge detection.

  b. Gradient Calculation: The algorithm then calculates the gradient of the image using filters such as the Sobel operator. The gradient measures the intensity change at each pixel and helps identify potential edges.

  c. Non-Maximum Suppression: After calculating the gradient, non-maximum suppression is applied to thin out the edges. This step ensures that only the strongest edges are retained and that the edges are one-pixel wide.

  d. Edge Tracing by Hysteresis: This step involves two threshold values— a high threshold and a low threshold. Strong edges above the high threshold are immediately marked as edges, while weaker edges are only considered if they are connected to strong edges. This step ensures that the algorithm detects true edges and eliminates spurious results.

  The result of the Canny edge detection process is an image that highlights the edges of objects, making it easier to analyze the structure and shape of features within the image.

  - It should be noted that this function will only draw the edge by the image.
  - Parameter “threshold_1” is gradient magnitude and Parameter “threshold_2” is hysteresis thresholding.
  - Parameter suggestion: Test Hysteresis lower thresholding and Hysteresis upper thresholding start from 80, and then decrease to generate more result output.
  - Read more: https://en.wikipedia.org/wiki/Canny_edge_detector

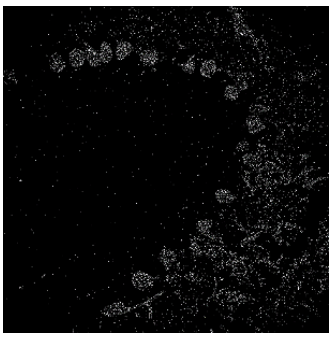

Fig. 37-1 Original image

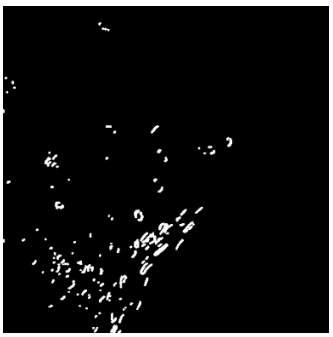

Fig. 37-2 Canny edge detection

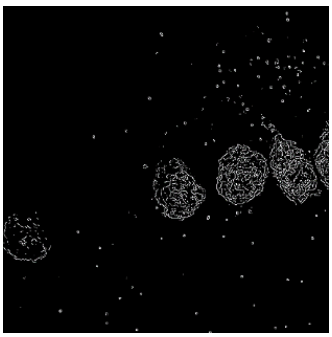

Fig. 37-3 Zoom-in

- **Watershed Algorithm**

  The Watershed Algorithm is a powerful image segmentation technique used to partition an image into distinct regions based on the topology of intensity levels. It treats the image as a topographic surface, where intensity values represent the height of the surface. The basic idea is to simulate flooding the surface from markers placed in the regions of interest, allowing the boundaries of flooded areas to define the segments.

  In the watershed process, markers are placed at local minima (or predefined regions) in the image, and the algorithm gradually "floods" the image from these markers. The flooding process is akin to water rising in a landscape, where basins represent regions of similar intensity. When the water

levels from different basins meet, a boundary (or watershed line) is formed, effectively dividing the image into distinct regions.

This algorithm is especially useful in situations where clear, well-defined objects or boundaries exist in the image. It's widely used for applications such as object detection, cell segmentation, and image analysis where regions need to be separated based on intensity or feature contrast.

However, the watershed algorithm can be sensitive to noise and over-segmentation, which is why preprocessing steps like noise reduction and proper marker selection are often used to improve the results. It can also be combined with other methods like morphological operations to refine the segmentation.

- It should be noted that this function will only draw the edge by the image.
- The Input foreground image needs to be the grayscale image (original image), while the input background image needs to be the binary image (mask image).
- Read more: https://en.wikipedia.org/wiki/Watershed_(image_processing)

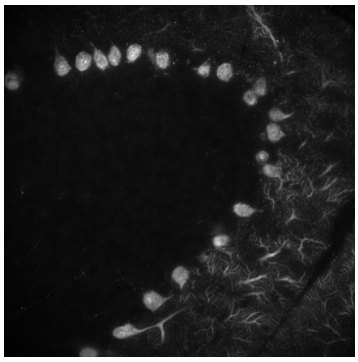

Fig. 38-1 Grayscale image

Fig. 38-2 Marker Mask

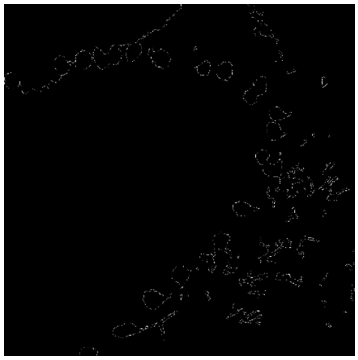

Fig. 38-3 Watershed

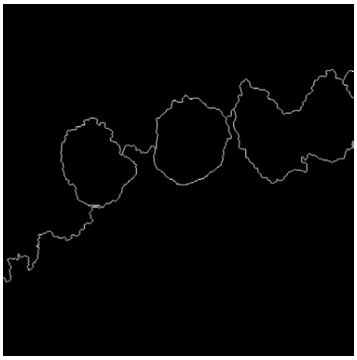

Fig. 38-4 Zoom-in

- **Distance Transformation**

  Distance transformation is a technique used to compute the distance of each pixel in a binary image to the nearest zero pixel (background). It is commonly used in image processing for tasks like shape analysis, object detection, and skeletonization.

  - Input image must be a binary image.

Fig. 39-1 Original binary image

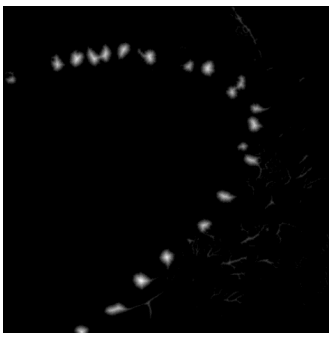

Fig. 39-2 Distance transformation

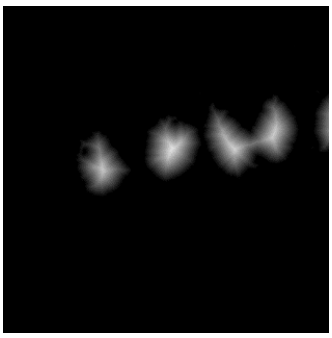

Fig. 39-3 Zoom-in

- **Blob Detection**

  Blob detection is a technique in image processing used to identify and locate regions in an image that differ in properties, such as intensity or texture, compared to their surroundings. These regions, or "blobs," can represent objects or features of interest, such as bright spots, dark regions, or other distinctive patterns. The primary goal of blob detection is to identify areas that are significantly different from their neighboring pixels based on certain criteria, such as intensity changes or texture variations.

  Blob detection typically involves analyzing the image at different scales to capture blobs of various sizes. Methods like Laplacian of Gaussian (LoG), Difference of Gaussian (DoG), or simple thresholding are often used to detect blobs. These methods work by filtering the image to highlight areas with strong contrasts, followed by applying a threshold to identify the blobs. The detected blobs can be characterized by their size, shape, and location, which can be useful in applications like object recognition, feature tracking, and image segmentation.

  Blob detection is widely used in computer vision tasks such as tracking moving objects, detecting shapes, or identifying regions of interest for further analysis.

  - Please note that this function will only draw the circle by the blob in the image.
  - "min pixel radius": The minimum threshold value for the pixel radius.
  - "max pixel radius": The maximum threshold value for the pixel radius.
  - "area filter": A flag indicating whether blobs should be filtered based on their area.
  - "min area": The minimum area of blobs to be considered.
  - "circularity filter": A flag indicating whether blobs should be filtered based on their circularity.
  - "min circularity": The minimum circularity of blobs to be considered.
  - "convexity filter": A flag indicating whether blobs should be filtered based on their convexity.
  - "min convexity": The minimum convexity of blobs to be considered.
  - "inertia filter": A flag indicating whether blobs should be filtered based on their inertia.
  - "min inertia ratio": The minimum inertia ratio of blobs to be considered.
  - Read more: https://en.wikipedia.org/wiki/Blob_detection

Fig. 40-1 Original binary image

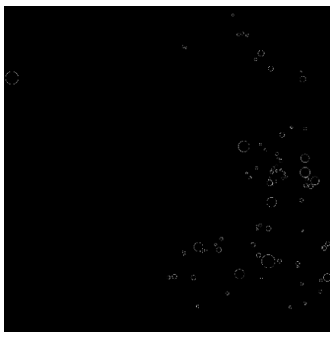

Fig. 40-2 Blob detection

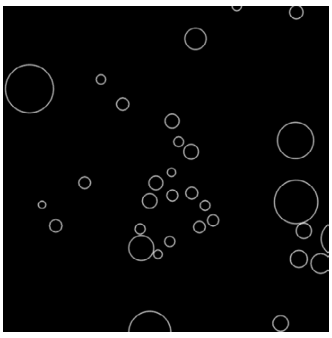

Fig. 40-3 Zoom-in

1.2.7 AI Plugin Mask Inference

- **AI plugin function (custom module)**

  Please be aware that the order of the input channel must be the same as the order of the input channel in the trained model.

APPENDIX 2 - EXAMPLES OF IMAGE PROCESSING APPLICATIONS

2.1 Example 1 - To create the mask for the overlapping regions of two images.

The protein-protein interaction, cellular organelle association or cell-cell contact play a critical role in signal transduction, cellular homeostasis or intercellular communication. In this context, we illustrate the fundamental steps in basic image processing used to generate the mask pattern for the overlapping regions of two cellular expression structures.

Step 1: Image preprocessing (optional) (cytoplasm protein image)

The 'Image Preprocessing' function can be used to optimize the images for further mask pattern generation. The tasks include brightness/contrast adjustment, noise reduction and image features enhancement. To perform these tasks, access the 'Image Preprocessing' menu and select 'Contrast Brightness' . Choose the desired source of the raw image to be processed in the dialog box.

The parameter range of contrast or brightness is -100 to 100. The adjustment result, as shown in the right panel, demonstrates an image with enhanced brightness and contrast. While the contrast/ brightness adjustment can improve the low-intensity region of the image, it is important to note that it may also amplify noise and lead to pixel saturation in the enhanced high-intensity regions.

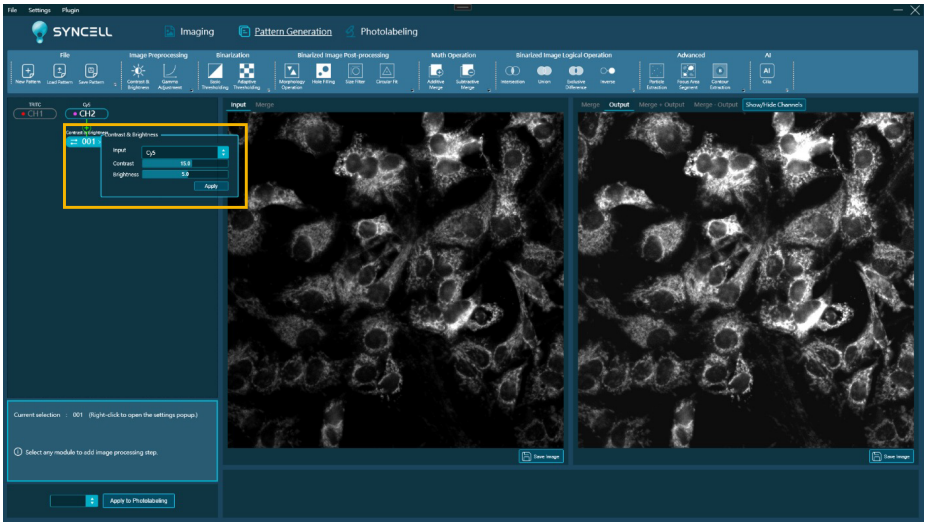

Fig. 1 Result of contrast and brightness adjustment

Considering the noise amplification and pixel saturation issue, the 'Contrast Limit Equalization' method proves more effective in improving local contrast and enhancing edge definition in each region of an image. To utilize this method, navigate to the 'Image Preprocessing' menu and select the 'Contrast Limit Equalization' item. In the next step, choose the first source image for processing.

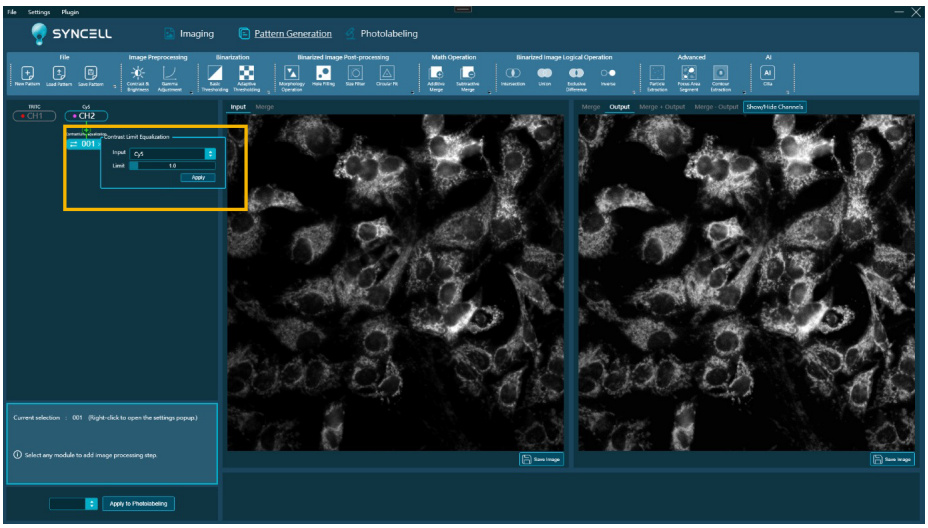

Fig. 2 Result of contrast limit equalization

The default parameter value (limit = 1) is selected in this example and the adjusted image is showed in the right panel. Different limit values may be used to achieve varying modifications that enhance local contrast while limiting noise amplification.

Step 2: Binarization (cytoplasm protein image)

The binarization processing can be used to convert the gray scale image into a binary image. The binary mask defines the region of interest (ROI) in an image. A typical approach to creating a mask from an image is to classify each pixel based on its intensity value. In the first step, you can select 'Simple Threshold' method by setting a threshold value. It retains the grayscale pixels whose intensity is above a specific threshold. In the next step, access the 'Binarization' menu and choose 'Simple Threshold'. Select the last processed image from the previous step as the source image.

You can set the 'Thresh Value' to mask pixels above this value. However, in this example, neither a low (45) nor a high (100) threshold value can adequately mask the image. This is due to the non-homogeneous expression pattern and intensity of the target protein.

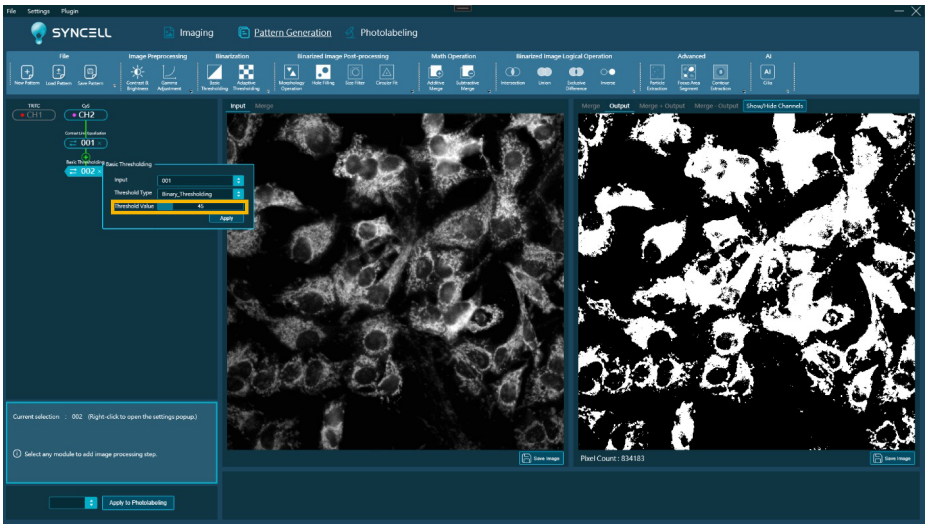

Fig. 3-1 Result of low threshold (45) by using simple thresholding

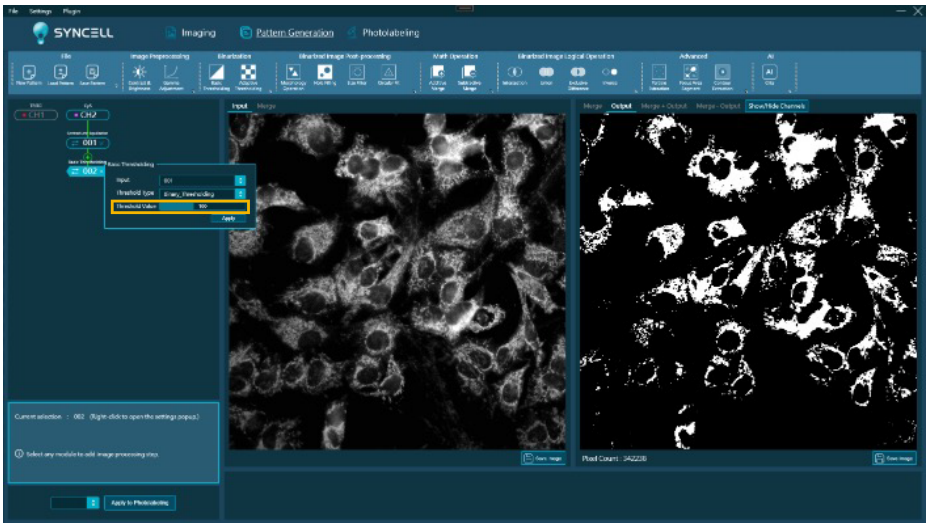

Fig. 3-2 Result of high threshold (100) by using simple thresholding

In this case, the 'Adaptive Threshold' method proves helpful in segmenting images under varying light conditions. Under the 'Binarization' menu, you can select 'Adaptive Threshold' option and load the last processed image as source image, followed by the value setting of the kernel size and offset to generate the binary mask image that fits to the expression pattern and shape of target (Figure 4). When using this method, ensure that the kernel size is a positive odd number (>=3). Negative values are permissible for the offset. It is a commonly used setting for small ROI targets with a small kernel size.

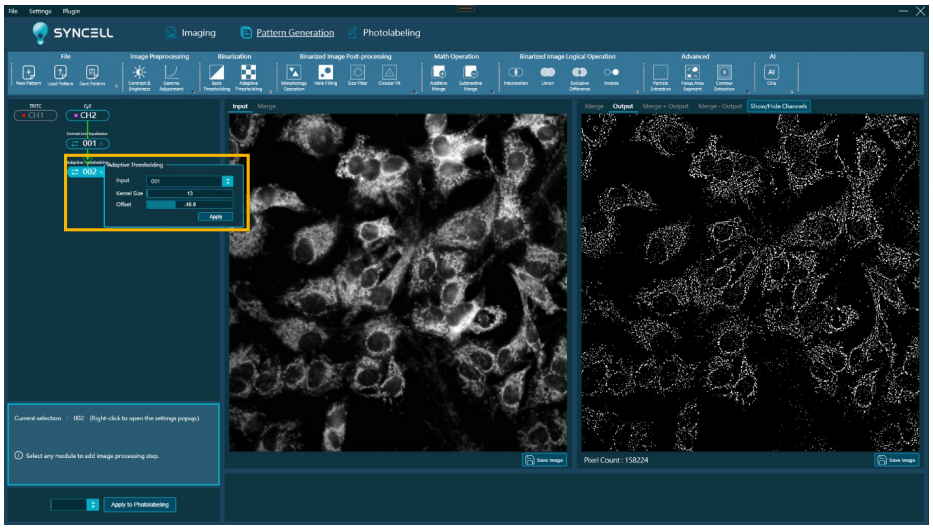

Fig. 4 Result of Adaptive thresholding

After the generation of the mask pattern, you can confirm its alignment with the original target protein pattern using the 'Merge-Result' function item in the right panel. To view one of the original images, simply uncheck the tick box of the image channel you wish to hide.

Step 3: Adaptive Threshold (vesicle-like structure)

The mask pattern generation of cytoplasm proteins has been completed. Next, the vesicle-like structure in the TRITC channel is loaded into the 'Adaptive Threshold' function to generate the binary image (Figure 5).

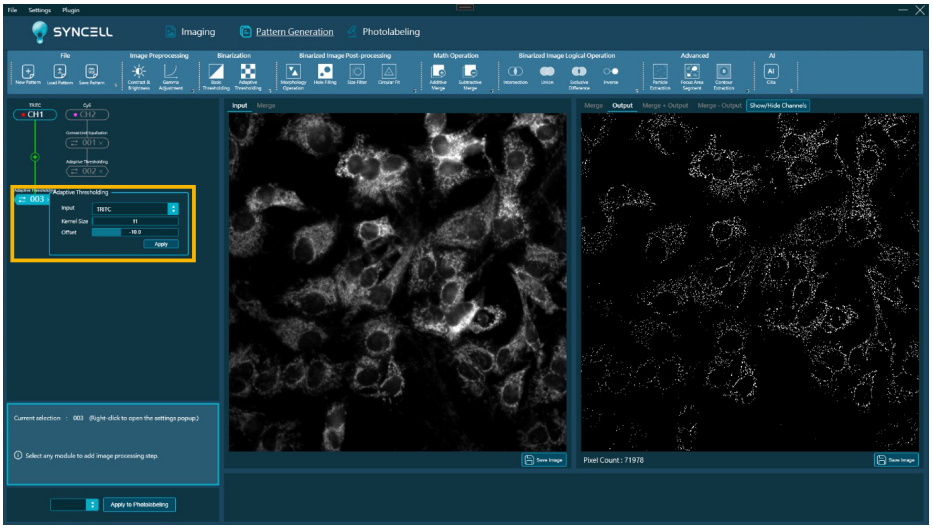

Fig. 5 Result of Adaptive thresholding

Step 4: Size Filter (vesicle-like structure)

Some small particles may be extracted after applying the adaptive threshold. Use the 'Size Filter' function to filter out or retain the white areas based on their size. The 'Size Filter' option is present in the 'Binarization Processing' menu. Load the last processed image and set the size range with lower and upper limits. For instance, in this example, the size range between 6 to 2000 is applied, and results shown in right panel reveal that removal of small size particles.

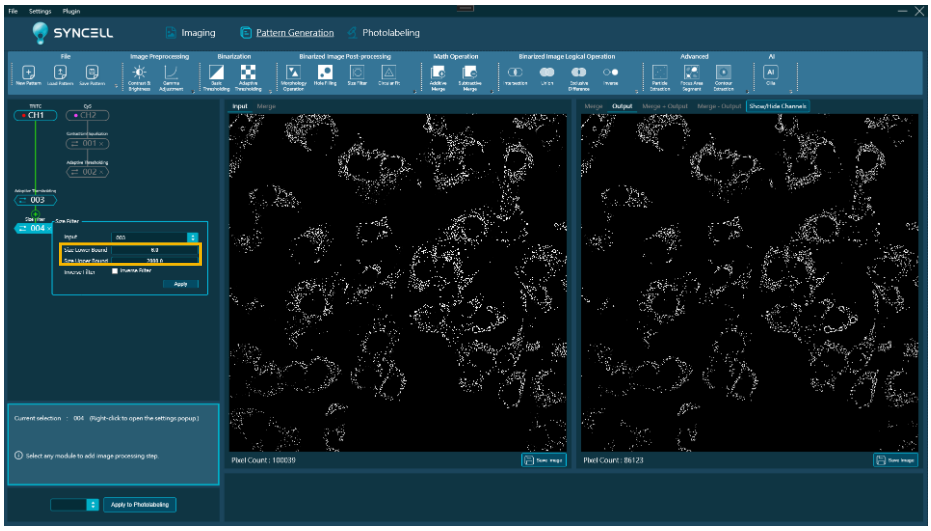

Fig. 6 Result of size filter

Step 5: Intersection

The mask pattern of two images has been generated respectively. The final step involves utilizing the 'Intersection' function to combine the two mask images and create an overlapping region mask. Access the 'Math Operation' menu and select the 'Intersection' item. The two processed images are imported as source (background) and source (foreground), respectively. The final mask result will be shown in the right panel (Figure 7).

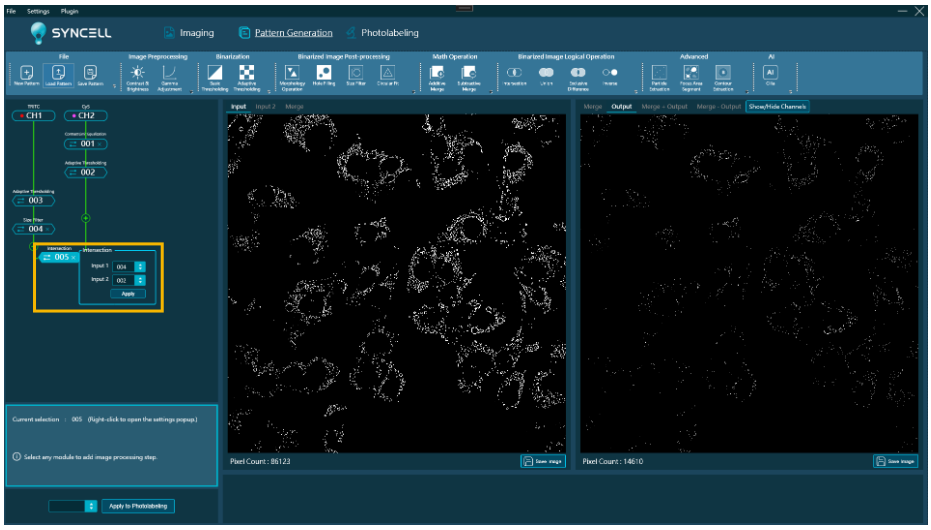

Fig. 7 Result of Intersection

2.2 Example 2 - Cilia

An in-depth understanding of the protein composition in the ciliary region is important for the study of primary cilia. Using Microscoop, thousands of known and novel proteins can be identified in the ciliary regions by laser photolabeling with a precise image mask. The purpose of this document is to demonstrate mask generation for the photolabeling of ciliary proteins. In this application, cells were seeded onto a chambered glass and proceed starvation to induce cilia formation. The cells were allowed to achieve 90% confluence before fixation, They were subjected to immunofluorescence staining with polyglutamylated tubulin as the target. The original staining image is showed in the left window in step 1- Laplacian Transform.

Step 1: Laplacian Transform

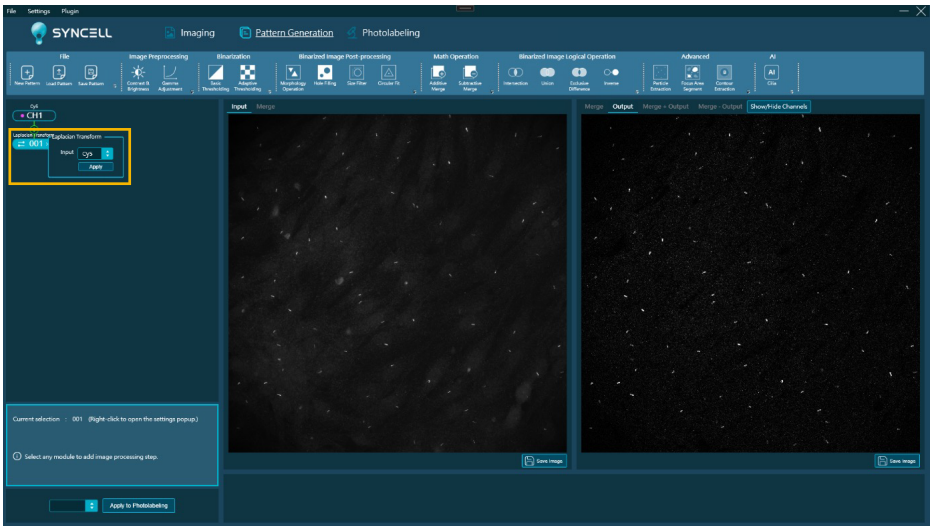

Fig. 8 Result of Laplacian Transform

The purpose of Laplacian transformation is to strengthen the gradient of intensity and highlight the area with specific intensity feature. With Laplacian transform, ciliary regions with higher intensity feature stand out from the background and out-of-focus (with lower intensity) regions. Apply a small kernel size on the small target region for more efficient elimination of background signals.

Step 2: Simple Threshold

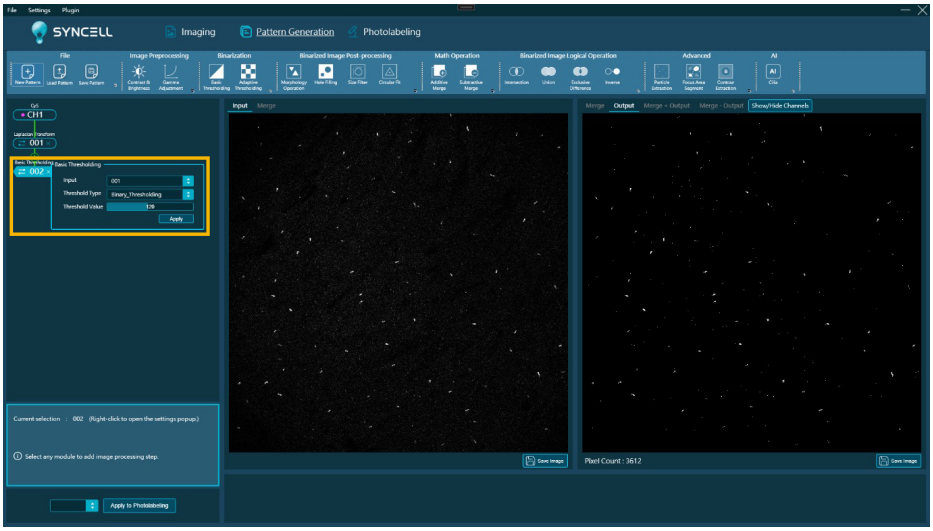

Fig. 9 Result of simple thresholding

Use simple threshold to binarize the gray-value image by setting a threshold value to keep pixels above the specific intensity. In this case, setting a threshold value of 120 separates regions with intensity above 120 from those having intensity below 120, is the latter composed mostly of the background signals.

Step 3: Size Filter

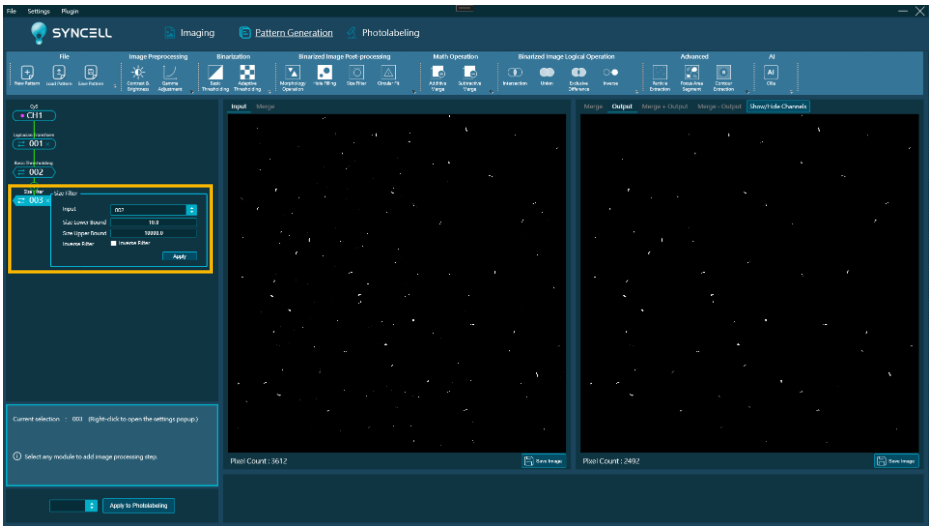

Fig. 10 Result of size filter

Even though a simple threshold is applied to distinguish the regions with different intensities, some high intensity background signals are still masked. The advantage of size filter is that a mask within a desired size can be generated by setting the lower and upper limits. In this case, only the regions of size 10-10000 are masked after the algorithm is applied.

Step 4: Adaptive Threshold

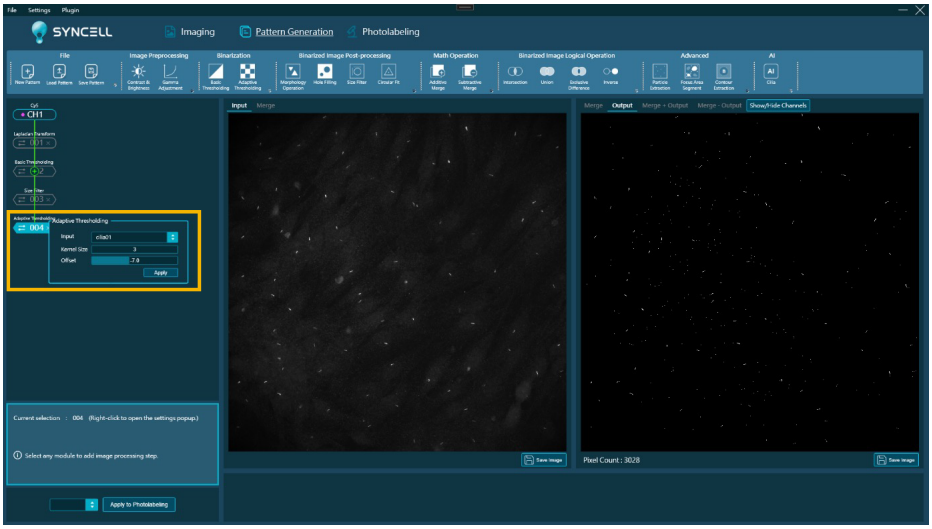

Fig. 11 Result of Adaptive thresholding

The output from steps 1 to 3 is aimed at marking the target regions, i.e., the ciliary regions in this case. However, the shape of these regions does not fit the real target signals, since the shape of the image is altered after Laplacian transformation. In step 4, the purpose of using adaptive threshold is to depict the shape of the ciliary regions with specific kernel size and offset. A small kernel size is recommended for small ROIs.

Step 5: Intersection

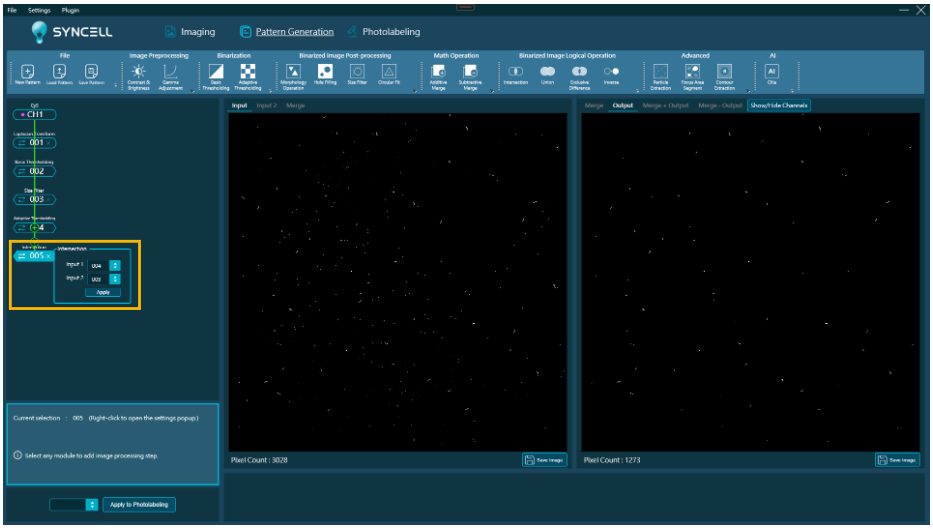

Fig. 12 Result of Intersection

This function applies “and” operation by pixels onto two masks, thereby merging the regions that exist in both masks. The combination of the output from steps 1-3, which marks the regions of the target signals, and the output from adaptive threshold (step 4), which determines the shape of the target signals, is used for generating the final mask for ciliary regions.

2.3 Example 3 - Microglia

Previous studies have shown that microglia are the brain’s immunocompetent macrophages, with a unique feature that enables the surveillance of the surrounding microenvironment and subsequent reactions to tissue damage, infection, or homeostatic perturbations (Leyh J et al., 2021). Under physiological conditions, microglial morphology is characterized by a small cell body and very fine, highly ramified processes (Tremblay et al., 2011; Nimmerjahn, 2012). The purpose of this document is to utilize the powerful image processing capability of Microscoop® for the independent isolation of target signals on soma or the microglial terminal signals, followed by the combination of two parts to generate a whole microglia pattern for high-precision photolabeling by Microscoop® . In this application, the brain section was subjected to immunofluorescence staining using Iba1 as target. The original staining image and following steps are shown below:

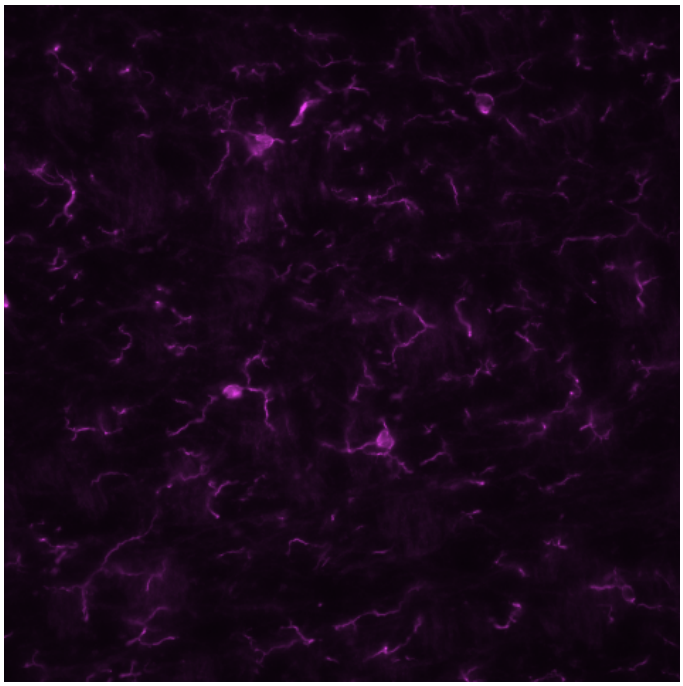

Fig. 13 Fluorescence image of Microglia

Step 1: Adaptive Threshold

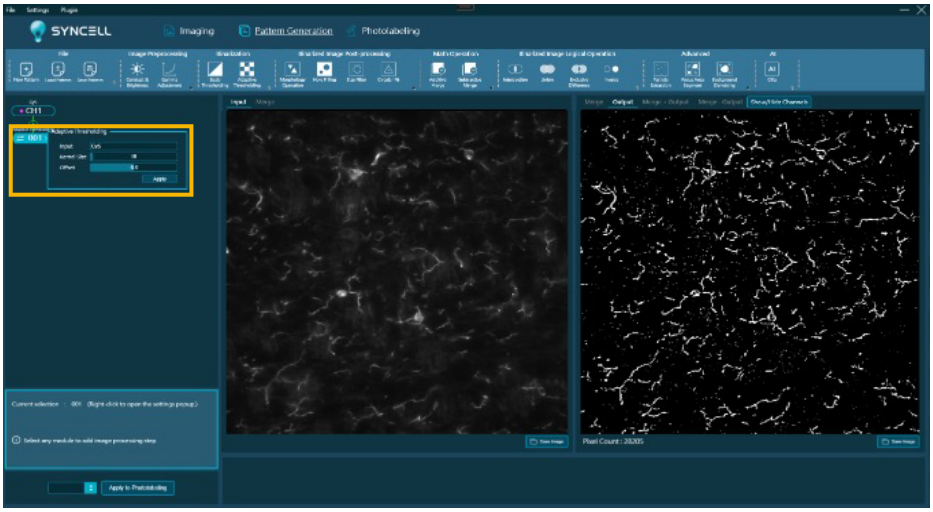

Fig. 14 Result of Adaptive thresholding

Use adaptive threshold to binarize the gray-value image by setting the kernel size and offset, in order to extract the maximum number of microglial signals. In this case, if the kernel size is set above 15, it becomes difficult to capture microglial terminal signals with a weak intensity. As for the offset parameter, this function can be fine-tuned to improve the contrast between signal and noise.

Step 2: Size Filter

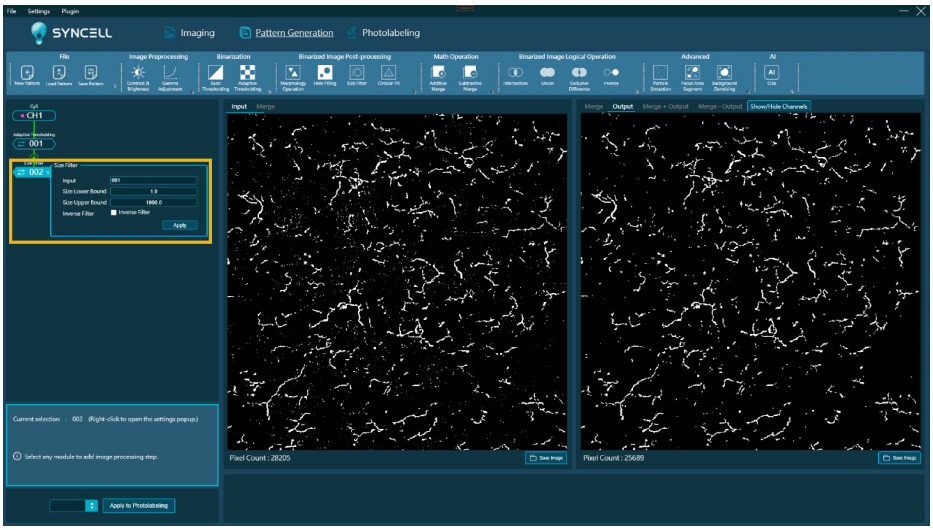

Fig. 15 Result of size filter

Even though an adaptive threshold is applied to distinguish the regions with differing intensities, some high intensity background signals are still masked. The advantage of the size filter is that it generates the mask within the desired size by setting the lower and upper bounds. In this case, the application of the algorithm masks only the regions sized between 1 and 1000.

Step 3: Contrast & Brightness

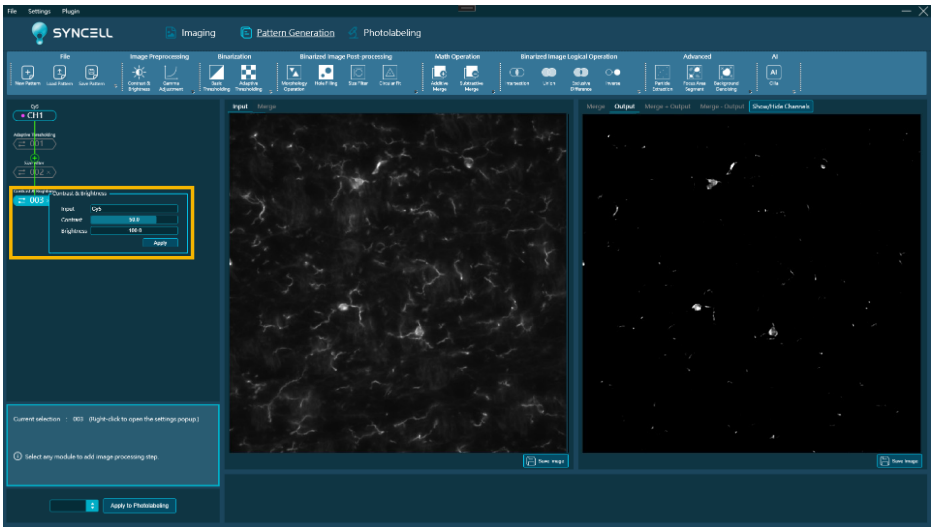

Fig. 16 Result of contrast and brightness adjustment

Even though steps 1 and 2 can extract a majority of microglial signals, it is still deficient in soma body signals, which may be extracted individually by adjusting contrast and brightness. In this case, a Brightness/Contrast value of 50/-100 can extract soma body signals.

Step 4: Simple Threshold

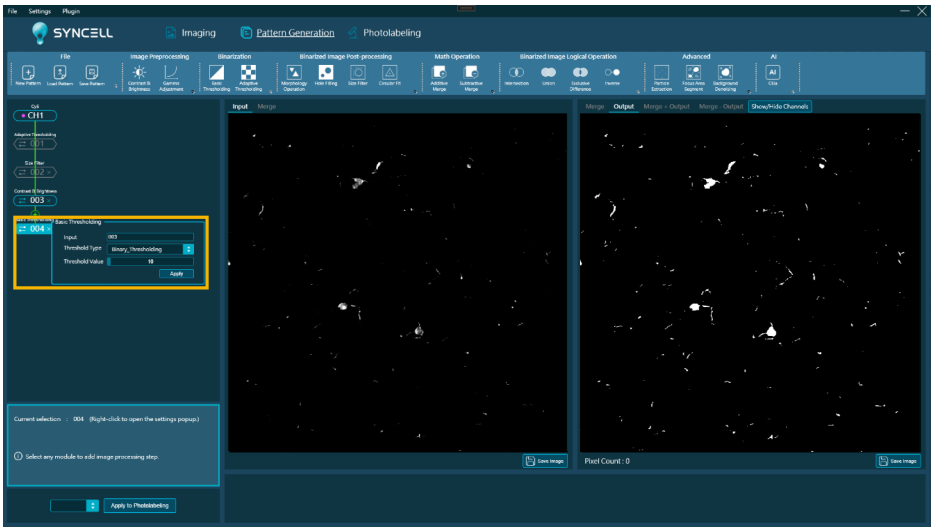

Fig. 17 Result of simple thresholding

Use simple threshold to binarize the gray-value image. This can be done by setting a threshold value to keep pixels above the specific intensity. In this case, setting a threshold value of 10 separates regions with intensity above 10 from that below 10, which is mostly composed of the background signals.

Step 5: Union

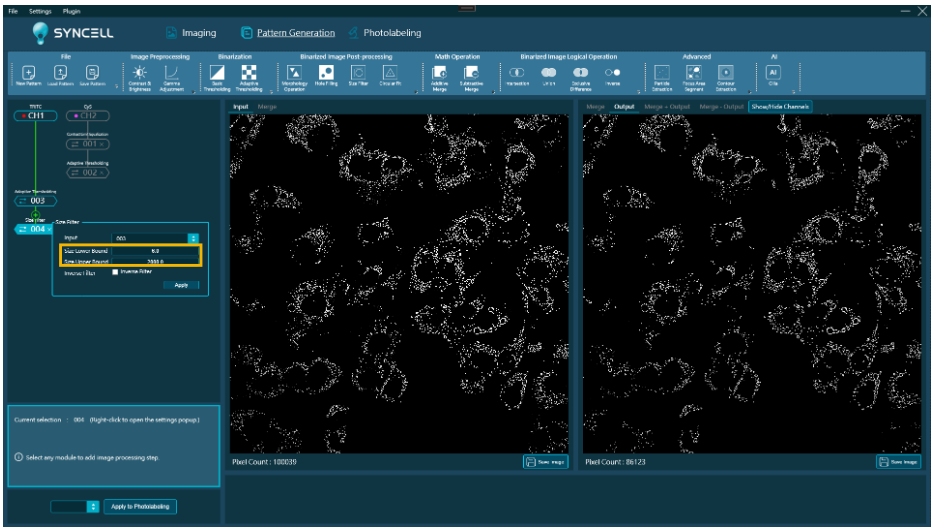

Fig. 18 Result of Union

This function applies “or” operation by pixels onto two masks, meaning all the regions in both masks merges. The combination of the output from steps 1-2, which marks the regions of the maximum microglial signals, and the output from steps 3-4, which marks the regions of soma body signals, generates the final mask for the microglial regions.

Step 6: Size Filter

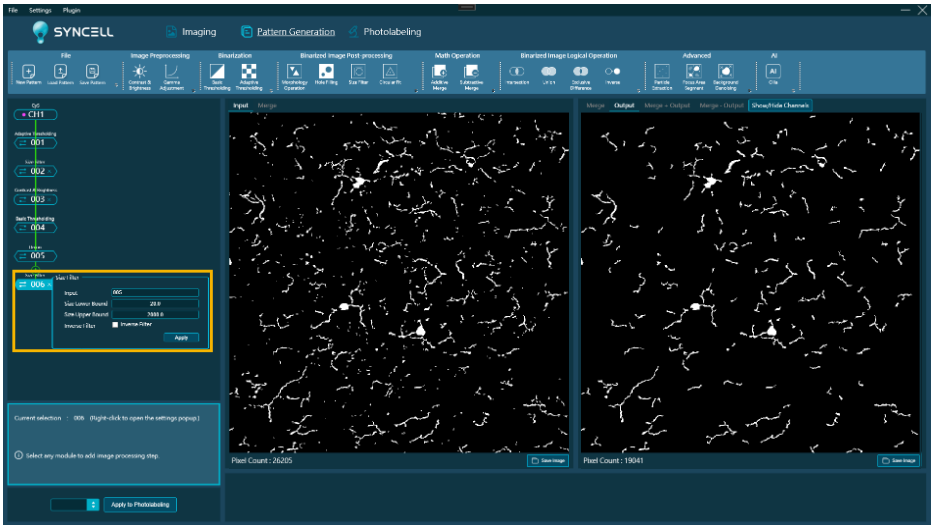

Fig. 19 Result of size filter

After the “Union” operation to generate the whole microglia pattern, there was still a lot of background noise of high intensity. To overcome this drawback, the region size may be set in the range 20-2000, which results in a pure microglial pattern mask for photolabeling.

APPENDIX 3 - AUTOSCOOP AI MODEL INTEGRATION GUIDE

3.1 Introduction

Autoscoop's AI function needs trained models to be converted into a common format and requires specific configuration files for loading. The success of this conversion relies on the TensorFlow version used during training and packaging. If third-party packages are used, ensure they are compatible with the versions needed.

3.2 Limitation

When preparing your model for integration into Autoscoop, consider the following limitations:

- Autoscoop supports integration of weighted-frozen pb format models, which are built on the TensorFlow framework.
- Ensure that the model can be inferred well by using TensorFlow 2.10 for compatibility with Autoscoop. Additionally, kindly note that the environment version for packaging needs to be lower than the environment version for inference if you intend to test the pb file in your environment.
- When Autoscoop executes an AI Inference, it will check the dimensions of the input layer. Dimensions include batch size (N), image height (pixels), image width (pixels) and channel.
  For example, if the input of the model is a multi-color image containing two channels of TRITC and FITC at a resolution of 800x800, the dimensions of the input layer will be (N, 800, 800, 2)
- Because Autoscoop currently only supports 800x800 and 1600x1600 resolutions. This means that when you train a model, you can only train at these two resolutions. To avoid image distortion caused by scaling, Autoscoop does not allow images of different resolutions to be input for inference.
- The value of each pixel of the model input and output must be a floating-point number between 0 and 1 after normalization.
- Autoscoop supports channel-last models only. If your model is channel-first, please incorporate an additional layer to reshape before transferring it to the frozen model.
- The number of layers in the AI model is not restricted in Autoscoop. However, the AI model must be capable of being loaded into a 12G GPU memory due to computer configuration.
- Autoscoop's pattern generation currently only supports contrast (LUT) adjusted images as image sources. Therefore, you must use contrast (LUT) adjusted images for model training to be close to the real situation of using Autoscoop. (Do image preprocessing before model training to ensure image is like the imaging in Autoscoop.)
- To fit the real cases for microscope image, we recommend implementing different contrast and brightness levels for image augmentation during model training.
- Autoscoop also has its limitations for model output. The dimensions of the model output must be (N, image height, image width, 1). This means that the resolution of the model output must be consistent with the input image, and it can only be output for a single channel. Autoscoop converts the output to a binarized image with a threshold of 127.
- Autoscoop's pattern generation can only load one AI model at a time. However, it allows adding Autoscoop's built-in image pre-processing and image post-processing methods before and after the AI step.

3.3 Load model into Autoscoop

To load an AI model into Autoscoop, you need to write a specific configuration file and place both the configuration file and the model file in specific locations. The following will explain how to write the configuration file.

- Create a text file using any text editor, name the file, and save it as a .ini file format. The content should include two sections: [MODEL_INFO] and [SOURCE], each containing their respective keys (properties). Fig. 1 presents an example of a configuration file.

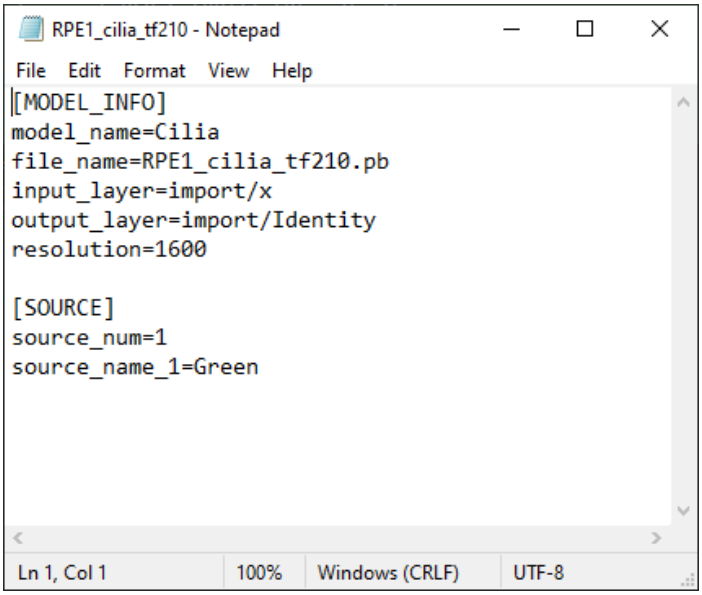

Fig. 1 Example of a Configuration File

- [MODEL_INFO] section includes the following Keys (properties), and each Key (property) is detailed as follows:
  a. model_name: The function name to display in Autoscoop. In this example, it is named as Cilia.
  b. file_name: The name of the model file to be read. In this example, the file name is RPE1_cilia_tf210.pb.
  c. input_layer: The name of the model's input layer. In this example, it is import/x.
  d. output_layer: The name of the model's output layer. In this example, it is import/Identity.
  e. resolution: Not applicable. Required before Autoscoop version 1.0.1.7, but not checked. Can be removed for Autoscoop version 1.0.1.7 and later (inclusive).
- [SOURCE] section includes the following Keys (properties), and each Key (property) is detailed as follows:
  a. source_num: The number of input sources (channels) corresponding to the input layer. Typically represents the number of channels, but there are no strict limitations. It can be defined based on the model's training requirements. For instance, utilizing the same channel after different image preprocessing as distinct inputs to the model. This parameter merely defines the selectable source quantity in the UI. An error will occur if set inconsistently with the dimensions of the input layer.
  b. source_name_N: The display name for each source (input) image in the UI. You can customize the name of each source (input) image. If not filled, default names will be provided by the system. In this example, we named source (input) image 1 as "Green."
- When the configuration file and model file are ready, both files need to be placed in the following paths:
  a. The configuration file should be placed in “C:\Program Files\Syncell\Autoscoop\AI”.
  (If the folders do not exist, you may need to create them manually)
  b. The model file needs to be placed in “C:\Program Files\Syncell\Autoscoop\AI\model”.
  (If the folders do not exist, you may need to create them manually)

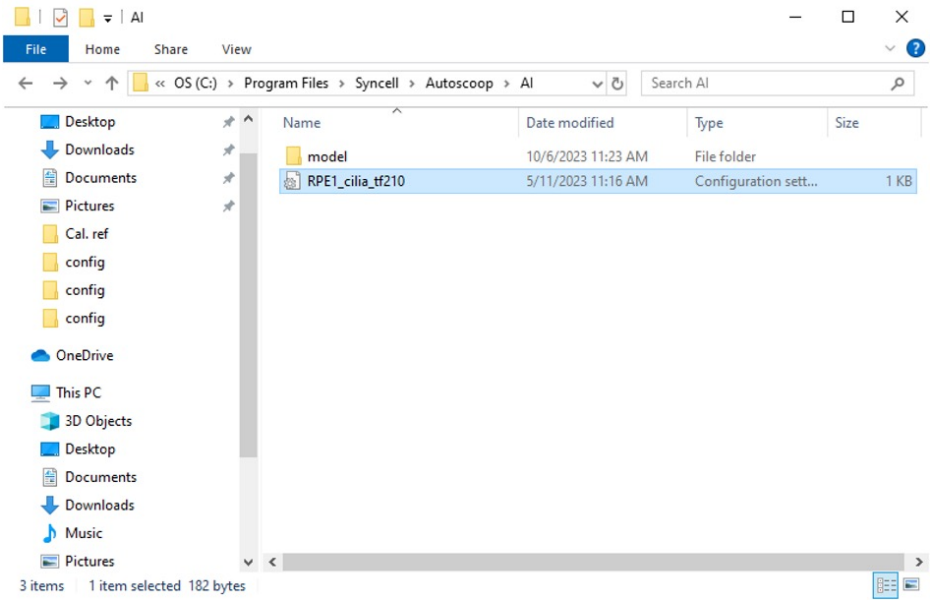

Fig. 2 Configuration File Path

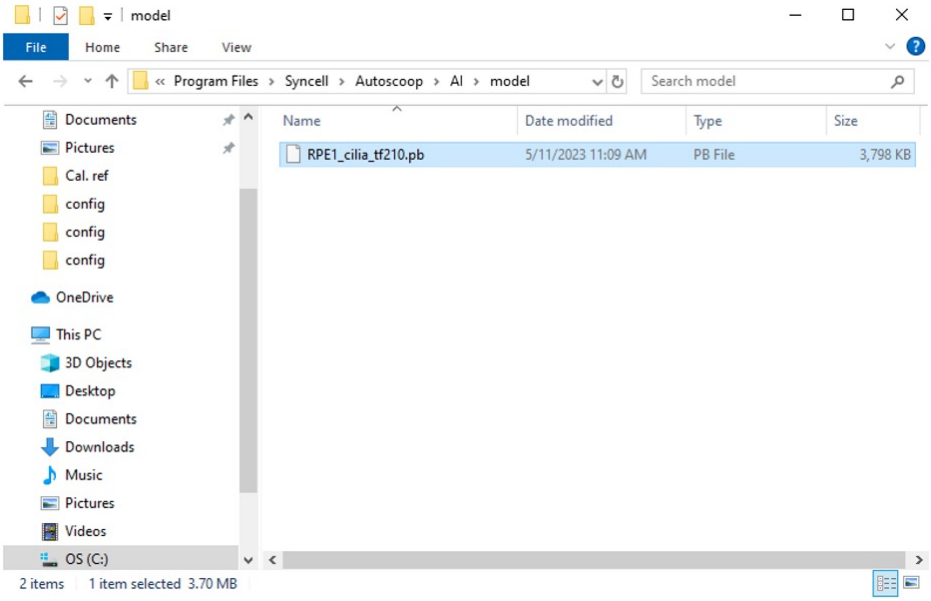

Fig. 3 Model File Path

Loading the AI model is part of the initialization process in Autoscoop. You need to restart the Autoscoop program to correctly load the AI model, and any changes require a program restart (e.g., modifying the content of the configuration file).

3.4 Performing AI Inference in Autoscoop

If Autoscoop has successfully loaded the configuration file, the Tool Bar in the Pattern Generation will display the toolbox for the AI functionality. Upon selecting the desired AI method, a corresponding user interface will appear, and Autoscoop will conduct model checks in the background. Like using the Pattern Generation feature normally, selecting an input image and clicking Apply or Run will produce the output result of the AI inference. (For how to use Autoscoop's Imaging and Pattern Generation functions, please reference to Chapter 5 and Chapter 6.)

Please note that the initial execution of AI inference might take longer. This delay is due to the system needing software initialization and hardware resource allocation. It does not imply that performing computations in the Photolabeling task will require a similar duration.

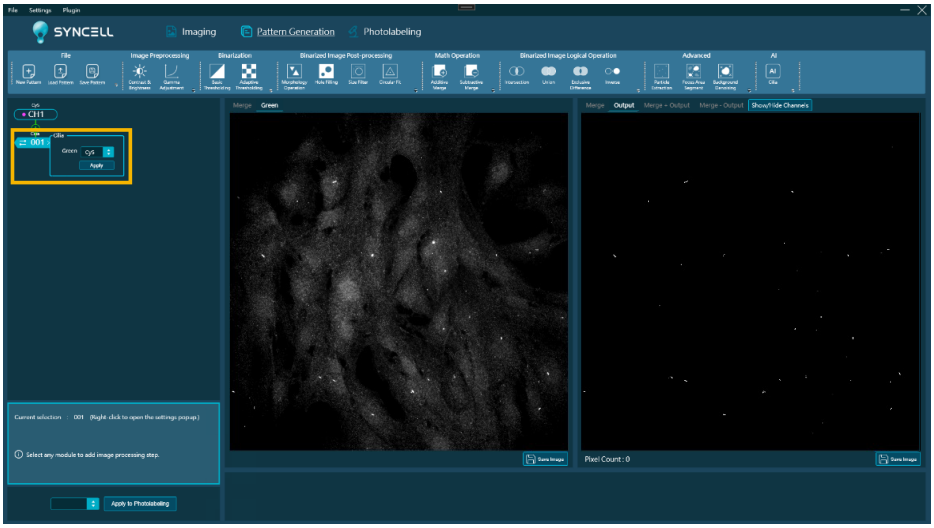

Fig. 4 Example of AI Inference in Autoscoop

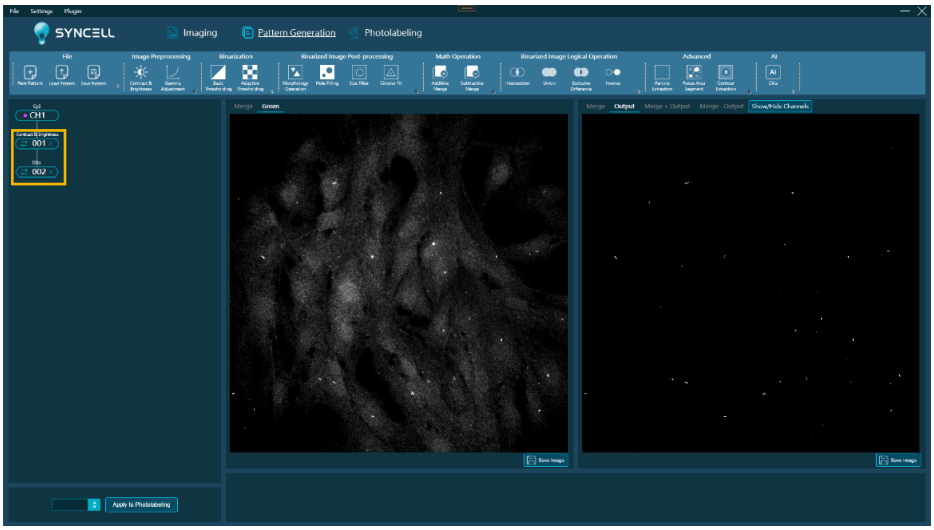

Fig. 5 AI inference example including image preprocessing in Autoscoop

3.5 Troubleshooting

This section outlines potential errors encountered while utilizing AI methods in Autoscoop and their respective resolutions. If you encounter an error during AI processing, refer to the following troubleshooting steps to resolve the issue. If the problem persists or if your issue is not listed, please contact the Syncell technical team for further assistance.

- **Model loading failure:**

  If you encounter the Model Loading Failure error message as shown in Fig. 6, it indicates a potential issue with the 'file_name' attribute in the [MODEL_INFO] section of the configuration file. It could also signify the absence of the specified file within the designated model folder. Please review the configuration file content or verify if the file exists in the specified directory.

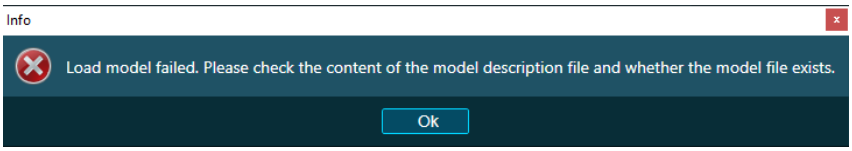

Fig. 6 Model Loading Failure Error Message

- **Can not find input layer:**

  When you encounter the error message shown in Fig. 7, it indicates that Autoscoop couldn't locate the corresponding input layer during model inspection. Please check the configuration file and ensure that the correct input layer name is provided.

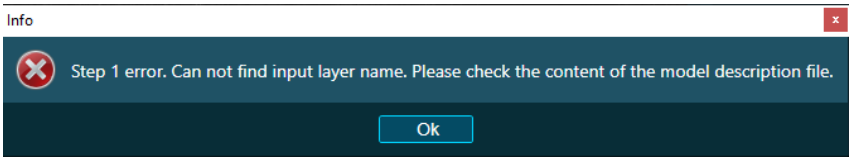

Fig. 7 Can Not Find Input Layer Error Message

- **Can not find output layer:**

  When you encounter the error message shown in Fig. 8, it indicates that Autoscoop couldn't locate the corresponding output layer during model inspection. Please check the configuration file and ensure that the correct output layer name is provided.

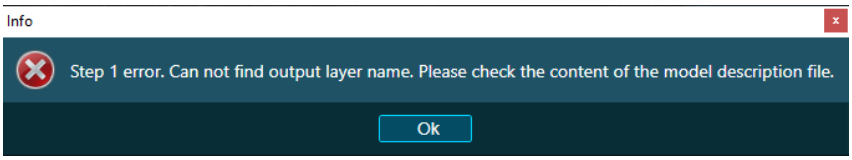

Fig. 8 Can Not Find Output Layer Error Message

- **Invalid input size:**

  When you encounter this error message, it suggests a failed dimension check in the input layer. Typically, it arises from incorrect image resolution or quantity. For instance, if the model is trained with a resolution of 1600x1600, but an image with a resolution of 800x800 is used for inference. Another scenario could be a mismatch between the "source_num" quantity specified in the configuration file and the actual number of inputs in the model's input layer.

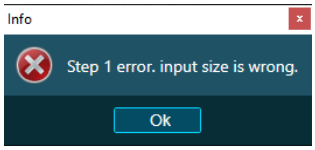

Fig. 9 Invalid Input Size Error Message

- **Invalid output size:**

  When you encounter this message, it indicates a failed dimension check in the output layer. Typically, it arises from incorrect image resolution or quantity in the model's output. The model's output must match the input's resolution and can only be a single image (channel). Encountering this error usually requires verifying whether your model output conforms to the specifications.

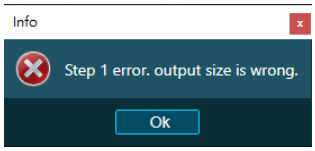

Fig. 10 Invalid Output Size Error Message

- **Execution timeout:**

  When you encounter this message, it indicates that the execution time of the image processing function is exceeding the set time limit. To prevent the system from hanging, Autoscoop will display an error message and terminate the operation. This protective mechanism may also be triggered when utilizing the AI functionality.

  Particularly, when loading larger-sized models, the initial AI inference may take an extended period, potentially surpassing the system's configured safe duration, resulting in this error. Please try applying the operation once again to confirm if the model can function correctly.

  If the message persists, it may suggest that your model is too complex for Autoscoop. Consider training the model with a different structure.

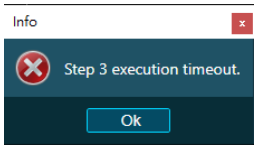

Fig. 11 Execution Timeout Error Message

3.6 Sample Code

Sample Code for transferring HDF5 (.h5) to Frozen Protobuf (.pb)

```
import tensorflow as tf
from tensorflow.keras.models import load_model

def convert_h5_to_pb(h5_model_path, output_pb_file):
    model = load_model(h5_model_path)
    concrete_func =
model.signatures[tf.saved_model.DEFAULT_SERVING_SIGNATURE_DEF_KEY]
    frozen_func = tf.function(concrete_func).get_concrete_function()
    frozen_graph_def = frozen_func.graph.as_graph_def()
    with tf.io.gfile.GFile(output_pb_file, "wb") as f:
        f.write(frozen_graph_def.SerializeToString())
    print("Frozen graph saved to:", output_pb_file)

# Example usage
h5_model_path = "your_model.h5"
output_pb_file = "output_model.pb"
convert_h5_to_pb(h5_model_path, output_pb_file)
```

Sample Code for transferring Protobuf (.pb) to Frozen Protobuf (.pb)

```
import tensorflow as tf

def convert_pb_to_frozen_pb(pb_file_path, output_pb_file):
    with tf.io.gfile.GFile(pb_file_path, "rb") as f:
        graph_def = tf.compat.v1.GraphDef()
        graph_def.ParseFromString(f.read())
    with tf.Graph().as_default() as graph:
        tf.import_graph_def(graph_def, name="")
        concrete_func = tf.function(lambda: [node.name for node in graph.get_operations()])
        frozen_func = concrete_func.get_concrete_function()
        frozen_graph_def = frozen_func.graph.as_graph_def()
    with tf.io.gfile.GFile(output_pb_file, "wb") as f:
        f.write(frozen_graph_def.SerializeToString())
    print("Frozen graph saved to:", output_pb_file)

# Example usage
pb_file_path = "your_model.pb"
output_pb_file = "output_model.pb"
convert_pb_to_frozen_pb(pb_file_path, output_pb_file)
```

Sample Code for transferring TensorFlow Checkpoint (.ckpt) to Frozen Protobuf (.pb)

```
import tensorflow as tf

def convert_ckpt_to_pb(ckpt_dir, output_pb_file):
    graph = tf.Graph()
    with graph.as_default():
        saver = tf.compat.v1.train.import_meta_graph(ckpt_dir + '.meta', clear_devices=True)
        session = tf.compat.v1.Session()
        saver.restore(session, ckpt_dir)
concrete_func = tf.function(lambda: [node.name for node in session.graph.get_operations()])
frozen_func = concrete_func.get_concrete_function()
frozen_graph_def = frozen_func.graph.as_graph_def()
    with tf.io.gfile.GFile(output_pb_file, "wb") as f:
        f.write(frozen_graph_def.SerializeToString())
    print("Frozen graph saved to:", output_pb_file)

# Example usage
ckpt_dir = "your_checkpoint_directory/model.ckpt"
output_pb_file = "output_model.pb"
convert_ckpt_to_pb(ckpt_dir, output_pb_file)
```