

DobotVisionStudio Software

User Guide

Issue: V1.3.2

Date: 2019.07.18

Chapter 1 Disclaimer

Copyright © ShenZhen Yuejiang Technology Co., Ltd 2019. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Yuejiang Technology Co., Ltd

Logical Disclaimer

To the maximum extent permitted by applicable law, the products described (including its hardware, software and firmware, etc.) in this document are provided AS IS, which may have flaws, errors or faults. Yuejiang makes no warranties of any kind, express or implied, including but not limited to, merchantability, satisfaction of quality, fitness for a particular purpose and non-infringement of third party rights. In no event will Yuejiang be liable for any special, incidental, consequential or indirect damages resulting from the use of our products and documents.

Before using our product, please thoroughly read and understand the contents of this document and related technical documents that are published online, to ensure that the robotic arm is used on the premise of fully understanding the robotic arm and related knowledge. Please use this document with technical guidance from professionals. Even if follow this document or any other related instructions, Damages or losses will be happen in the using process, Dobot shall not be considered as a guarantee regarding to all security information contained in this document.

The user has the responsibility to make sure following the relevant practical laws and regulations of the country, in order that there is no significant danger in the use of the robotic arm.

Shenzhen Yuejiang Technology Co., Ltd

Address: 3F, Building NO.3, Tongfuyu Industrial Town, Nanshan District, Shenzhen, China

Website: <http://www.dobot.cc/>

Table of Contents

Chapter 1 Updates	1
Chapter 2 Overview	2
2.1 Key Features	2
2.2 System Requirements.....	2
2.3 Installation.....	3
2.3.1 Install DobotVisionStudio Software	3
Chapter 3 Software Introduction	6
3.1 Welcome Page	6
3.2 Home Page	6
3.2.1 Menu	8
3.2.2 Toolbar	10
3.3 Tool	16
3.3.1 Tool Introduction	16
3.3.2 General Parameter	16
Chapter 4 Image Acquisition.....	18
4.1 Image Files	18
4.2 Camera Image	19
4.3 Save Image	20
Chapter 5 Vision Tools	22
5.1 Location	22
5.1.1 Feature Matching	23
5.1.2 Find Circle.....	29
5.1.3 Find Line	32
5.1.4 Blob	34
5.1.5 Caliper	37
5.1.6 Find Edge.....	39
5.1.7 Find Edge Width.....	40
5.1.8 Position Correction.....	41
5.1.9 Find Rect	43
5.1.10 Find Peak.....	43
5.1.11 Caliper Corner	45
5.1.12 Find Parallel Lines.....	46

5.2 Measure	47
5.2.1 L2C Measure.....	47
5.2.2 L2L Measure	48
5.2.3 C2C, P2C, P2L And P2P Measure	49
5.2.4 Circle Fit and Line Fit.....	49
5.2.5 Intensity Measure	52
5.2.6 Pixel Count	53
5.2.7 Histogram Tool	54
5.2.8 Geometry Creation.....	55
5.3 Rognition	55
5.3.1 2D Code	56
5.3.2 1D Code	58
5.3.3 OCR.....	59
5.3.4 VeriCode	62
5.4 Deep Learning	63
5.4.1 DL Training Tool	64
5.4.2 DL Character Recognition	73
5.4.3 Defect Detection	74
5.4.4 DL Text Detection	76
5.4.5 DL Classification	77
5.4.6 DL Object Detection	77
5.5 Calibration	79
5.5.1 N-Point Calibration.....	79
5.5.2 CalibBoard Calibration	83
5.5.3 CalibBoard Calibration Transformation.....	85
5.5.4 Scale Transformation.....	87
5.5.5 Distortion Calibration.....	88
5.5.6 Image Correction.....	89
5.6 Alignment.....	90
5.6.1 Camera Alignment.....	90
5.6.2 Point Alignment.....	91
5.6.3 Points Alignment	91
5.6.4 Line Alignment	92
5.7 Image Processing.....	93
5.7.1 Image Combination	94

5.7.2 Image Morphology	95
5.7.3 Image Binarization	96
5.7.4 Image Filter	97
5.7.5 Image Enhancement	98
5.7.6 Image Math	99
5.7.7 Image Sharpness	99
5.7.8 Image Fixture	100
5.7.9 Shade Correction.....	101
5.7.10 Affine Transformation	102
5.7.11 Polar Unwrap.....	103
5.7.12 Copy and Fill.....	105
5.7.13 Frame Mean	106
5.7.14 Image Normalization	107
5.8 Color Processing	108
5.8.1 Color Extraction.....	108
5.8.2 Color Measure	108
5.8.3 Color Transformation	109
5.9 Defect Detection	111
5.9.1 OCV.....	111
5.10 Logical Tools	116
5.10.1 If Module	116
5.10.2 Branch	117
5.10.3 Branch String Comparison	118
5.10.4 Save Text.....	119
5.10.5 Logic Module.....	120
5.10.6 Format Module	121
5.10.7 Calculator	123
5.10.8 String Comparison.....	124
5.10.9 Script	125
5.10.10 Loop Control.....	133
5.10.11 Points Assemble	135
5.10.12 Time Consuming	136
5.10.13 Wait Time.....	137
Chapter 6 Communication.....	139
6.1 Receive Data.....	139

6.2 Send Data	142
6.3 TCP Communication	144
6.4 IO Communication	145
6.5 ModBus Communication.....	146
6.6 PLC Communication	147
Chapter 7 Dobot Tools.....	149
7.1 Overview	149
7.2 Dobot Magician Tools.....	149
7.2.1 Conveyor Belt	149
7.2.2 Sliding Rail	150
7.2.3 Moving To a Point.....	151
7.2.4 Suction Cup Switch.....	152
7.2.5 Gripper Switch.....	153
7.2.6 Laser Switch.....	154
7.2.7 Home Calibration	154
7.2.8 I/O Input	155
7.2.9 I/O output	156
7.2.10 Speed Ratio	156
7.3 Dobot M1 Tools	157
7.3.1 I/O Input	157
7.3.2 I/O Output	158
7.3.3 Moving to a point	158
7.3.4 Dobot Arm Orientation	160
7.3.5 Speed Ratio	161
Chapter 8 Operation Interface.....	163
8.1 Editing Interface	163
8.2 Controls	163
8.3 Editing Area	164
8.4 Parameter Settings	164
8.5 Role Interface Settings	166
Chapter 9 Examples	169
9.1 Location for USB Hole	169
9.2 Detection for Detective Metal.....	176
9.3 Calibration and Detection	181

9.4 Loop Function.....	184
9.5 Script Function	186
9.6 Multi-Flow Communication	189
Chapter 10 Dobot Magician Examples.....	193
10.1 Robot Calibration	193
10.2 Color Sort	196
10.3 Character Defect Detection.....	201
10.4 Measure and Sort.....	208
10.5 RectangleTemplate Match	215
10.6 Circle Template Match	220
Chapter 11 Dobot M1 Examples	226
11.1 Robot Calibration	226
11.2 Color Sort	229
11.3 Character Defect Detection.....	235
11.4 Measure and Sort.....	243
11.5 RectangleTemplate Match	251
11.6 Circle Template Match	258

Chapter 2 Updates

Table 2.1 Update description

Version	Description
DobotVisionStudio 1.3.2	Modify some bugs
DobotVisionStudio 1.3.0	<p>The updated contents:</p> <ul style="list-style-type: none">● Add the descriptions of the DobotStudio● Add the wait time tool● Add the speed ratio tool● Add the Dobot arm orientation tool of the DobotM1
DobotVisionStudio 1.2.0	<ul style="list-style-type: none">● Original version

Chapter 3 Overview

DobotVisionStudio Software integrates various algorithm components of machine vision. It is designed to quickly combine algorithms to find, measure the object and detect defection and so on, suitable for a variety of application scenarios.

The DobotVisionStudio software features a powerful library of visual analysis tools, which can be used to build machine vision application solutions, without programming. It can meet the needs of vision applications such as visual positioning, measurement, detection and identification. It has the characteristics of rich function, stable performance and friendly user interface.

3.1 Key Features

- Easy to Use: With drag-and-drop component, you can build a visual application without programming.
- Wide Compatibility: Supports multiple operation systems including Windows 7/10 (32/64-bit).
- User-Friendly Interface: Provides clear, simple and visualized user interfaces.
- Flexible Display: Maximally save the limited screen display space.

3.2 System Requirements

Make sure the computer on which you install the client software at least meets the minimum requirements.

Notes:

- Dongle is required for running this software. Before using it, please install the Dongle driver and relevant machine vision drivers.
- We recommend adding the software to the whitelist of the antivirus software or closing the antivirus software, in case of being recognized as virus.

Recommended

- Operating System: Microsoft Windows 7/10 (32/64-bit)
- .NET Running Environment: .NET4.5 and above
- CPU: Intel Pentium IV 3.0 GHz and above
- Memory: 8G and above
- NIC: Intel i210 and above
- Discrete Graphics Card: 1G or above (4G is required for DL training tool fro deep learning)

- USB Interface : USB3.0 interface is required

Minimum

- Operating System: Microsoft Windows 7/10 (32/64-bit)
- CPU: Intel Pentium IV 2.0 GHz
- Memory: 4G and above
- NIC: Gigabit Ethernet
- Discrete Graphics Card: 1G or above (4G is required for DL training tool for deep learning)
- USB Interface : USB3.0 interface is required

3.3 Installation

3.3.1 Install DobotVisionStudio Software

The installation steps are shown below:

1. Double-click the installation package to install software, as shown below, click the **next** to start the next step.

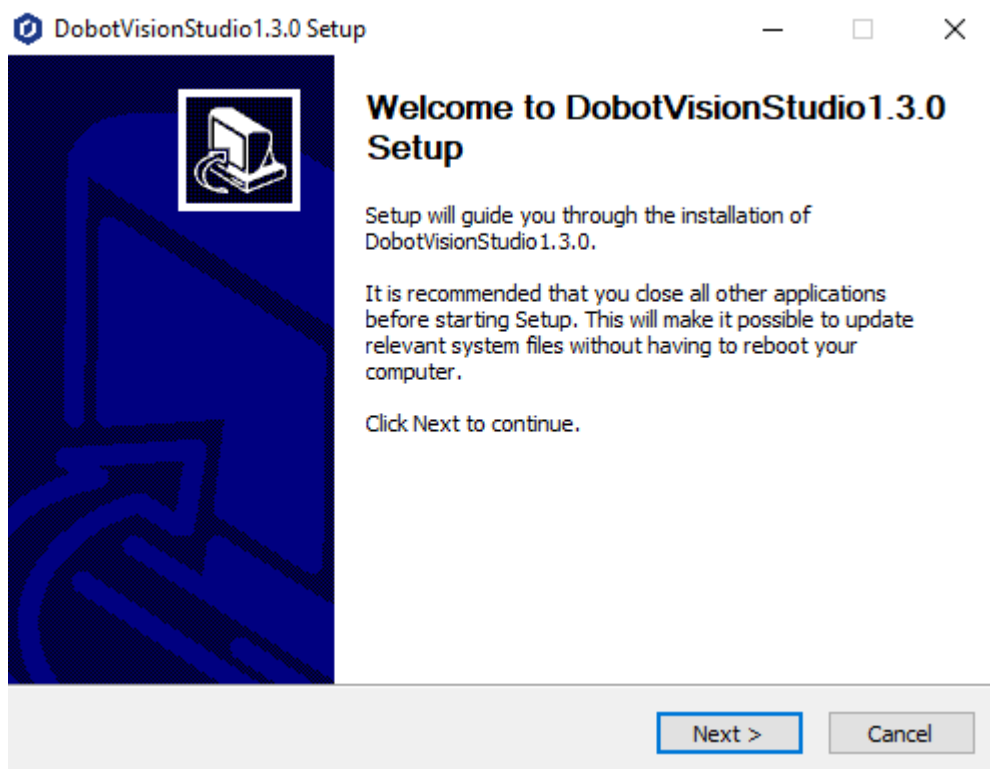


Figure 3.1 Install program

2. Set the installation path and choose whether to install the **DogDrive** and the **SDK** before installation, as shown below.

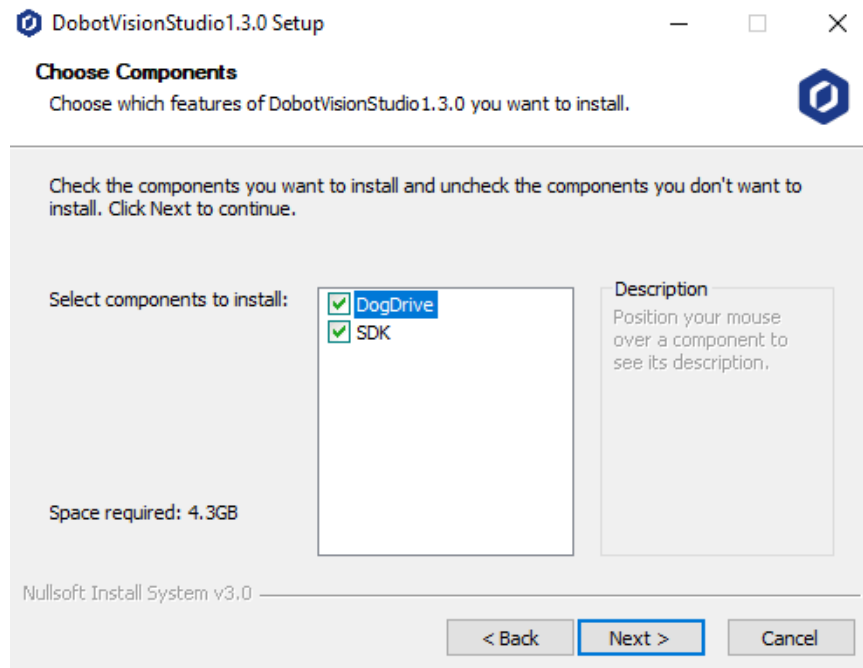


Figure 3.2 Install DogDrive and SDK

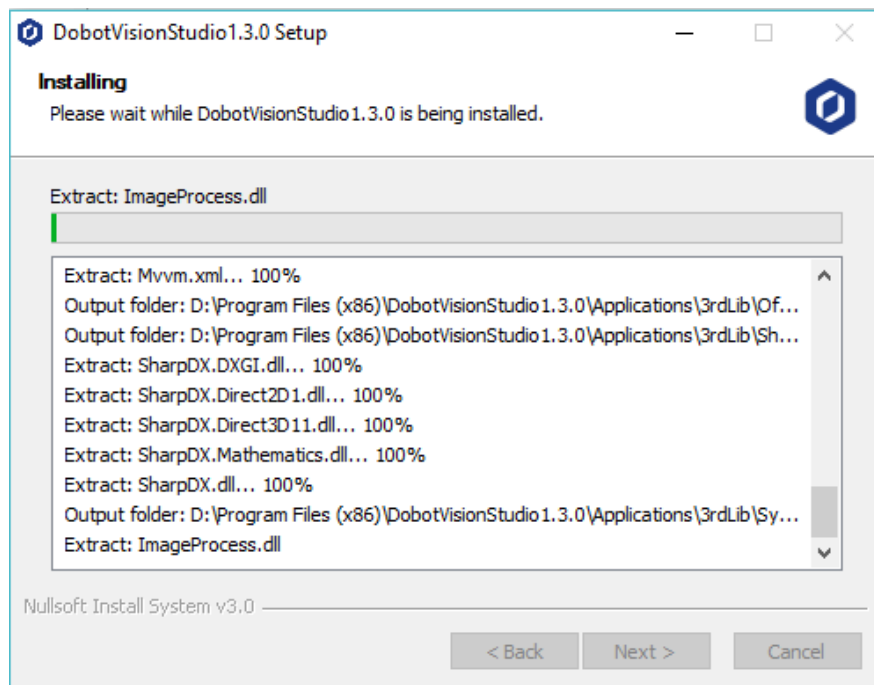


Figure 3.3 Start installation

3. After the installation is complete, choose whether to open DobotVisionStudio. As shown below.

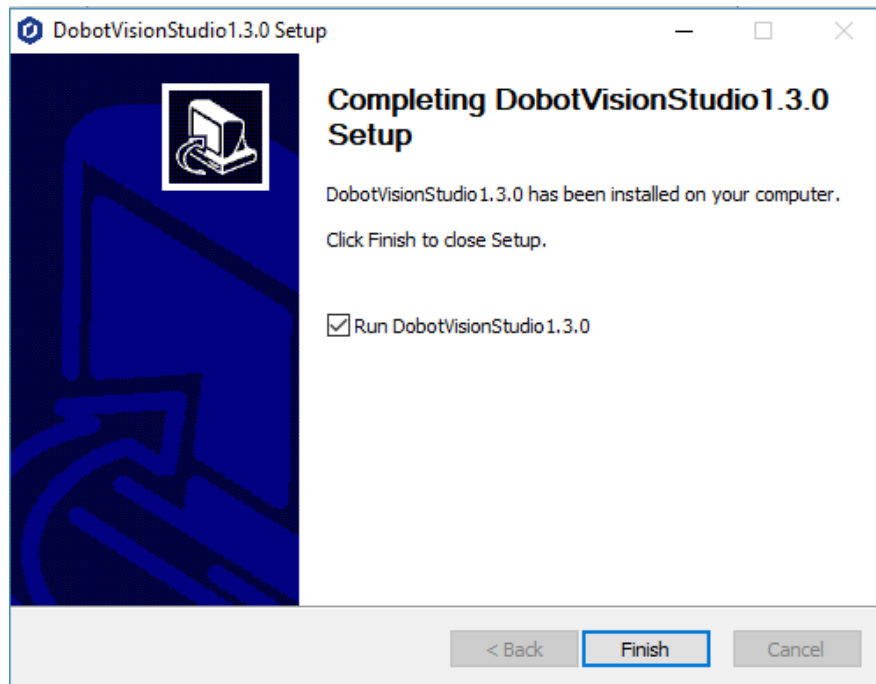



Figure 3.4 Finish installation

Chapter 4 Software Introduction

4.1 Welcome Page

Steps:

Double click , and then the welcome page will pop up.

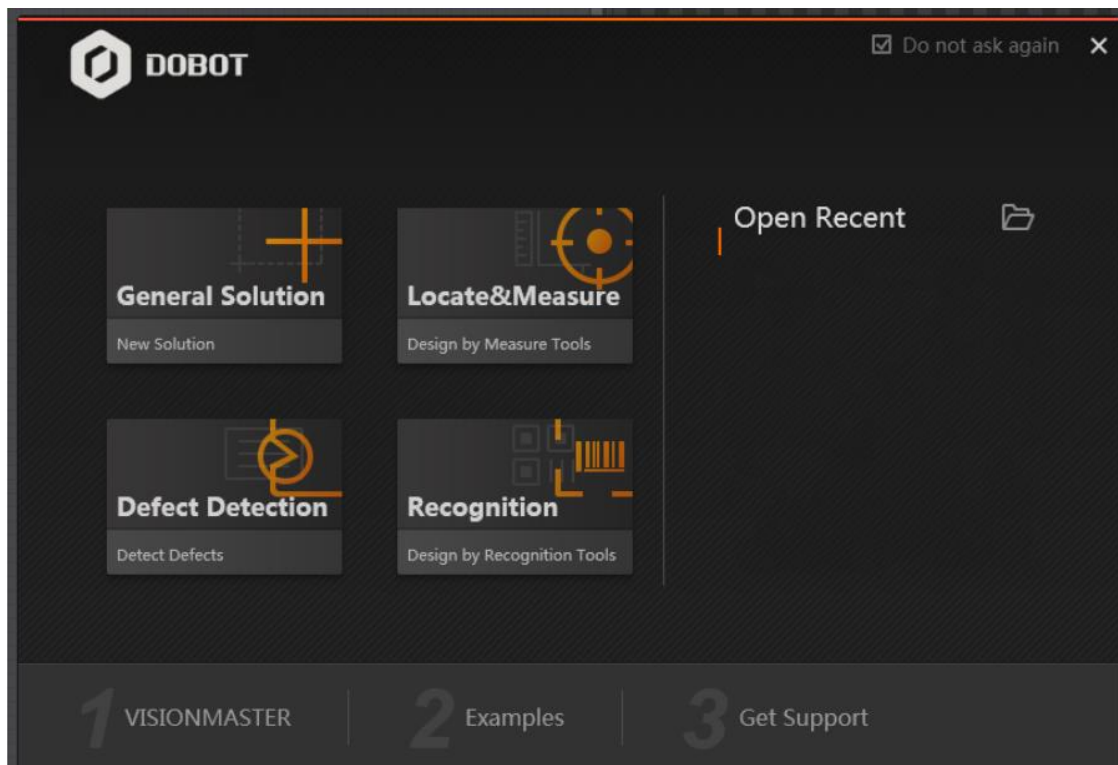


Figure 4.1 The welcome page

- There are four modules selectable: **General Solution**, **Location & Measure**, **Defect Detection**, and **Recognition**.
- **Open Recent**: Open the recently-used solution.

Note:

You can go to **Help->Welcome Page** to open it.

4.2 Home Page

The home page of the software is shown as follows:

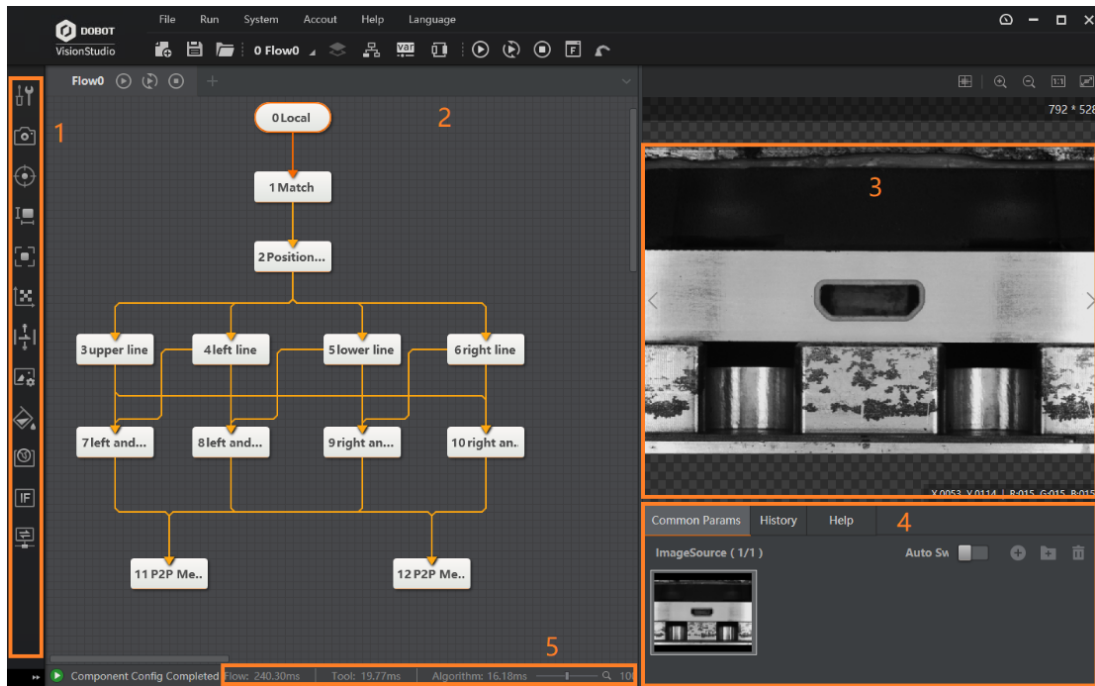


Figure 4.2 The home page

The following table shows the description of each part of the home page.

Table 4.1 Home page description

No.	Area Name	Description
1	Tools	Functions include image collection, positioning, measurement, identification, calibration, position, image processing, color processing, defect detection, logic tools, communication and other functions .
2	Flow	You can edit the flow here.
3	Images	Image is displayed here.
4	Results	View current results, historical results, common parameters, and help information
5	Status	Display the individual tool run time, total process uptime, and algorithm time.

4.2.1 Menu

The menu is shown as below.

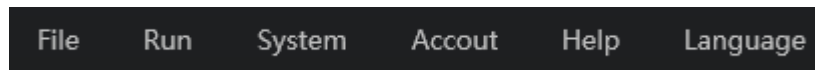


Figure 4.3 Menu

File

- You can make new solutions or save solutions here. When saving the solution, you can click **Encryption Option** to encrypt the solution;
- For **Start-up Settings**, you can set the **Delayed Start** time (20 seconds recommended) to run the program after PC start-up, and you can also select the solution for auto running once you open the program.

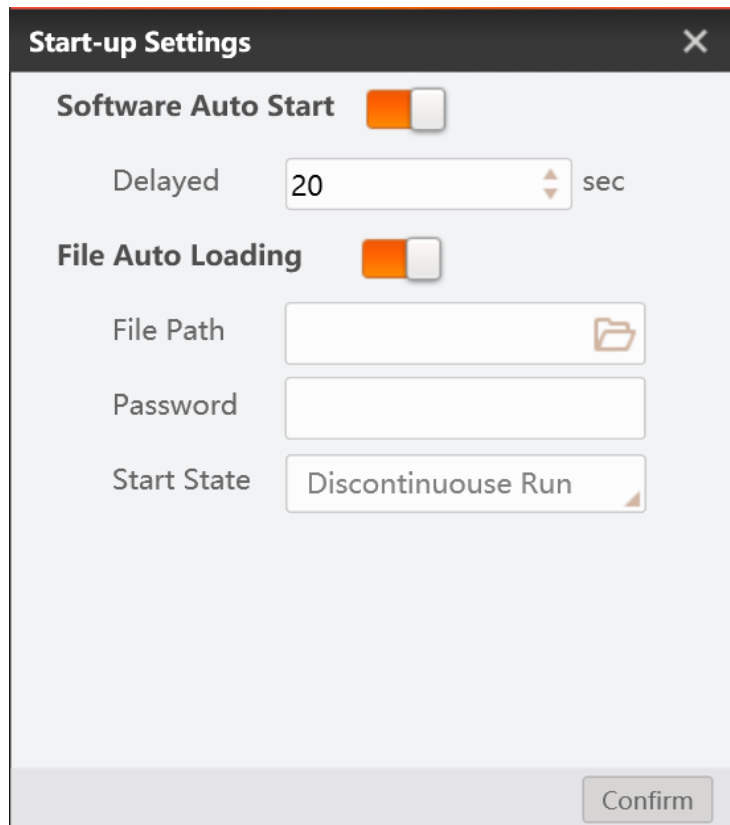


Figure 4.4 Start-up setting

For **Solution Management**, you can make the solutions switch automatically.

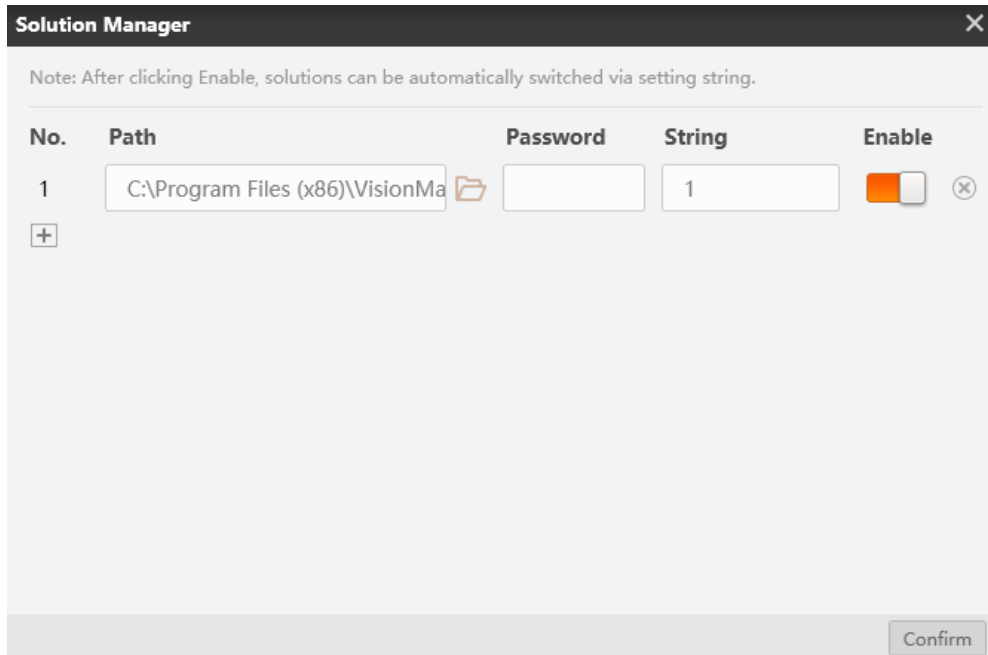


Figure 4.5 Solution manager

Run

You can control the current solution here.

- Click **Once** to execute the solution once.
- Click **Continuously** to execute the solution continuously.
- Click **Stop** to make solution stop.
- Click **Operation** Interface to display the operation window.

System

- Click **Log** to display the error and warning information.
- Click **Communication Management** to add the device.
- Click **Exit Settings** to set the program background running or exit the program.

Account

You can set the admin password and enable the admin authority here.

Help













- Click **Help** to get the DobotVisionStudio material.
- Click **Version** to learn about the DobotVisionStudio version.
- Click **Welcome Page** to display the welcome page.




Language


Select the language here.

4.2.2 Toolbar

Table 4.2 Toolbar description

Icon	Name	Description
	New	Click to add the solution.
	Save	Click to save the solution.
	Open	Click to open the solution.
	Script Function	Click to switch the flow view.
	Up Layer	Click to go to the previous layer. This function only works in the Group.
	Main Flow	Click to display all flows that you have created. You can click the window for further operations.
	Flow	Under the main flow window, drag the Flow in the left to the center and release it so as to add new flow.
	Data Queue	Under the main flow window, drag the Data Queue in the left to the center and release it so as to add a data queue. The queues should have values, or else the returning module may return error. .
	Global Variable	Click to add the global variable. Up to 32 global variables can be set.
	Communication Management	Click to add the device and set the communication protocol and other parameters.
	Once	Click to execute the flow once. Work for all flows.
	Continuously	Click to execute the flow continuously. Work for all flows.

	Stop	Click to stop the flow. Work for all flows.
	Operation Interface	Click to display the operation window.
	DobotStudio	Click to display the DobotStudio page.

- **Main Flow:** after creating multi flows, click  to view all the flows you created, as shown below. You can click one flow or multi flows to operate them, view operation times and time period for single operation. Right-click one flow, you can set run internal, delete, rename this flow.

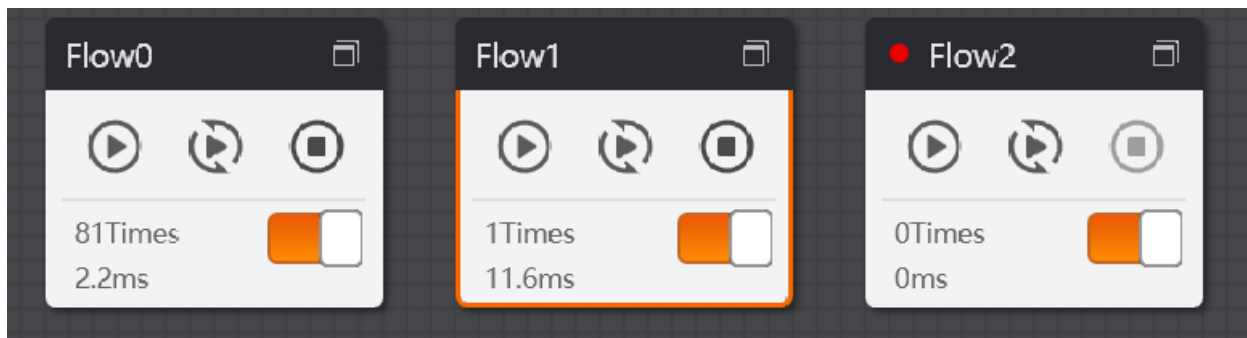



Figure 4.6 Main flow

- **Global Variable:** after entering global variable interface, you can add 32 global variables at most by clicking . Setting name, type, and current value is available. After enabling communication initialization, you can set communication string. For example, if the variable is var0, you can set its value as 0 by send SetGlobalValue: var0=0 via communication tools, as shown below.

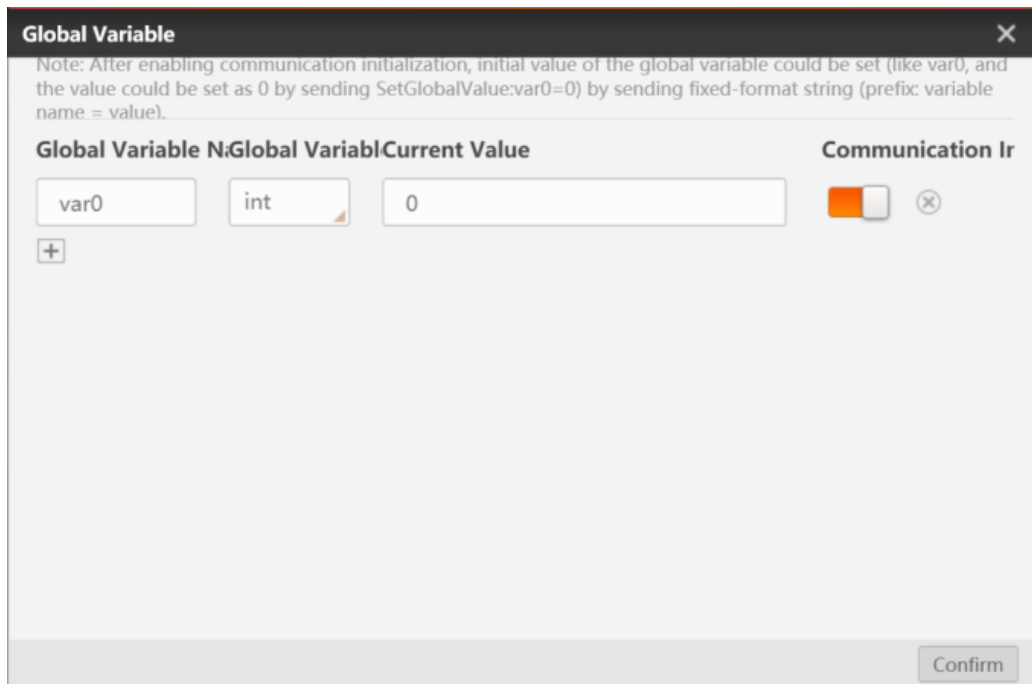


Figure 4.7 Global variable

- **Communication Management:** it sets communication protocol and parameters, supports four types, including TCP Client, TCP Server, UDP and Serial Port, as shown below.

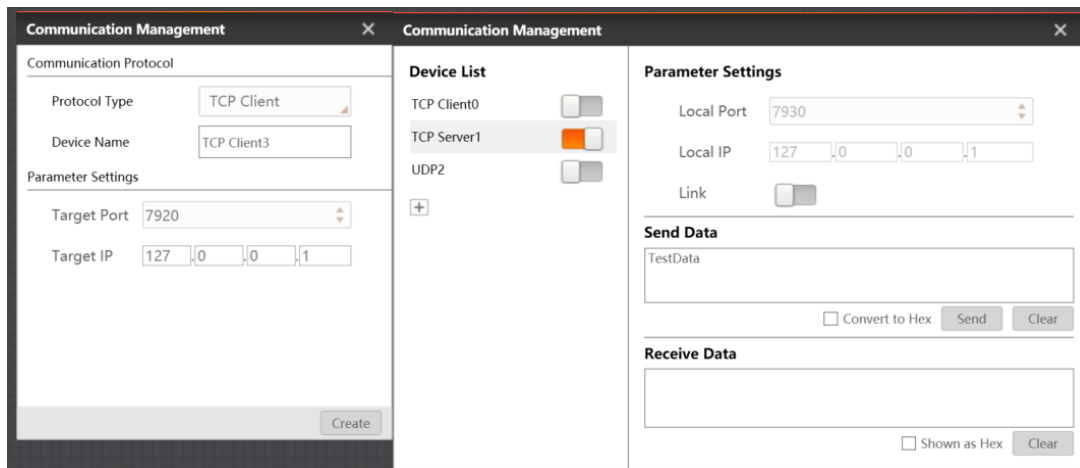


Figure 4.8 Communication Management

- **Data Queue:** the data queue is used to transmit data. Setting data queue is required before receiving data. Click Main Flow, select data queue, and add data queues of different types. Only there is a value in the existing queue can data queue receive data, otherwise the receiving module returns an error.

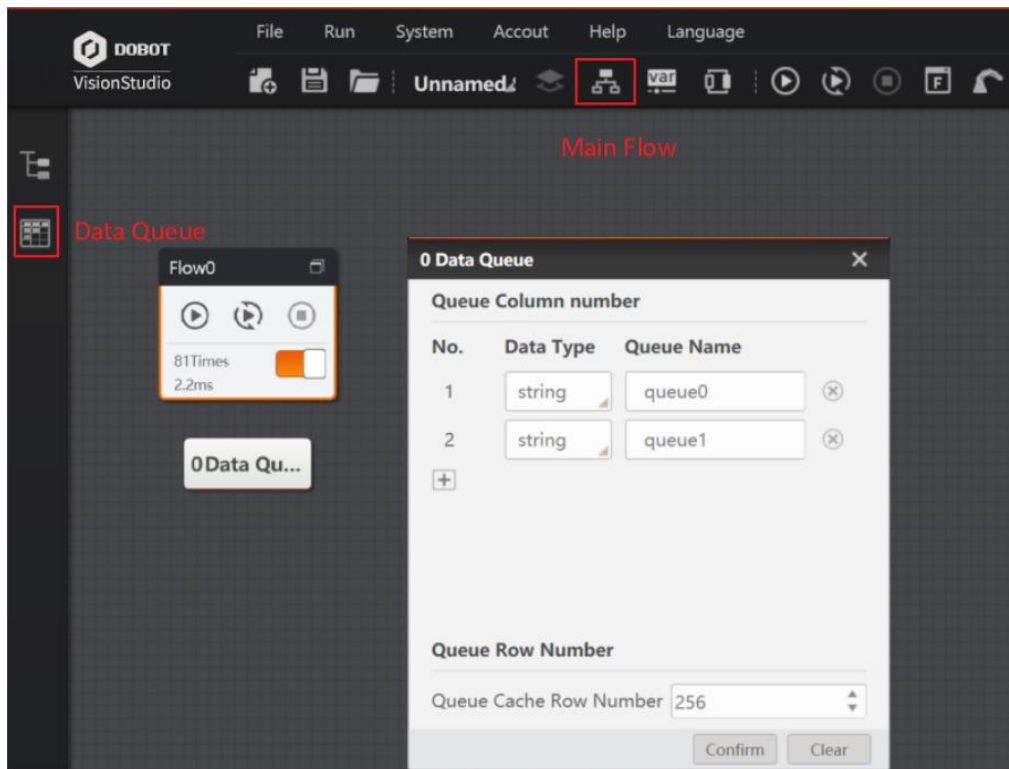


Figure 4.9 Data Queue

- DobotStudio: Click  to enter the DobotStudio control page.

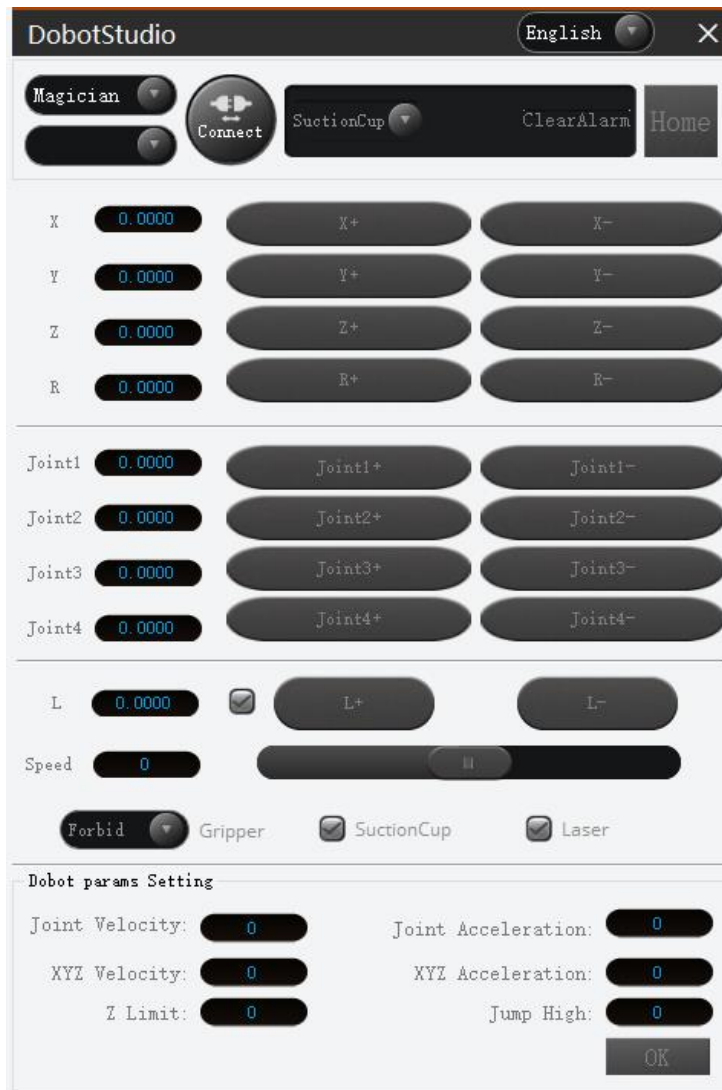

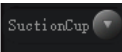
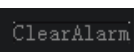


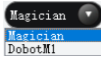


Figure 4.10 DobotStudio

Table 4.3 DobotStudio description

Function	Description
Robotic arm connection	Click  to connect or disconnect the dobot arm
End-effector selection	Click  to select the end-effector of the robotic arm
Alarm clear	Click  to clean alarm

Function	Description
Homing	Click  to execute the homing function. When the robotic arm is moving, if its movement is hindered by an obstacle or the stepper motors lost steps, perform the homing operation
Language	Click  to switch system language
Robotic arm selection	Click  to select robotic arm to Magician or DobotM1 (only support serial port connection).
Coordinate jogging	Jog the Dobot Magician by clicking X (X+/-), Y (Y+/-), Z (Z+/-), or R (R+/-) in the Cartesian coordinate system
Joint jogging	Jog the Dobot Magician by clicking J1+/-, J2+/-, J3+/-, or J4+/- in the Joint coordinate system
Sliding rail control	When the sliding rail is enabled, click L+/- to move the robotic arm along the rail. Value range: 0 mm - 1000 mm
Gripper control	When the end-effector is chosen as a Gripper , you can set the gripper to open, close, or disable in the Gripper drop-down box
Suction cup control	When the end-effector is chosen as a Suction Cup , select SuctionCup to power on the air pump. If unselected, the air pump is powered off
Laser control	When the end-effector is chosen as a Laser , select Laser to turn on the laser. If unselected, the laser is turned off
Jogging speed control	Set the jogging speed percentage Default value: 50% Value range: 1% - 100%

Function	Description
Robotic arm parameter setting	Set the robotic arm parameters, such as Joint Velocity , Joint Acceleration , XYZ Velocity , XYZ Acceleration , Z Limit , and Jump High

4.3 Tool

4.3.1 Tool Introduction

There are different tools in the left for operations.

- You can add frequently-used tools to the **Common Tool**.
- **Collection** includes **Local**, **Camera** and **Save Image**.
- Visual processing tools include **Location**, **Measurement**, **Reorganization**, **Calibration**, **Contraposition**, **Image Processing**, **Color Processing**, **Defect Detection**, and **Logical tool**. You can select the tool as you desire.
- **Communication** supports communication IO, ModBus protocol communication, PLC communication. IO communication is designed for Dobot vision box.

4.3.2 General Parameter

Purpose

This section introduces the genera parameters, and the detailed parameters will be introduced in the following chapters.

Steps

1. Select the tool you desire, and drag it to the flow and release it so as to add the tool to the flow.
2. Double click the tool to display the parameter setting window.

Basic Parameters

Image Input: Select the image input source here.

ROI Area: You can draw the ROI area by the shape you have selected, or you can inherit certain featured area. Please refer to 5.2.10 for details.

Note: For the tools that use ROI, the searching direction is the ROI area direction. ROI arrow direction is the x axis. The x axis direction refers to the direction of "from left to right".

Mask Region: The bland region will not be processed. Up to 32 vertexes cab be set.

Position Correction: It works with the position correction tool.

Run Parameters

Run parameters differ according to different tools. Please refer to specific chapter for these parameters.

Result Show

Check Result: Enable the parameters and input the specific value for checking. The exceeding part will display as red.

Image Display: Enabled by default.






Text Display: Set the text content, OK color, NG color and etc.

Chapter 5 Image Acquisition

You can select the image files, acquire image via camera or save the image here.

5.1 Image Files

Steps:

1. Move to  and drag the **Image Files** icon to the right area and release it.
2. Move to  to add the image. You can also move to  to load the image folder or move to  to delete the image.
3. Move to  to display the image parameter setting interface.

The interface is shown as below.

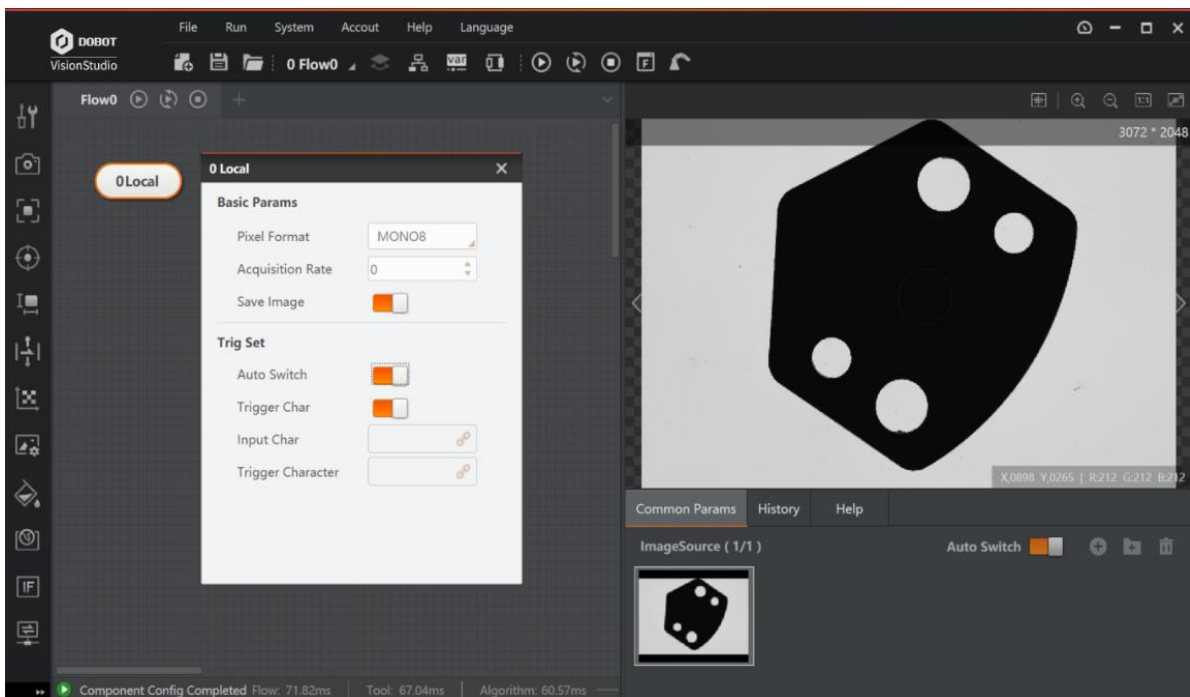


Figure 5.1 Image files

Pixel Format: You can select the MONO8 or RGB24.

Acquisition Interval: The time interval for collecting the image, unit: ms.

Save Image: Once enabled, when saving the solution, the image will be saved.

Auto Switch: Once enabled, the images will be switched automatically.



Trigger Character Filter: Once enabled, you can control this function via external communication control.

Input Character: Select the input character.

Trigger Character: Once set, only the specific characters can trigger this function.

5.2 Camera Image

Steps:

1. Click  and drag the **Camera** icon to the right area and release it.
2. Click Choose Camera under Common Para.
3. Double click  to display the camera parameter setting interface.

The camera image interface is shown as below.

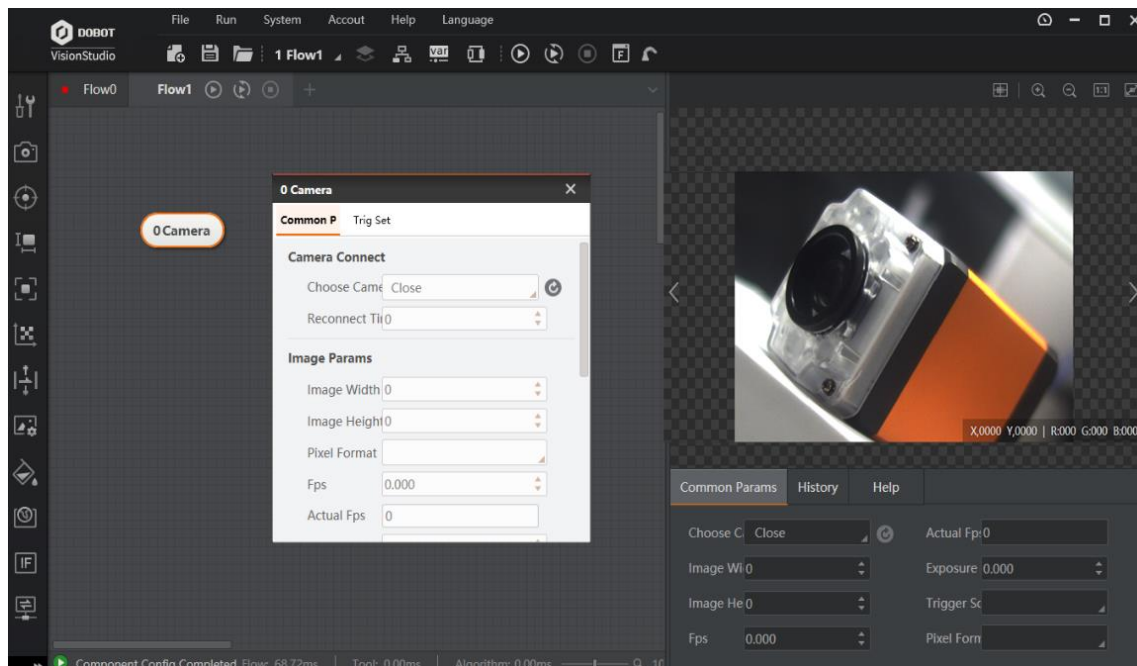



Figure 5.2 Camera Image

- **Choose Camera:** You can select the machine vision camera that is in the same network segment with the PC.
- **Image Width/Height:** View the image width and height here.
- **FPS:** Set the frame rate of the connected camera, which influences the image collecting speed.

- **Actual FPS:** The real-time frame rate of the camera.
- **Exposure Time:** The camera exposure time, which influences the image brightness.
- **Pixel Format:** Mono8 and RGB8Ppacked available.
- **Gain:** Brighten the image via gain when not adding the exposure.
- **Gamma:** For 0° to 1°, the dark part of the image will be brightened; for 1° to 4°, the dark part of the image will be darkened.
- **Horizontal Frequency:** For the line scan camera.
- **Actual Horizontal Frequency:** For the line scan camera.
- **Trigger Source:** Software trigger means the DobotVisionStudio controls the camera; hardware trigger means the external hardware controls the camera.
- **Trigger Delay:** After receiving the triggering signal, the trigger will be delayed according the time set here.
- **Trigger Character Filter:** Once enabled, you can control this function via external communication control.
- **Input Character:** Select the input character.
- **Trigger Character:** Once set, only the specific characters can trigger this function.

5.3 Save Image

Steps:

1. Click  and drag the **Save Image** icon to the right area and release it.
2. Double click it to display the saving image setting interface.

The saving image interface is shown as below.

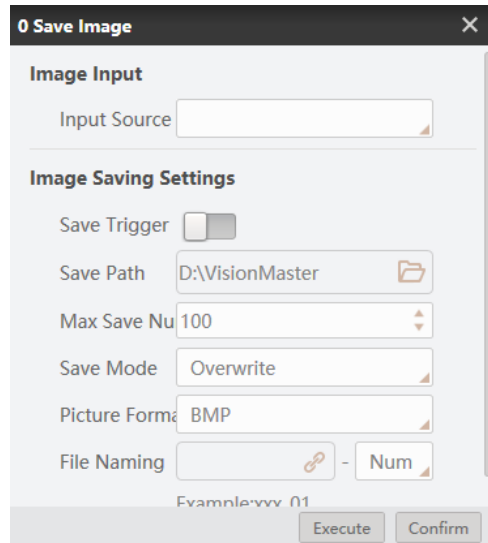


Figure 5.3 Save image

- **Input Source:** Select the image for saving.
- **Save Trigger:** Once enabled, you can save the image according to the triggering conditions.
- **Save Path:** The path of saving the image.
- **Max. Saving Number:** The number of the maximum image saved.
- **Save Mode:** When the images saved reach the maximum number, you can select **Overwrite** or replace the images you have saved or **Stop** to stop the saving.
- **Picture Format:** BMP and JPEG available.
- **File Naming:** For naming the image.

Chapter 6 Vision Tools

Processing function area includes all vision processing tools, and it is convenient for users to use. Processing function includes ten tool groups such as location, measurement, recognition, calibration, contraposition, image and color processing, defect detection, logic tool and communication tool, as shown below.

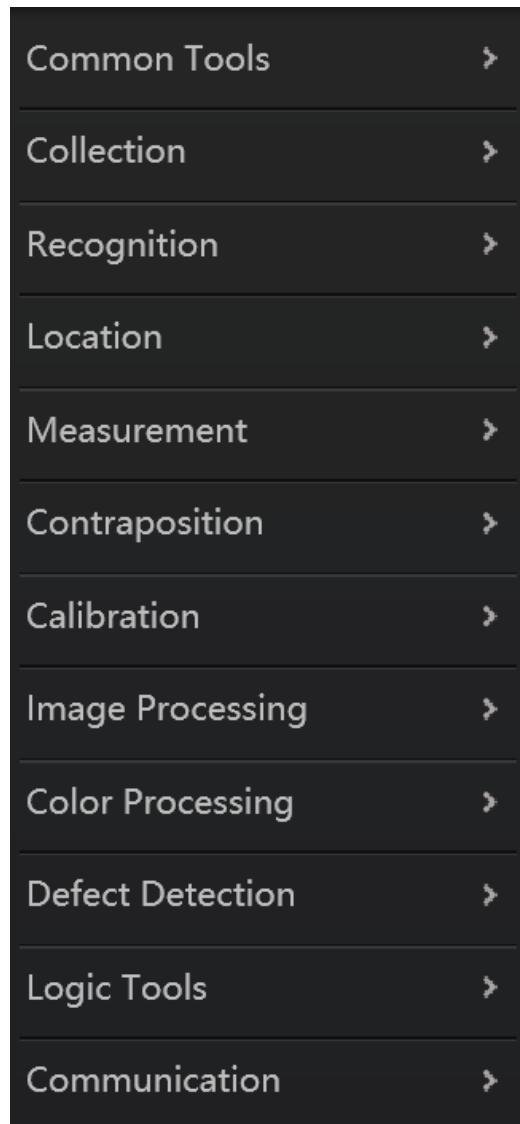


Figure 6.1 Vision tools

6.1 Location

There are 13 tools in location menu as shown below. The main function of these tools is to locate or detect certain features in images.

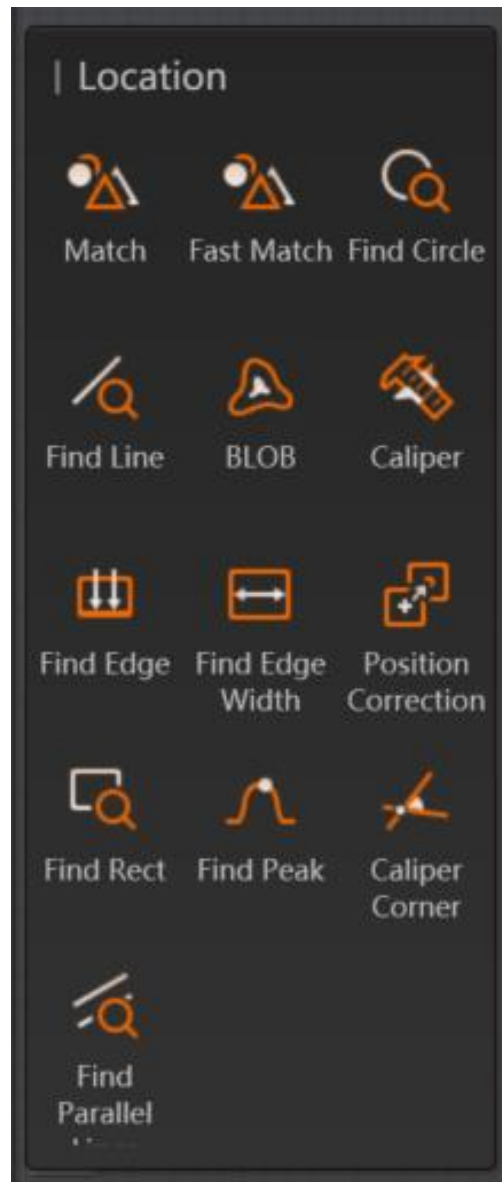


Figure 6.2 Location

6.1.1 Feature Matching

Feature matching includes match and fast match. This tool uses the edge feature of image as template, confirms search space in accordance with preset parameters, and searches targets that are similar with the template among images. This tool can be used to locate, calculate and verify.

Match has high precision, but its operation speed is slower than fast match. Feature matching function searches target images by geometric feature learning model of template image, and it is used together with images, as shown below.

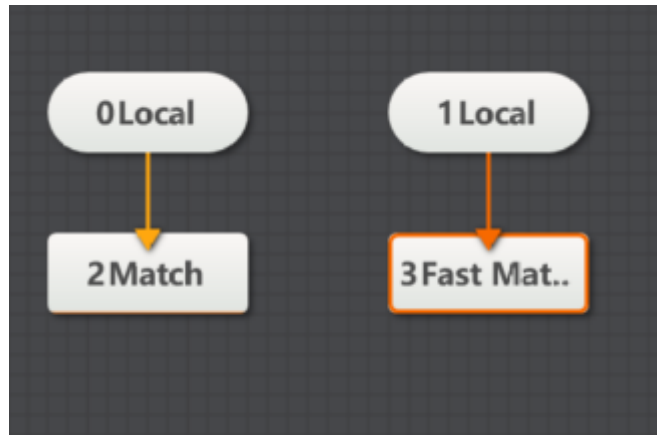


Figure 6.3 Feature matching

Double-click feature matching module to set parameters. These parameters include basic parameters, feature template, operation parameters, result display and etc. For basic parameters and result display, you can refer to Tools Application Example Section. Here we introduce feature template and operation parameters.

Feature Template

The feature template is used to extract image features. For initial use, you need to edit its template. Select the template area that needs to be edited, and click training model after setting parameters, as shown below.

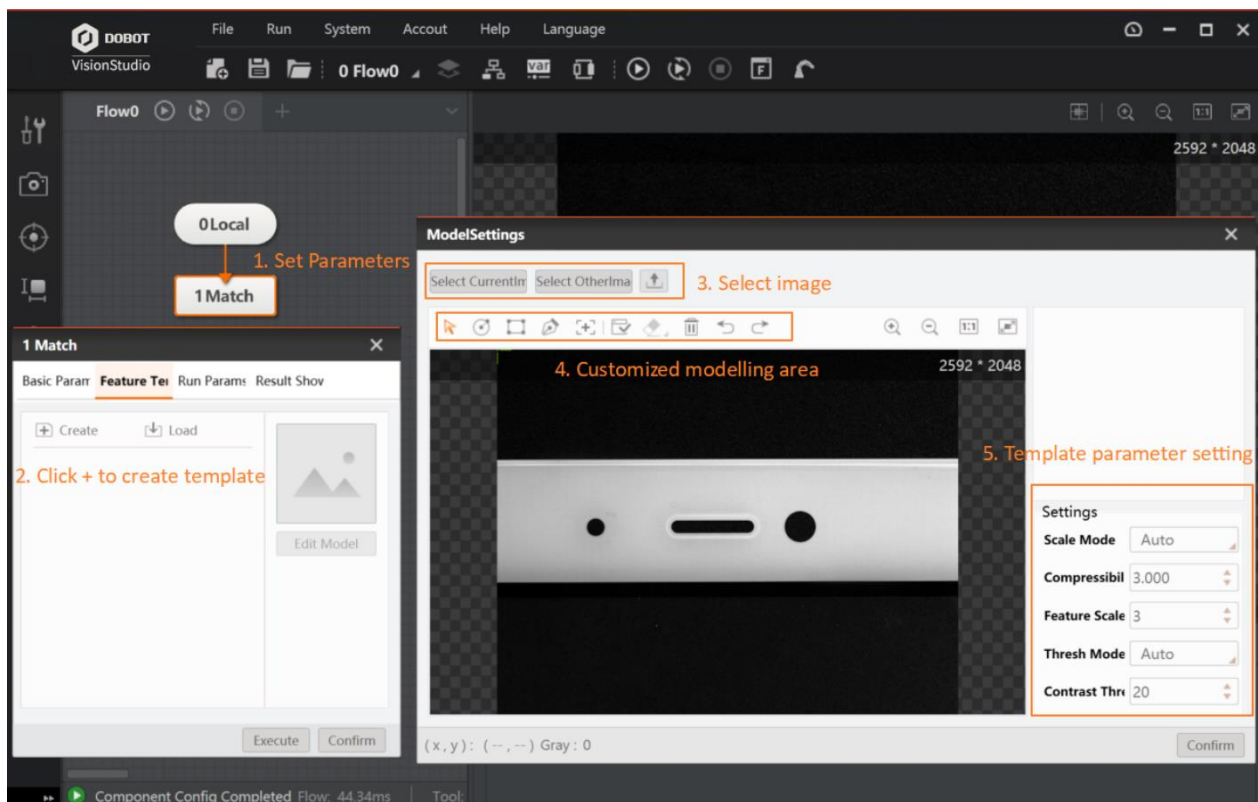


Figure 6.4 Feature template

The shortcuts shown in area 4 mean moving image, creating fan circle mask, creating rect mask, creating polygon mask, selecting model match center, building model, erasure contour point, clearing all model ling area, undo and back from left to right.

- **Match Point:** it is used to create location standard. Click Select Model Match Center, and set match point by yourself.
- **Scale Mode:** when its auto mode can meet demands, it will not adjust. if not, it will switch to manual mode.
- **Roughness scale:** the higher the roughness scale parameter, the greater feature scale, and its respective edge points will be sparse. But, in this way, it can speed up feature matching speed.
- **Feature Scale:** it is used to extract the fine level of feature granularity. The fine scale can be integer only and it cannot be larger than roughness only. When the value is 1, it means the finest. After adjustment, the contour point quantity will change, as shown below.

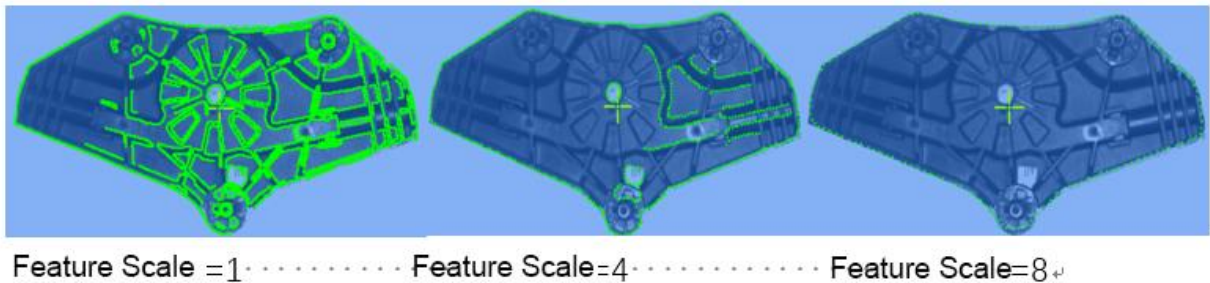


Figure 6.5 Feature scale

- **Threshold Mode:** when its auto mode can meet demands, it will not adjust. if not, it will switch to manual mode.
- **Contrast Threshold:** it is related with feature points and gray value of background. The larger the value, more feature points will be eliminated.

Running Parameter

It is used to set parameters of feature match in order to set search space. The targets in specific search space can be searched only, as shown below.

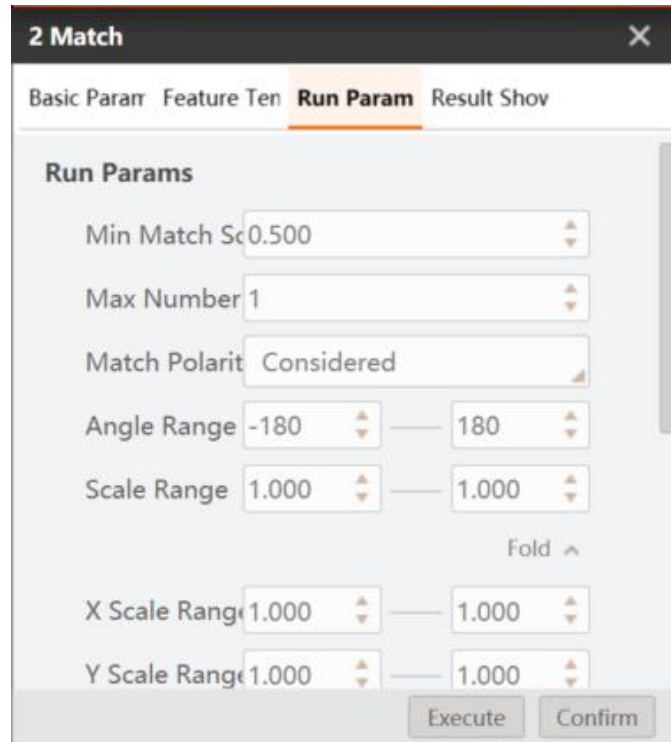


Figure 6.6 Running parameter

- **Min. Match Score:** the match score means the similarity between feature template and target in searched images. It is also called as similarity threshold. Only when the target's similarity reach similarity threshold, it can be searched out. The max. value is 1 which means totally matched.
- **Max. Match Number:** it is the max. target quantity can be searched.
- **Match Polarity:** polarity means the color transition from feature image to background. When the edge polarity of the searched target does not match with feature template polarity, but you want to make sure the target is searched, then you need to set match polarity as ignored. if not, you can set as considered, and search time can be reduced.
- **Angle Range:** it means the angle changing range when comparing the target and created template. If you want to search the proactive target, and then you need to set respectively, and the default range is -180° to 180° .
- **Scale Range:** it means scale changing range when comparing the target and created template. If you want to search the target with same scale changing, and then you need to set respectively, and the default range is 1.0 to 1.0.

Advanced parameters can set some special parameters. If the basic parameters can meet demands, you do not need to set advanced parameters.

- **X/Y Scale Range:** it is the range scaled along the x/y direction. If it is used with the Scale Range function, the final result is multiplied. This parameter is only valid in match.

- **Overlap Threshold:** when searching for multiple targets, two detected targets overlapping with each other, and the maximum overlap ratio allowed by the matching frames is overlap threshold., The larger the value, the greater the degree of overlapping between the two targets, and the range is 0° to 100°, as shown below.

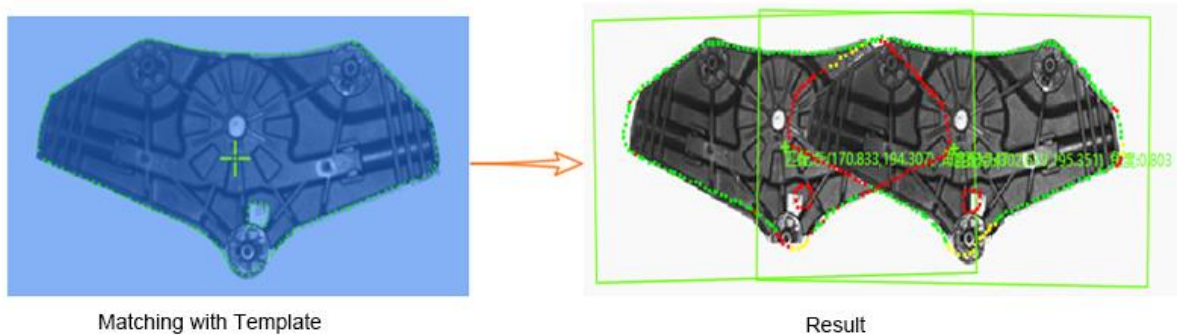


Figure 6.7 Overlap threshold

- **Sort:** After the matching is completed, the matched image will be displayed in the current result. When there is more than one matching feature image, the sorting is ranked according to the setting in the sorting type.
 - Sort by score in descending order: sorted by the score of the feature match in descending order.
 - Sort by angle in descending order: in descending order of relative angular deviation in the current result.
 - Sort by X from small to large: the current result has the X coordinate of the matching box center. According to the X coordinate, the order is rounded up from small to large. When the X coordinates are the same, then is sorted by Y from big to small, and the Y axis has the same operation with X axis.
 - X is from small to large, and Y is from small to large too: the current result has the matching box center X/Y coordinates. According to the X coordinate, the order is rounded up from small to large. When the X coordinates are the same, then the order is sorted according to Y from small to large .

Here, we take the order sorted in descending order by angle as example. The feature template is created by image 1, and the matching point is used as the origin point. When the target image is transformed, the matching point will also change. The angle is the matching angle of the target image compared with the target image. It is important to emphasize that the angle is positive after clockwise rotation, and the angle is negative after counterclockwise rotation. The changed angle of the target image of No. 3 is 110.192° compared with the feature image.

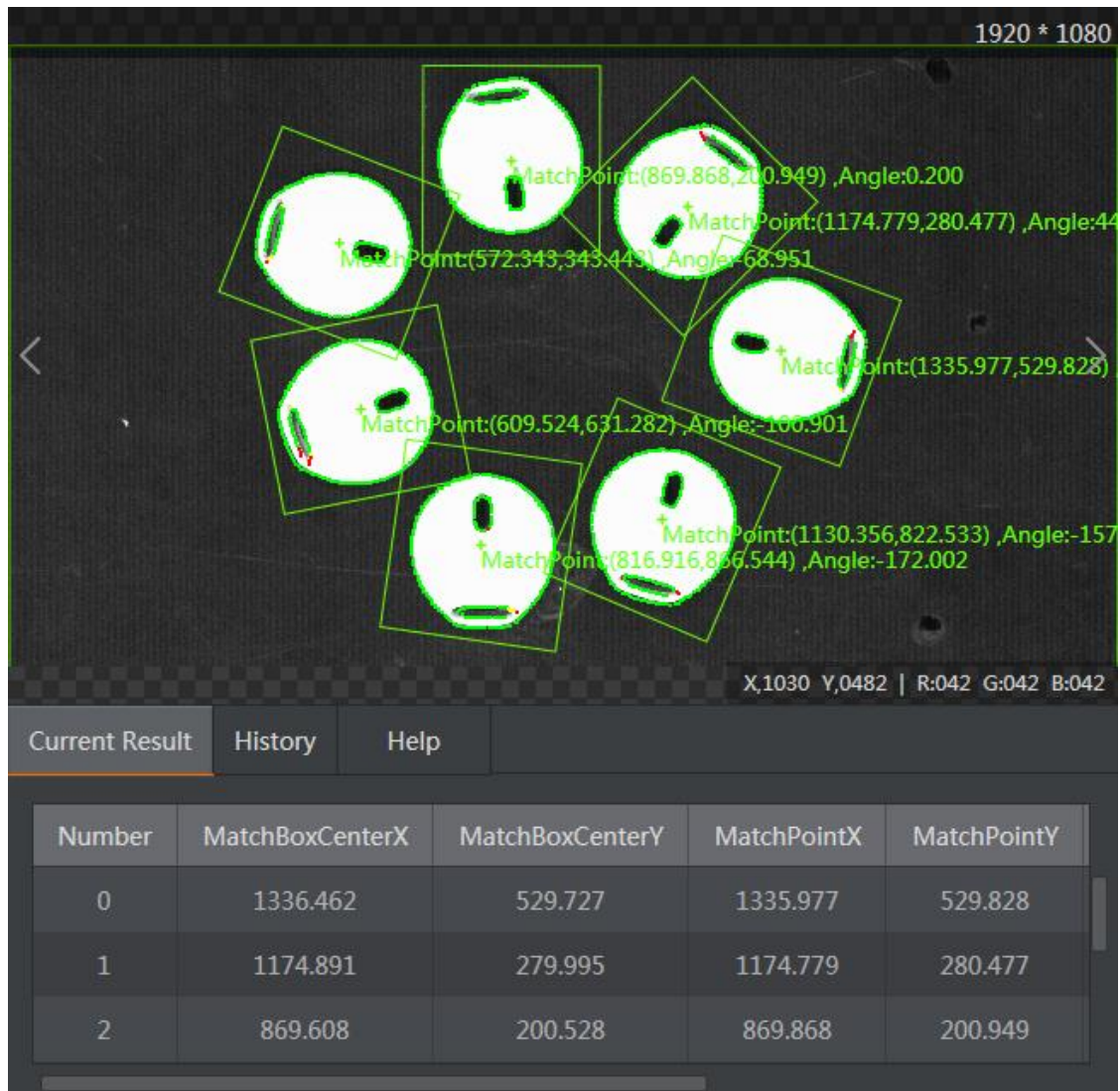


Figure 6.8 Sort

- **Threshold Type:** it indicates contrast threshold mode in the matching stage, and threshold type has three modes as shown below.
 - Automatic threshold: it automatically determines the threshold parameter according to the target image.
 - Template threshold: it uses the contrast threshold of the template as the contrast threshold in matching phase.
 - Manual threshold: it uses the threshold set by the user as the threshold parameter for searching.
- **Match Point:** it is used to create location standard. Click Select Model Match Center, and set match point by yourself

- **Extension Threshold:** it is the ratio that the portion of missing feature compared with the complete feature when the image is incompletely displayed at the edge of the image. When the searched target appears in the edge of the image, the extension threshold can make sure the image is found, as shown below. As long as the extension threshold is set to be larger than 20, the top target image can be found.

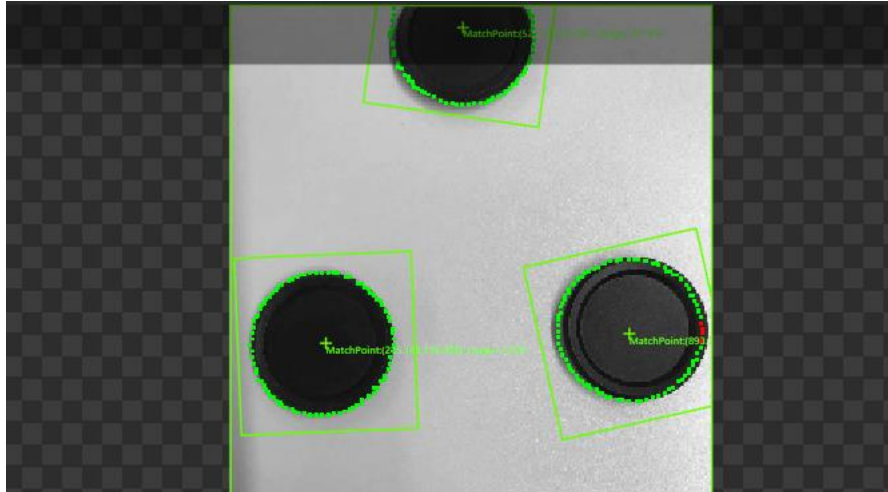


Figure 6.9 Extension threshold

- **Clutter Considered:** If mottle considered is enabled, the algorithm will consider the mottle feature. If the feature has burrs, the score will decrease.
- **Timeout:** When the time exceeds the time set by the overtime control, the search will stop and no search result will be returned. The value range of overtime control is 0 ms to 10000 ms, and 0 means that the overtime control is disabled.
- **Contour Enabled:** If contour enabled is enabled, the contour feature points of the template are displayed. If it is disabled, the feature points are not displayed, and only the matching frame is displayed, which can reduce operation time.

6.1.2 Find Circle

The tool of finding circle detects multiple edge points first, and then fits them into a circle. This tool is used for location and measurement of the circle. The basic parameters and result display have been explained in the Section 3.3.2. Here we take running parameters as an example, as show below.

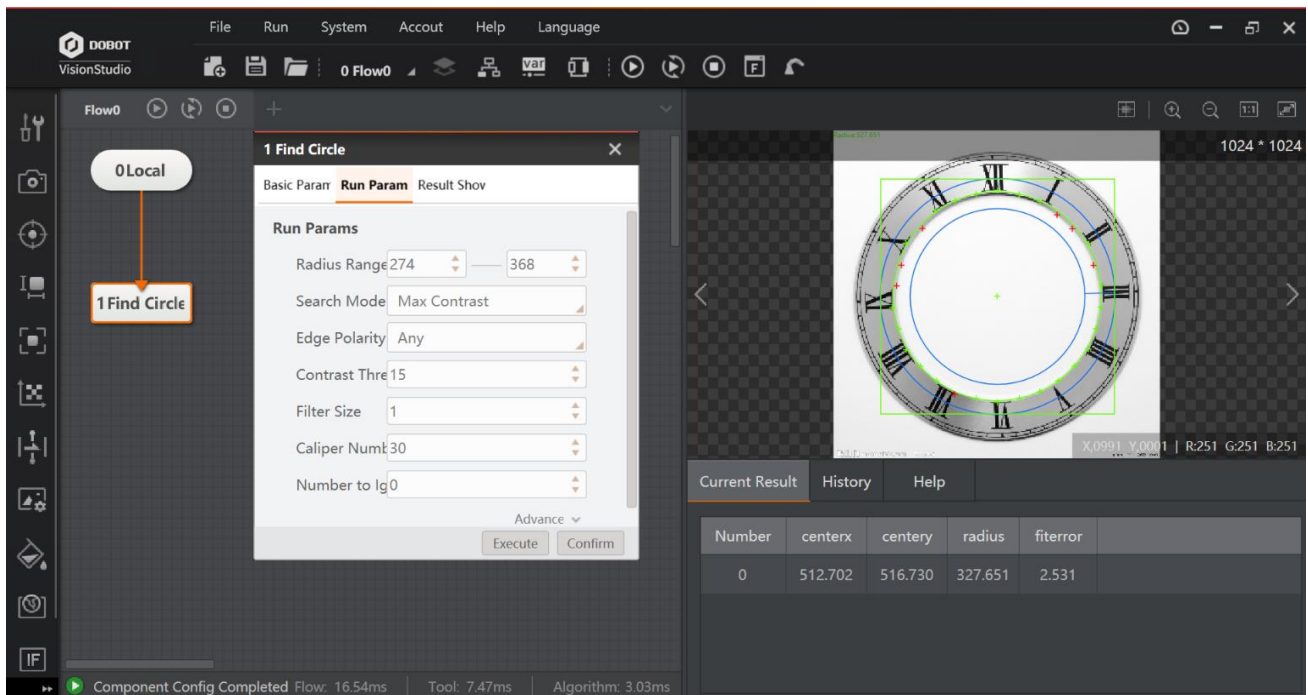


Figure 6.10 Find circle

- **Radius Range:** it includes the minimum radius and the maximum radius, and the radius of the target circle must be within this range.
- **Search mode:** it has three mode, including max contrast mode, max. radius mode and min. radius mode.
 - Max contrast: it means that only edge points of the largest gradient threshold within the scan range is detected and the circle is fitted.
 - Max. radius: it means that only edge points of the longest distance from the circle center within the scan range is detected and the circle is fitted.
 - Min. radius: it means that only edge points of the shortest distance from the circle center within the scan range is detected and the circle is fitted.
- **Edge Polarity:** it has three modes, including dark to light, light to dark and any.
 - Dark to light: it means that the transition from the area with a low gray value to the edge of the area with a high gray value.
 - Light to dark: it means the transition from the area with a high gray value to the edge of the area with a low gray value.
 - Any means both edges are detected.
- **Contrast Threshold:** it is also called gradient threshold, and its range is from 0 to 255. The edge point whose gradient threshold is larger than this value can be detected. The larger the value,

the stronger noise resistibility, the smaller the number of edges obtained, and even the target edge points are filtered out.

- **Filter Size:** it is used to enhance the edge and suppress noise, and its minimum value is 1. When the edge is blurred or there is noise interference, you can increase its value to make the detection result more stable. If the distance between the edge and the edge is smaller than the filter size, it will affect the accuracy of the edge location or even lose edge. This value needs to be set based on the actual situation.
- **Calipers Number:** it is used to scan ROI area quantity of edge points, as shown below.

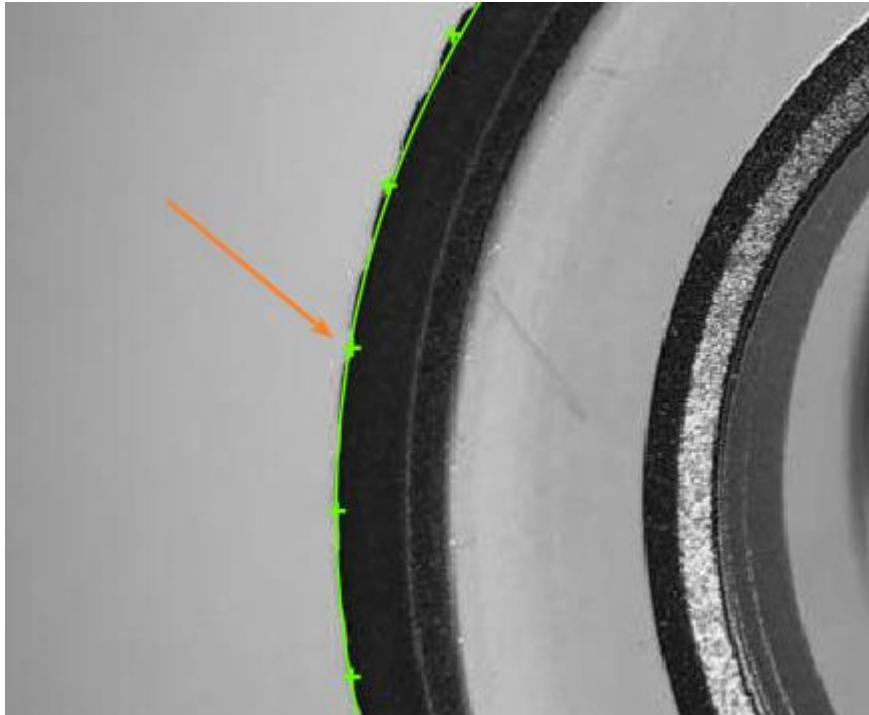


Figure 6.11 Calipers number

- **Number to Ignore:** it means that the number of minimum points that have high error to be excluded from fitting. In general, if there is a great number of points excluded from fitting, its value should be set higher. For better results, it is recommended to use it in combination with the parameter of distance to remove.
- **Distance to Ignore:** it means that the maximum pixel distance from the outlier to the fit circle. The smaller the value, the more points are excluded.
- **Caliper Length:** it is used to describe the width of the area where the scanning edge points find the ROI. Within certain range, increasing this value can create more stable edge point.
- **Subsampling coefficient:** it is also called downsampling, which means that the number of sampling points is decreased. For an image of $N \times M$, if the subsampling coefficient is K , then one dot is taken every K points of each row and column in the original image to create an

image. Therefore, the larger the subsampling coefficient, the more sparse the contour points will be, and the less detailed contour will be. It is recommended that the value should not be set too large.

- **Locating Sensitivity:** it is used to eliminate the interference points. The larger the value, the stronger the ability to eliminate noise interference, but it is also easy to cause the initial location of circle failure.
- **Initial Fit Type:** it has two types, including global fit and local optimum.
 - Local optimum: local optimum is to fit the circle according to the local feature points. If the local features more accurately reflect the location of the circle, the local optimum is adopted, otherwise the global fit is preferred.
 - Global fit: it is used to find global feature points and fit circle.
- **Fit Mode :** it has three types, including least squares, huber and tukey. These three fitting modes are different in way of calculating the weight only. As the number of outliers and the distance from the group increases, it is recommended to use least squares, huber, and tukey successively.

Note:

This tool can only find one circle at one time. If you want to find multiple circles, it is recommended to use it together with the loop function.

6.1.3 Find Line

The tool of finding line is used to find a line with certain features in the image. It uses known feature points to form feature point group, and then fits into a line. The basic parameters and result display have been explained in the Section 3.3.2. Here we take running parameters as an example, as show below.

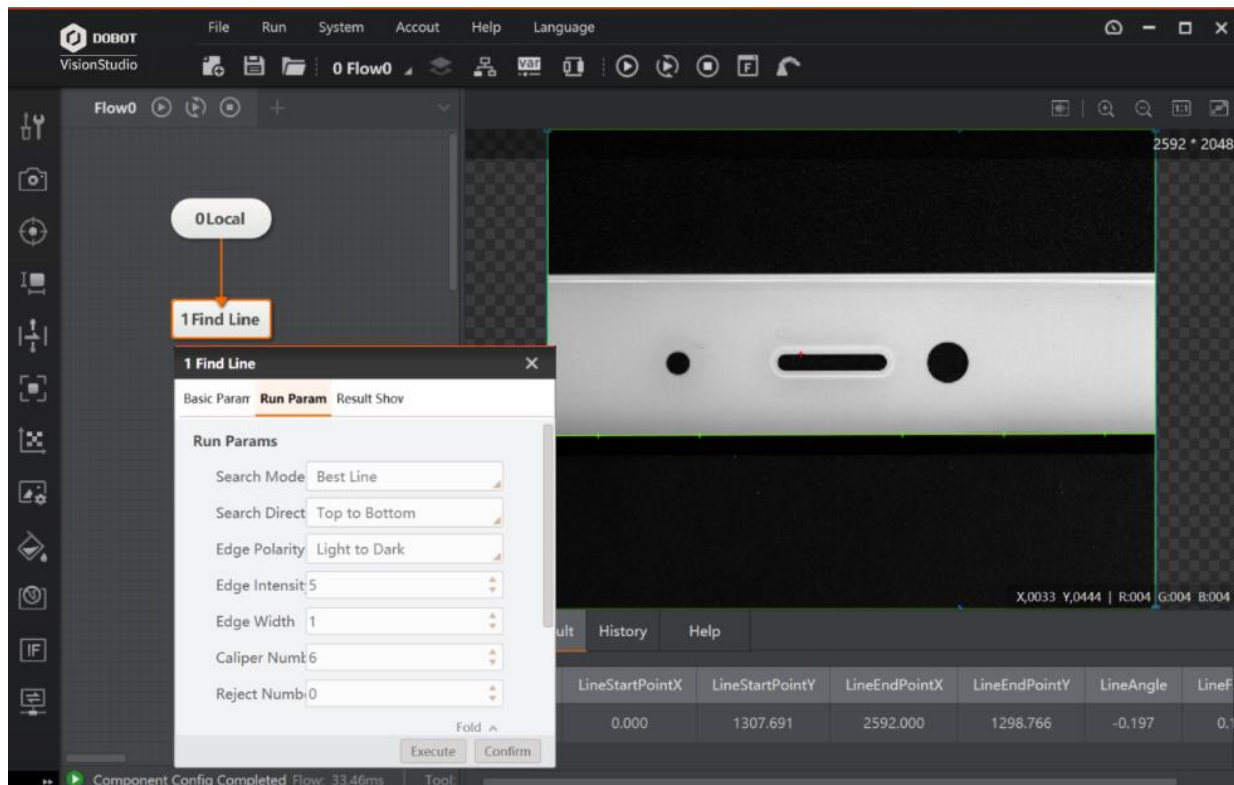


Figure 6.12 Find line

- **Search Mode:** it has three modes, including best line, first line and last line.
 –Search best line: it means that find the set of edge points with the largest gradient threshold, and then fit them into a line.
 –Search first line and last line mean that finding the first and last line that meet searching condition.
- **Search Direction:** on the basis of rectangular ROI area, and it has two methods, including from top to bottom and left to right.
- **Edge Polarity:** it has three polarities, including dark to light, light to dark and any.
- **Edge Strength:** it is used to filter the noise. The larger the value, the stronger the ability to eliminate noise interference, the less obtained edge quantity, but it is also easy to filter target edges.
- **Edge Width:** it means the edge span. The larger the span, the slower the edge transition speed will be. It is recommended to set a large value. But if the two edges are too close, the large value may result in edge losing.
- **Projection Width:** it is the caliper width. It is used to describe the area width of searching edge point ROI. Within certain range, increasing this value can create more stable edge point, as shown below.

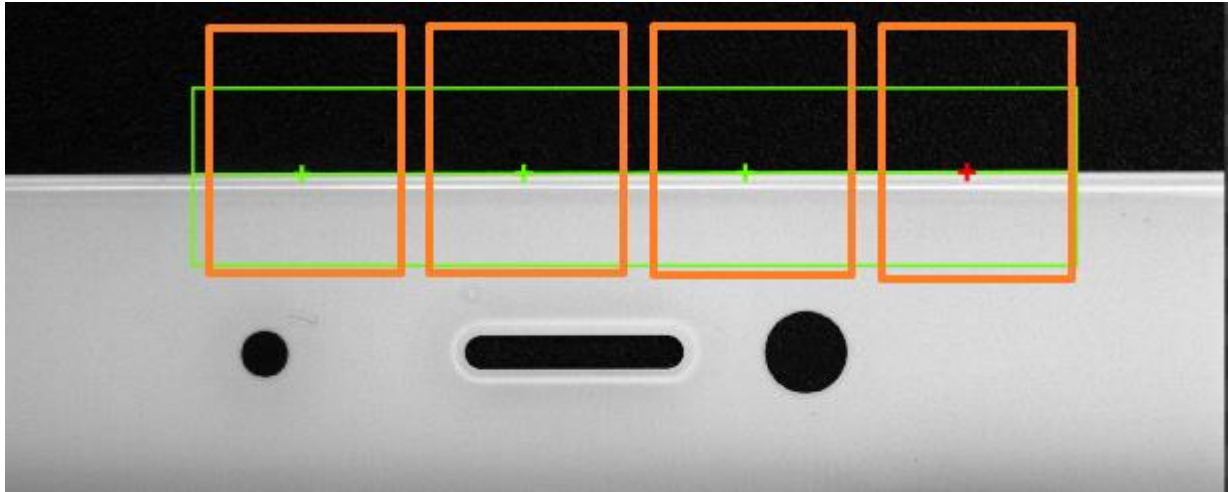


Figure 6.13 Projection width

6.1.4 Blob

Blob analysis is a process of detecting, locating, or analyzing a target object in an image area where the pixel is a limited gray level. The Blob tool provides information about certain features of the target object in the image, such as presence, quantity, position, shape, orientation, and topological relation between blobs. The basic parameters and result display have been explained in the Section 3.3.2. Here we take running parameters as an example, as show below.

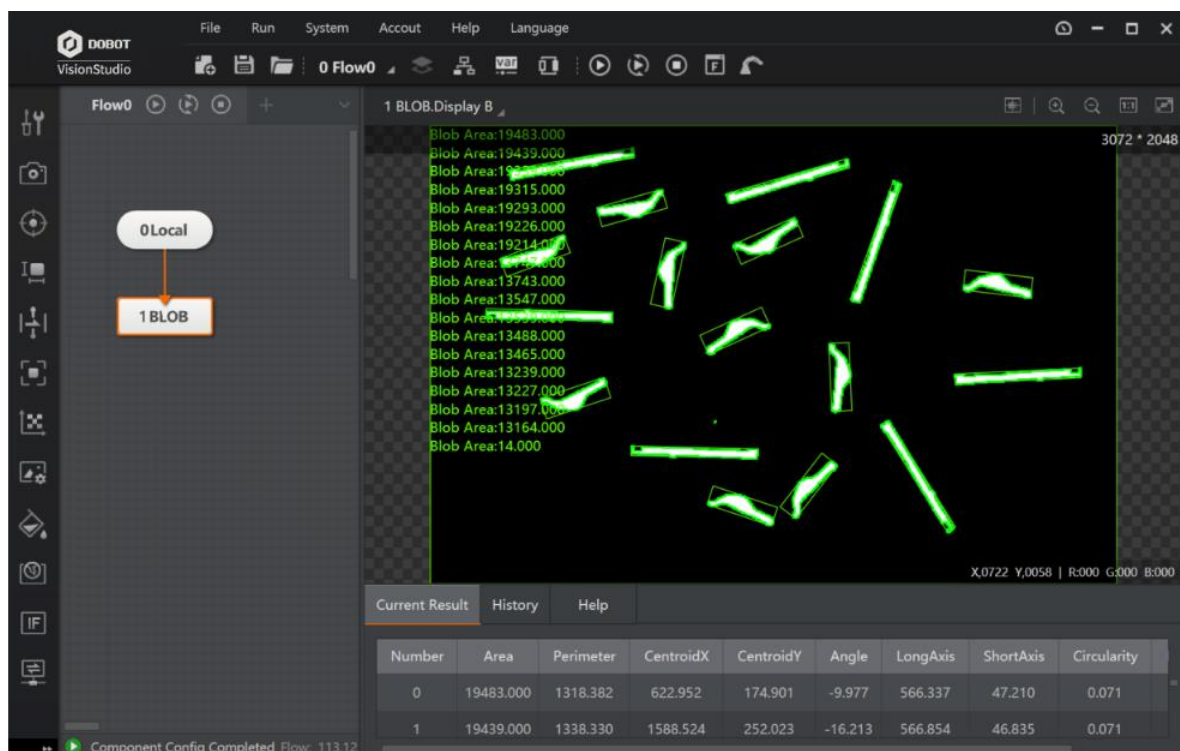


Figure 6.14 Blob

- **Threshold Mode:** When the input image is a binary image, it is optional to not execute binarization. In other cases, there are five options, including single threshold, double threshold, auto threshold, soft threshold (fixed) and soft threshold (relative).
- When single threshold, double threshold, or auto threshold is selected.
 - Low threshold: it is used to set lower limit of threshold.
 - High threshold: it is used to set upper limit of threshold.
 - When the high threshold is higher than the low threshold, the target gray range is [low threshold, high threshold]. When the low threshold is set higher than the high threshold, the target gray range is [0, high threshold] and [low threshold, 255].
- When soft threshold (fixed) is selected.
 - Softness: It is a parameter that varies between the low threshold and the high threshold.
- When soft threshold (relative) is selected.
 - Softness: it is a parameter that varies between the soft threshold low threshold and the high threshold.
 - Low tail: it is the left tail of the threshold, which can be removed according to percentage.
 - High tail: it is the right tail of the threshold, which can be removed according to percentage.
 - Low scale coef: the low threshold ratio within the soft threshold range.
 - High scale coef: the high threshold ratio within the soft threshold range.
- **Polarity:** it has two modes, including dark blobs, light background, and light background dark background.
 - Dark blobs, light background is that the feature image pixel value is lower than the background pixel value.
 - Light background, dark background is that the feature image pixel value is higher than the background pixel value.
- **Threshold Range:** it is used to set the lower limit and upper limit of threshold. The target whose edge threshold within the threshold range, and it can be found in Blob area.
- **Number of Fund Blobs:** it is used to set the number of Blob graphs to find.
- **Min. Hole:** it refers to the mini. non-Blob area size in Blob area. If the parameter is not larger than this value, the hole padding is the Blob.
- **Enable:** this function means that only feature images within the parameter settings range can be found.
 - Area: the area of the target image.
 - Contour: the circumference of the feature image.

- Major and minor axis: the length and width of the smallest circumscribed rectangle, as shown below.

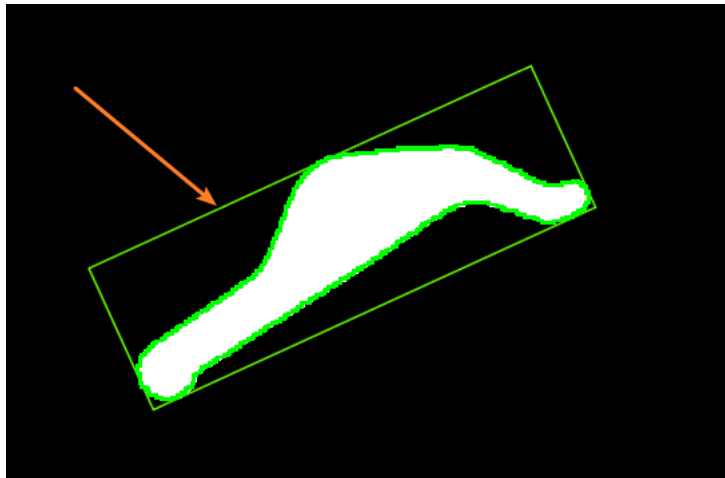


Figure 6.15 Major and minor axis

- Circularity and rectangularity: the similarity to a circle or rectangle.
- Barycenter offset: the pixel point of barycenter offset.
- **Connectivity:** In image processing, the object we are interested in is usually a combination of interconnected pixels. Therefore, in order to obtain an area, we must calculate all the connected areas that are contained in the area after division. On a rectangular pixel grid, there are two definitions of connectivity. The first one is that the two pixels have a common edge, that is, one pixel is above, below, to the left or to the right of the other pixel, which is called 4-connectivity. The second definition is the extension of the first definition. The adjacent pixels on the diagonal line are also included, which is called 8-connectivity. The relation between these two kinds of connectivity is shown below.

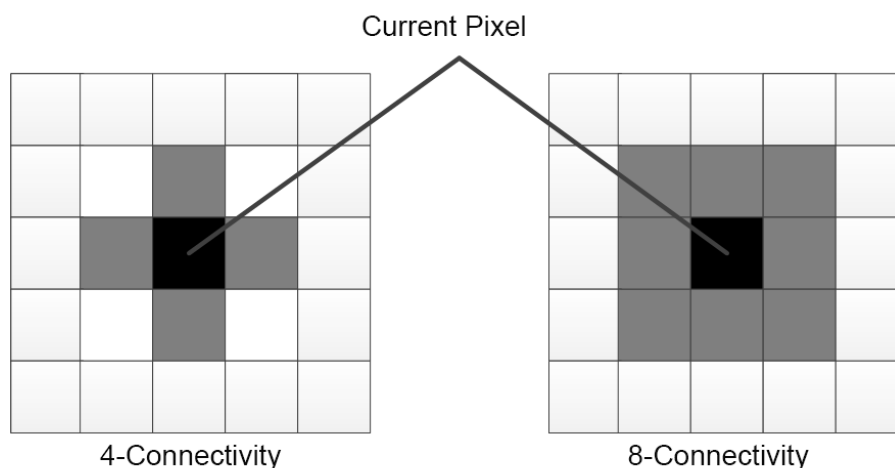


Figure 6.16 Connectivity

6.1.5 Caliper

The caliper is a visual tool that measures the width of a target object, edge location, the location of the feature or edge pair, and the distance between the edge pair. Unlike other visual tools, the caliper requires the user to confirm the approximate area of the measurement or position, and the target object or edge feature. The caliper tool can position edge or edge pair via selecting different edge modes. The basic parameters and result display have been explained in the Section 3.3.2. Here we take running parameters as an example, as show below.

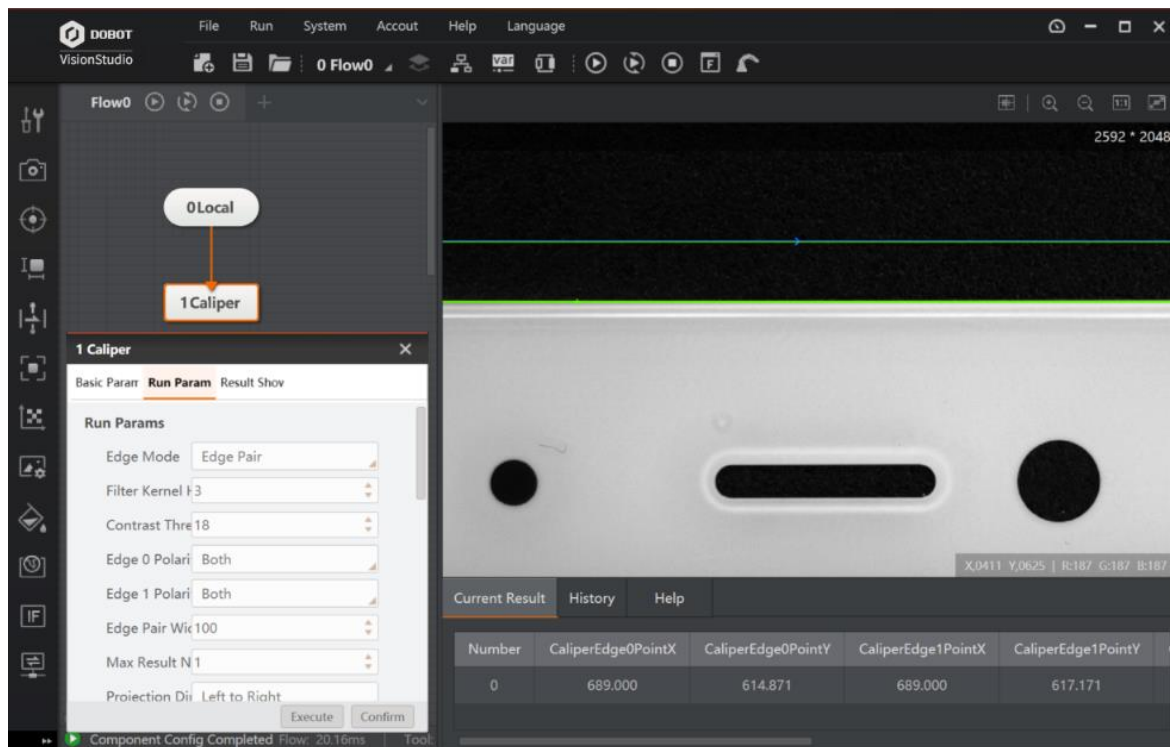


Figure 6.17 Caliper

- **Edge Mode:** it includes single edge and edge pair modes.
 - Single edge mode detects the edge position in the specific area and can be used to locate, count and verify the presence or absence.
 - Edge pair mode detects edge spacing in a specific area.
- **Filter Kernel Half-width:** it is used to enhance the edge and suppress noise, and the minimum value is 1. When the edge is blurred or there is noise interference, you can increase its value to make the detection result more stable. If the distance between the edge and the edge is smaller than the filter size, it will affect the accuracy of the edge location or even lose edge. This value needs to be set based on the actual situation.
- **Contrast Threshold:** it is also called gradient threshold, and its range is from 0 to 255. The edge point whose gradient threshold is larger than this value can be detected. The larger the value,

the stronger noise resistibility, the smaller the number of edges obtained, and even the target edge points are filtered out.

- **Edge Polarity:** it has three modes, including dark to light, light to dark and both.
- **Edge Pair Width:** it means that the size of the pixel spacing of the edge pairs that are expected to be output. Adjusting this parameter separately cannot directly screen out the expected edge pairs. Only when one or more score methods, including current position normalization score, relative position normalization score, space score, space gap score, and relative space gap score, are enabled, adjusting this parameter is meaningful.
- **Max. Result Number:** it is the maximum number of edge pairs expected to be output. If the logarithm actually found is greater than this parameter, the edge pair of this parameter is output according to the score from high to low. Otherwise, the actual edge pair quantity is output.
- **Projection Direction:** it includes from top to bottom and from left to right, the search direction is fixed relative to the search ROI.

– From top to bottom means that the edge point is searched in the left and right direction relative to the search ROI. If the search ROI is rotated by 90°, and then the edge point is searched from bottom to top direction.

– Left to right means that the edge point is searched in the top and bottom direction relative to the search ROI. If the search ROI is rotated by 90°, and then the edge point is searched from left to right direction.

Note:

Single edge mode has three kinds of scores enabling, as show below.

- **Contrast Score Enable:** it is scored by edge contrast or edge pair contrast. The user can execute contrast score on the candidate edges. The edge contrast here reflects the intensity of the grayscale change at the edge. The maximum score is 1.0, which is equivalent to an edge contrast of 255 (the maximum value of the edge contrast). If it is the search mode of edge pair, the average contrast of the two edges is used as the scoring factor.
- **Position Score Enable:** it is scored by the absolute position difference of the edge or edge pair center relative to the center of the ROI area.
- **Relative Position Enable:** it is scored by the absolute position difference (positive or negative) of the edge or edge pair center relative to the center of the ROI area. Each score enabling has six parameters, including curve type, starting point, midpoint, ending point, highest value, and lowest value of score. Curve type includes descending and ascending. The target edge point has the highest score by setting these six parameters, as shown below.

Edge pair mode has eight kinds of scores enabling, as show below.

- Contrast score enable, position score enable and relative position enable are same with that of single edge mode.

- **Normalized Position Score Enable:** it is scored by the absolute position difference of the edge or edge pair center relative to the center of the ROI area. The denominator of normalization is the edge pair width.
- **Normalized Relative Position Score Enable:** it is scored by the position difference of the edge or edge pair center relative to the center of the ROI area. The denominator of normalization is the edge pair width.
- **Spacing Score Enable:** it scores in accordance with actual distance of edge pair / edge pair width.
- **Spacing Difference Enable:** it scores by single edge in accordance with (actual distance of edge pair - edge pair width) / edge pair width.
- **Relative Spacing Difference Enable:** it scores by both edges in accordance with (actual distance of edge pair - edge pair width) / edge pair width.

6.1.6 Find Edge

The tool of finding edge is used to find edge with certain features in the image. The basic parameters and result display have been explained in the Section 3.3.2. Here we take running parameters as an example, as show below.

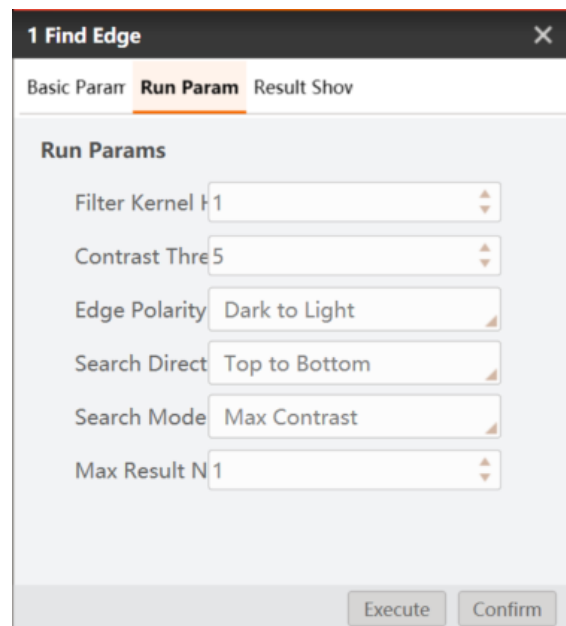


Figure 6.18 Find edge

- **Search Direction:** it includes from top to bottom and from left to right, the search direction is fixed relative to the search ROI.

- From top to bottom means that the edge point is searched in the left and right direction relative to the search ROI. If the search ROI is rotated by 90°, and then the edge point is searched from left to right direction.
- Left to right means that the edge point is searched in the top and bottom direction relative to the search ROI. If the search ROI is rotated by 90°, and then the edge point is searched from top to bottom direction.
- **Search Mode:** it has four modes, including max, contrast, first edge, last edge and all edges. Among them, max, contrast means that detect the set of edge points of the largest gradient threshold within scanning range.

6.1.7 Find Edge Width

The tool of finding edge width is used to detect the spacing between two feature edges. This tool searches edges that meet condition first, and then measures distance, as shown below. Here we take running parameters as an example.

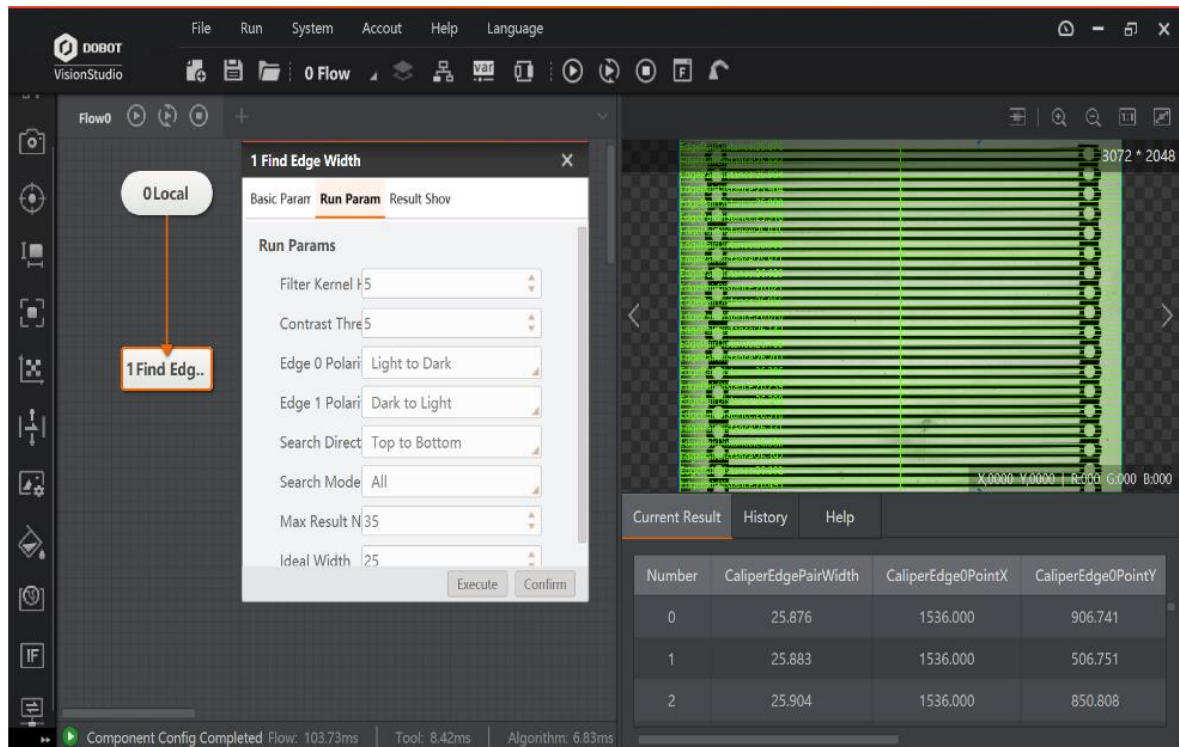


Figure 6.19 Find edge width

- **Search Mode:** it has eight modes, including widest edge pair, narrowest edge pair, max, contrast, min. contrast, first edge, last edge, nearest, and farthest
- Widest edge pair means that the edge pair that has the largest spacing within the detection range.

- Narrowest edge pair means that the edge pair that has the smallest spacing within the detection range.
- Max. contrast means that the edge pair that has the largest average gradient within the detection range.
- Min. contrast means that the edge pair that has the smallest gradient within the detection range.
- First edge means that the edge pair that its center has closest spacing to the search start point within the detection range.
- Last edge means the edge pair that its center has farthest spacing to the search start point within the detection range.
- Nearest means that the set of edge pair that has closest width to ideal width within the detection range.
- Farthest means that the set of edge pair that has least, closest width to ideal width within the detection range.

● **Ideal Width:** it is the pixel spacing size of expected output edge.

6.1.8 Position Correction

The tool of position correction is used to assist positioning and precise positioning, and modify target motion deviation. It establishes the reference of position deviation according to match points and match frame angle in the template matching result, and then realize ROI detection coordinate rotation deviation according to the relative position deviation of running point and fiducial point in the template matching result, as shown below.

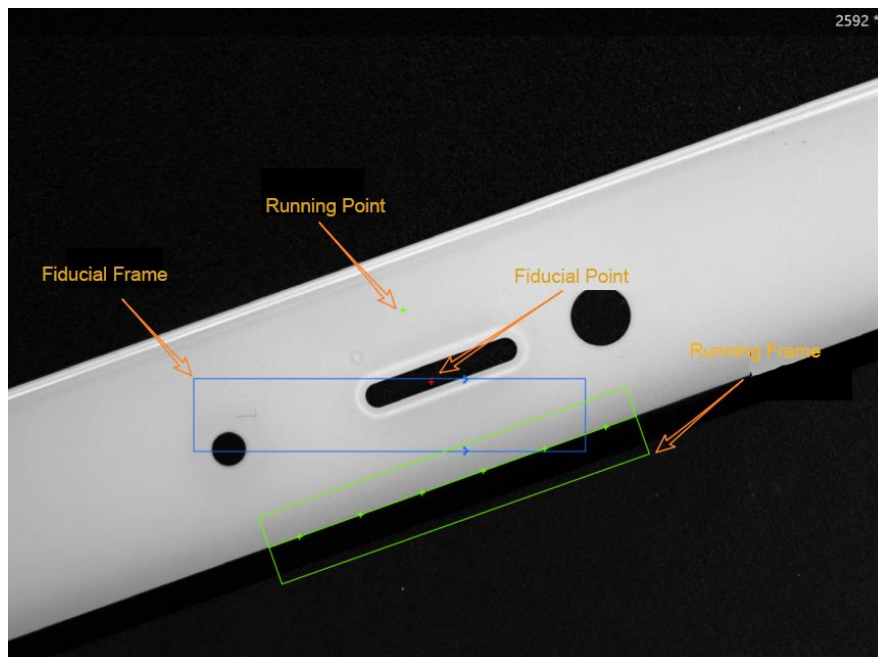


Figure 6.20 Template matching

Fiducial point and fiducial frame are the match point and match frame when creating reference. Running point and running frame are the match point and match frame when target image is matching with features. According to fiducial point and running point, the pixel deviation of image can be determined, and according to fiducial frame and match frame, angle deviation can be determined. Thus, the ROI area can keep up with the changing of image angle and pixel.

The position correction is shown below.

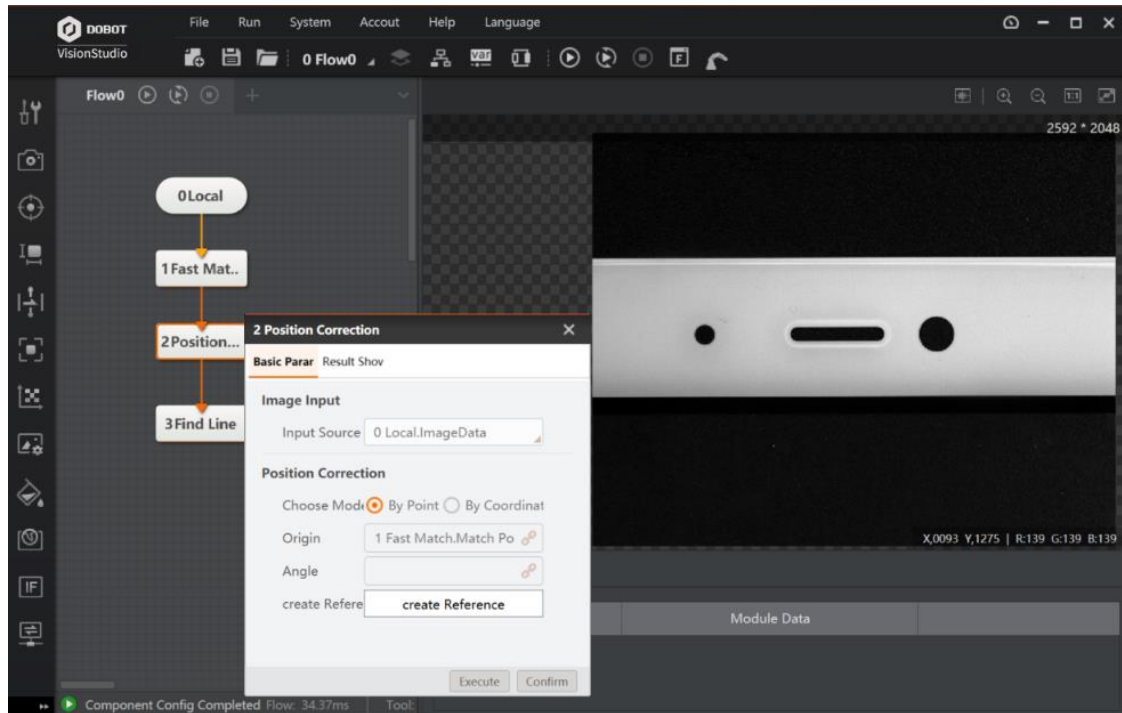


Figure 6.21 Position correction

There are two ways to correct position, and they are position correction by point and position coordinate by coordinate. By point, the position of the point has been determined. By coordinate, X and Y are used to determine the position of the point. It should be noted that whether it is a point or a coordinate, its position information is transmitted from the previous module, and its role is to determine the deviation of the pixel and the angle.

Here is a set of comparison images before and after position correction, as shown below. The images above are after position correction, and images below are before position correction.

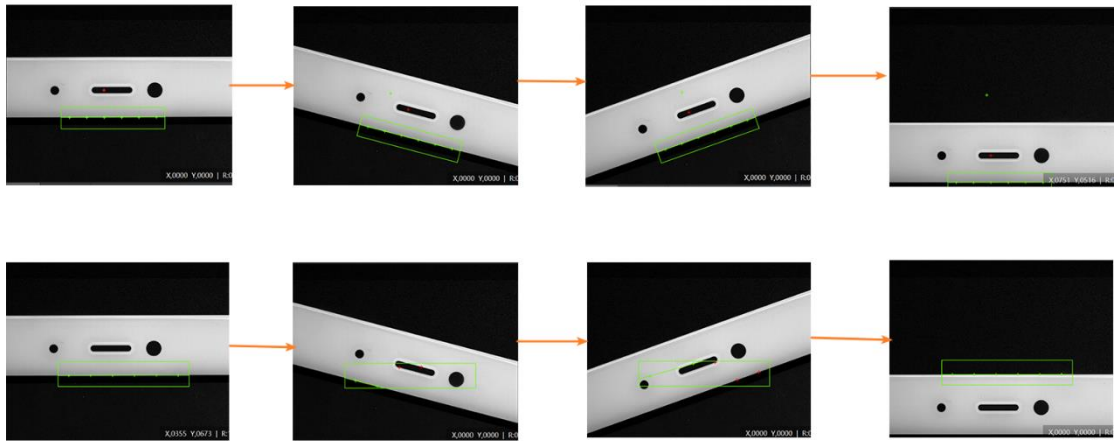


Figure 6.22 Comparison images

6.1.9 Find Rect

The tool of the find rect is used to detect the rectangle in the target image. The basic parameters and result display have been explained in the Section 3.3.2, and running parameters have explained in other sections. Here we take a solution demonstration as an example, as show below.

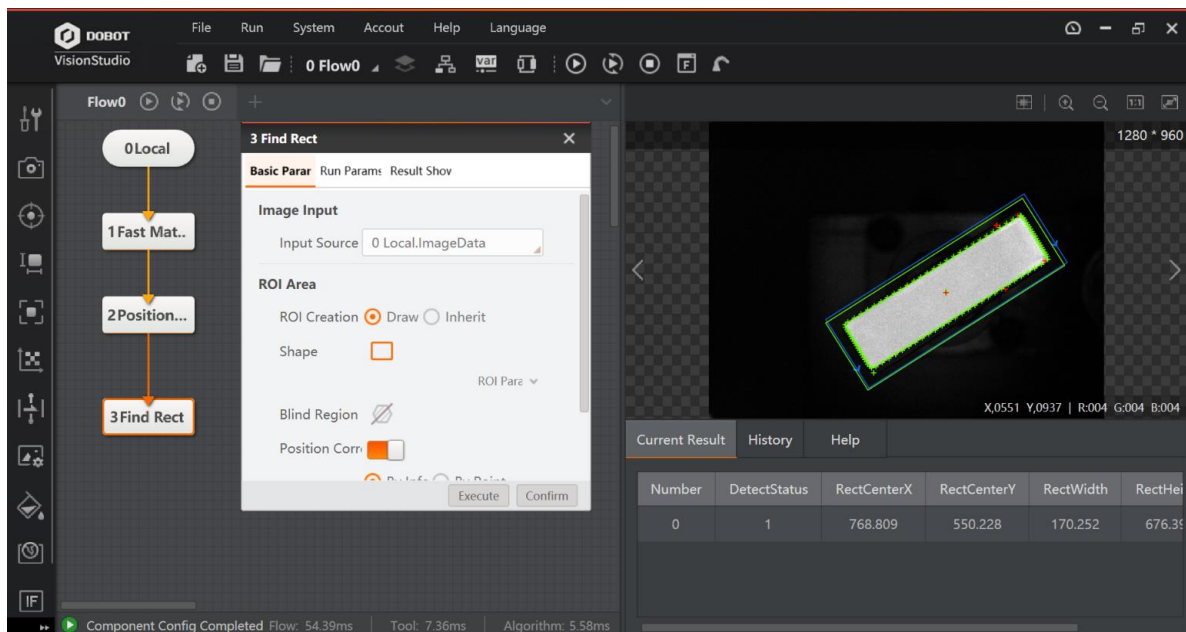


Figure 6.23 Find rect

6.1.10 Find Peak

The tool of finding peak is used to detect the peak of an object in specific area, as shown below.

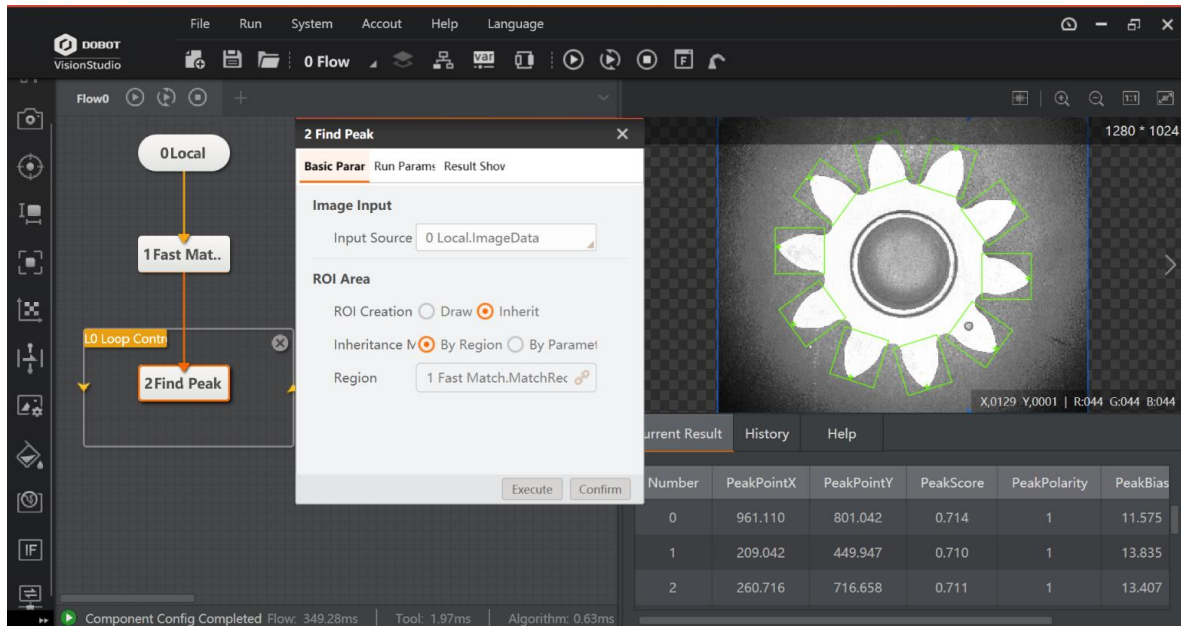


Figure 6.24 Find peak

- This software has integrated all the required drivers. As a result, you can install the software easily without installing driver separately.
- We recommend adding the software to the whitelist of the antivirus software, in case of being recognized as virus.

ROI Area: inheriting means that the previously defined area is inherited. Here we inherit the match frame of fast match as an example, and the tool detects ten ROIs and finds each peak via loop searching.

Scan Width: it is used to describe the area width of searching edge point ROI, and the min. value is 1. Within certain range, increasing this value can reduce the quantity of edge points.

Result Show: it is set as shown below.

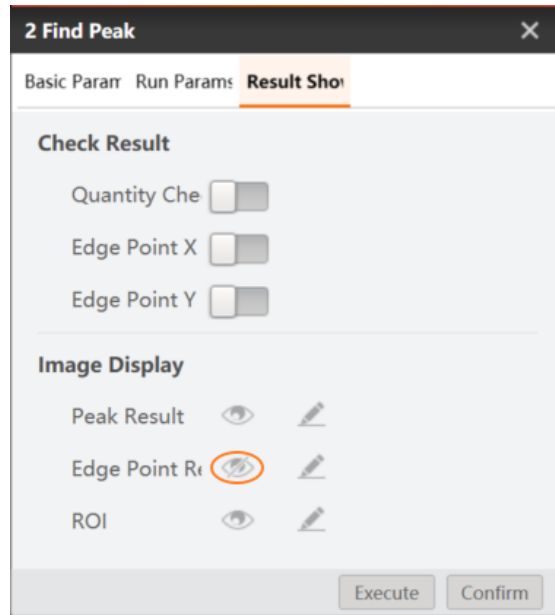


Figure 6.25 Result show

Note:

In result showing tab, edge point and peak are green by default. If you want to highlight peak, it is recommended to customize the color.

6.1.11 Caliper Corner

Caliper Corner

The tool of caliper corner is used to find the intersection of two edges, and the search direction and polarity can be set according to the required intersection. The basic parameters and result display have been explained in the Section 3.3.2. Here we take running parameters as an example, as show below.

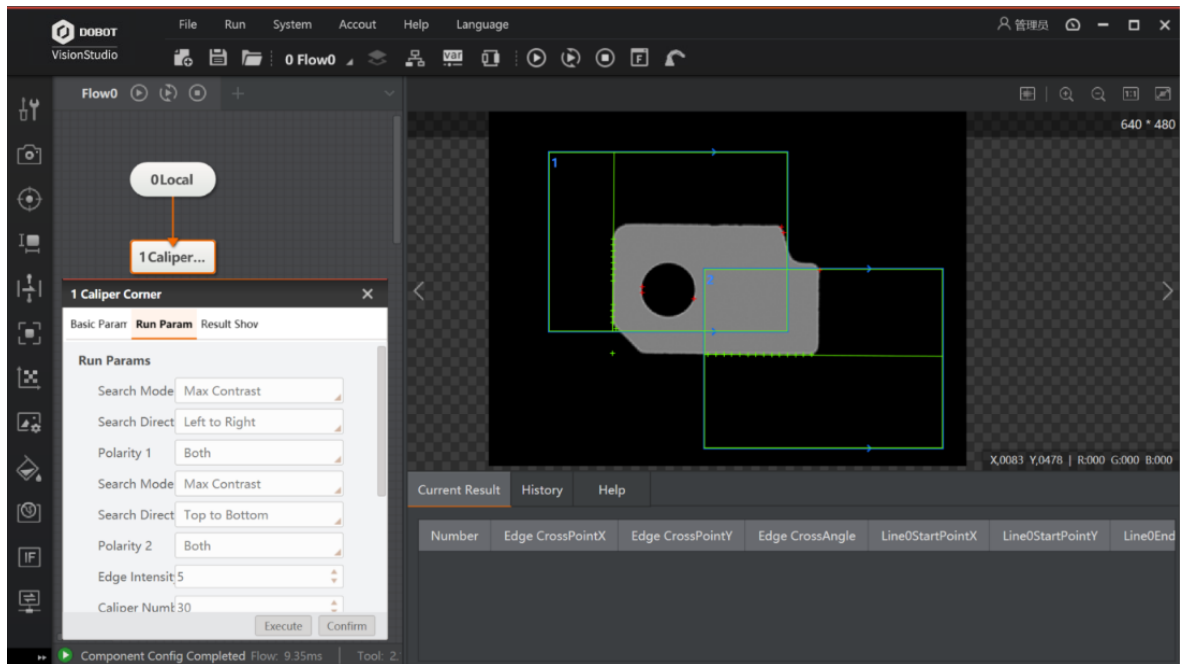


Figure 6.26 Caliper corner

- **ROI Area:** setting one or two ROIs is available. When setting one, it is to find the intersection of two lines in one ROI area. When setting two, it is to find the line in two ROI areas separately, and then find the intersection of these two lines.
- **Search Mode:** edge1 search mode is to find the line in ROI 1 area, and edge2 search mode is to find the line in ROI 2 area.

6.1.12 Find Parallel Lines

The tool of finding parallel lines is used to find lines that are approximately parallel within the tolerance angle range. The centerline of the two parallel lines is also displayed in the middle, as shown below. For running parameters, you can refer to Section 5.2.3.

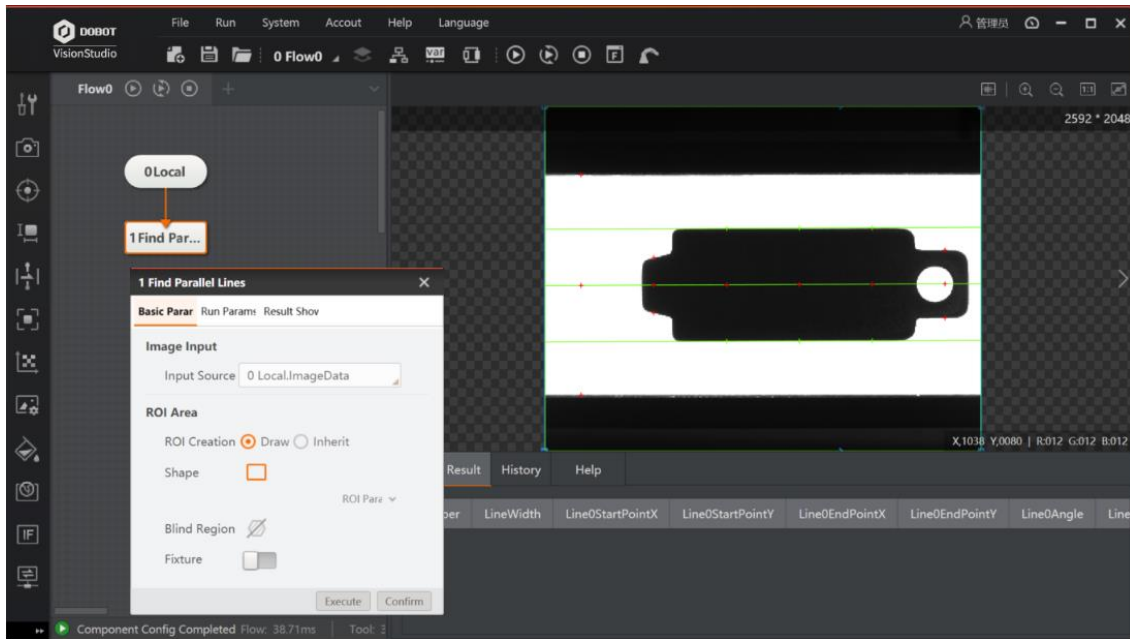


Figure 6.27 Find parallel lines

- **Angle Tolerance:** if the angle of target lines is smaller than this parameter value, the target lines will be determined as parallel lines. Otherwise, the target lines will be determined as non-parallel lines.
- **Filter Kernel Size:** it is used to enhance the edge and suppress noise, and the minimum value is 1. When the edge is blurred or there is noise interference, you can increase its value to make the detection result more stable. If the distance between the edge and the edge is too close, it will affect the accuracy of the edge location or even lose edge. This value needs to be set based on the actual situation.

6.2 Measure

6.2.1 L2C Measure

The tool of L2C measurement returns the vertical distance and the intersection point coordinate of the line and the circle in the image. This tool finds the line and the circle in the image of the object to be measured first, as shown below.

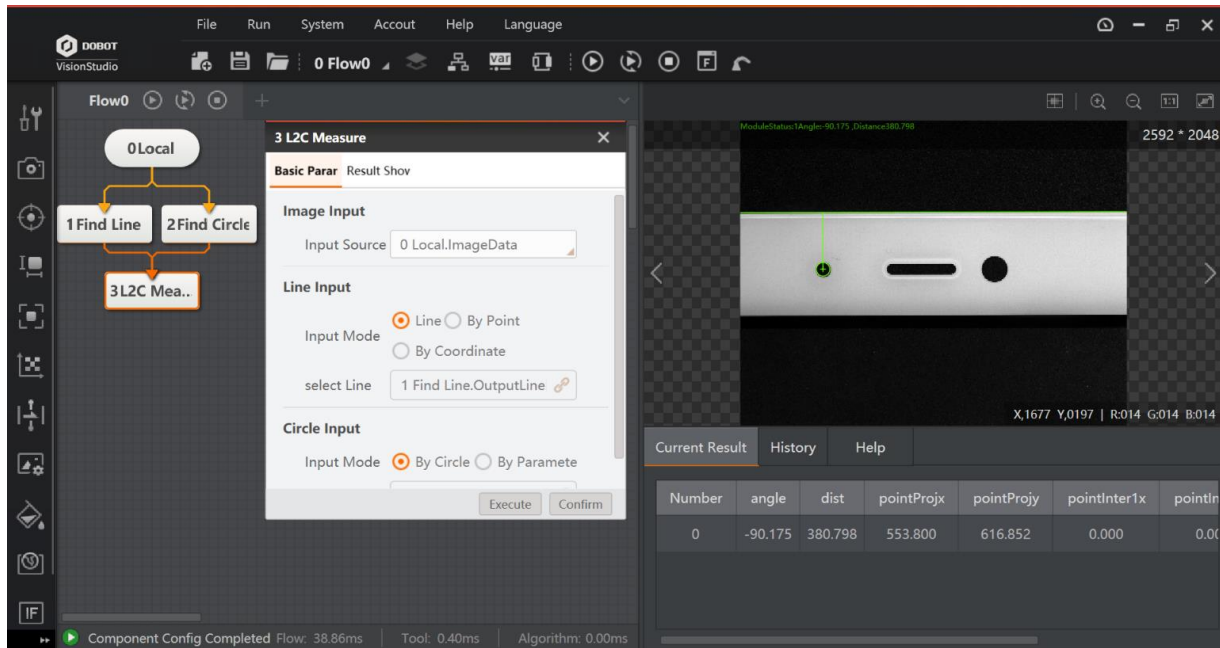


Figure 6.28 L2C measure

The specific steps are described as follows:

Steps:

1. Solution Creation: select the image source in the operation area, drag the algorithm module of finding line, finding circle and line2C measurement into the operation area, use the operation line to connect the above-mentioned modules successively, and click Run Once to complete the solution creation, as shown above.
2. After finding the target circle and line respectively, and set line input and circle input. There are several ways to input line and circle, as shown below.
 - By line or circle: select the result of finding line and circle for input source.
 - By point: you can customize or link the start point, end point and angle of the line.
 - By coordinate: you can customize or link the X/Y coordinate of the line's start point, and end point.
 - By parameter: you can customize or link the coordinate of circle center and radius length.
3. Inputting result: the vertical distance between the line and circle, intersection point coordinate of the line and the circle, and projection coordinate of circle center.

6.2.2 L2L Measure

In general, two lines are not absolutely parallel, and thus line to line measurement distance is calculated by the average distance from four end points of the line to another line. Line to line measurement is divided into distance and absolute distance, and the positive and negative of the

distance can represent the relative position relation between the two lines. The first line can only be left and right or top above and below to the second line. The left or above is positive, and right or below is negative, as shown below.

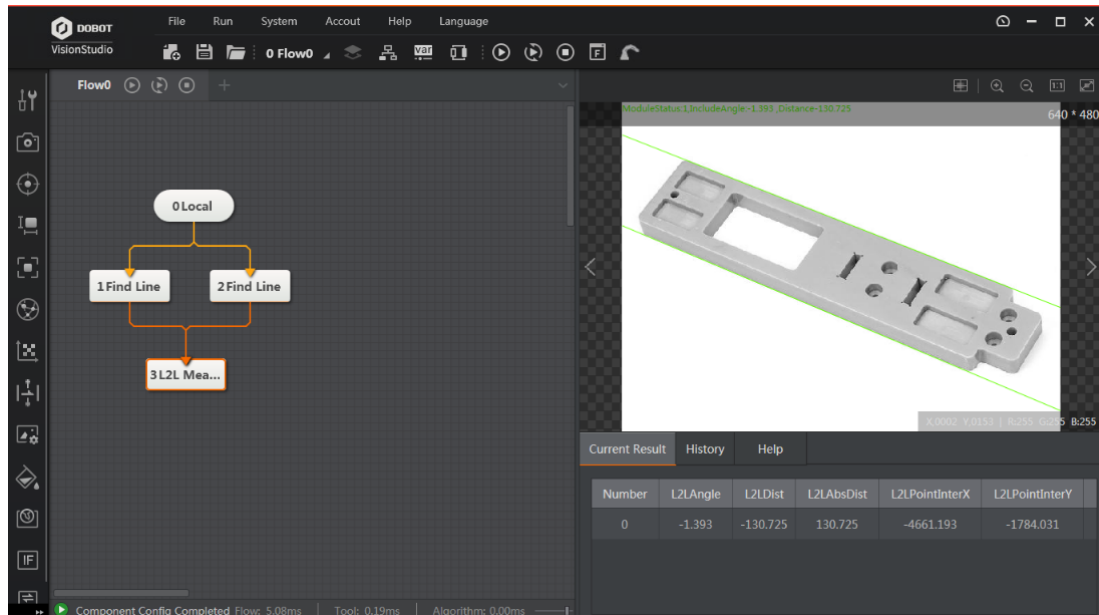


Figure 6.29 L2L measure

- **By Line:** inputting source is the result of finding line.
- **By Point:** you can customize or link the start point, end point and angle of the line.
- **By Coordinate:** you can customize or link the X/Y coordinate of the line's start point, and end point.
- **By Angle:** the angle difference between two lines.
- **Distance:** the average distance from four end points of the line to another line.
- **Absolute Distance:** the absolute value of distance.
- **Intersection Point X:** the intersection point X coordinate of extension lines of two lines.
- **Intersection Point Y:** the intersection point Y coordinate of extension lines of two lines.

6.2.3 C2C, P2C, P2L And P2P Measure

For C2C, P2C, P2L and P2P measurement, please refer to Section **L2C Measure**. According to different demands, you can set these tools accordingly.

6.2.4 Circle Fit and Line Fit

On the basis of three points or above, this tool fits circle, as shown below.

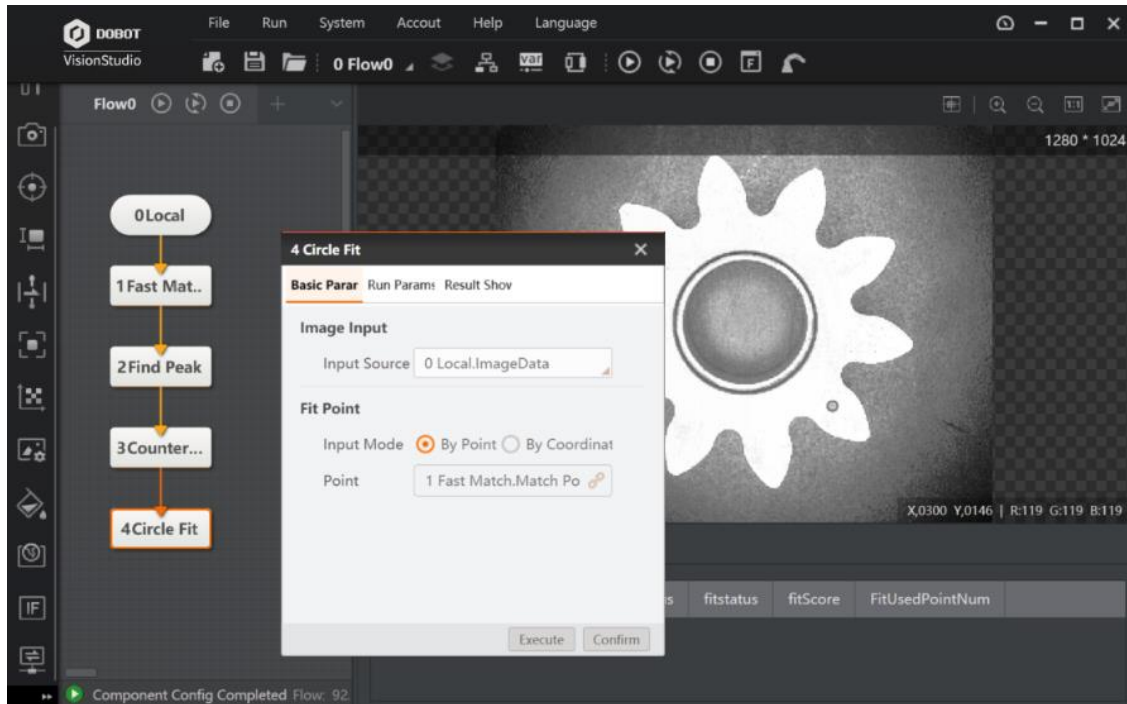


Figure 6.30 Circle fit and line fit

Basic parameters:

- **Image Input:** select collected images.
- **Fit point:** select collected point in the flow as fitting source.

Running parameters:

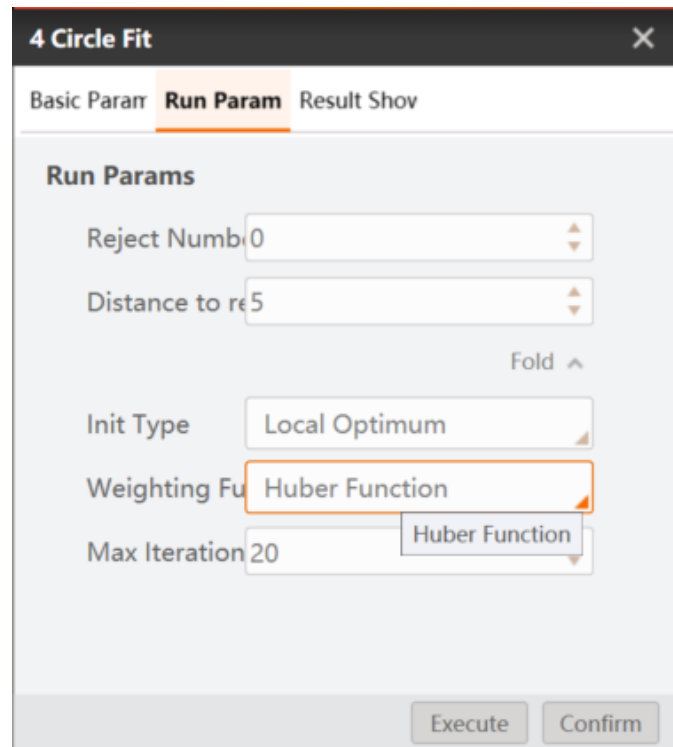


Figure 6.31 Circle fit

- **Number of Points to Ignore:** it means that the number of minimum points that have large error to be excluded from fitting. In general, if there is a great number of points excluded from fitting, its value should be set higher. For better results, it is recommended to use it in combination with the parameter of distance to remove.
- **Distance to Remove:** it means that the maximum pixel distance from the outlier to the fit circle. The smaller the value, the more points are excluded.
- **Initial Type:** it includes global optimum and local optimum.
- **Weighting Function:** it has three types, including least squares, huber function and tukey function. These three fitting modes are different in way of calculating the weight only. As the number of outliers and the distance from the group increases, it is recommended to use least squares, huber, and tukey successively.
- **Max Iteration Times:** The max execution times for fitting algorithm.

Line fitting needs two fitting points, and its principle is similar to circle fitting. For specific parameters, please refer to circle fitting mentioned above. Here we take a solution demonstration as an example, as show below.

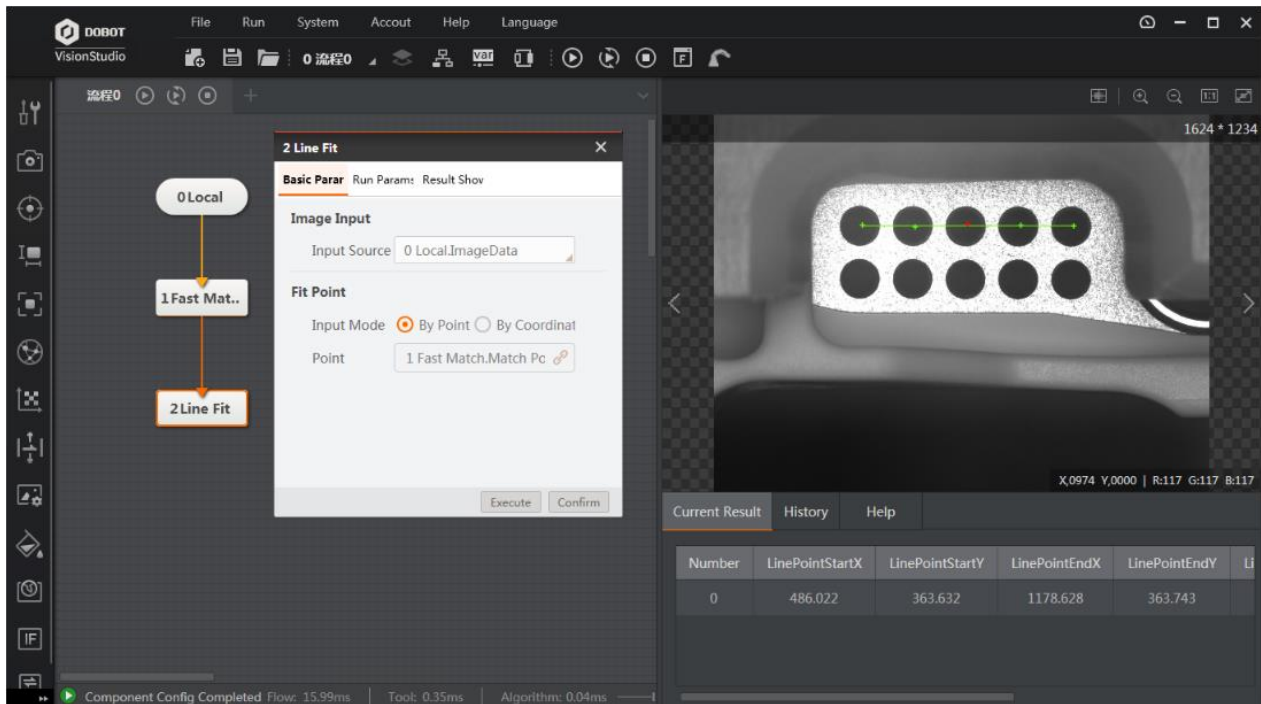


Figure 6.32 Max iteration times

6.2.5 Intensity Measure

The tool of intensity measurement is used to measure grayscale average value and standard deviation of all pixels within the ROI of the measured image. Select the image source in the operating area first, drag the intensity measurement algorithm module into the operating area, and use the operating line to connect the module in turn. You can use ROI tool to select area and narrow search range, and click Run Once to complete, as shown below.

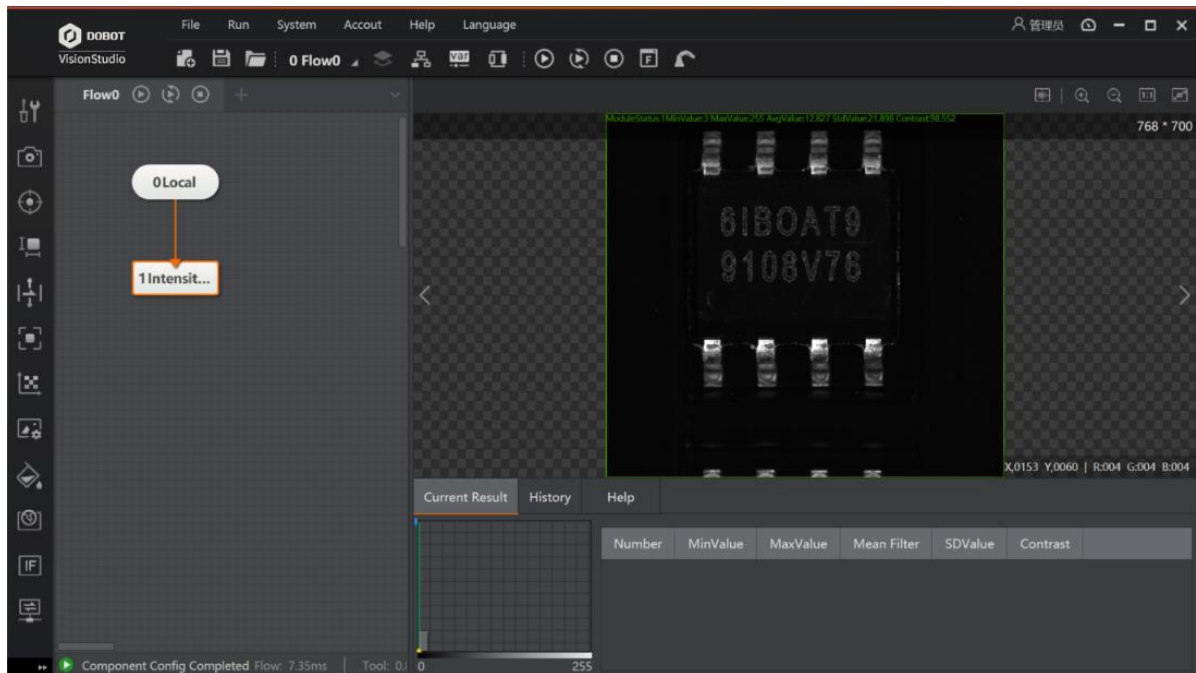


Figure 6.33 Intensity measure

6.2.6 Pixel Count

The tool of pixel count is used to count pixel within certain grayscale value in specific area, as shown below.

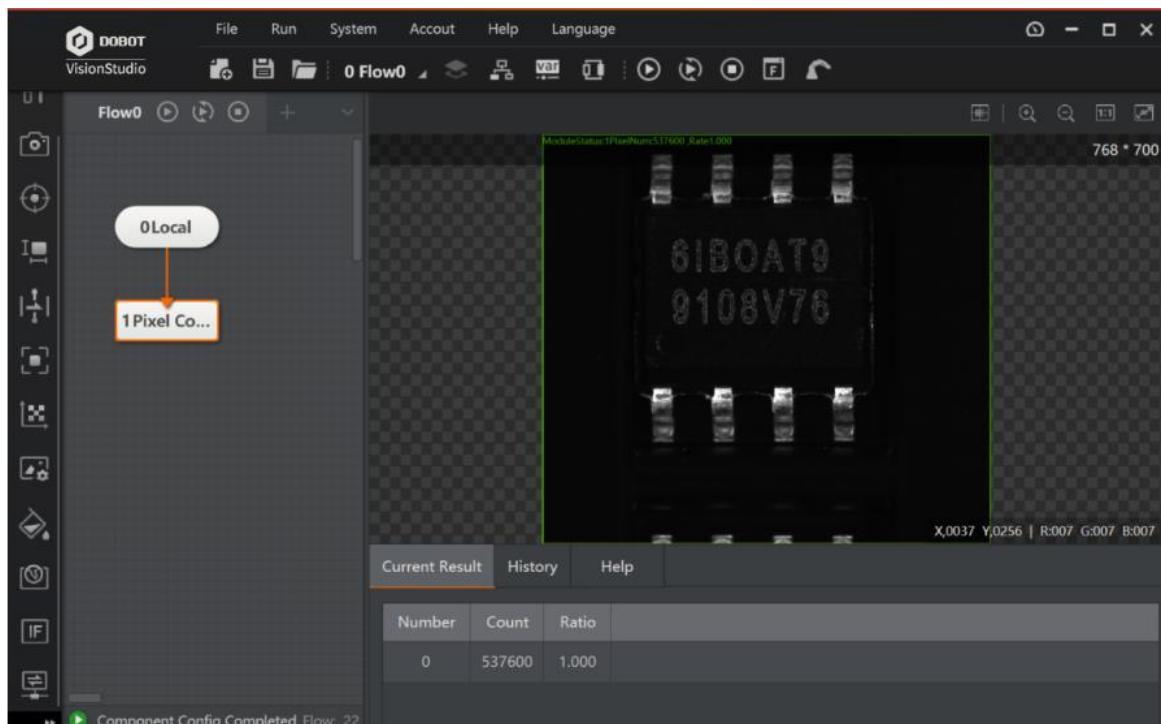


Figure 6.34 pixel count

- **Low Threshold:** the grayscale value in the area that needs count should be larger than low threshold value.
- **High Threshold:** the grayscale value in the area that needs count should be smaller than high threshold value.
- **Rate:** pixel point rate in this area.

Note:

If low threshold is larger than high threshold, the value of pixel should be between [0, high threshold] and [low threshold, 255]. If low threshold is smaller than high threshold, the value of pixel should be between [low threshold, high threshold].

6.2.7 Histogram Tool

The tool of histogram is used to count pixel quantity, average value, min. value, max. value, peak value, standard deviation and contrast of gray. It also generates a gray histogram, which clearly shows the distribution of pixel points under each gray value, as shown below.

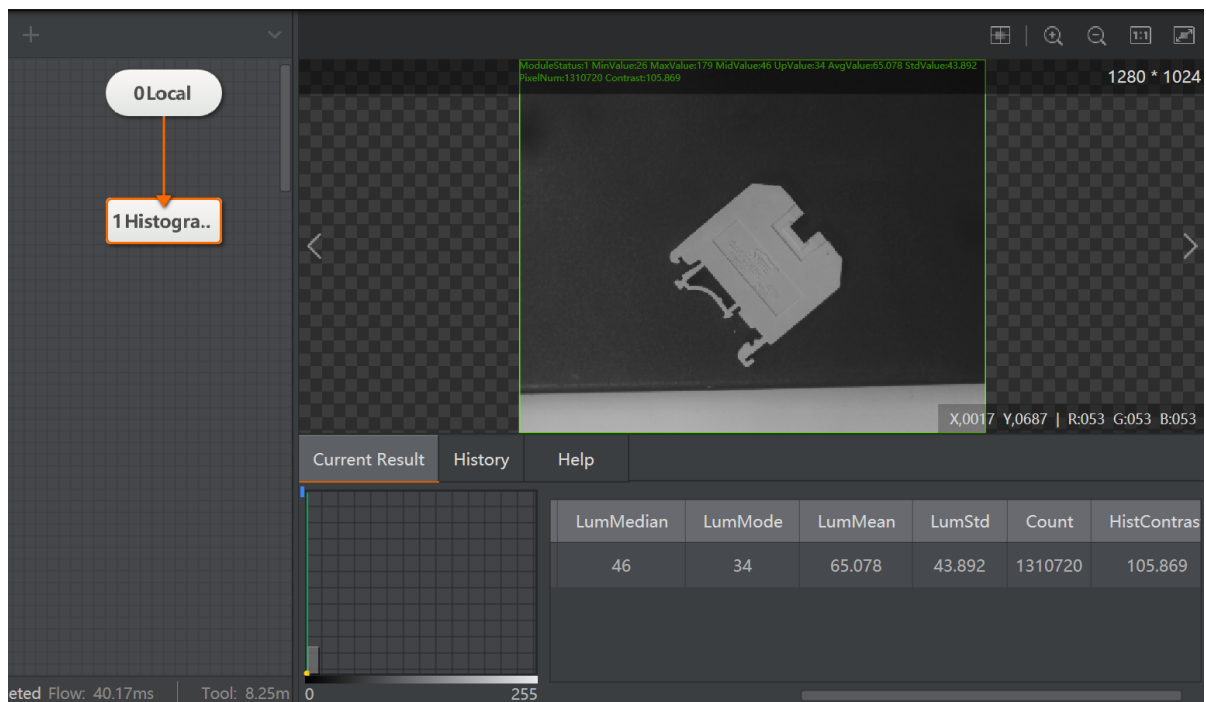


Figure 6.35 Histogram tool

6.2.8 Geometry Creation

The tool of geometry creation is used to freely create auxiliary points, line, and circle. When it is difficult to locate some points, lines or circles, you can create them according to the graphic features, as shown below.

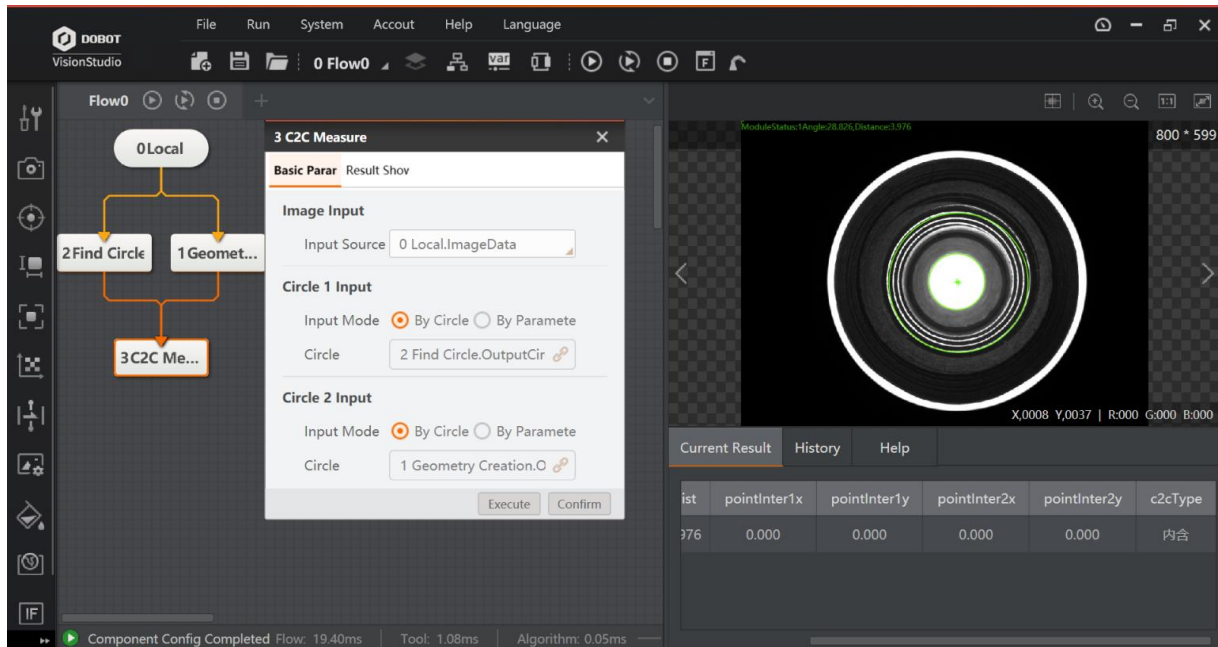


Figure 6.36 Geometry creation

The tool of geometry creation can create point, line and circle, and specific method is shown below.

- **Create Pont:** select point as type, click the image and then there will be a point in the image. Drag the point or modify the X/Y coordinate to change the location.
- **Create Line:** select line as type, and drag the mouse on the image to create a line. Drag the end point of the line or modify the X/Y coordinate to change the location.
- **Create Circle:** select circle as type, and drag the mouse on the image to create a circle. Drag the end point of the circle or modify the X/Y coordinate of circle center to change the location.

6.3 Rognition

The tool of recognition includes 2D bar code (2D BcR) recognition, bar code (BcR) recognition, OcR recognition and VeriCode recognition, as shown below.

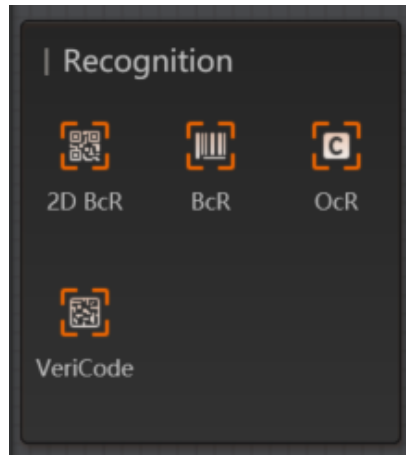


Figure 6.37 Recognition

6.3.1 2D Code

The tool of 2D code recognition is used to identify the 2D code in the target image, and outputs 2D code information in the form of character. It can efficiently and accurately identify multiple 2D code at a time. Currently, only QR code and DataMatrix code are supported, as shown below.

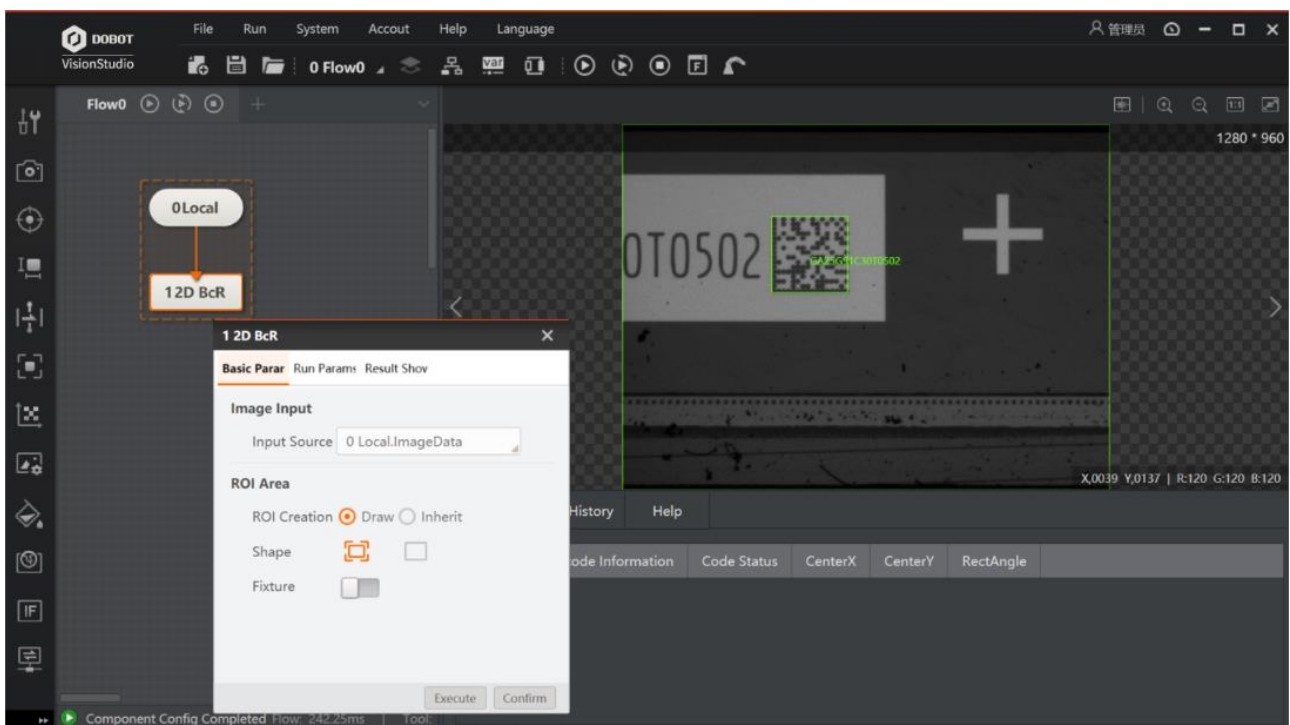


Figure 6.38 2D Code

- **QR Code and DM Code:** after enabling, the software can recognize QR code and DataMatrix code in the image. When you do not sure about the code type, it is recommended to enable both of them.
- **2D Code Number:** the max quantity of 2D codes that are expected to be searched and output. If the quantity of actually found is smaller than this parameter, the actual quantity of 2D codes is output. Sometimes the quantity of 2D codes in the scene is variable. To identify all the 2D codes that appear, you should set this parameter as the max. quantity of 2D codes in the scene. In some applications, the background texture is complicated, and the current parameters can be appropriately larger than the quantity of 2D codes to be recognized, which will reduce efficiency.
- **Polarity:** it has three types, including dark on white, white on dark and any. you can select them according to your demands.
- **Edge Type:** it has three types, including continuous, discrete and compatible type, as shown below. The left one means continuous type, right one means discrete type, and compatible type can be compatible with continuous and discrete type.



Figure 6.39 Edge type

- **Subsampling Ratio:** it is the image subsampling coefficient. The larger the value, and the higher the efficiency of the algorithm will be. But the recognition rate of the 2D code is reduced. The min. module of the 2D code supported by the algorithm library accounts for a max. of 16 pixels. Thus, when the smallest module in the scene accounts for more than 16 pixels, such as 2 times subsampling, you should set this parameter or change image acquisition method. When the min. module in the scene does not exceed 16, you should adjust this parameter according to the scene effect.
- **Code Width Range:** the pixel width that 2D codes occupy, and it includes the pixel width of the largest and smallest code.
- **Mirror Mode:** it means the X direction mirror of image, including mirrored and non-mirrored mode.
- Mirror mode enable switch, which refers to image X-direction mirroring, including "mirror" and "non-mirrored" modes. This parameter should be enabled when the image is collected from reflected mirror. Otherwise, this parameter should not be enabled.

- **QR Distortion:** this parameter needs to be enabled when the 2D code to be identified is printed on the bottle or bags of logistics industry.
- **Timeout-Period to Exit:** if the running time of the algorithm exceeds this parameter value, it will exit directly and unit is ms. When this parameter is set as 0, and the timeout exit time will be disabled.
- **Application Mode:** the normal mode is adopted in the normal scene. The expert mode is reserved for the 2D codes that are difficult to identify. When the application scene is simple, the code is clear, and area is large and clean, the fast speed mode can be adopted.
- **DM Code Type:** it has three types, including square, rectangle and compatibility mode.

6.3.2 1D Code

The tool of 1D code recognition is used to locate and identify the bar code in the specified area, and tolerate the target barcode to rotate at any angle which has a certain amount of angle tilt. It supports CODE39 code, CODE128 code, Codabar code, EAN code, alternating 25 code and CODE93 code. The specific operation is shown below.

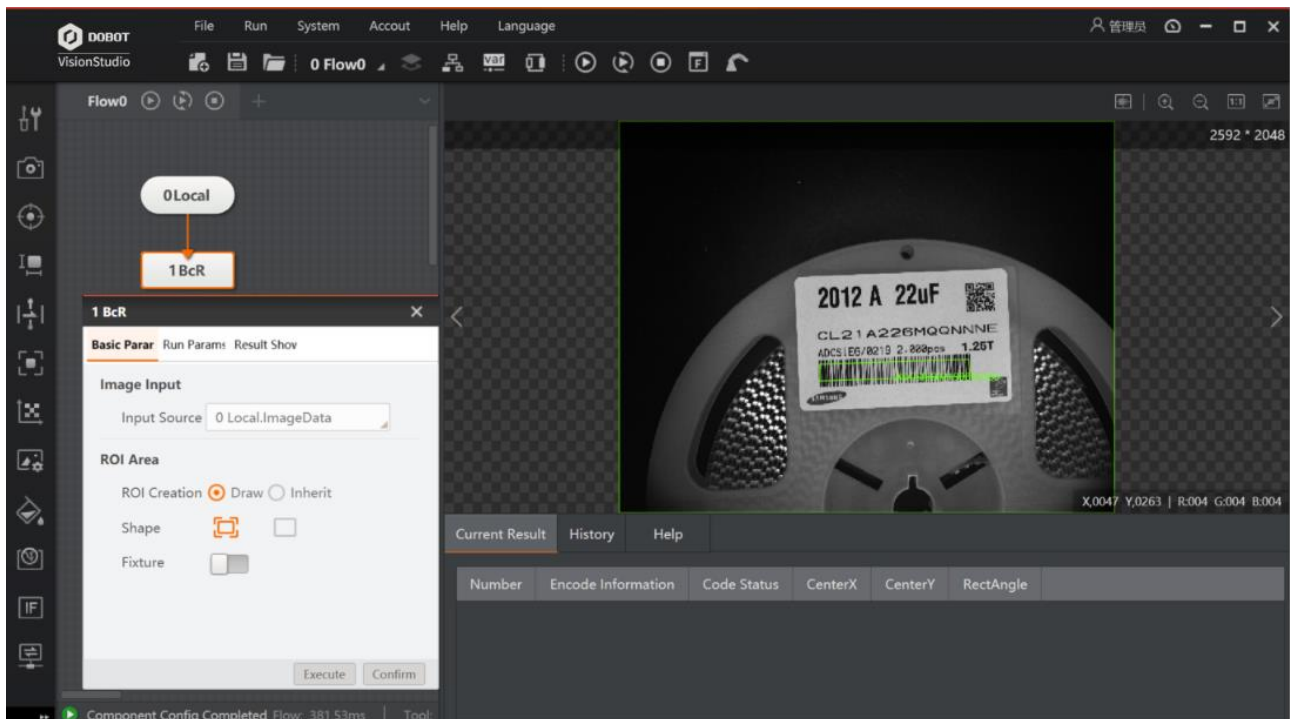


Figure 6.40 1D Code

- **Code Enabling:** it supports CODE39 code, CODE128 code, Codabar code, EAN code, alternating 25 code and CODE93 code.

- **Bar Code Number:** the max. quantity of bar codes that are expected to be found and output. If the quantity of actually found is smaller than this parameter, the actual quantity of bar codes is output.
- **Subsampling coefficient:** it is also called downsampling, which means that the number of sampling points is decreased. For an image of $N \times M$, if the subsampling coefficient is K , then one dot is taken every K points of each row and column in the original image to create an image. Therefore, the larger the subsampling coefficient, the more sparse the contour points will be, and the less detailed contour will be. It is recommended that the value should not be set too large.
- **Detection Window Size:** it is the size of bar code area location window size. The default value is 4, and when the blank space in the bar code is relatively large, you can set it larger, for example 8. But, you have to make sure that the bar code height is greater than 6 times of that of the window size, and the value range is from 4 to 65.
- **Quite Zone Width:** it refers to the width of the blank area on the left and right sides of the bar code. The default value is 30. If the blank area is sparse, you can set the value as 50.
- **Bar Code Height Range:** the minimum bar code height and the maximum bar code height that algorithm is able to recognize. The default value is from 30 to 200.
- **DfkSize:** the minimum bar code height and the maximum bar code height that algorithm is able to recognize. The default value is from 30 to 2400.
- **Timeout-Period to Exit:** if the running time of the algorithm exceeds this parameter value, it will exit directly and unit is ms. When this parameter is set as 0, the actual time that algorithm costs should prevail, and the unit is ms.
- **Distortion:** it is recommended to enable this parameter when the bar code has distortion.
- **White Gap:** it is recommended to enable this parameter when the bar code has bad printing quality.
- **Light Spot:** it is recommended to enable this parameter when the bar code has reflection under light.

6.3.3 OCR

The tool of OCR recognition is used to read the character text on the label. This tool requires character training. The specific steps are shown below.

Steps:

1. Drag the character recognition module in to the flow operating area and set relevant parameters after double-clicking.
 2. Set parameters before character training.
- **Character Polarity:** it has two types, including dark on light and light on dark.

- **Character Width Range:** it is used to set the min. width and max. width of character.
- **Width Type:** it has two types, including changeable type and aequilatus type. When the character width is the same, it is recommended to select aequilatus type. Otherwise, select changeable type.
- **Character Height Range:** it sets the min. height and max. Height of the character.
- **Binary Ratio:** it is the parameter of binaryzation threshold.
- **Area of Fragment:** the area range of single character.
- **Accept Threshold:** it refers to the min. score of a character that can be recognized.
- **Distance Threshold:** the distance from the Blob to the text baseline. If the actual value is greater than this parameter, it will be deleted.
- **Ignore Borders:** whether to delete the blob that is in touch with the ROI boundary.
- **Main Direction Range:** the search range of text tilt angle.
- **Tilt Range:** the max range that allows characters to be tilted.
- **Gap Merge:** whether to merge the min. gap between characters.
- **Min. Gap:** the min. gap between characters.
- **Max Width and Height Ratio:** the largest width and height ratio.
- **Classification Method:** it has three methods, including the nearest distance, the highest weight and the fastest rate.
- **Stroke Width Filter Enable:** Whether to enable stroke width filter or not.
- **Stroke Width Range:** the width range of a single stroke, and it can only take effect after the width filter is enabled.
- **Similarity Type:** it supports euclidean distance and cosine distance. The former one describes the linear distance of two feature vectors, and the latter describes the angle between the two feature vectors.

3. Double-click fronts training.

- Select the target character area.
- Click Extract Character and it will show characters that are divided by red box.
- Click Train Characters, input respective characters to character library, as shown below.

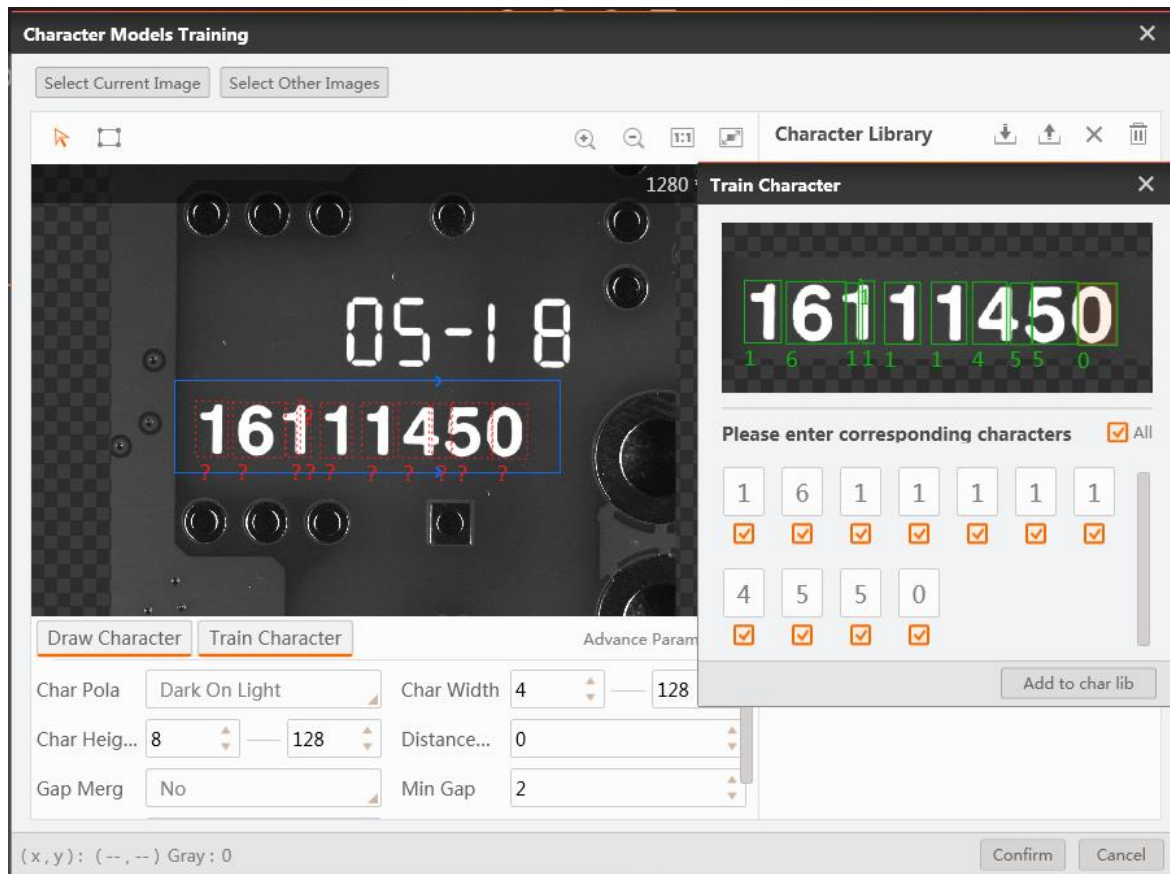


Figure 6.41 Train character

4. Click **Run Once** to complete, as shown below.

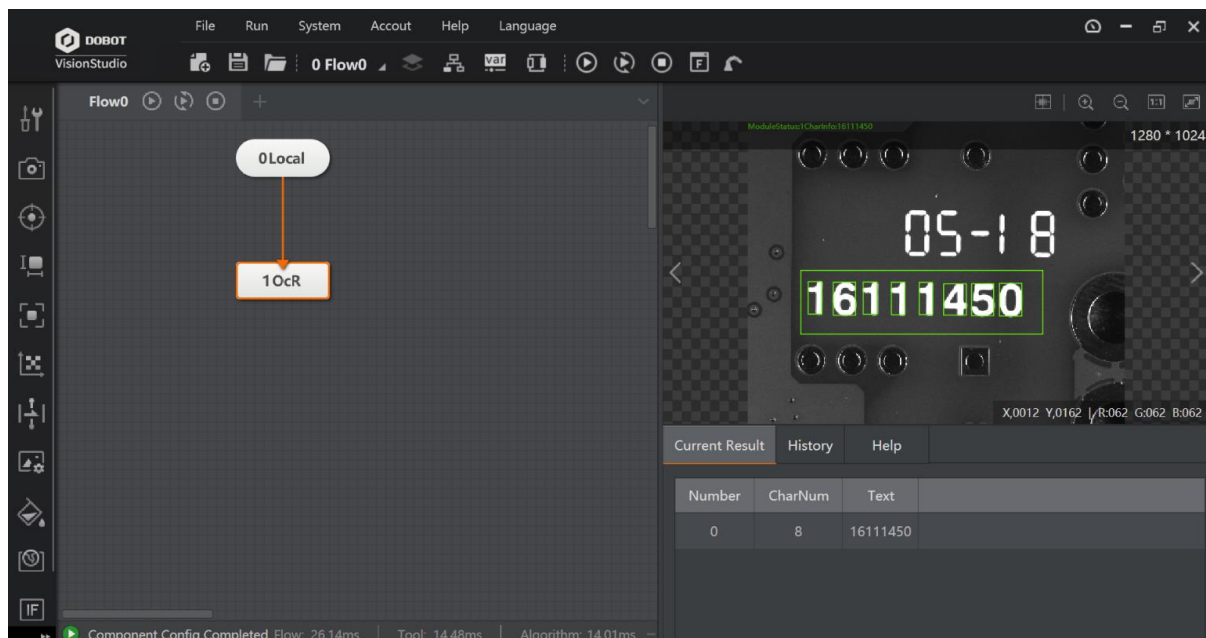


Figure 6.42 Run once

6.3.4 VeriCode

The tool of VeriCode recognition is used to detect and read VeriCode in specific area, and it allows the target barcode to rotate at any angle which has a certain amount of angle tilt. Here we take running parameters as an example, as shown below.

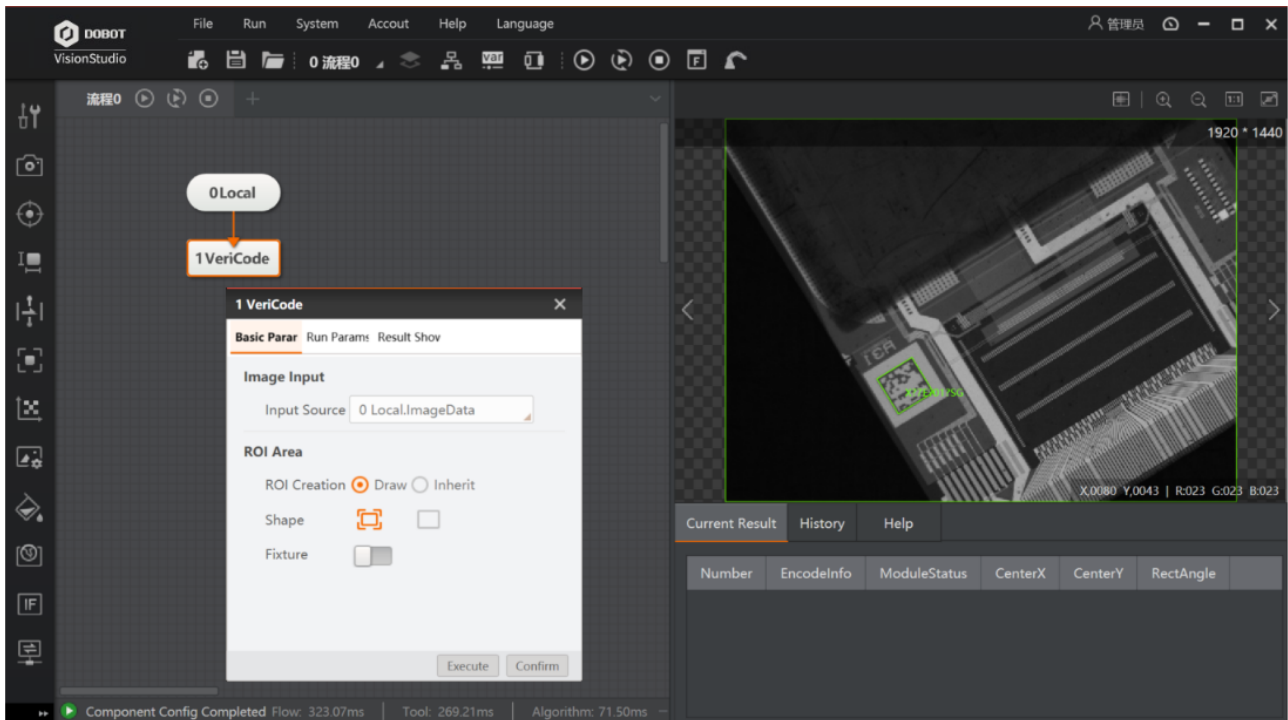


Figure 6.43 VeriCode

- **Symbol Type** : it has four types, including normal code, black code, white code and outline code.
 - White code means that only the white code over the black background in the specified area is detected.
 - Black code means that only the black code over the white background in the specified area is detected
 - Normal code means that detecting the white and black codes in the specified area.
 - Outline code means that only the outline code in the specified area is detected.
- **Max. Runtime**: When it is set as 0, it means that no timeout. When it is set as > 0, it means that the program will exit when the program takes longer than max. runtime.
- **Code Size Range**: it means that the number of black and white squares or dots in a row or column of a 2D code, including the edges on both sides. The code size of 2D code is fixed.
- **Cell Size Range**: it means that the range of pixels that the width or height of a single module in a VeriCode code has. The lower limit is 3, indicating that the single module of the min.

detectable code has a width and height of 3 pixels. The upper limit is 50, indicating that the single module of the max. detectable code has a width and height of 50 pixels.

- **Contrast Threshold:** the difference between the max and min. brightness of the corners of the Vericode code, and the default value is 15. When the brightness of the image is very dark, it is necessary to reduce this parameter value to increase the detection rate, but the operation time will increase.
- **Neighborhood Radius Range:** pixel data with a symmetrical corner shape in the 2D code. The optimal value of the lower limit of this parameter is set to $\frac{1}{3}$ of the module size, and the min. limit value is 2. The optimal value of the upper limit of this parameter is set to $\frac{1}{2}$ of the module. For most images, the upper and lower limits are set to 2~4 can meet the demand (this parameter is not open to outline code).
- **Outline Code Mode:** this parameter enables the Normal code of positioning thin edge. if the recognized edge width of the Normal code is smaller than other module sizes, the parameter can be enabled to improve the positioning success rate (this parameter is not open to edge code).
- **Auto:** it includes contrast, sampling width, cell bit, min. code size, max. code size, EDAC error correction level, and compression mode.

Note:

The tool of VeriCode recognition should be used together with VeriCode authorization code.

6.4 Deep Learning

Deep learning is a type of machine learning algorithm which is developed by traditional neural networks. Through deep learning model which is similar to brain neural network, computers, like human beings, is able to absorb, learn and understand complicated information in the real world, and complete difficult recognition tasks. Deep learning is used to locate character, recognize and detect defects, classify image, and detect target.

Deep learning is based on a certain data foundation, thus it is necessary to train a great number of data before performing deep learning. The data groups that participate in the training need to be labeled, and the data diversity needs to be ensured as much as possible. But, deep learning has lower requirement for resolution. Taking character location as an example, deep learning requires at least 150 images a sample.

Note:

- This function module has higher requirements for computer configuration. Deep learning classification, object detection, text detection, defect detection, character recognition modules require 2G memory or above, and 1060 graphics card or above. Defect detection and training tool of deep learning require 4G video memory or above, and 1080 graphics card or above. It is recommended to use Gigabyte GTX1080Ti, ASUS GTX1080Ti, and NVIDIA GTX1080Ti graphics cards.

- DL text detection, classification and object detection has two ways of G and C. The suffix of module name of GPU version is G, the suffix of module name of CPU version is C, and the CPU version does not depend on the graphics card.
- This function should be used together with special dongle for deep learning.

6.4.1 DL Training Tool

The training tool is mainly used to train samples. When using DL training tool, it is necessary to complete the three steps of selecting training file and type, label data, and model training, as shown below.

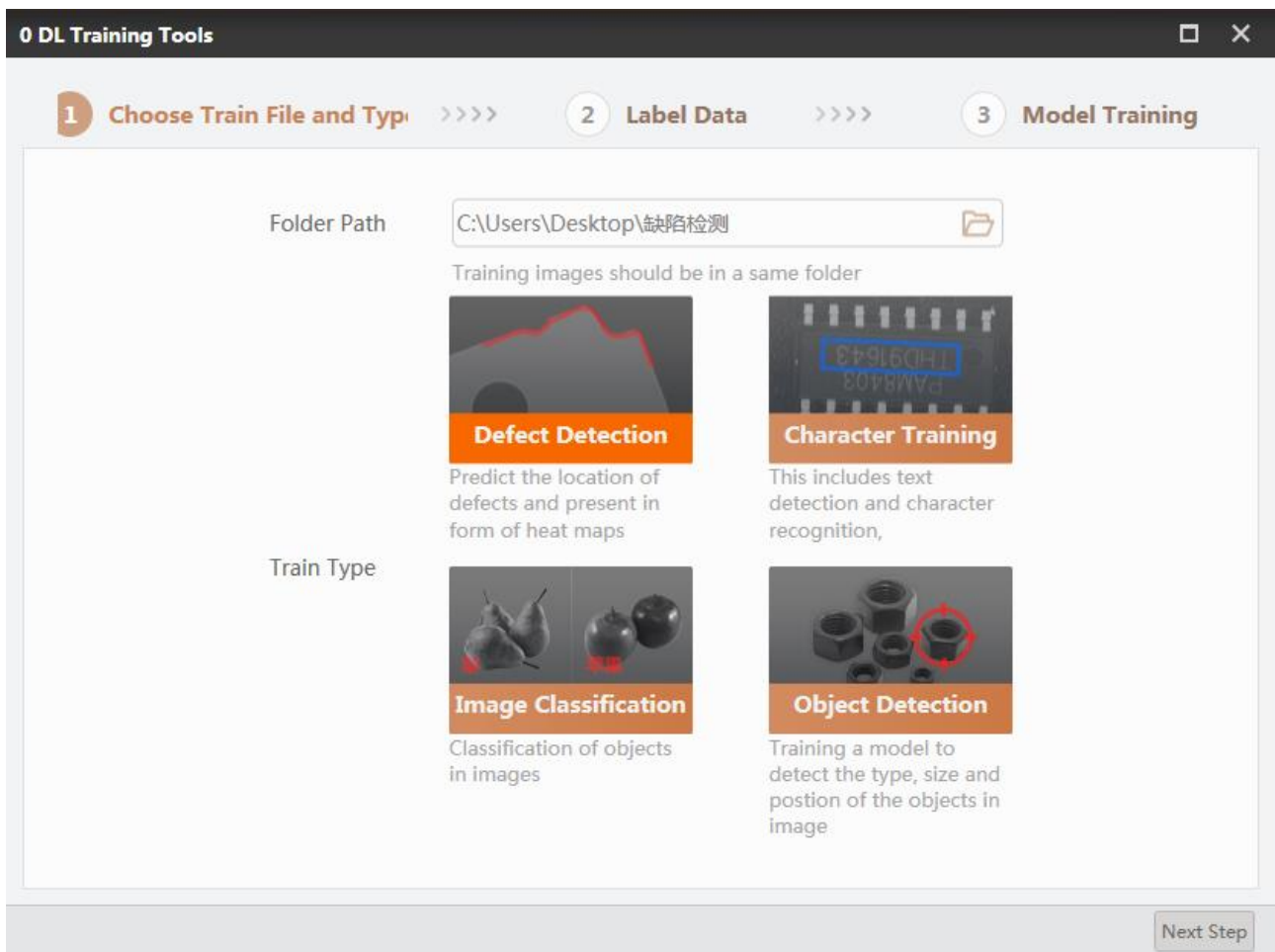


Figure 6.44 Train tools

Defect Detection

Defect detection is mainly used to train samples before DL defect detection. The richer the sample size, the more accurate the positioning of defects will be. Specific steps are shown below.

1. Select **Defect Detection** as training type. Load the image file that is to be trained in file folder address. The image file must contain all the sample images. Click **Next** to enter the label data.

2. Enter the label data interface, as shown below.

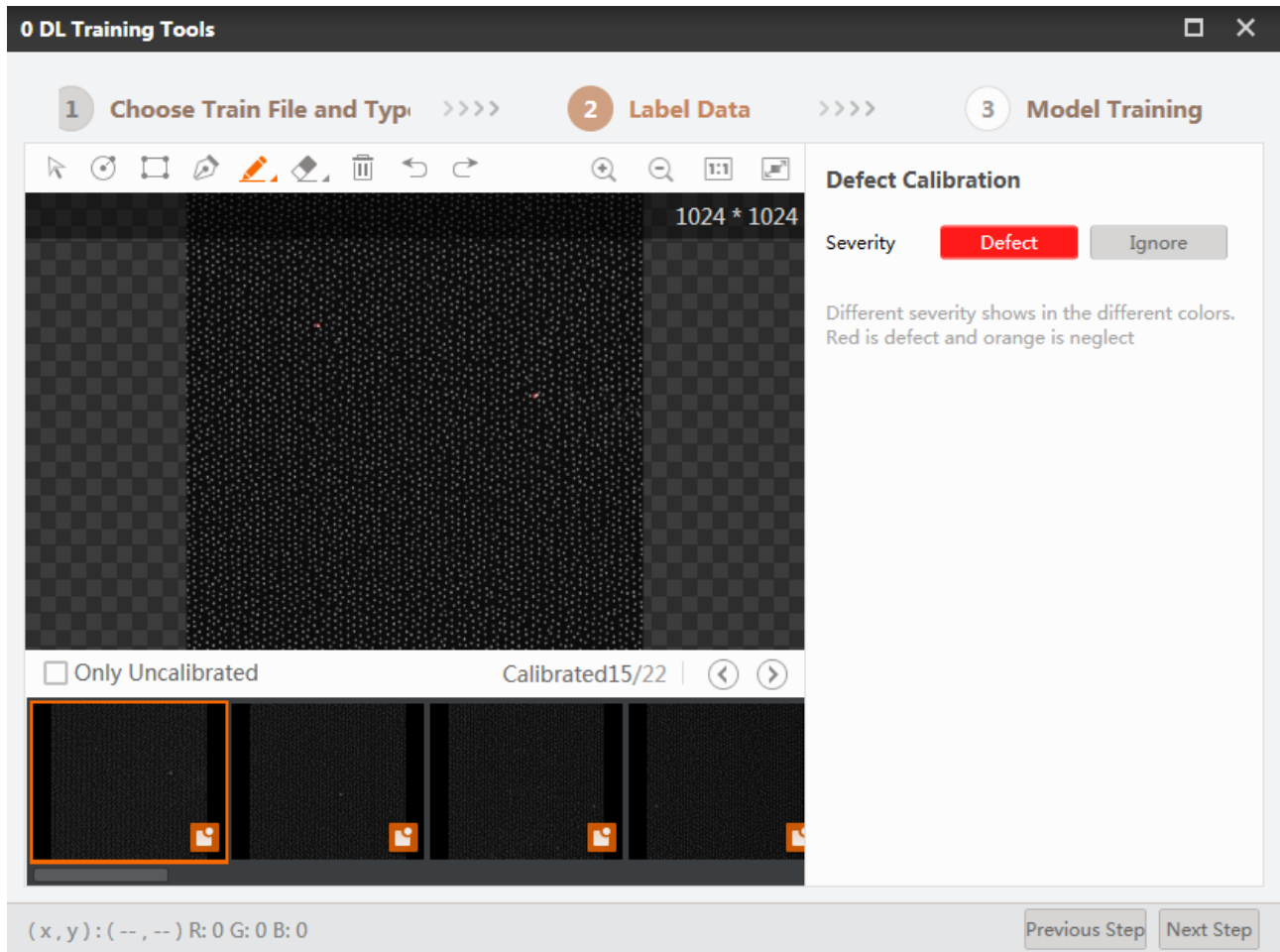


Figure 6.45 The label data interface

– Mark one or many areas that have defects in the defect image, and the marked area will turn into orange red. The area can be defined as defect or ignore in severity. The defect refers to define the selected area as the defect area, and ignore refers to some features that are very close to the defect and are easy to misjudge, and need to be ignored.

– After a calibration is completed, switch to the next one. If the calibration quantity is less than 11 pieces, the software will remind you that the calibration quantity is not sufficient. After all the samples have been calibrated, click Next to enter the model training interface.

In the shortcut toolbar above the interface, they are, from left to right, the moving image, calibrating circle defect area, calibrating rectangle defect area, polygon defect area, self-defined defect area, deleting, undo and back.

3. Enter model training interface and set related parameters, as shown below.

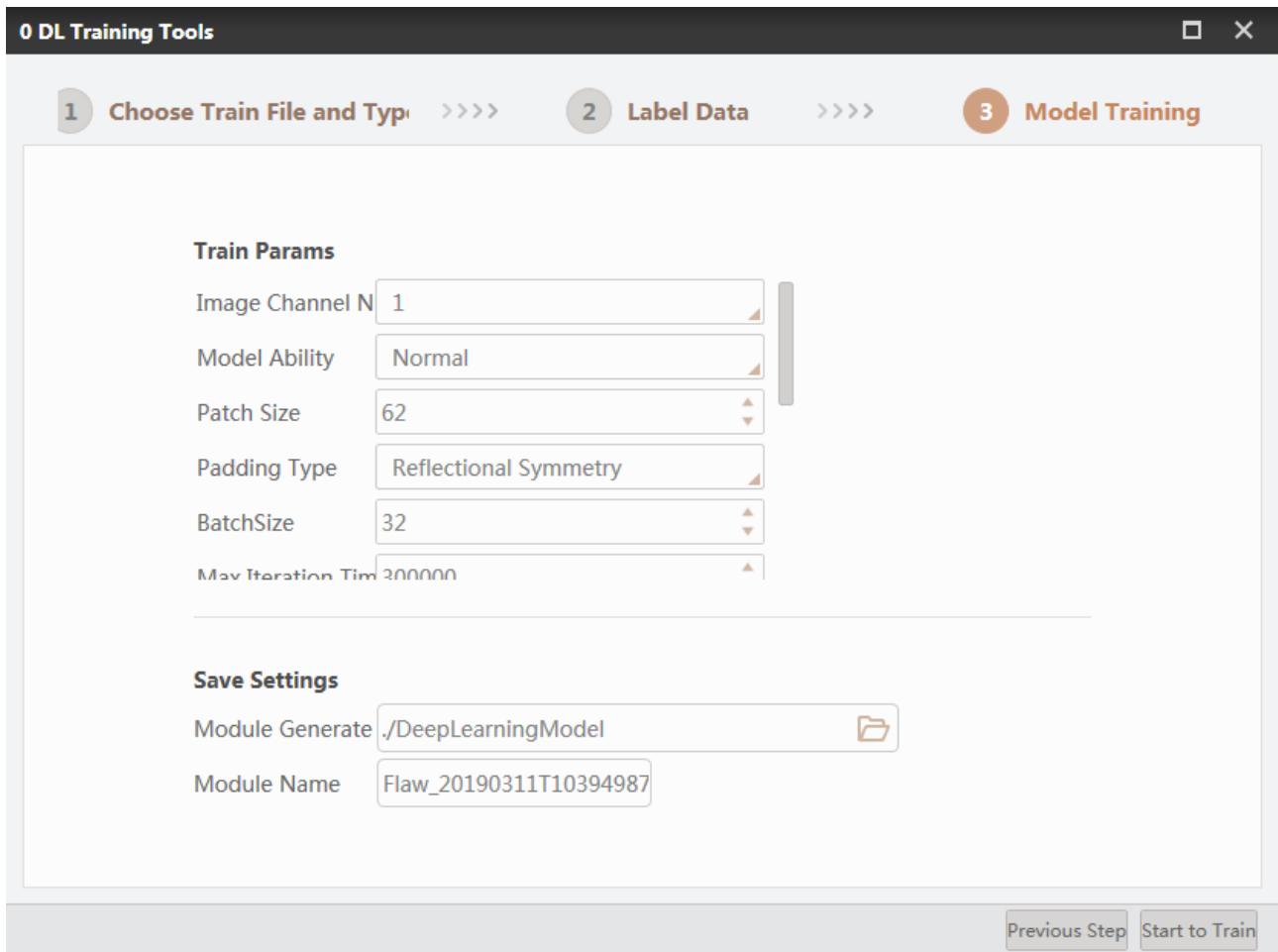


Figure 6.46 The model training interface

You can set training parameters as follows:

- **Image Channel No.:** 1 stands for grayscale and 3 stands for color image.
- **Model Ability:** the fineness level of detecting defects, and it has three models, including Small, Big, and Normal. The small model can cope with single, low-complexity scenes, and its training and detection operations are efficient. The big model can cope with very complex scenes. The normal model is between small and big. By default, the model ability is normal mode.
- **Patch Size:** it is the size of the patch.
- **Padding Type:** the filling method of image edge.
- **Batch Size:** the edge fill ratio.
- **Max. Iteration:** the max number of training images. The larger the value, the longer time it takes.
- **Learning Rate:** this parameter determines the speed rate of training. The larger the value, the faster the training loss will fall, but there may be a large fluctuation of the training loss, and

even the training loss becomes infinity and the training fails. The smaller the value, the slower the training loss will fall. If the value is too small, there may be little drop in training loss.

You can set angle range as follows:

- **Angle Range:** it determines the angle range of the image.
- **Scale Ratio Range:** it is the scale ratio range of the image.
- **Brightness:** it is the value of the image brightness.
- **Contrast:** it is the contrast size.
- **Flip Type:** Image rotation changes.
- **Model Saving Path:** you can customize the model saving path.
- **Model Name:** you can customize the name of model.

During the training process, attention should be paid to training loss during the training process, as shown below.

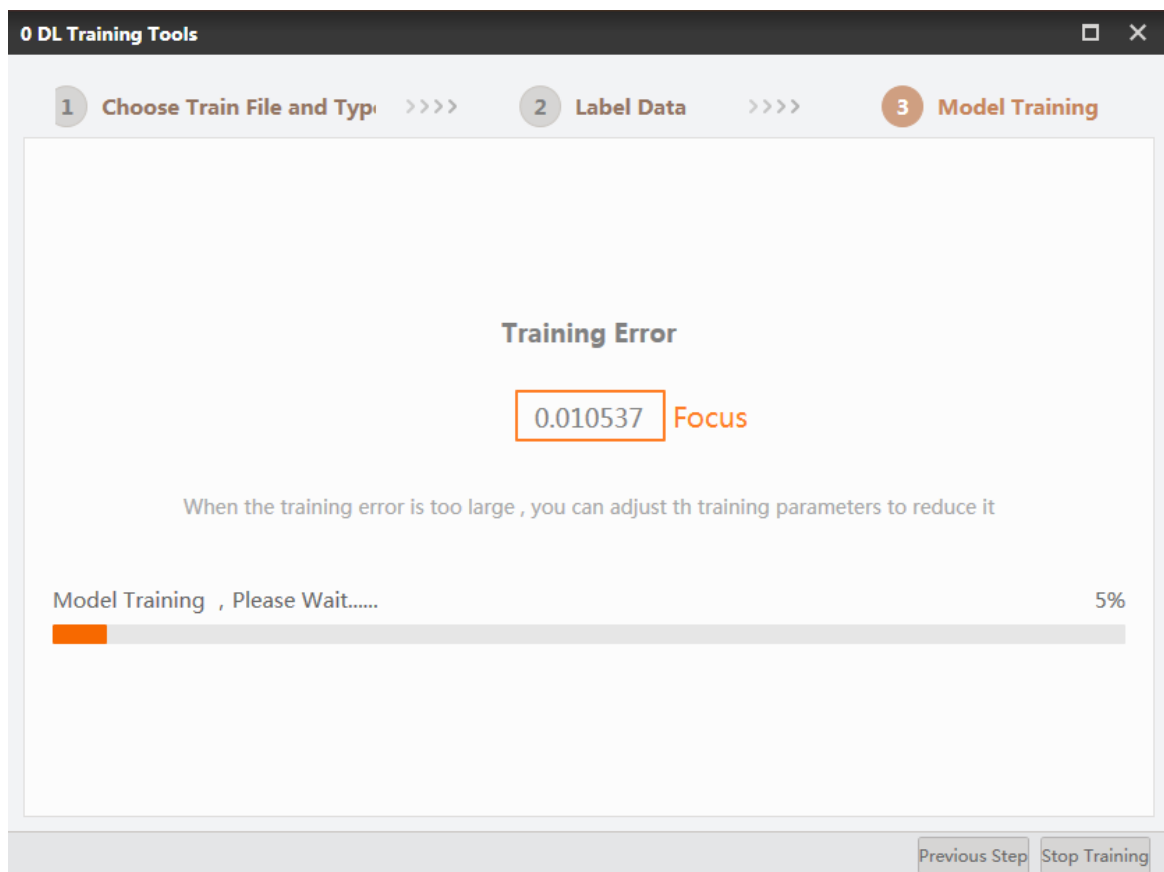


Figure 6.47 Training error

- Change too slowly: you should consider adjusting the learning rate.

- The error is still falling, but the training has ended: you should increase the max. number of iterations.
- The error has always been large: it is recommended to check if the dataset is qualified.
- The error can be large and small without any rule: if the amplitude is not large, it is normal. Otherwise, it is recommended to check the dataset or reduce the learning rate.

Character Training

Character training is mainly used for DL recognition or literacy teaching of samples before locating. The richer the sample size, the more accurate the recognition of characters will be.

Specific steps are as follows:

1. Select **Character Training** as training type. Load the image file that is to be trained in file folder address. The image file must contain all the sample images. Click **Next** to enter the label data.
2. Enter the label data interface, as shown below.

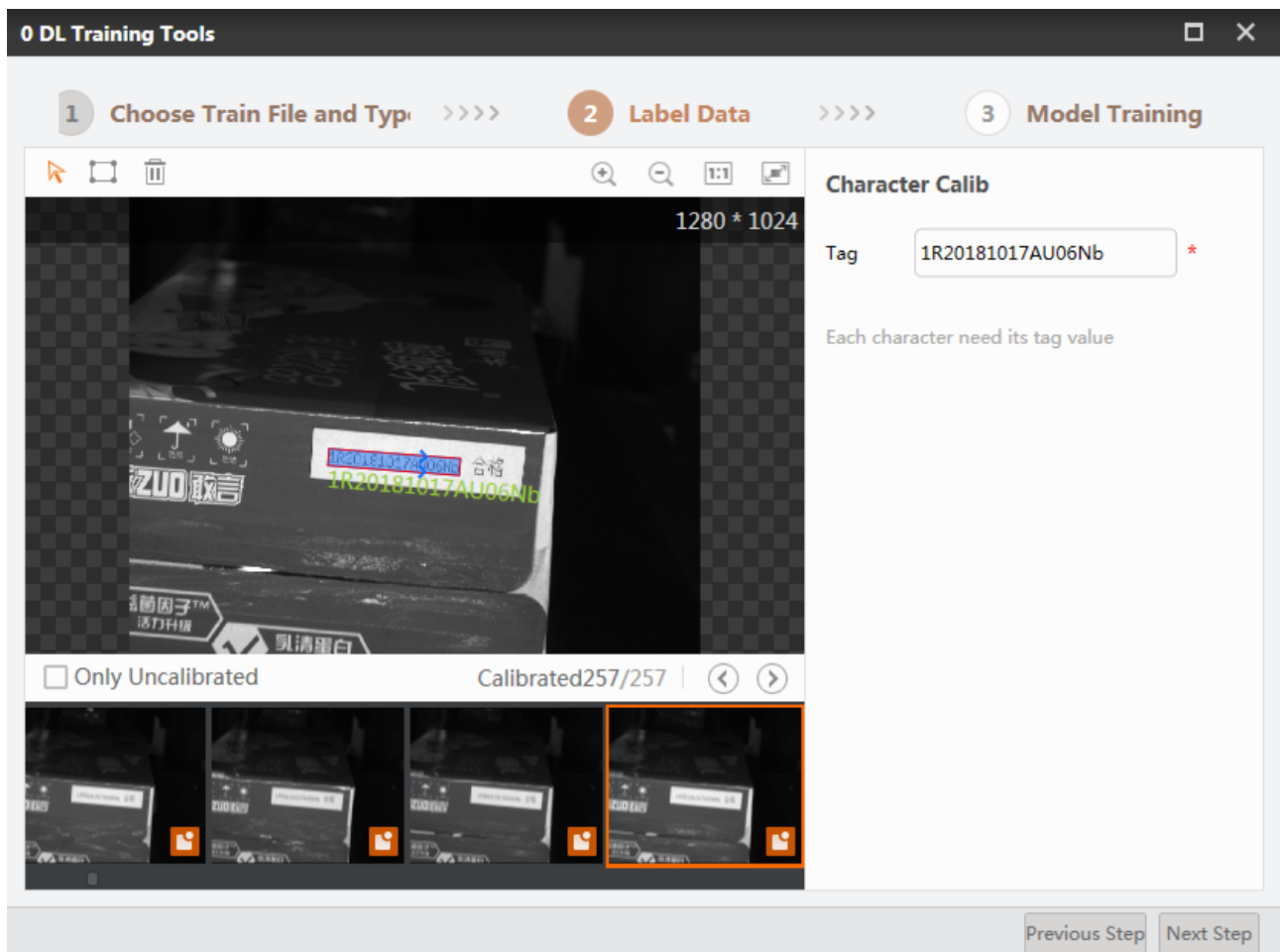


Figure 6.48 Enter the label data interface

Select the target character first, you can select multiple areas at one time. Then enter the true value in the tag, and press the enter key and calibrate the next one.

3. After entering the model training module, set the training parameters, and then start the model training. When selecting the character recognition, as shown below.

The screenshot shows the 'DL Training Tools' application window. It has three tabs: '1 Choose Train File and Type', '2 Label Data', and '3 Model Training'. The '3 Model Training' tab is active. Inside the tab, there are two sections: 'Train Params' and 'Save Settings'. The 'Train Params' section contains five input fields: 'Character Model' (set to 'OcR'), 'Model Type' (set to 'High Precision'), 'Iteration' (set to '3'), 'epoch' (set to '200'), and 'Accuracy Threshold' (set to '1.000'). The 'Save Settings' section contains two input fields: 'Module Generate' (set to './DeepLearningModel') and 'Module Name' (set to 'OCR_20190311T11255114'). At the bottom right of the window, there are two buttons: 'Previous Step' and 'Start to Train'.

Figure 6.49 Set the training parameters

- **Model Type:** it has three types, including high precision, Light-weight, and normal type.
- **Iteration Time:** the cycle times of dataset.
- **Epoch:** The times that the entire dataset participating in the training. It should be determined according to the complexity of the dataset.
- **Iteration:** The number of times an epoch is trained, and it is recommended to increase as the data set increases.
- **Accuracy Threshold:** when the character training accuracy reaches the threshold, the training will stop. In actual applications, it is recommended to set it as 0.98 and above.

When selecting text detection as character model, the interface is shown below.

0 DL Training Tools

1 Choose Train File and Type >>>> 2 Label Data >>>> 3 Model Training

Train Params

Character Model Text Detection

Max Iteration Time 2000

Base Learning Rate 1

Resize Mode Resize After Padded

Pad Value 0

Save Settings

Module Generate ./DeepLearningModel

Module Name OCR_20190311T11255114

Previous Step Start to Train

Figure 6.50 Select text detection

- **Max Iterations:** the max times of training images. The larger the value, the longer time it takes.
- **Base Learning Rate:** this parameter determines the speed rate of training.
 - The larger the value, the faster the training loss will fall, but there may be a large fluctuation of the training loss, and even the training loss becomes infinity and the training fails.
 - The smaller the value, the slower the training loss will fall. If the value is too small, there may be little drop in training loss. It may lead to bad training effect in the set max. iterations times.
- **Resize mode:** It has three modes, including direct resize, resize after padded and pad after resized.
- **Pad Value:** it refers to the value filled during the resize process.

Classification

Image classification is mainly used for image recognition teaching of sample images of deep learning. The richer the sample size, the more accurate the image classification will be. Specific steps are shown below.

1. Select **Image Classification** as training type. Load the image file that is to be trained in file folder address. The image file must contain all the sample images. Click **Next** to enter the label data.
2. Enter the label data interface as shown below.

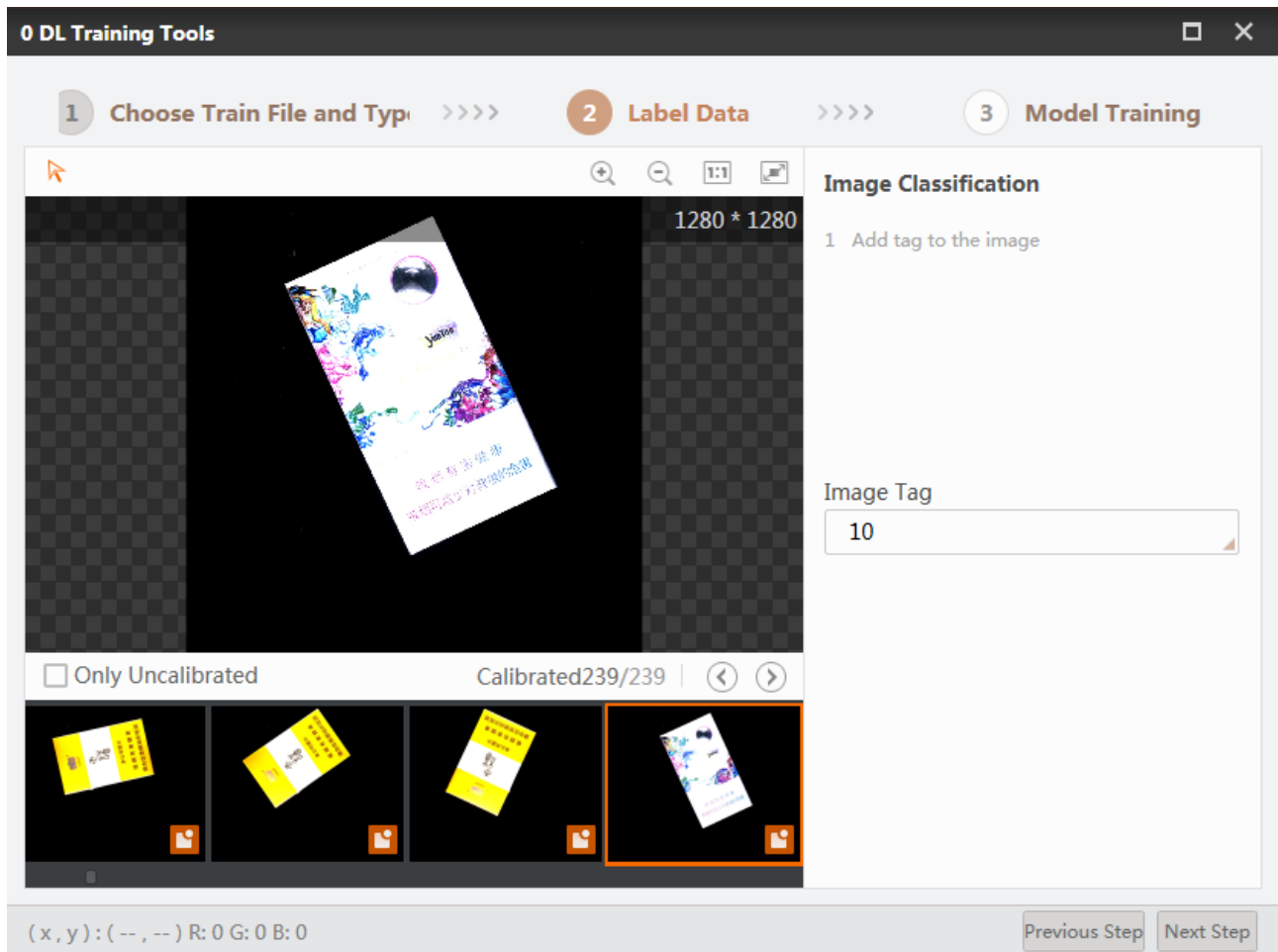


Figure 6.51 Image Classification

You can input corresponding label in the **Image Tag** when the image is displayed on the left side, and the Chinese character input is not supported currently. After completing image classification for one image, you can do the next one.

3. After entering the model training module, set the training parameters, and then start the model training. The settings for image classification training are shown below.

When selecting the character recognition, as shown below.

- **Max Iteration:** the max. times of iterations. The fixed value of batch_size is 4, and it is calculated by the size of training and epoch.
- **Base Learning Rate:** this parameter determines the speed rate of training. If you adjust this parameter to a very small value, it may lead to a infinite loop. If you adjust this parameter to a very large value, it may make the training process very fast.

Object Detection

Object detection is mainly used for target training before deep learning detection. The richer the sample size, the more accurate the object detection will be. Specific steps are shown below.

1. Select **Target Detection** as training type. Load the image file that is to be trained in file folder address. The image file must contain all the sample images. Click **Next** to enter the label data.
2. Enter the label data interface as shown below.

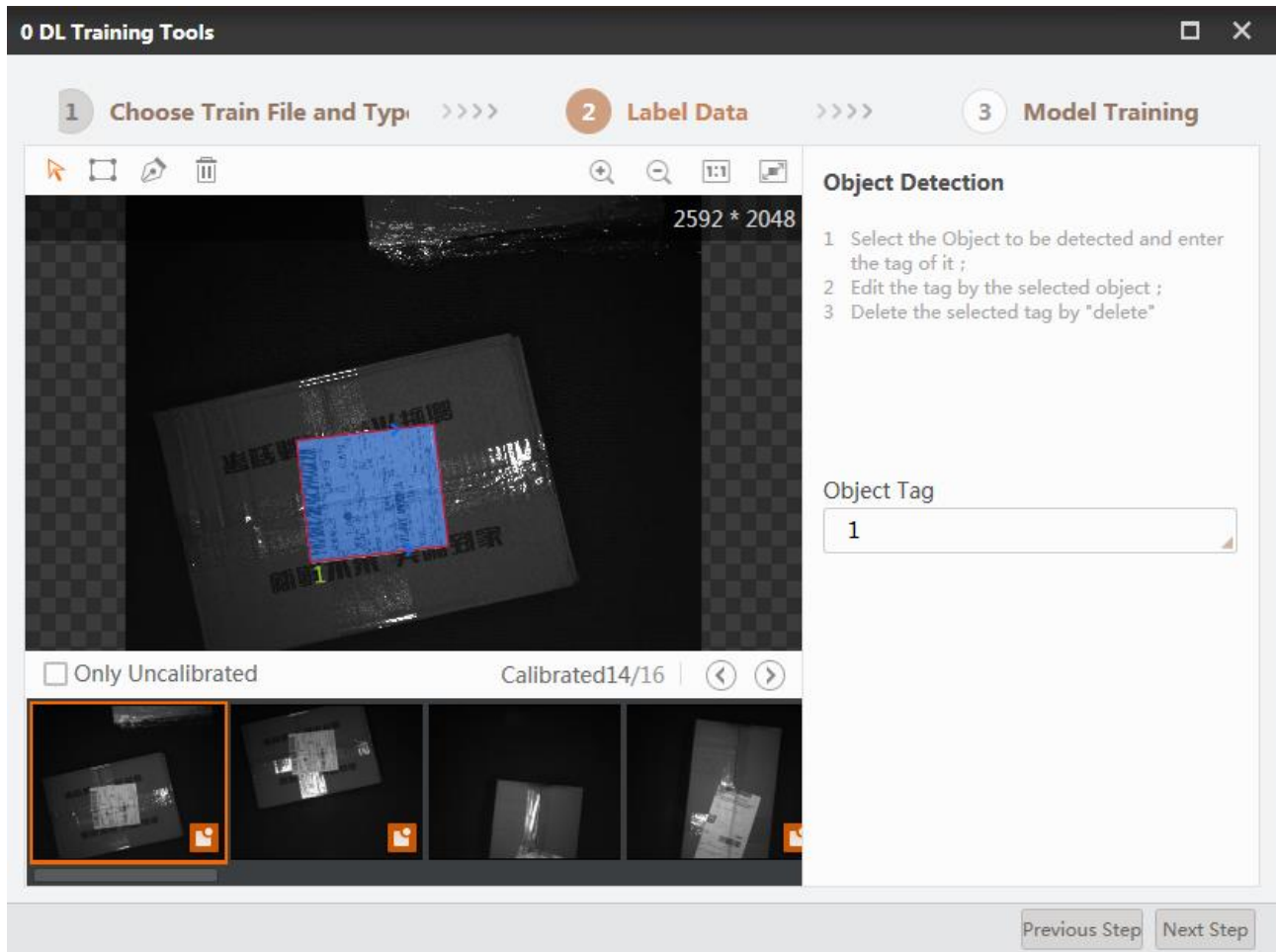


Figure 6.52 Enter the label data interface

You can customize object area, and the area will be blue after completing it. Inputting object tag is available. After completing object detection for one image, you can do the next one.

3. After entering the model training module, set the training parameters, and then start the model training. The settings for object detection training are shown below.

- **Max Iterations:** the max. times of iterations. The fixed value of batch_size is 4, and it is calculated by the size of training and epoch.

- **Base Learning Rate:** this parameter determines the speed rate of training. If you adjust this parameter to a very small value, it may lead to a infinite loop. If you adjust this parameter to a very large value, it may make the training process very fast.

6.4.2 DL Character Recognition

The tool of character recognition refers to the process of determining the shape of characters by detecting light and dark mode, and then translating the shape into character by character recognition.

This tool has the features of low rejection rate, low error rate, fast recognition speed, high stability, friendly user interface, and etc. It is widely applied to dot and matrix character recognition, IC chip character recognition, inkjet character recognition, character of bank card recognition, and etc. Character training is required before character recognition, and it is recommended to use this tool together with character location. The settings of character recognition are as shown below.

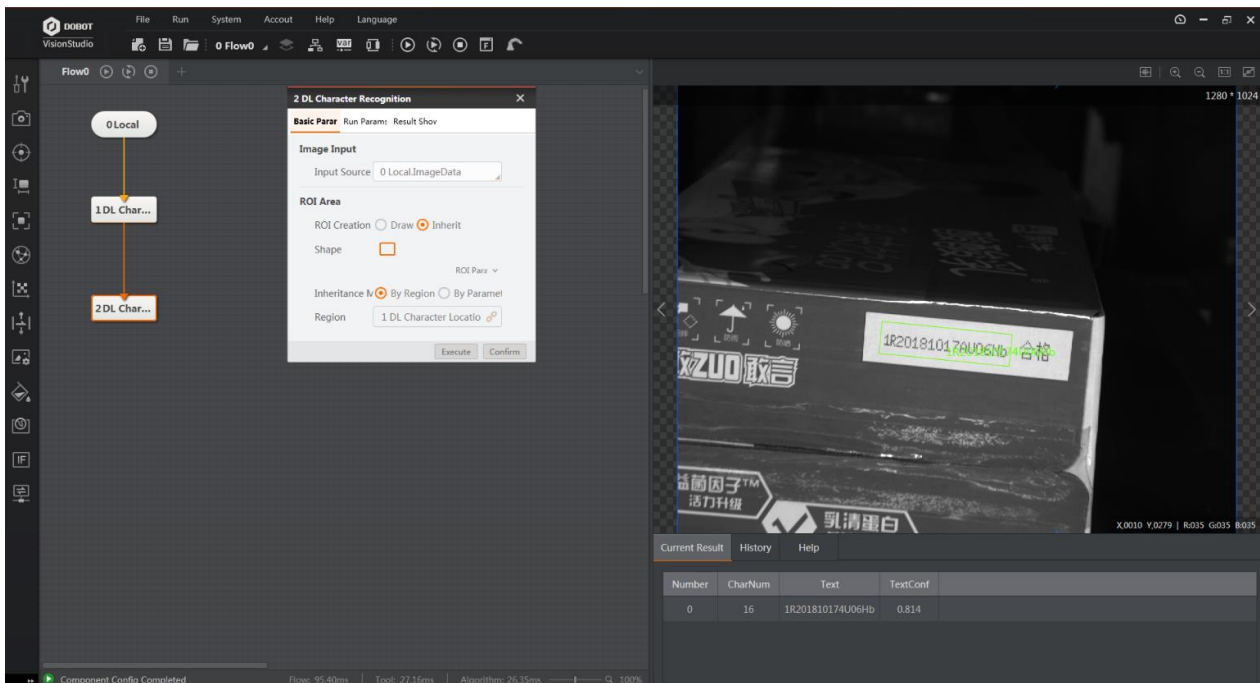


Figure 6.53 The settings of character recognition

- **Basic Parameter:** in this setting, the ROI area where the target character is located must be selected. When the image location is changed, it is recommended to use character recognition together with character location.
- **Running Parameter:** the setting of running parameter is shown below.

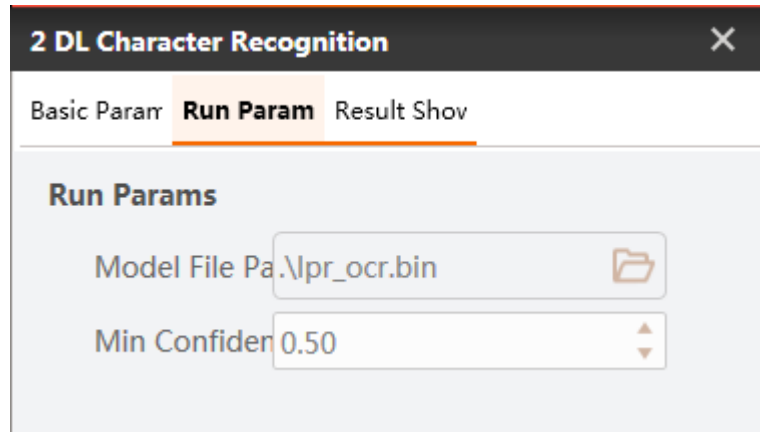


Figure 6.54 Running Parameter

- **Model File Path:** the model file provides a default model, and the user can also load the model file created by the previous character training.
- **Min. Confidence:** the min. score for the location frame.

6.4.3 Defect Detection

The tool of defect detection uses the model of defect training, and sets its parameters to detect defects, as shown below.

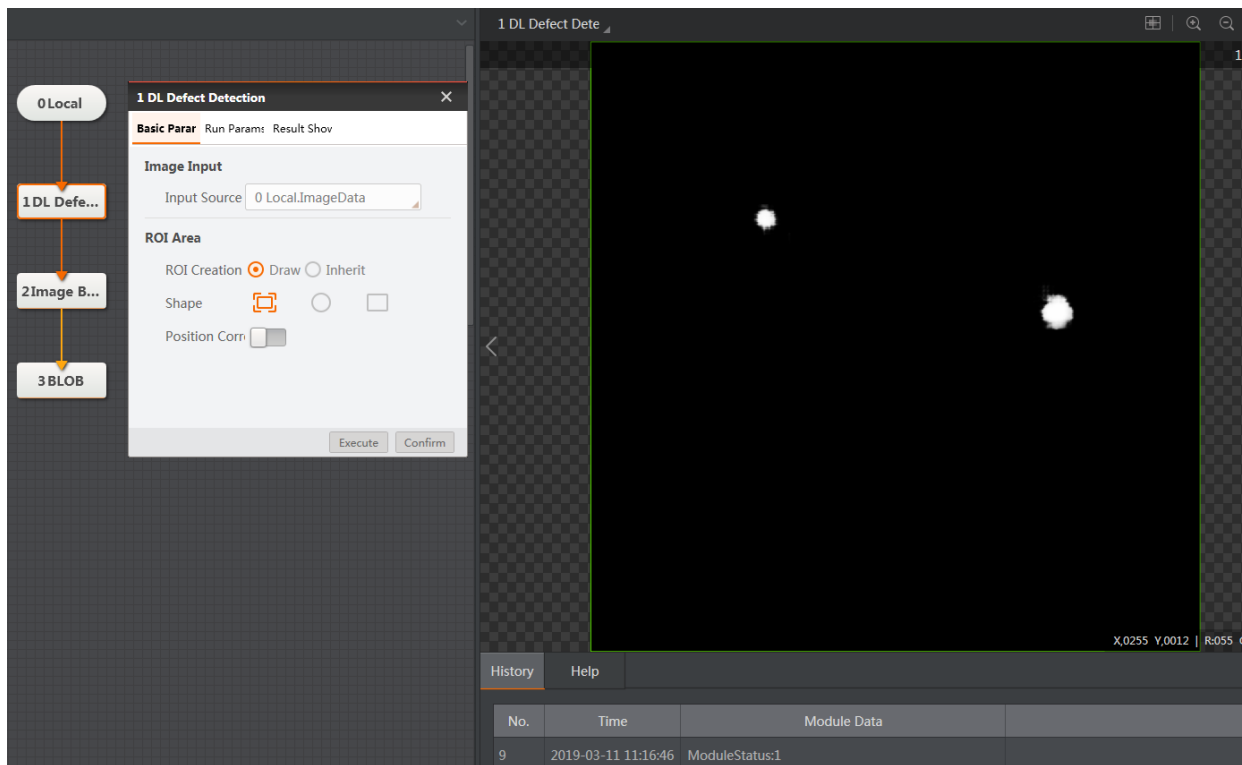


Figure 6.55 Defect detection

For defects with high pixel value, most of interference can be removed by subsequent processing or secondary training. After subsequent processing, the output is as shown below.



Figure 6.56 The output

The setting of running parameter is shown below.

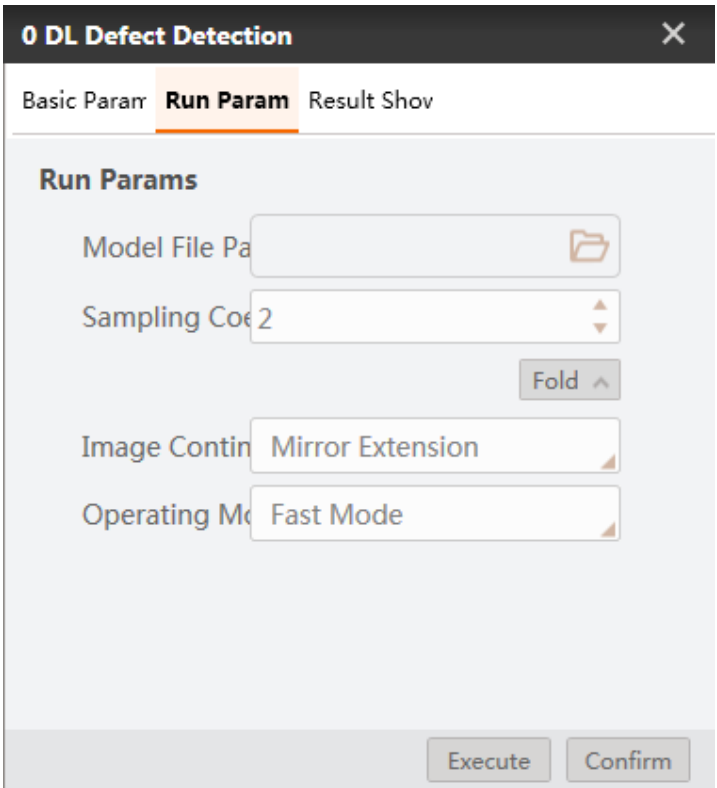


Figure 6.57 Run param

- **Model File Path:** select the model file created by the previous character training.
- **Sampling Coefficient:** the larger the sampling coefficient, the more sparse the contour point will be. If you increase this parameter value, the quantity and size of defect Blob will increase. Thus, it is recommend to adjust the values according to the algorithm and the default is 4.
- **Image Padding:** there are two types of image padding, including mirror padding and zero padding. When there is a defective image created by mirror imaging, and it is mirror padding by default.
- **Running Mode:** it has two modes, including high performance mode and fast mode. It is high performance mode by default.

6.4.4 DL Text Detection

The tool of text detection determines the location of characters by matching the character library information. The specific process is shown below.

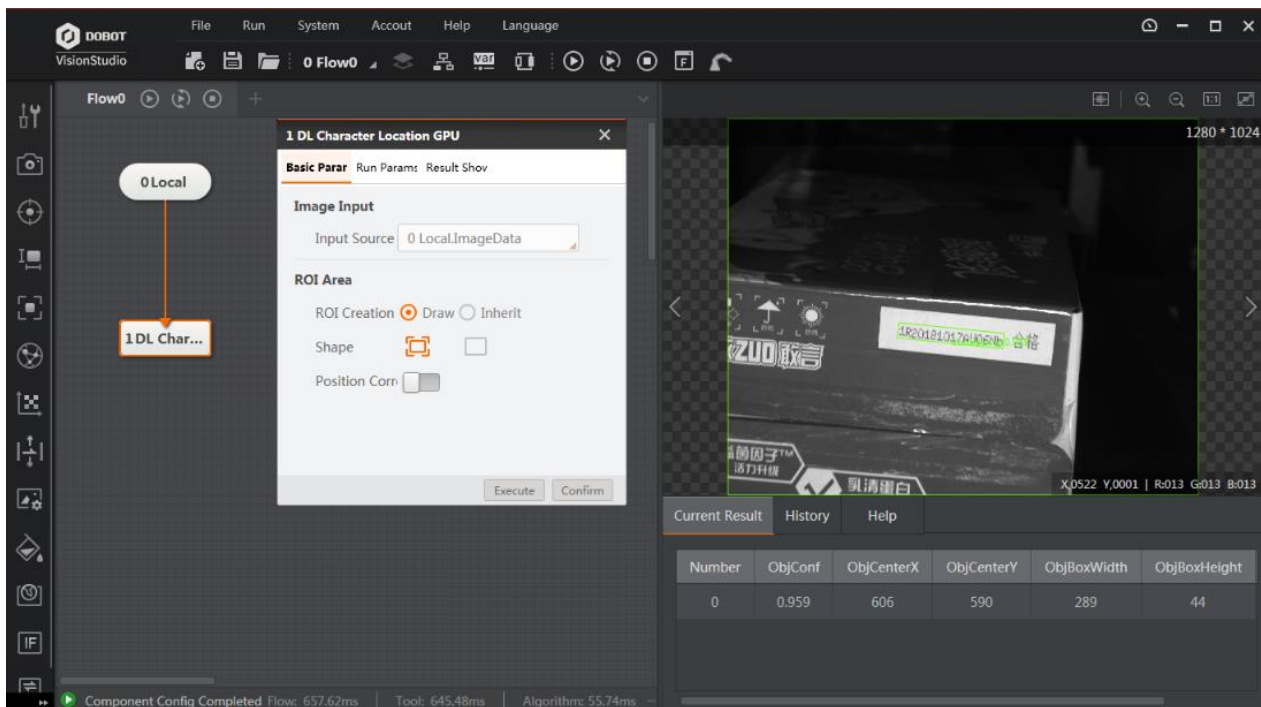


Figure 6.58 Text detection

- **Model File Path:** select the model file created by the previous character training.
- **Max Number to Find:** the max. Quantity for searching target.
- **Min. Confidence:** the min. Score for the location frame.
- **Character Angle Enable:** It is to set the relative angle range tolerance value of the target character. To search for a target with rotation change, you need to set it accordingly, and the default range is from -180° to 180°.

- **Character Width/Height Enable:** the targets whose character width/height are within this range can be detected.

6.4.5 DL Classification

Deep learning classification refers to the image processing method for distinguishing the targets of different types according to the different features reflected in image information. It has two ways, including DL classification G and DL classification C. The suffix of module name of GPU version is G, the suffix of module name of CPU version is C, and the CPU version does not depend on the graphics card. Deep learning classification uses computer to quantitatively analyze images, and classifies each pixel or region in an image into one of several categories so as to replace human beings' visual judgment. It has been widely applied into object recognition and sorting. The specific process is shown below.

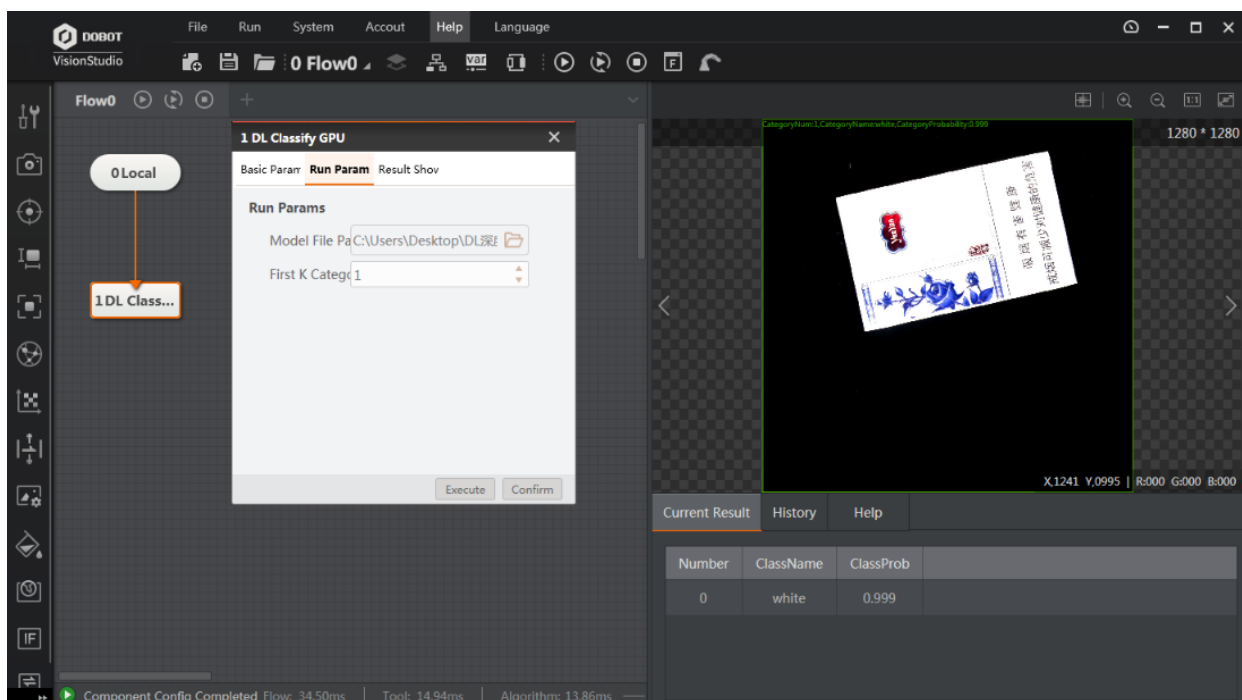


Figure 6.59 DL Classification

- **Model File Path:** select the model file created by the previous character training.
- **Top-K:** it outputs the index number and respective confidence of the highest confidence score of K categories.

6.4.6 DL Object Detection

Object detection is an image segmentation based on object geometry and statistical features, and it combines segmentation and recognition of objects into one. Its accuracy and real-time performance are an important capability of the whole system. It has two detection methods, including DL object detection G and DL object detection C. The suffix of module name of GPU

version is G, the suffix of module name of CPU version is C, and the CPU version does not depend on the graphics card. In complex scenes, when multiple objects need to be processed in real-time, automatic extraction and recognition of objects are particularly important.

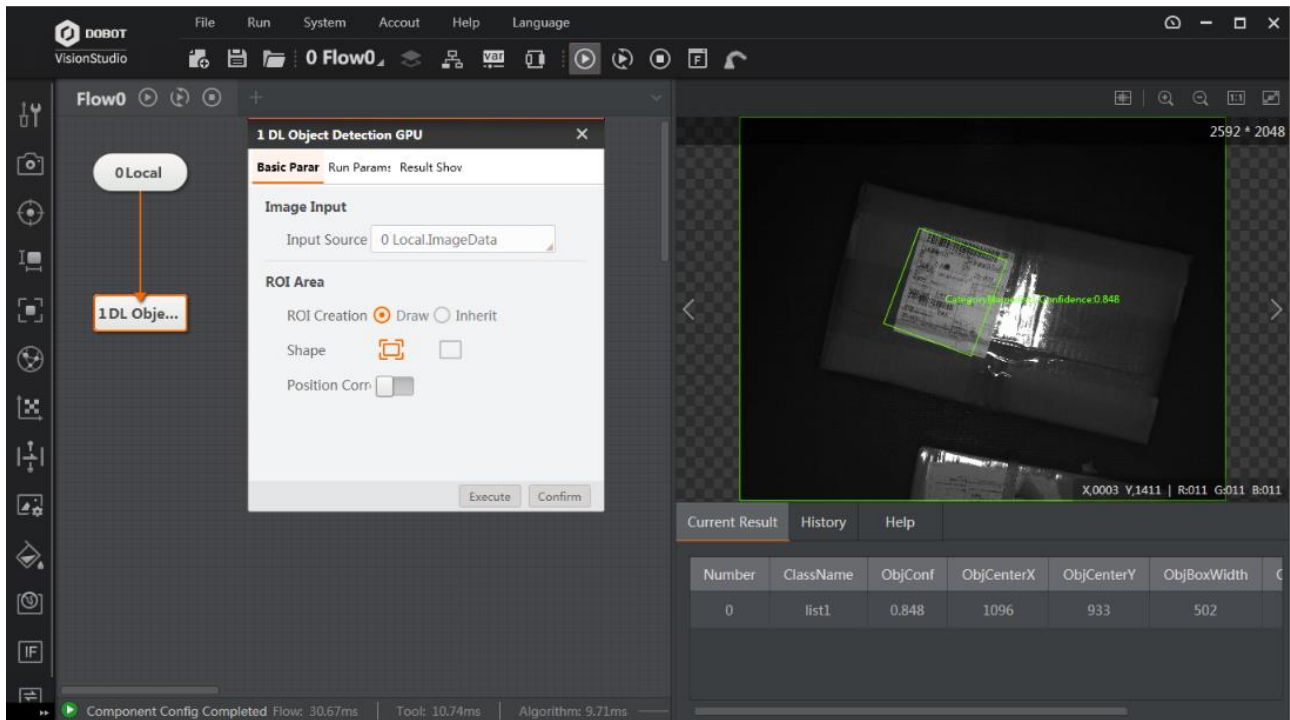


Figure 6.60 DL object detection

- **Model File Path:** select the model file created by the previous character training.
- **Max Number to Find:** the max. Quantity for searching object.
- **Min. Confidence:** the min. Score for the location frame.
- **Max. Overlap Threshold:** the max. Proportion that the object images allow to be sheltered.
- **Character Angle Enable:** it is to set the relative angle range tolerance value of the object character. To search for an object with rotation change, you need to set it accordingly, and the default range is from -180° to 180° .
- **Character Width/Height Enable:** the objects whose character width/height are within this range can be detected.

Note:

The deep learning classification and object detection category can output specific tag value.

6.5 Calibration

6.5.1 N-Point Calibration

The calibration is mainly used to determine the conversion relation between the camera coordinate system and the robot arm coordinate system. The N-point calibration realizes the conversion between the camera coordinate system and the executing structure coordinate system and generates calibration file via N-point pixel coordinate and physical coordinate. The N needs to be larger than or equal to 4.

In the actual situation, there are mainly two calibration methods, including upper camera capture and lower camera alignment, as shown below.

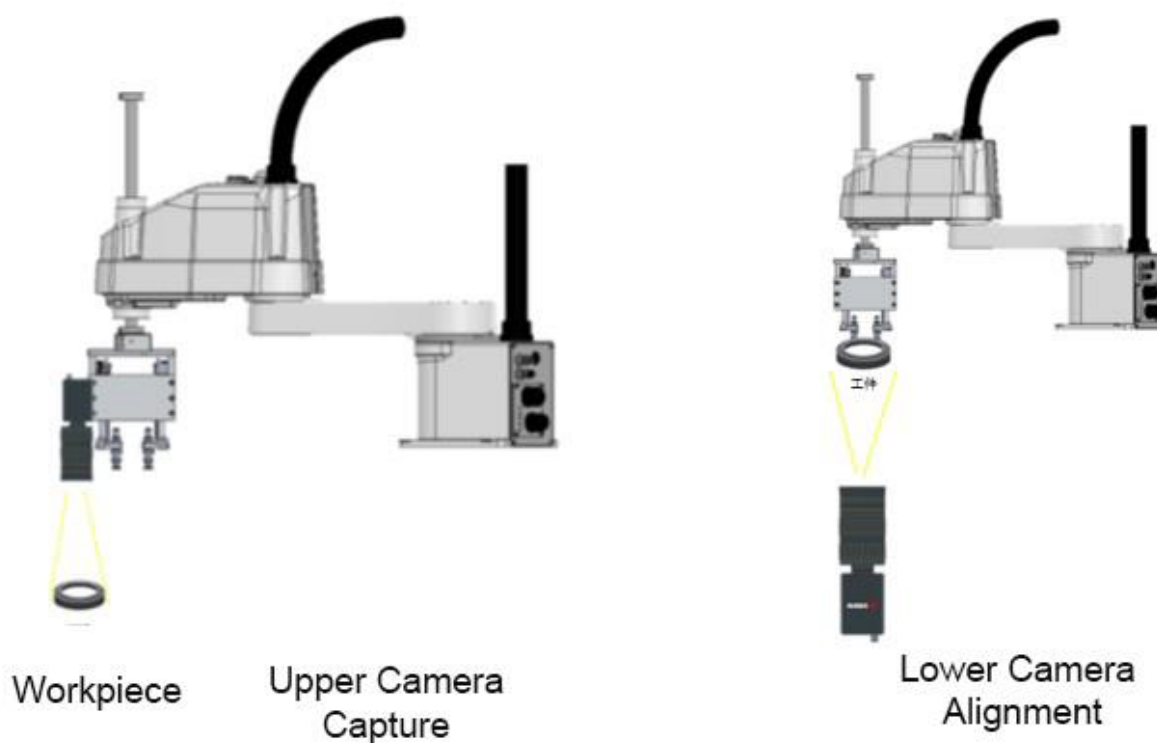


Figure 6.61 Two calibration methods

The calibration solution is shown below. The role of branch module is used to determine whether feature matching is successful or not. If it is successful, the flow enters N-point calibration. Otherwise, format module is a specific character.

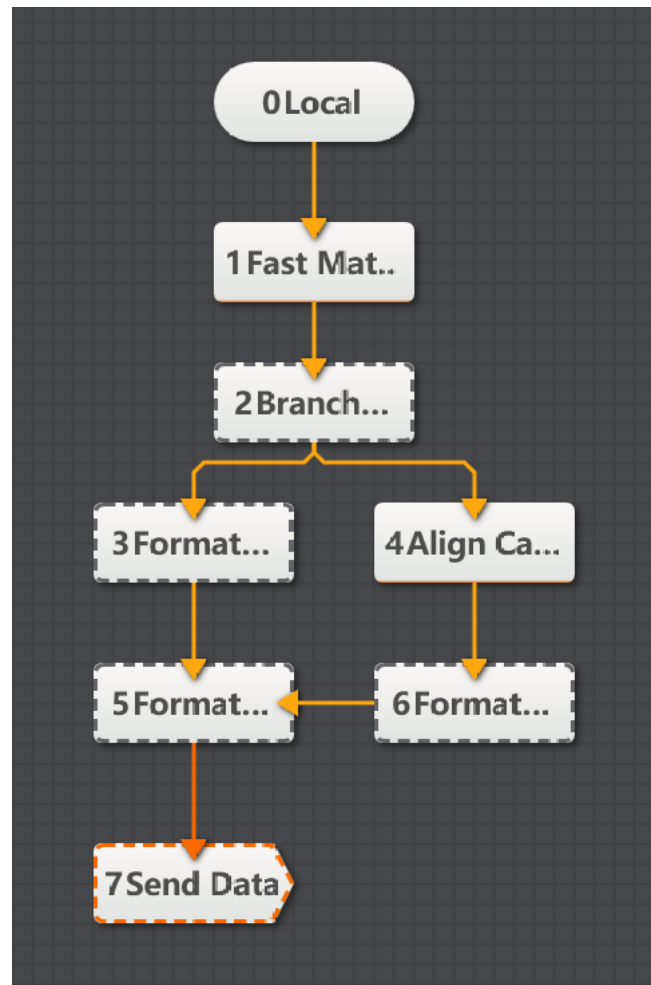


Figure 6.62 Calibration solution

Taking lower camera alignment as an example, N-point calibration is executed by the camera that is driven by the mechanical arm to move in the direction set by the parameter. Each movement will trigger the camera to capture a picture. At this time, the calibration module in the solution is synchronously calibrated. At last, the calibration file is created. The specific parameter settings are shown below.

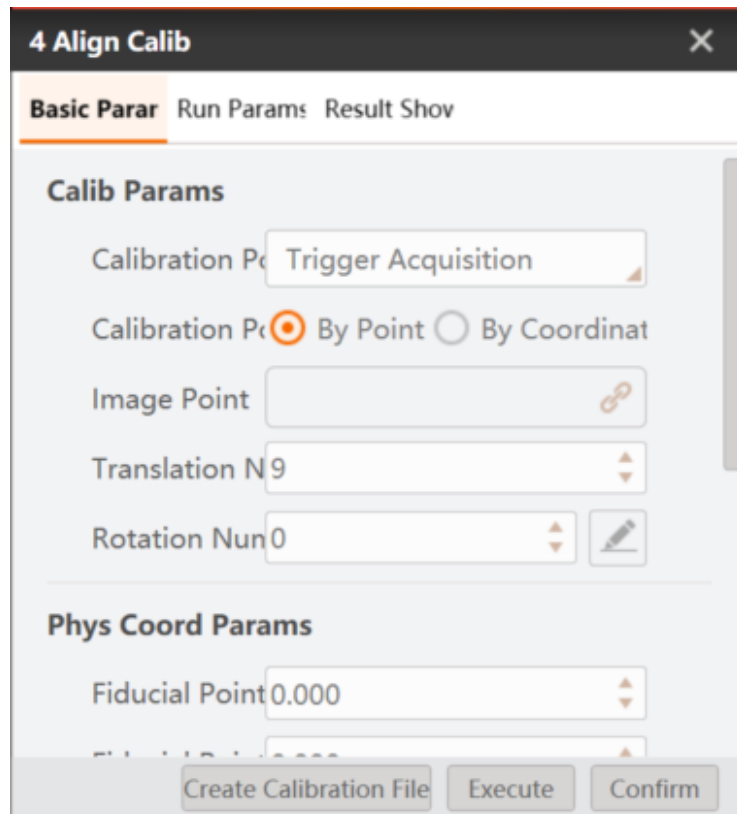


Figure 6.63 Align calib

- **Calibration Points Acquisition:** select trigger acquisition or manual input.
- **Calibration Points Input:** select by point or by coordinate.
- **Image Point:** the calibration point of N-point.
- **Translation Times:** the times for obtaining calibration by moving. It is related with X/Y movement only, and it is set as 9 by default.
- **Rotation Times:** the number of rotations is required when the rotation axis is not coaxial with the center of the image, and it is recommended to set as 3. The rotation happens at the 5th location.
- **Calibration Origin:** it is recommended to set as 4.
- **Fiducial Point X and Fiducial Point Y:** physical coordinate of the fifth point of the robot arm movement.
- **Offset X and Offset Y:** the amount by which the arm moves in one direction per movement.
- **Shifting Mode:** it sets the movement priority direction for mechanical operation each time.
- **Shifting Times:** the times of robot arm changes direction after moving.

- **Fiducial Angle:** the initial offset angle of the camera to the calibration. If it rotates 3 times, and rotation angle is from -10 degrees to 0 degree, and then to +10 degrees. The fiducial angle is -10 degrees and offset angle is 10 degrees.

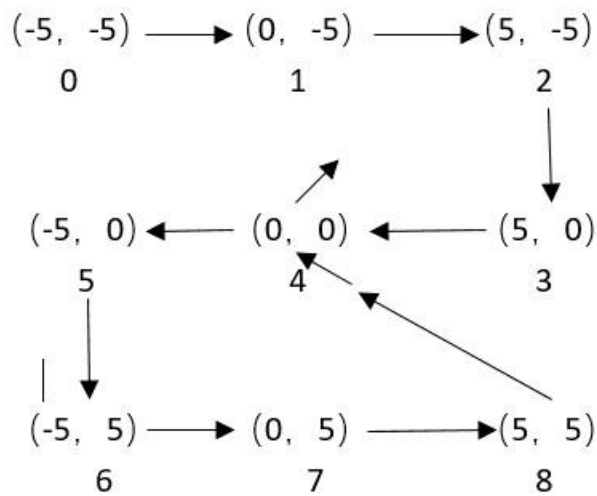


Figure 6.64 Fiducial angle

In the image above, movement is made towards X or Y direction for 9 times, and rotation is made towards other direction for 3 times. It is X axis movement first and the rotation times is 3.

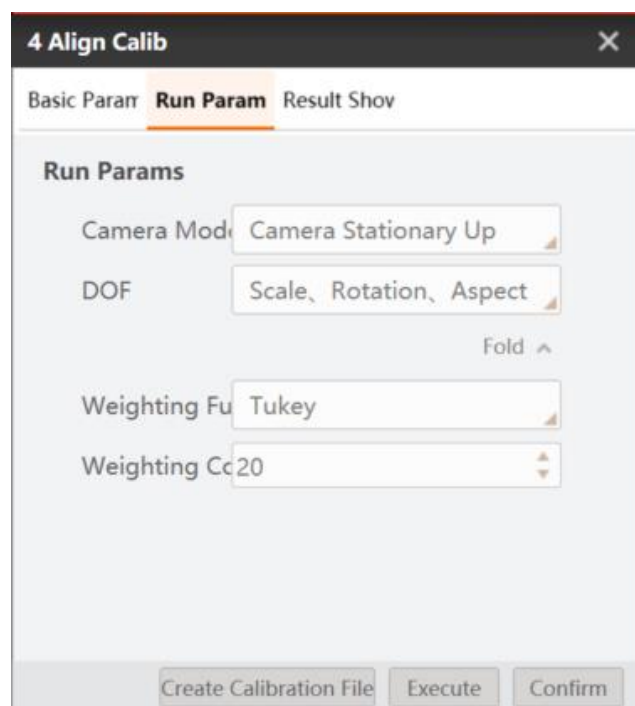


Figure 6.65 Set run param

- **Camera Mode:** it has three types, including static camera upper position or static camera lower position, and moving camera.
- **DOF:** you can select according to your demands, including scale, rotation, aspect ratio, tilt, translation and transmission, and etc,
- **Weighting Function:** you can select least squares, Huber, Tukey, and Ransac algorithm functions. It is recommended to use default parameter.
- **Weighting Coefficient:** it is the parameter setting item when Tukey or Huber weight function is selected.
- **Distance Threshold:** it is the parameter setting item when Ransac weight function is selected, which means that the distance threshold for eliminating the error point. The smaller the value, the stricter the point set selection will be. When the point group's accuracy is not high, this threshold can be appropriately increased. It is recommended to use the default value.
- **Sampling Rate:** it is the parameter setting item when Ransac weight function is selected. When the point group's accuracy is not high, the sampling rate can be appropriately reduced. It is recommended to use the default value.

Note:

N-point calibration module can be independently used without any other module. You just need to drag the module from the left menu to the operation interface, and input calibration coordinates, the software will automatically output calibration result.

6.5.2 CalibBoard Calibration

The calibration board calibration is divided into two types, including checkerboard and circle grid board.

Here we take the checkerboard as an example. After inputting the checkerboard grayscale image and its dimension parameters, the software will calculate the mapping matrix, calibration error, and calibration status between the image coordinate system and the checkerboard physical coordinate system. This tool can generate one calibration file to be used by the calibration conversion. By clicking Creating Calibration File, you can select save path for created calibration file, as shown below.

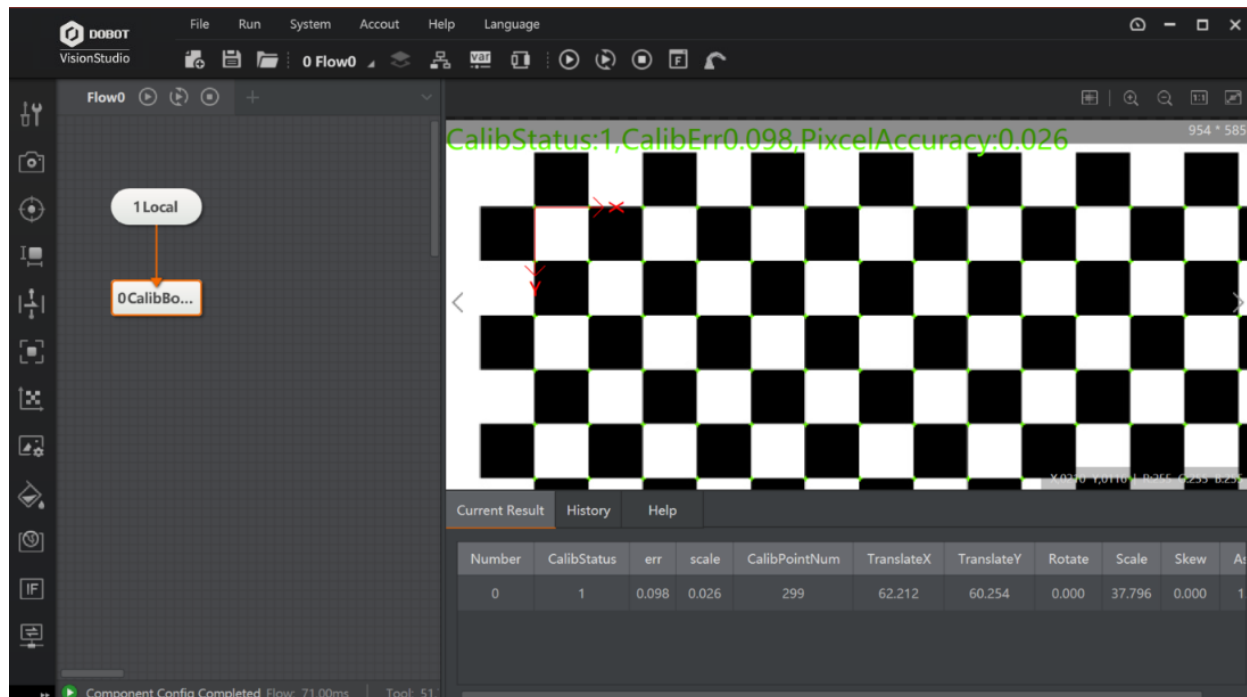


Figure 6.66 CalibBoard calibration

- **Create Calibration File:** select the creating calibration file storage path.
- **Origin (X) and Origin (Y):** this origin point is the origin point of the physical coordinates.
- **Rotation Angle:** the rotation angle of calibration board.
- **Coordinate System Mode:** select the left-hand coordinate system or the right-hand coordinate system.
- **Physical Size:** the side length of each black and white grid of the checkerboard or the center distance of two adjacent centers of the circular plate, and the unit is mm.
- **Calibration Board Type:** checkerboard and circle grid board.
- **Subpixel Window:** it refers to the widow that is used when extracting checkerboard angle point.
- **Set Window Size:** it refers to the size of subpixel window that user sets.
- **Grayscale Contrast:** it refers to the min. value of contrast in the window.
- **Median Filter Status:** It refers to the image median filter preprocess before extracting angle point.
- **DOF:** you can select according to your demands, including scale, rotation, aspect ratio, tilt, translation and transmission, and etc,
- **Dot Circularity:** it is the threshold of circle detection. The larger the value, the much rounder circle will be detected.

- **Edge Low Threshold:** it is used to extract the low threshold of the edge.
- **Edge High Threshold:** it is used to extract the high threshold of the edge. Only the edge point whose edge gradient threshold is between edge low threshold and edge high threshold can be detected.
- **Weighting Function:** you can select least squares, Huber, Tukey, and Ransac algorithm functions. It is recommended to use default parameter.
- **Weighting Coefficient:** it is the parameter setting item when Tukey or Huber weight function is selected.

6.5.3 CalibBoard Calibration Transformation

After the calibration is completed, the conversion between the camera coordinate system and the robot arm coordinate system can be realized by the calibration conversion module.

Click Load Calibration File Path in Calibration Transformation interface, and select calibration file path. The flow is shown below.

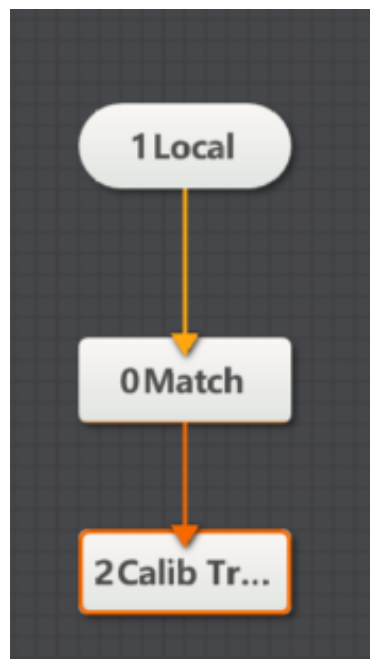


Figure 6.67 Calibration flow

Find the position of the work piece in the camera coordinate system via the feature matching template, load the saved calibration file, and lick Run Once to complete the operation.

The position of the work piece in the robot arm coordinate system after the calibration conversion is shown below.

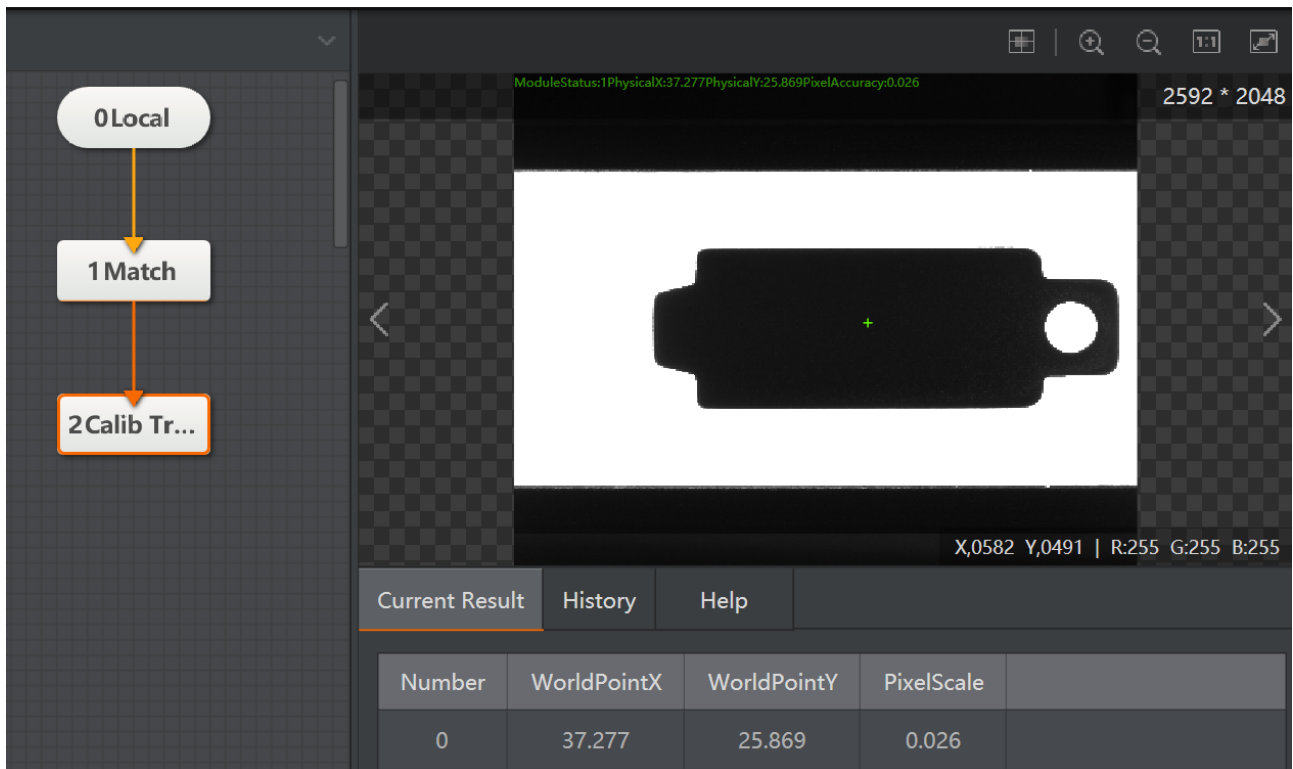


Figure 6.68 Find the work piece

Through external communication, the camera is controlled to capture images. The pixel coordinate positioning of the work piece image is realized by using feature template function. The generated calibration file is loaded in the calibration conversion module, and the pixel coordinates are output as robot arm coordinate. Through formatting, the external communication tells the robot arm unit to complete the function of the control robot arm.

- **Image Coordinate Input:** select the input method by point or by coordinate, and the input source.
- **Align File:** load the calibration file.

The basic flow is shown below.

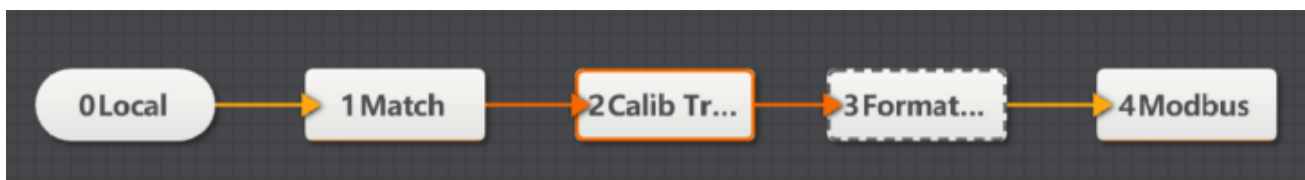


Figure 6.69 The basic flow

6.5.4 Scale Transformation

The tool of scale transformation is used to convert pixel units like distance and width to physical units. For specific use, you only need to load the calibration file and set the distance to be converted, as shown below.

2 Scale Transformation [X]

Image Input

Input Source: 0 Local.ImageData

Scale Transformation

Pixel Interval: 1 Find Circle.fiterror[]

Align File

Load Calibration File: D:\VM3.0.2\B1\VisionM...

Execute Confirm

Figure 6.70 Load calibration file

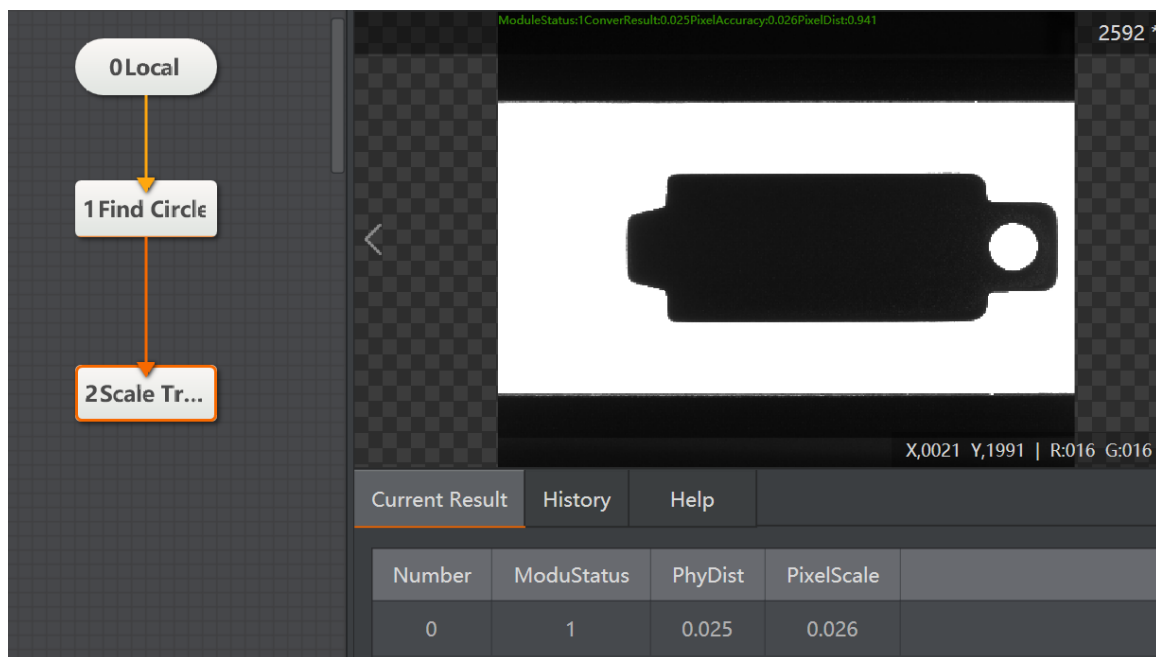


Figure 6.71 Scale transformation

6.5.5 Distortion Calibration

The tool of distortion calibration is used to calibrate calibration board images that have calibration by inputting gray scale calibration template image. It generates calibration file, outputs calibration error and calibration status, as shown below.

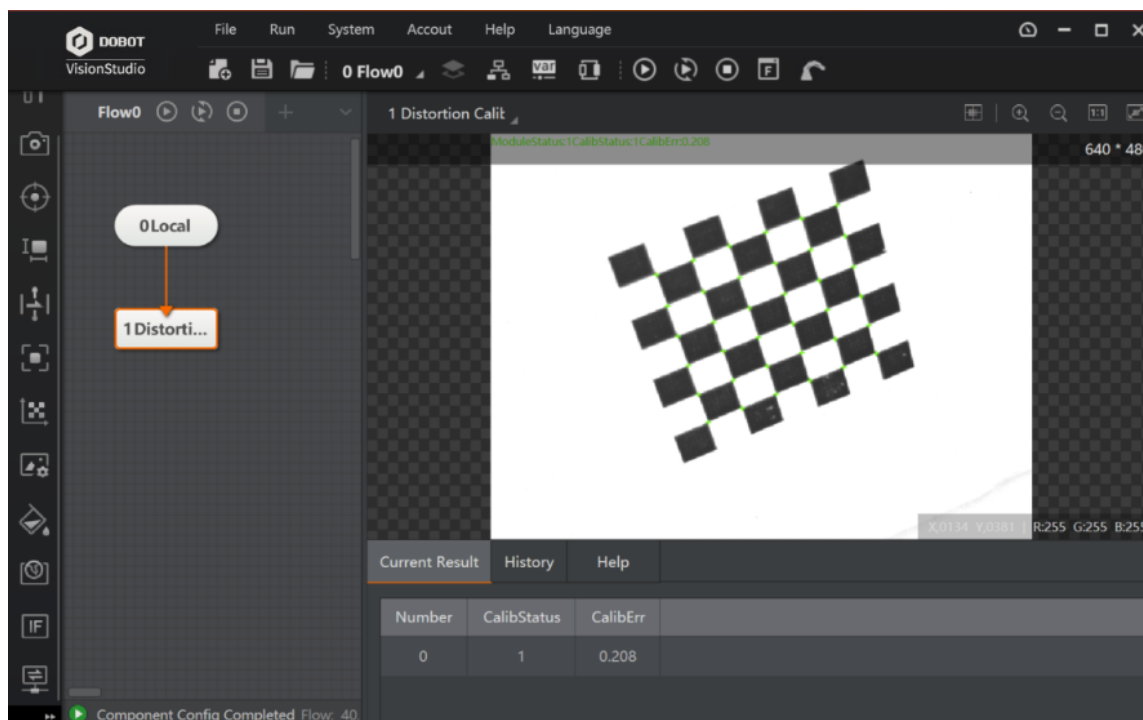


Figure 6.72 Distortion calibration

- **Distortion Type:** it has three types, including perspective distortion, radial distortion and radial perspective distortion.
- **Calibration Board Type:** select checkerboard calibration board, or circle calibration board.
- **Correction Center Point:** It refers to perspective center coordinate when executing perspective distortion correction.
- **Dot Circularity:** it is the threshold of circle detection. The larger the value, the much rounder circle will be detected.
- **Edge Low Threshold:** it is used to extract the low threshold of the edge.
- **Edge High Threshold:** it is used to extract the high threshold of the edge. Only the edge point whose edge gradient threshold is between edge low threshold and edge high threshold can be detected.
- **Weighting Function:** you can select least squares, Huber, Tukey, and Ransac algorithm functions. It is recommended to use default parameter.
- **Weighting Coefficient:** it is the parameter setting item when Tukey or Huber weight function is selected.

6.5.6 Image Correction

The tool of image correction is used to correct images by loading distortion calibration file, and outputs the corrected image. After loading distortion calibration file and setting parameters of calibration board, you can use click Run Once to operate. The image comparison between the original image and the corrected image is shown below.



Figure 6.73 Image correction

- **Correction Mode:** it has three types, including perspective distortion correction, radial distortion correction and radial perspective distortion correction.
 - Perspective distortion correction: it is used to solve the perspective transformation matrix of the image. It is applied to the tilted calibration board (which is not perpendicular to the camera optical axis), the lens distortion is relatively small, and you can obtain an image without perspective distortion.
 - Radial distortion correction: it is used to solve the radial distortion parameters of the image. It is applied to estimate the radial distortion coefficient of the lens. If you do not need to remove the perspective distortion of the image, but need to remove the radial distortion, and you can select this mode.
 - Radial perspective distortion correction: it is applied to general scenes, and it can solve both perspective distortion and radial distortion.

6.6 Alignment

6.6.1 Camera Alignment

The camera alignment module calibrates the conversion relation between the two camera coordinate systems via the corresponding pixel pair of the two cameras, and outputs the calibration file, the calibration status, and the calibration error, as shown below.

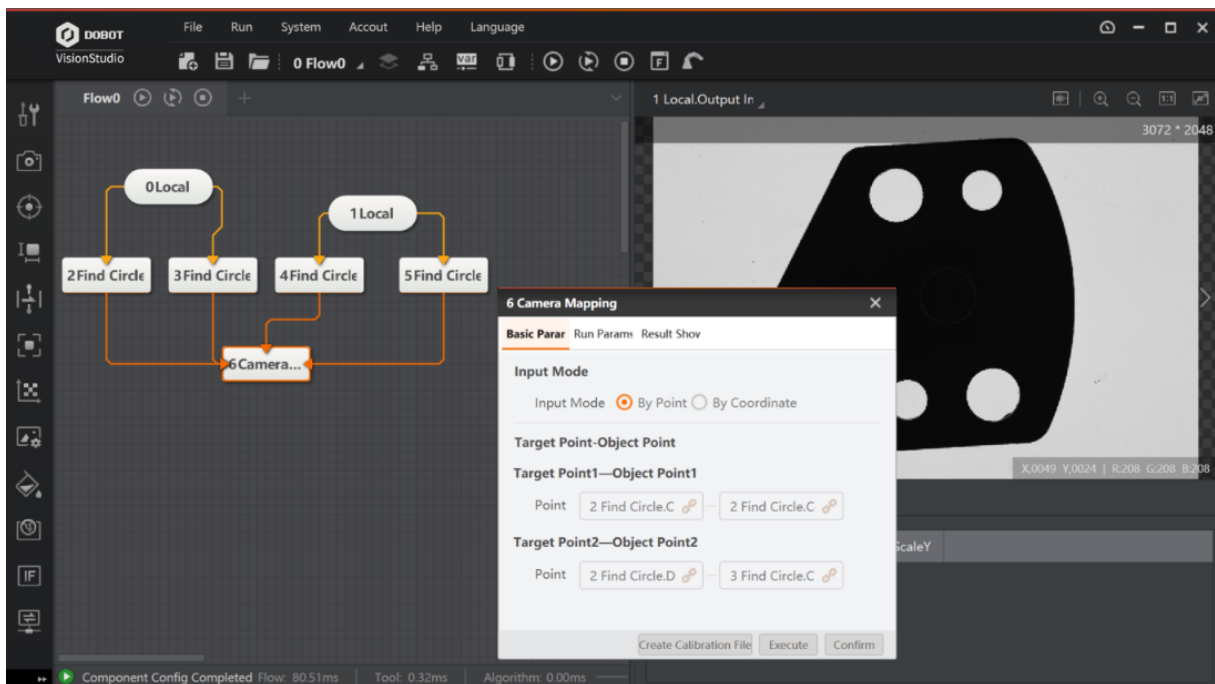


Figure 6.74 Camera alignment

- **Input Mode:** select by point or by coordinate.

- **Target Point-Object Point:** select target point and object point, which need to be at least 1 pair.
- **Create Calibration File:** it outputs calibration file.

6.6.2 Point Alignment

According to the input target point position (X_0 , Y_0) and the direction and the object point position (X_1 , Y_1) and direction, the tool of point alignment calculates the movement amount required from the object point to the target point, including the position movement amount and angle movement amount. The alignment module inputs the physical coordinates, so it needs to be used together with the calibration conversion, as shown below.

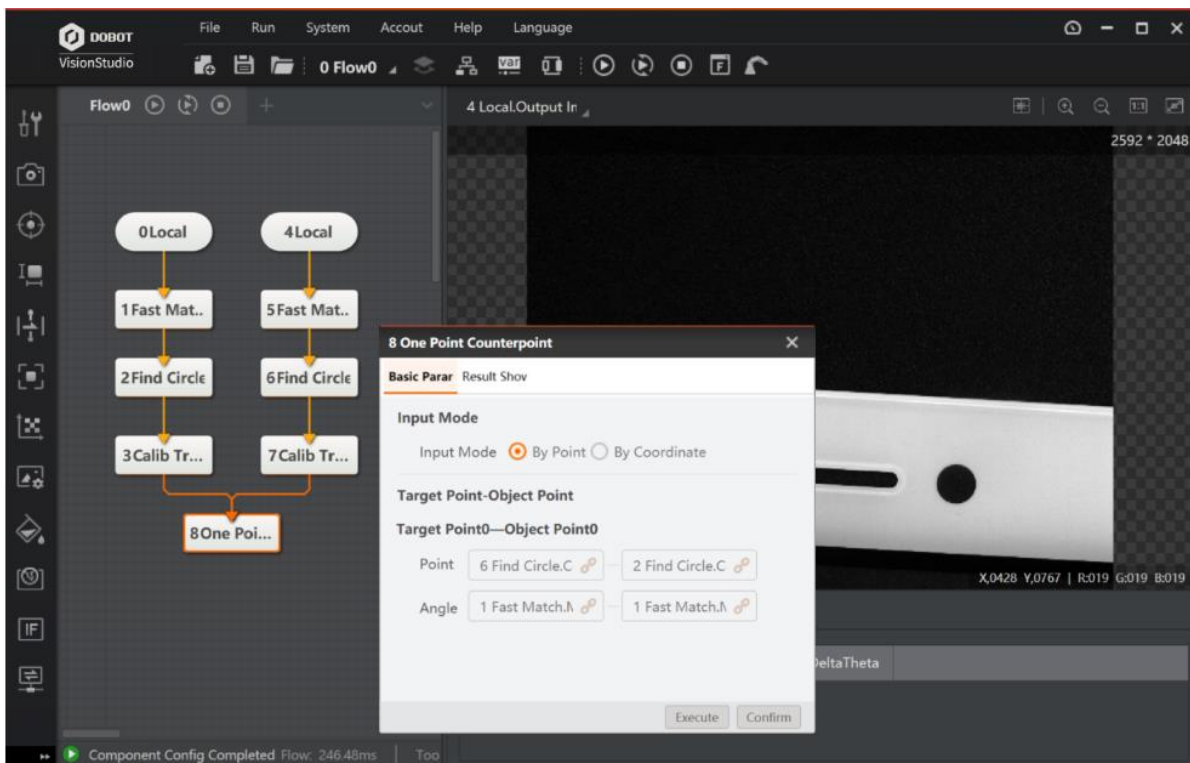


Figure 6.75 Point Alignment

- **Input Mode:** select by point or by coordinate.
- **Target Point-Object Point:** select target point and object point.

6.6.3 Points Alignment

According to the X array and Y array of input target points, and the X array and Y array of object points, the tool of points alignment calculates the movement amount required from the points to the target point, including the position movement amount and angle movement amount. The alignment module inputs the physical coordinates, so it needs to be used together with the calibration conversion, as shown below.

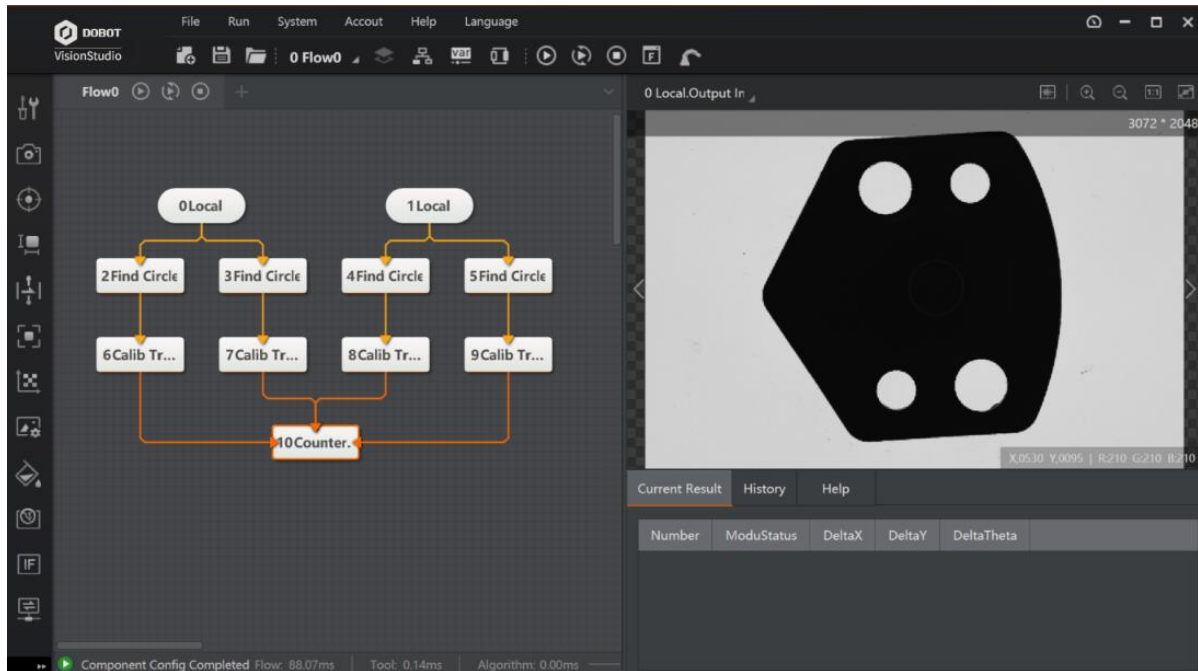


Figure 6.76 Points Alignment

- **Input Mode:** select by point or by coordinate.
- **Target Point-Object Point:** select target point and object point, and the number of points should be larger or equal to 2.

6.6.4 Line Alignment

According to the input target lines and object lines that are both line arrays, the tool of line alignment calculates the movement amount required from the points to the target point, including the position movement amount and angle movement amount, as shown below.

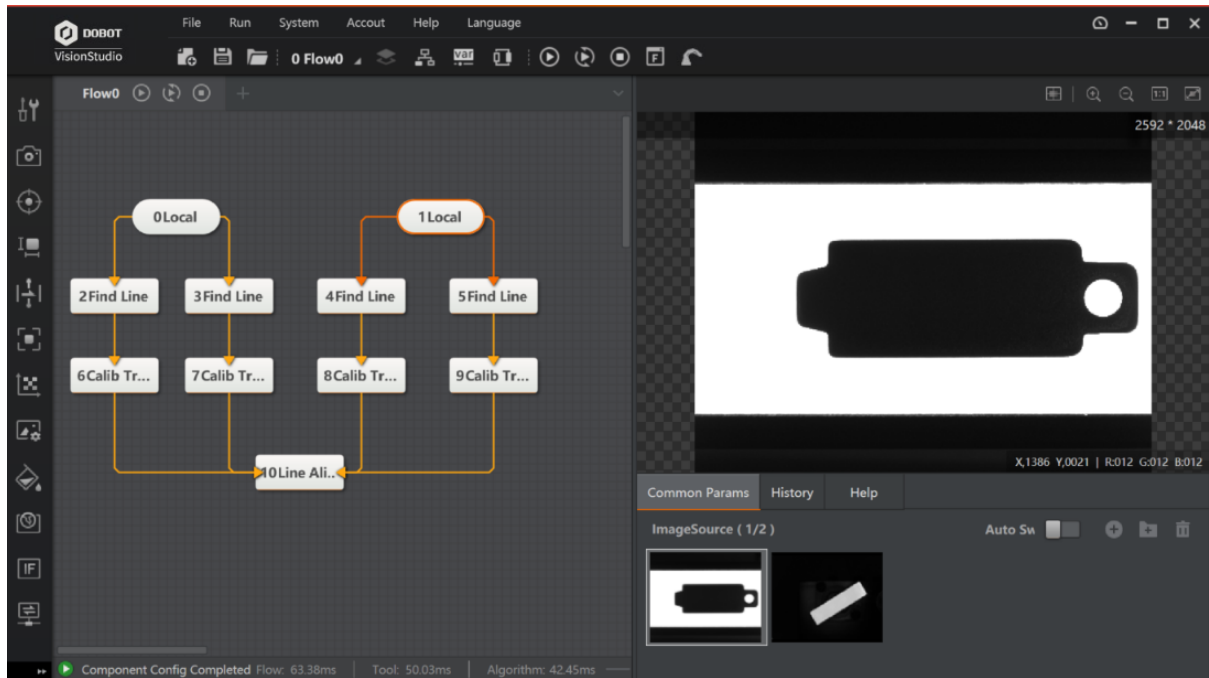


Figure 6.77 Line alignment

- **Input Mode:** select by point or by coordinate.
- **Target Point-Object Point:** select target point and object point, and the number of points should be larger or equal to 2.
- **Alignment Shape:** according to the shape of lines, select opening or closing. If the shape of lines is closed, select closing.

6.7 Image Processing

The tool of image processing is used to process the target image. The sub-menu of image processing includes image combination, image morphology, image binarization, image filter, image enhancement, image math, image sharpness, image fixture, shade correction, affine transformation, polar unwarp, copy fill, frame mean, and image normalization, as shown below.

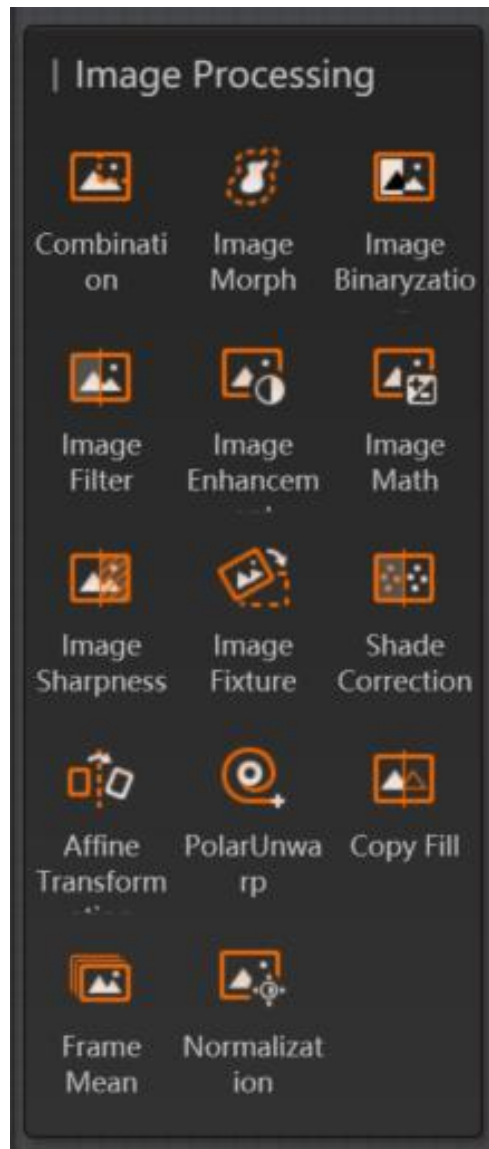


Figure 6.78 Image processing

6.7.1 Image Combination

The image combination combines with five image processing modules, including morphological processing, image binarization, image filter, image enhancement and shade correction. Its operation process is shown below.

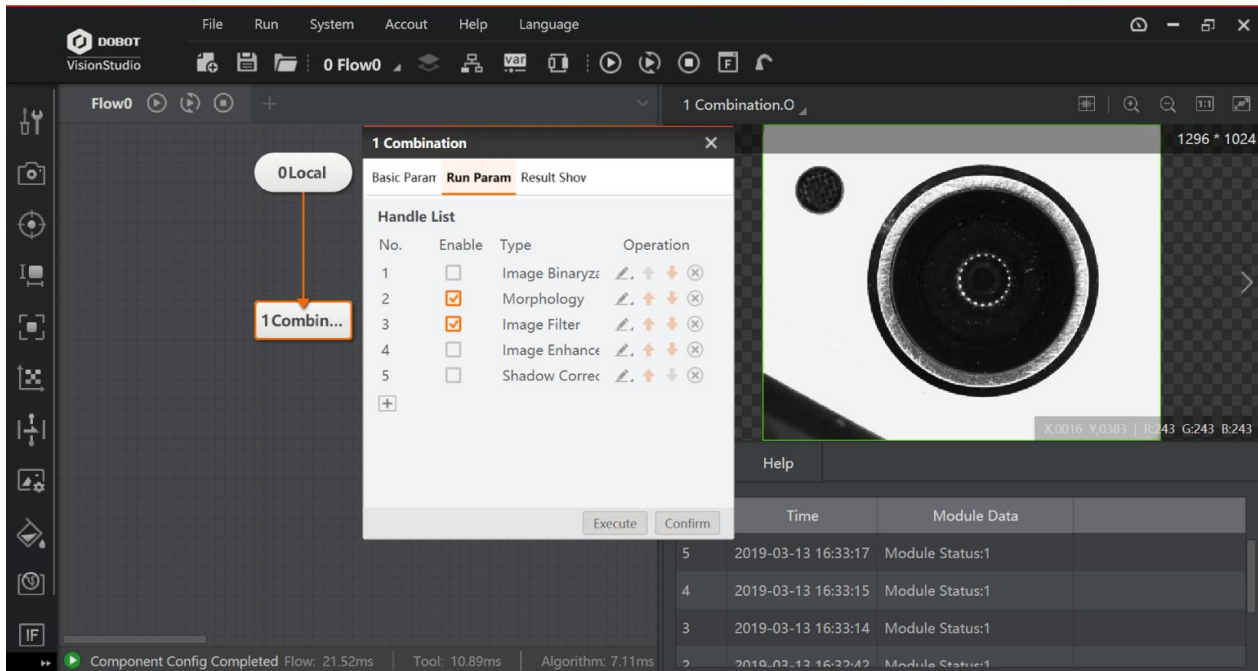




Figure 6.79 Image combination

Steps:

1. Add different image processing modules to the processing list.
2. Check Enable to enable the corresponding module function.
3. Click  of the corresponding module to set the running parameters.
4. Click  and then output.

Note: the order in which the corresponding modules operate can be adjusted.

6.7.2 Image Morphology

The tool of image morphology is used to extract image components that are meaningful to the expression and description of area shape, so that the subsequent recognition work can obtain the most essential shape features like edge and connected area of the target object. This tool copes with the white pixels in the image, and its effects are shown below. These effects, from left to right, are the original image, the effect after dilation, the effect of erosion, the effect of the opening operation, and the effect of the closing operation respectively.



Figure 6.80 Image morphology

- **Morphology Type:** it supports dilation, erosion, opening and closing.
 - Dilation is to separate independent image elements, and connect adjacent elements.
 - Erosion is a corresponding operation to dilation.
 - Opening operation means erosion first and then dilation.
 - Closing operation means dilation first and then erosion.
- **Structuring Element:** it is a common parameter, and it supports rect, ellipse and cross.
- **Iteration Times:** the max. running times of the morphological operation algorithm, and it only takes effect after setting dilation and erosion.
- **Element Width:** it is the width of adjacent elements.
- **Element Height:** it is the height of adjacent elements.

6.7.3 Image Binarization

The image binarization refers to the creation of binarized images that reflect the overall and local features of images from 256 grayscale images of the brightness level via appropriate threshold selection. The image binarization effect is shown below. These effects, from left to right, are the original image, the hard threshold binarization effect, the mean binarization effect, the Gaussian binarization effect, and the auto effect respectively.



Figure 6.81 Image binarization

- **Binarization Type:** it refers to the segmentation method of binarization, including hard threshold binarization, mean binarization, Gaussian binarization, and auto binarization. When setting as hard threshold binarization, the grayscale is between low threshold and high threshold, and the respective binarization image pixel is white. Otherwise, it is black. When setting as auto binarization, you do not need to adjust other parameters, area with large grayscale in binarization is shown as white area, and area with small grayscale in binarization is shown as black area.
- **Hard Threshold Binarization:**
 - Low threshold: the min. threshold set by this threshold
 - High threshold: the max. threshold set by this threshold.
- **Mean Binarization Effect:**

- Filter kernel width: the average value of filter kernel width. Increasing this value can make the average filtered image smoother.
- Filter kernel height: the average value of filter kernel height. Increasing this value can make the average filtered image smoother.
- Comparison type: it has four comparison types, including greater than or equal to, less than or equal to, equal to, and not equal to. If the condition is met, it will be displayed as a white area in the binarized image, otherwise it will be displayed as a black area.
- Threshold offset: the compensation value added to the current pixel value during the mean binarization or Gaussian binarization, the compensated pixel value and the average image or Gaussian filter image.

- **Gaussian Binarization:**

- Gaussian filter kernel: it refers to the size of the filter kernel. Increasing this value can make the image after Gaussian filter smoother.
- Gaussian standard deviation: it refers to the level of Gaussian filter.

6.7.4 Image Filter

The tool of Image filter is used to suppress the noise of the target image while reserving the image detail features as much as possible, and it is an indispensable operation in image pre-processing. Its processing effect will directly affect the effectiveness and reliability of subsequent image processing and analysis. The effects of image filter are shown below.

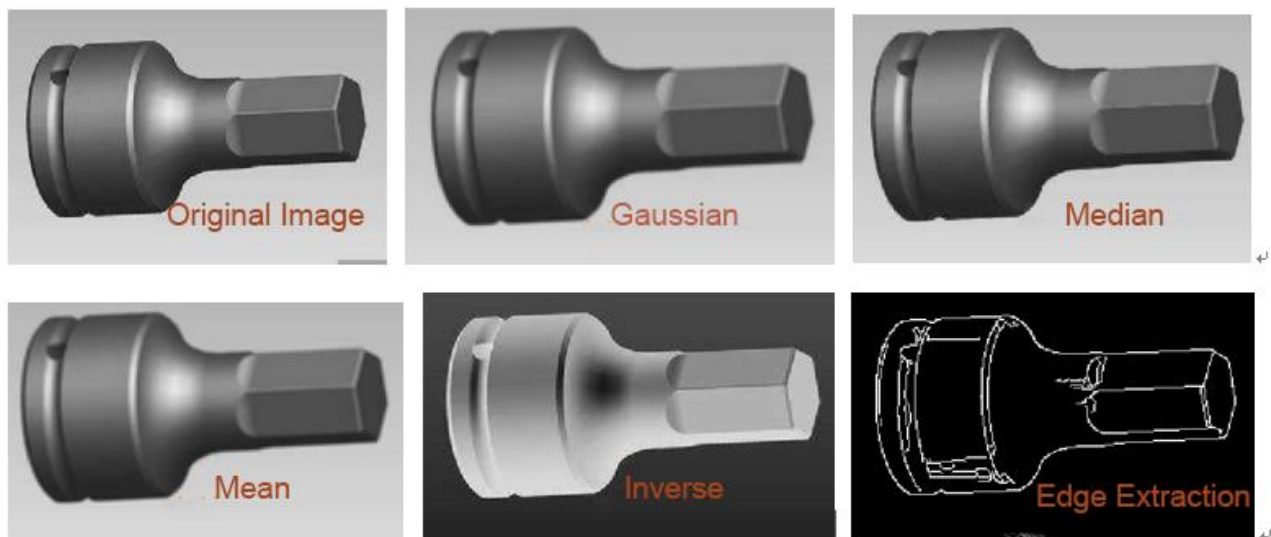


Figure 6.82 Image Filter

- **Image Filter Type:** it includes Gaussian filter, median filter, mean filter, inverse filter, edge extraction.

- **Gaussian filter kernel:** increasing this value can make the image smoother.
- **Filter Kernel Width:** the average or median value of filter kernel width. Increasing this value can make the average filtered image smoother in the horizontal direction.
- **Filter Kernel Height:** the average or median value of filter kernel height. Increasing this value can make the average filtered image smoother in the vertical direction.
- **Edge Threshold Range:** when extracting edge of image, setting this parameter is required, and it determines the extent range of edge extraction. In binary image obtained from edge extraction, the white points stand for edge and the black points stand for background points.

6.7.5 Image Enhancement

The tool of image enhancement includes sharpness, contrast adjustment, Gamma adjustment, and brightness adjustment. The image enhancement effects are shown below. These effects, from left to right, are the original image, sharpness effect, contrast adjustment effect, Gamma adjustment effect, and brightness adjustment effect respectively.



Figure 6.83 Image enhancement

- **Image Enhancement Type:** it includes sharpness, contrast, Gamma, and brightness.
- **Sharpness Intensity:** it is a common parameter, and 1000 refers to coefficient as 1. 0 refers to effect without sharpness. The larger the value, the more sharper will be.
- **Kernel Size:** the size of kernel, and its range is from 1 to 51. This parameter decides the size of sharpness area.
- **Contrast Coefficient:** it is a common parameter, and is used to control contrast coefficient. 100 stands for not adjustment, and the number is larger than 100, it means that contrast increases. Otherwise, it means that contrast reduces.
- **Gamma:** it is a common parameter, and 1 refers to effect without adjustment. Gamma adjustment is used to edit the gamma curve of the image to achieve non-linear tone editing of the image. It can detect the dark portion and the light portion of the image signal, increase their ratio, thus improving Image contrast effect. But, the too large parameter will make the picture dark.
- **Gain:** it is used to increase the overall pixel brightness of the image screen. The default value is 0 and the adjustment range is from 0 to 100.

- **Compensation:** it is used to add or subtract this value for overall pixel of the image. The default value is 0 and the adjustment range is from -255 to 255.

6.7.6 Image Math

The tool of image math can add, subtract, and calculate image absolute value, max. value, min. value, mean value of two images. This tool requires two images with same size and it supports full image operation. The principle of image operation is to calculate the gray value of the same coordinate pixel of two images and then get a new image, as shown below.

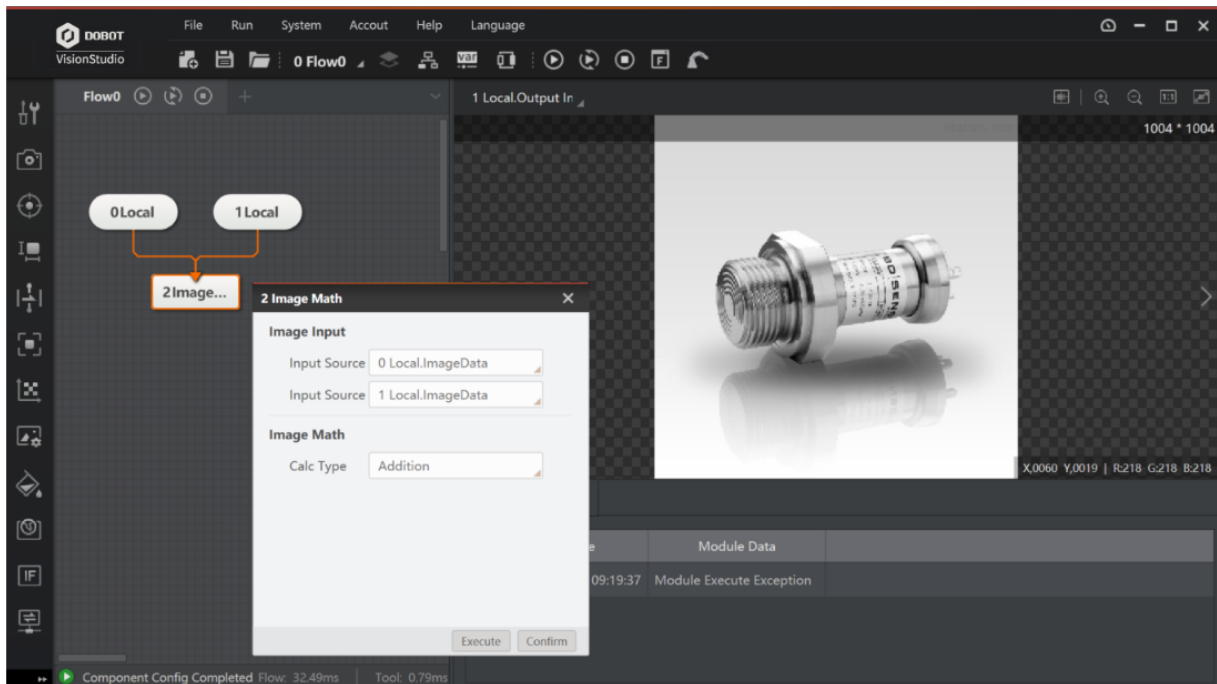


Figure 6.84 Image math

- **Image Input:** you can select the input source image between input source 1 and input source 2 to perform image calculation on these two images.
- **Image Math:** you can select image addition, subtraction, absolute difference, max. value, min. value, mean value, and, or, and xor.

6.7.7 Image Sharpness

The tool of image sharpness is used to assess the sharpness of the image and output the image sharpness score via inputting an image. This tool can judge whether the camera is clearly focused, as shown below.

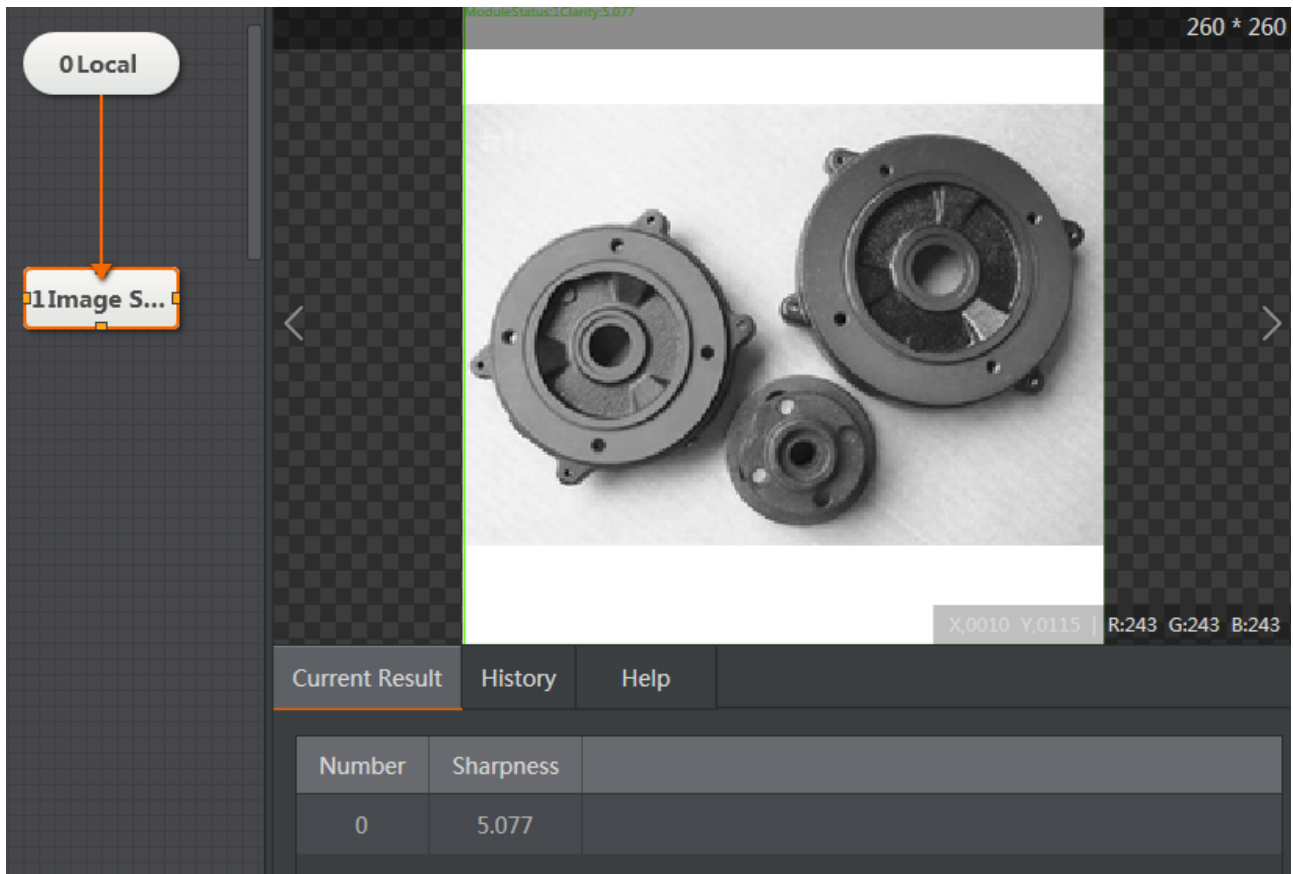


Figure 6.85 Image Sharpness

- **Assess Mode:** it has two modes, including autocorrelation and gradient square. The autocorrelation mode is applied to autofocus scene, and the gradient square mode is applied to the scene with rich image edge information.
- **Noise Level:** the default value is 0. When the image gray level difference is large, you can appropriately increase this parameter. It is valid in the autocorrelation mode only.

6.7.8 Image Fixture

The tool of image fixture is mainly used to correct the position of the image. When the position of the detected target changes, the target needs to be positioned. Image fixture corrects image deviation by setting the initial information of the fiducial point and the change of the fiducial point (position and angle) detected at the time of the operation. This tool make the corrected image and the reference image have the same posture, and it is usually used together with location correction, as shown below.

Note: Image fixture does not affect the resolution of the image.

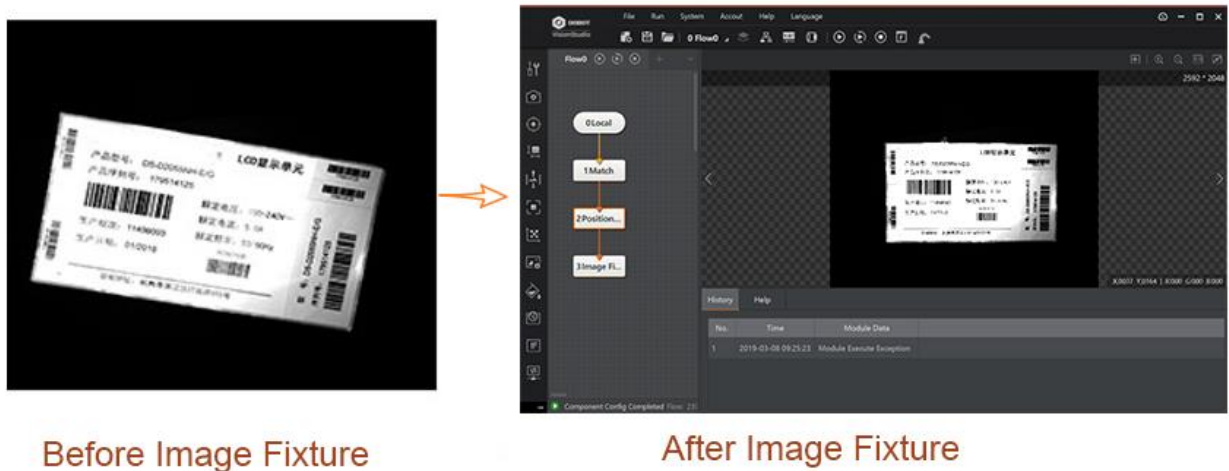


Figure 6.86 Image fixture

- **Position Correction Information:** you can select to correct the position by point or by coordinate. The fiducial point can be obtained by the location tool, and fiducial point information includes its location and angle.

6.7.9 Shade Correction

The tool of shade correction is used to filter the input image, and perform reflectance noise reduction for the difference image which is created by the input image and the filtered image. This tool is widely applied to scenes with uneven image illumination, and the image is made even by adjusting the ratio, gain, compensation and noise, as shown below.

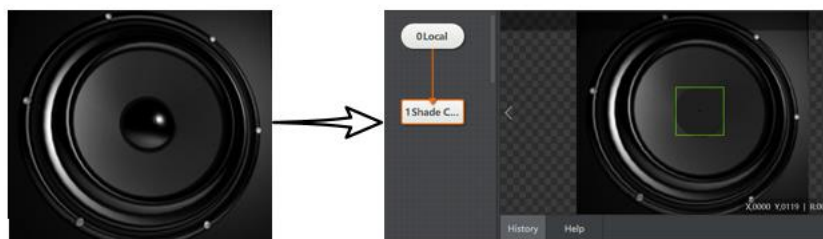


Figure 6.87 Shade correction

- **Filter Kernel Size:** it refers to the size of the filter kernel, and its range is from 1 to 50.
- **Gain:** it refers to the gain of signal amplification, and its range is from 0 to 100.
- **Compensation:** it refers to the brightness compensation value, and its range is from 0 to 255.

- **Noise:** it is used to eliminate noise. If the signal is lower than this parameter value, it will be set as 0, and its range is from 0 to 255.
- **Direction:** it has three types of direction, including X, Y, XY, and all of them indicate the direction of filter kernel size.

6.7.10 Affine Transformation

The tool of affine transformation is used to correct image direction via location and location deviation so as to make image rotate, move, and scale to the template status, as shown below.

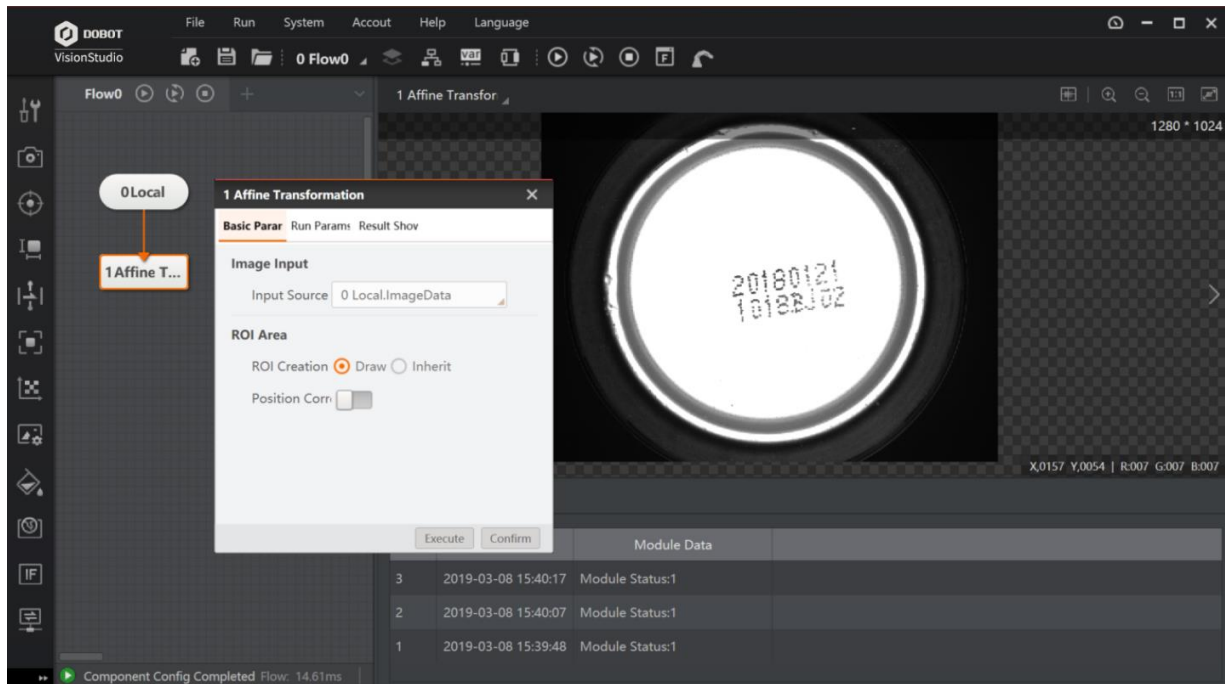


Figure 6.88 Affine transformation

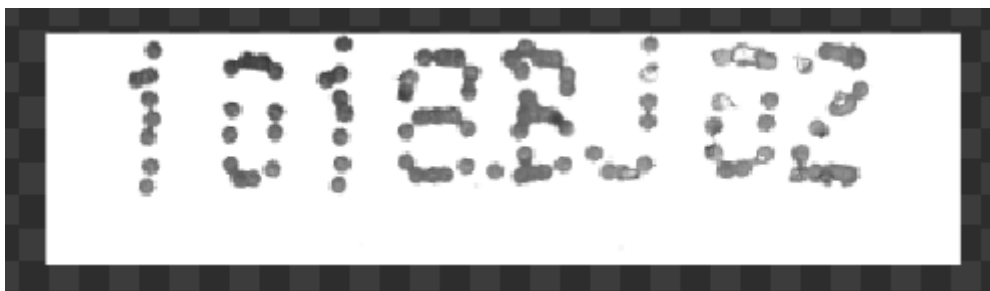


Figure 6.89 The result

- **Scale:** it refers to the image's scale coefficient.
- **Aspect Ratio:** it refers to the ratio of width and height.

- **Interpolation:** it includes two types, including NN and bilinear.
- **Fill Mode:** it includes two types, including constant and copy nearby.
- **Fill Value:** its range is from 0 to 255.

6.7.11 Polar Unwrap

In the polar unwrap interface, select the local image in the input source, otherwise the ROI cannot be selected. After that, in the specific rectangular area, polar unwrap can be performed, as shown below.

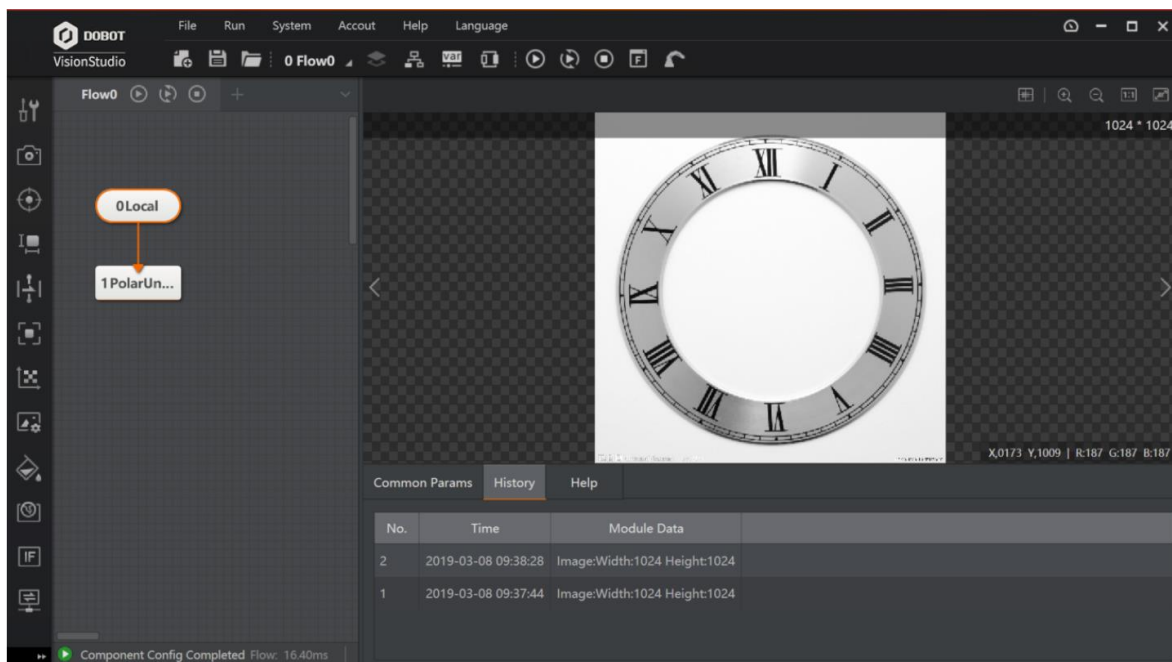


Figure 6.90 Polar unwrap

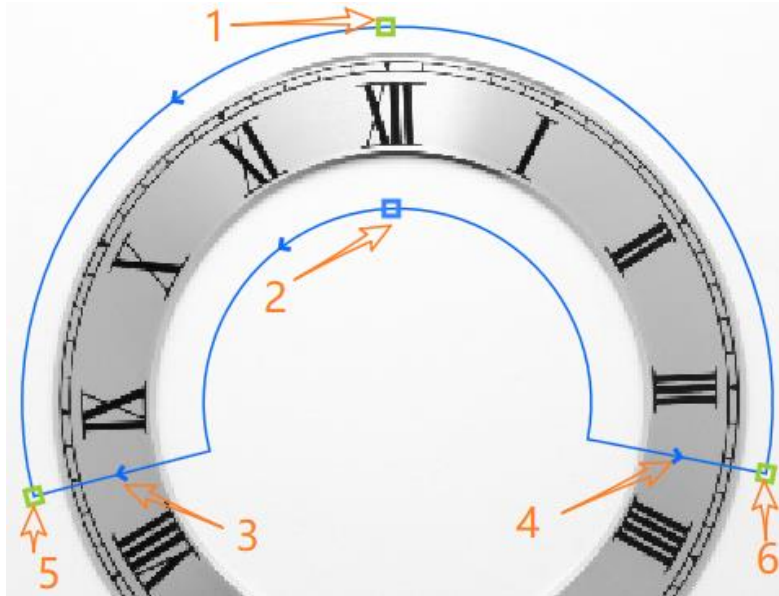


Figure 6.91 The ROI

- Arrow 1: adjust the radius of the outer circle and the inner ring will not change.
- Arrow 2: adjust the curvature of the inner and outer rings at the same time.
- Arrow 3/4: adjust the arc length of the inner and outer rings at the same time.
- Arrow 5/6: when selected one, the other is used as the fiduciary point for rotation and scaling.
- Direction of the circle: It has two direction, including counterclockwise and clockwise. The arrow direction is same of the inner and outer circles. The image above shows the counterclockwise direction, and you can adjust the direction of the circle by the arrow 3 or 4 direction.
- Radial direction: it has two directions, including from the inside to the outside and from the outside to the inside. You can adjust arrow direction via the location arrow 2.

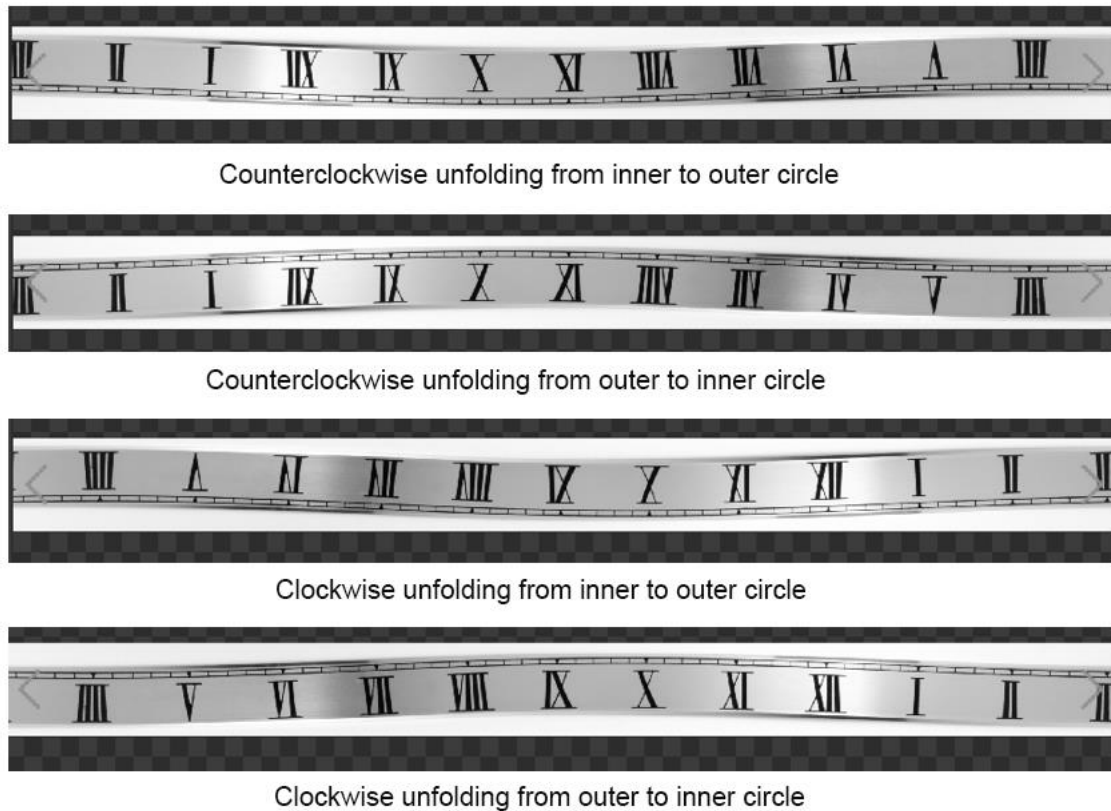


Figure 6.92 The unwrap result

6.7.12 Copy and Fill

The tool of copy and fill processes the ROI and ROI's min. circumscribed rectangle. The copy tool is to copy the image in the ROI and fill the area inside the min. circumscribed rectangle outside the ROI. The fill is to fill inside and outside the ROI, and the circumscribed rectangle. The effect of copy and fill can also be seen when the rectangle has certain angle, as shown below.



Figure 6.93 Copy and fill

- **Process Type:** you can select fill or copy.
- **Fill Value outside the Region:** it refers to the filling outside ROI region, and its range is from 0 to 255. The fill effect can be seen in the result display only when the ROI is not a rectangle or the entire image.
- **Fill Vale inside the Region:** it refers to the filling inside ROI region, and its range is from 0 to 255. This parameter is valid only when the process type is fill.

6.7.13 Frame Mean

The tool of frame mean is used to calculate the average value of the pixels of ROI region with the same size set by the multi-frame images, and output the average image of the region.

Steps:

1. Select the ROI, and perform a frame mean operation on the ROI or on the full image.
2. Click Current Image Count to accumulate. Click Current Image Ignore, you can skip the frame mean operation of the current image, as shown below.

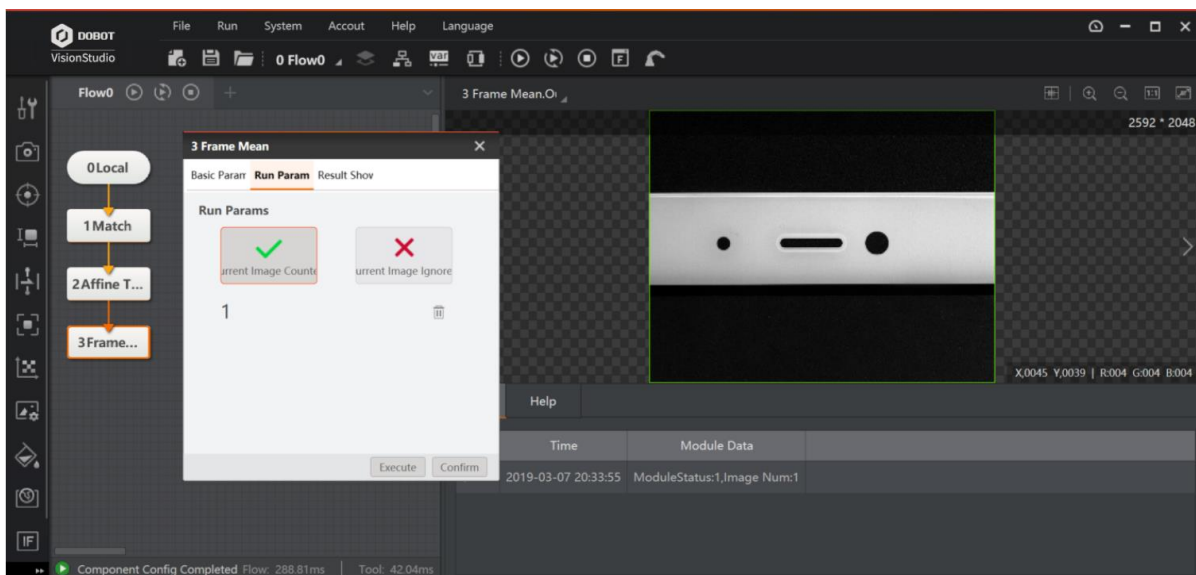


Figure 6.94 Frame mean

- **Accumulate:** it counts 100 input images at most, and output average images.
- **Ignore:** It skips the accumulated statistics for the current input image.
- **Clear Statistics Image:** it clears the average images obtained by the current statistics. Before executing statistical operation for ROI region, you need to clear the statistics image.

6.7.14 Image Normalization

The tool of image normalization is used to perform gradation transformation for input images. The effects of this tool are shown below. These effects, from left to right, are the original image, histogram equalization effect, histogram normalization, and MeanStd normalization.



Figure 6.95 Image normalization

- **Image Normalization Type:** it has three types, including histogram equalization, histogram normalization, and standard deviation normalization.
- **Histogram Equalization:** it makes image gray distribution even.
- **Histogram Normalization:** it transforms histogram distribution of inputting image that is removed left and right-end ratios into the specified histogram distribution, and the sum of the left and right-end ratios cannot exceed 1.0.
 - Left-end scale: the percentage of the left end of the histogram, and its range is from [0.0, 1.0]. 0.0 refers to the min. value of histogram gray scale, and 1.0 refers to the max. value of histogram gray scale.
 - Right-end scale: the percentage of the right end of the histogram, and its range is from [0.0, 1.0]. 0.0 refers to the max. value of histogram gray scale, and 1.0 refers to the min. value of histogram gray scale.
 - Gray value range: it refers to the gray scale range of the histogram that removes the left and right-end ratios. The min. value is the gray scale of the left end of the histogram, and the max. value is the right end of the histogram.
- **MeanStd Normalization:** it transforms the input image gray mean standard deviation to the standard deviation gray level.
 - Target mean value: it refers to the target mean value.
 - Target standard deviation: it refers to target standard deviation.

6.8 Color Processing

6.8.1 Color Extraction

The color space of the color extraction tool can be RGB, HSV or HSI. According to the brightness of each channel to be extracted, you can set parameter ranges of different channels. This tool extracts the pixel part of the specified color range from the color image and outputs the 8-bit binary image, and the extraction is a process of binarization. The color extraction effect is shown below. The parameters of three channels stand for their gray value, and its range is from 0 to 255.

Steps:

1. Input an RGB format image.
2. Select extraction range of three channels in RGB space.
3. After image processing, the image after color extraction is created.



Figure 6.96 Color extraction

- **Color Space:** you can select RGB, HSV or HSI.
- **Channel Upper Tolerance Value and Lower Tolerance Value:** it sets the color extraction range of three channels.

6.8.2 Color Measure

The tool of color measure is used to measure the color information in the specified area of the color image, including the max. value, min. value, mean value and standard deviation of each channel, as shown below.

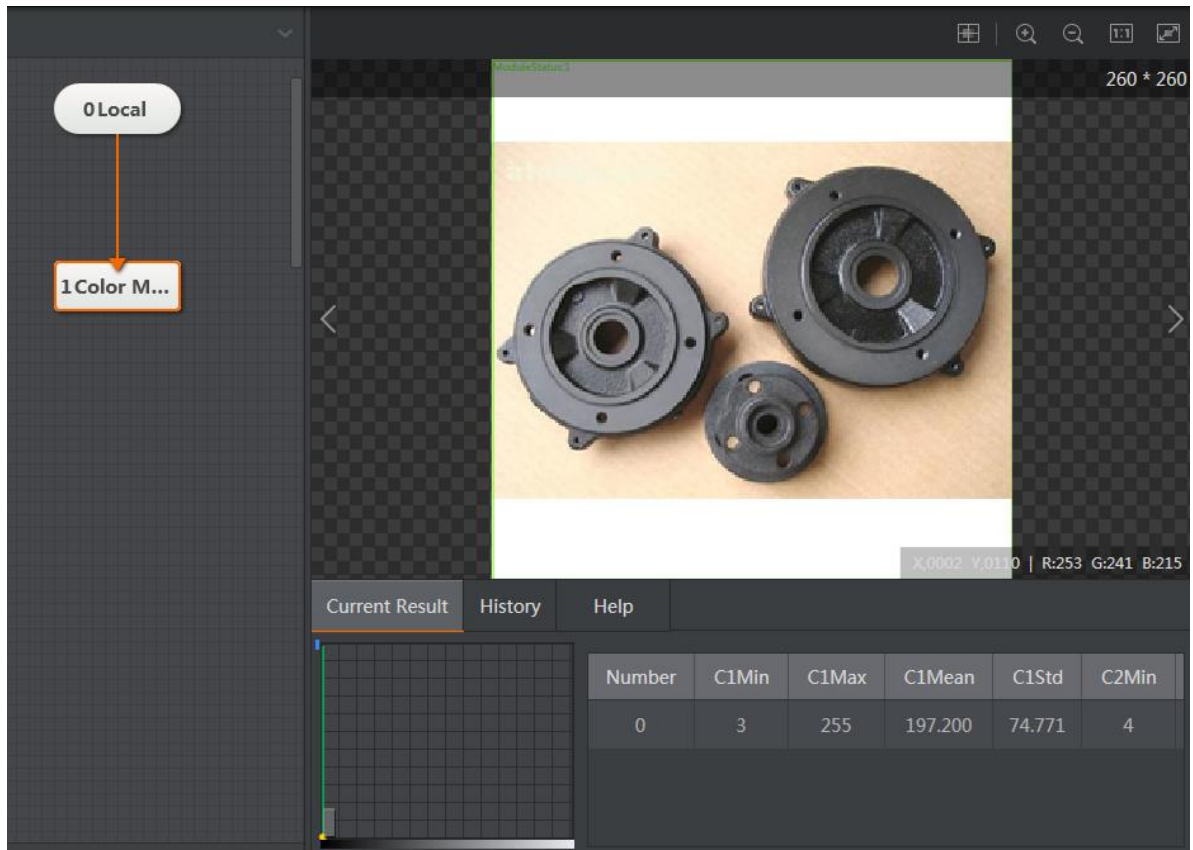


Figure 6.97 Color measure

Color Space: you can select RGB, HSV or HSI.

6.8.3 Color Transformation

After inputting a color image, you can choose transfer type, including RGB to grayscale, RGB to HSV, RGB to HSI, and RGB to YUV. The tool of color transformation transforms the color space in the specified area of the color image, and outputs the gray image of specific color channel after transformation in this region, as shown below.



Figure 6.98 Color transformation

- **Transfer Type:** this parameter sets color transformation type, including RGB to grayscale, RGB to HSV, RGB to HSI, and RGB to YUV.
- **Transfer Ratio:** it includes general transfer ratio, average transfer ratio, channel min. value, channel max. value, user-defined transfer ratio, channel R, channel B, and channel G.
 - General transfer ratio: $0.299r + 0.587g + 0.114b$. r stands for the gray value of the R channel, g stands for the gray value of the G channel, and b stands for the gray value of the B channel.
 - Average transfer ratio: $(r + g + b) / 3$.
 - User-defined transfer ratio: $i*r + j*g + k*b$, and r, j, k are ser-defined coefficients.
 - Channel max. value: $\max(r, g, b)$.
 - Channel min. value: $\min(r, g, b)$.
 - Channel R: $r + 0*g + 0*b$.
 - Channel B: $0*r + 0*g + b$.
 - Channel G: $0*r + g + 0*b$.
- **Displayed Channel:** when selecting RGB to HSV, RGB to HSI, and RGB to YUV as transfer type, you can select channel 1, channel 2 or channel 3.
 - When transforming RGB to HSV, you can select grayscale images among the three channels H, S and V.

- When transforming RGB to HSI, you can select grayscale images among the three channels H, S and I.
- When transforming RGB to YUV, you can select grayscale images among the three channels Y, U and V.

6.9 Defect Detection

6.9.1 OCV

The tool of OCV compares the target image with the standard image to detect whether printed characters and patterns have defects like characters missing and redundant . This tool is widely applied to packaging, printing, semiconductor and other manufacturing areas.

Defect detection is a process which compares the target image with the standard image, and thus standard images need to be trained before defect detection. The specific process is shown below.

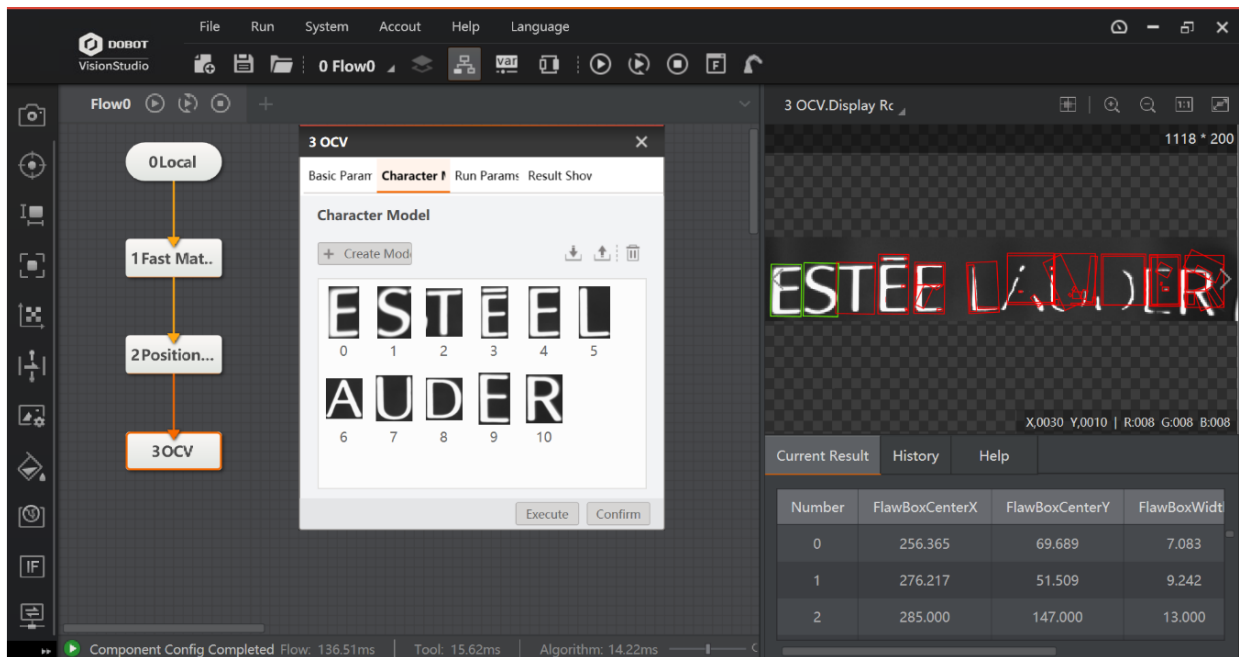


Figure 6.99 OCV

Steps:

1. Load standard image file in local image interface, and enable **Auto Switch**.

Note: The image file should at least have 10 to 20 standard images.

2. Select **Draw** as **ROI Creation** in OCV basic parameter interface.

Note: The ROI area should contain the target character to be detected.

3. Click **Create Model** in character model interface to extract character, as shown below.

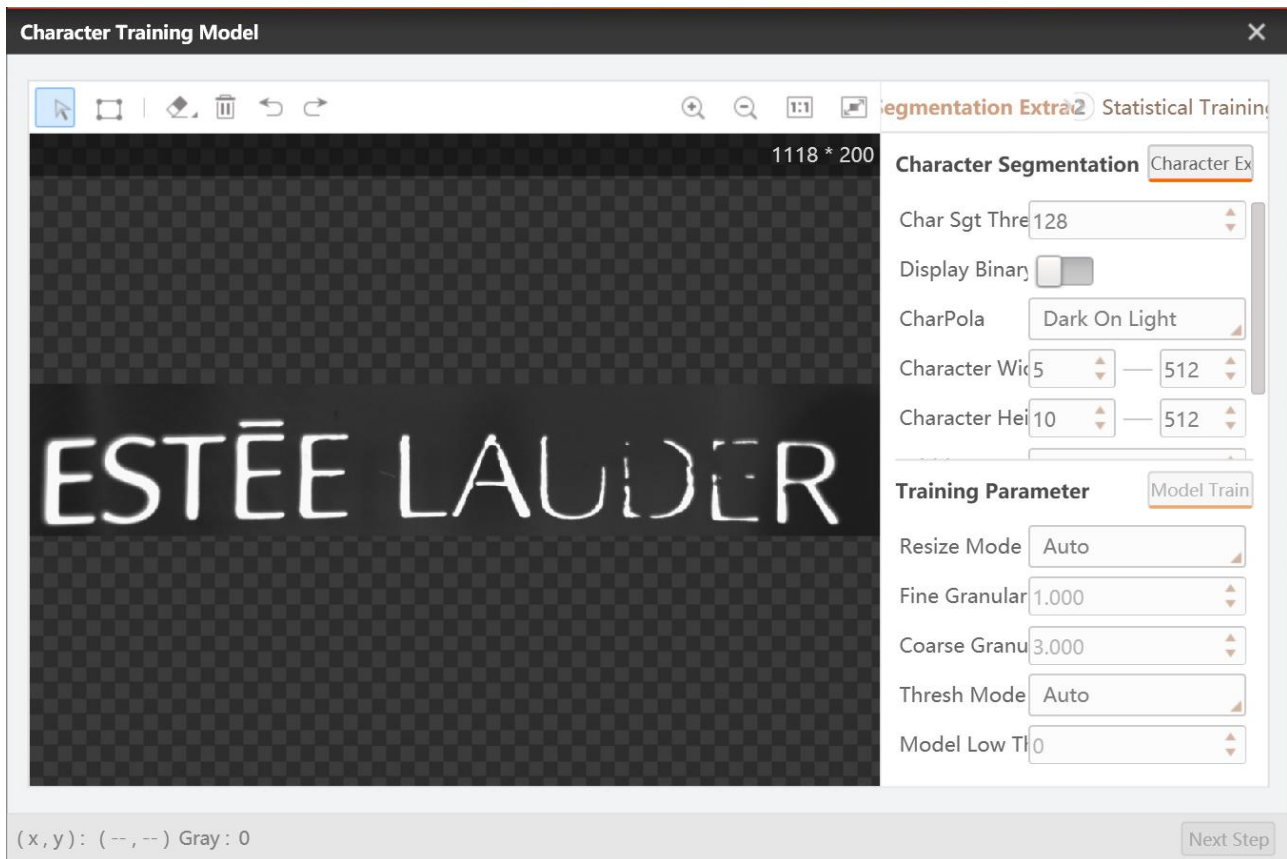


Figure 6.100 Create model

- **Character Segmentation Threshold:** it is the critical value of character segmentation. If the value is too small, extra feature graphics will be easily obtained. If the value is too large, the target graphics will be lost. It is recommended to set it as 128.
- **Display Binary Picture:** After this parameter is enabled, the function of image binarization will be executed, and the difference between the background image and the character gray scale will be more obvious.
- **Character Polarity:** it has two types, including dark on light and light on dark.
- **Character Width/Height Range:** it sets the width and height of target character.
- **Width/Height Increment:** the added value of width and height of target character area.

The tools in area 1, from left to right, are moving image, creating rect mask, mask list, clear, cancel and back respectively. When the target character is omitted, you need to create self-defined rect character extraction frame. When certain features are redundant, you can set mask region.

4. Execute statistical training and generate character templates: in statistical training module, select rational **OCV Model Type**, and click **Accumulate**.

If there is image that does not require statistical training, click **Ignore**. The number of training images should be at least 10 to 20, and the more training images, the more accurate the results will be, as shown below.

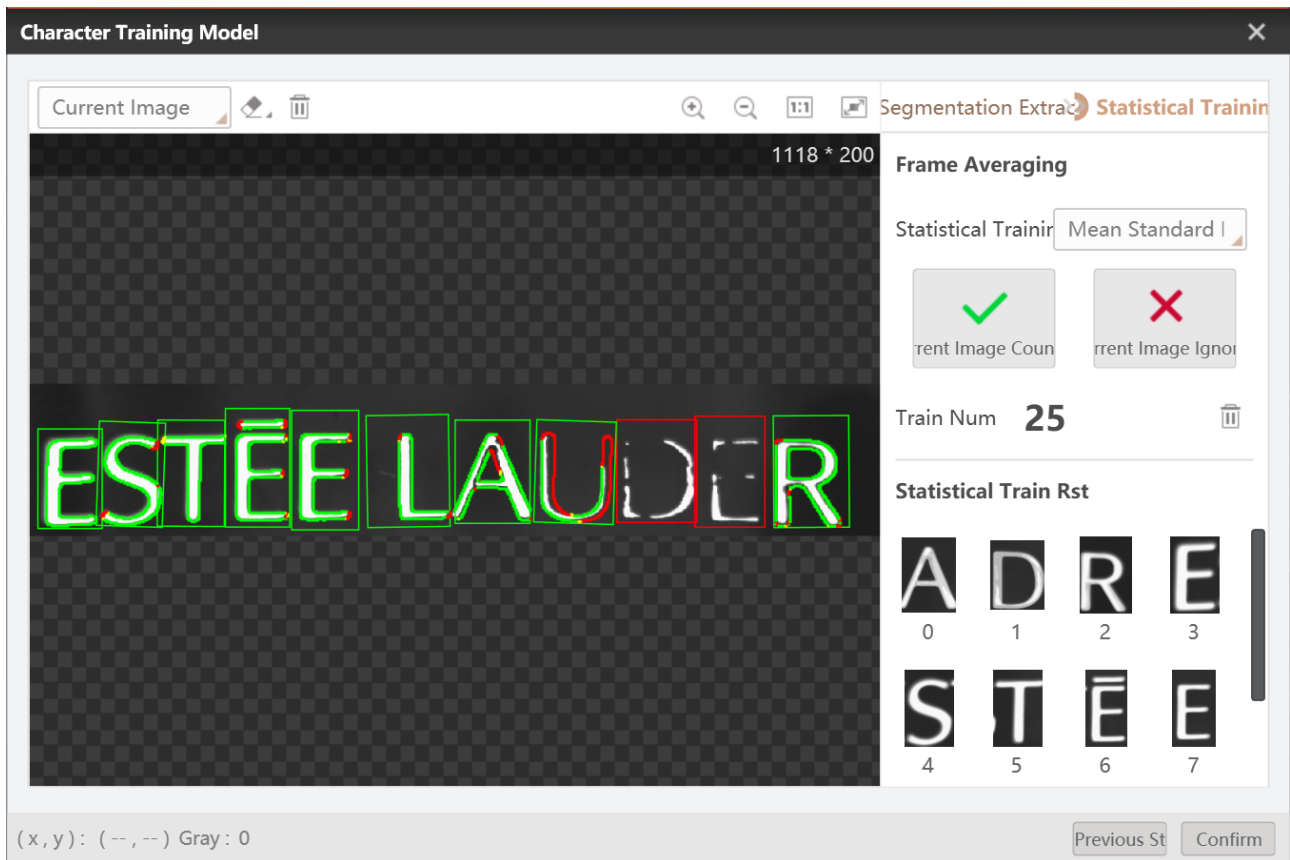


Figure 6.101 Train model

- **Current Image:** it displays the current image.
- **Average Image:** select one character in OCV model list, and select average image to see the training result, as shown below.

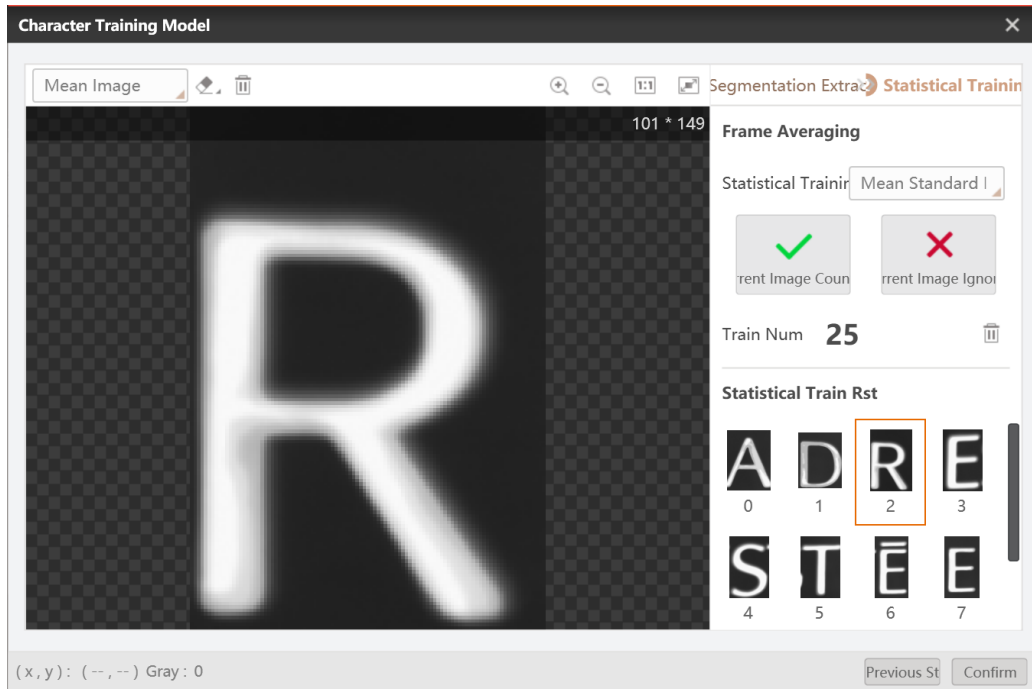


Figure 6.102 Train model

- **Mean and Std:** compare the average image and the mean and std image are obtained through multiple images with the normal images that are calculated with the high and low ratio offset coefficients.
- **Correlation:** defect judgment is executed by calculating the gradation correlation between the average image and the input detected image.

5. The running parameter setting is shown below.

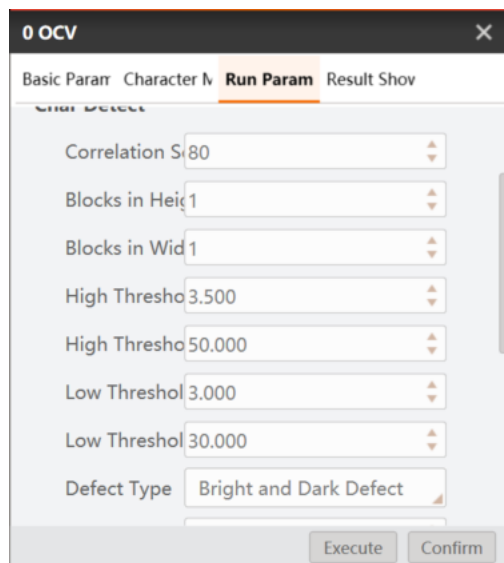


Figure 6.103 Set running parameter

- **Normalization Types:** it has four normalization modes, including histogram equalization, histogram normalization, standard deviation normalization and not deal.
- **Correlation Score Threshold:** if the target below the score threshold is considered to have defects.
- **Block in Width/Height:** it executes block score for characters. If block in width and block in height are both 1, and then it means not block.
- **High Threshold Ratio/Tolerance:** this ratio parameter is used to increase the foreground gray value of the high threshold image, and the tolerance parameter is used to increase the overall gray value of the high threshold image. These two parameters correspond to the bright defect detection. When the target image foreground has false bright defect, it is recommended to increase the high threshold ratio and high threshold tolerance value. When the target image foreground has omitted bright defect, it is recommended to reduce the high threshold ratio and high threshold tolerance value.
- **Low Threshold Ratio/Tolerance:** this ratio parameter is used to reduce the foreground gray value of the high threshold image, and the tolerance parameter is used to reduce the overall gray value of the high threshold image. These two parameters correspond to dark defect detection. When the target image foreground has false dark defect, it is recommended to increase the high threshold ratio and high threshold tolerance value. When the target image foreground has omitted dark defect, it is recommended to reduce the high threshold ratio and high threshold tolerance value.
- **Defect Type:** it has three types, including bright defect, dark defect, and bright and dark defect.

Area Threshold: the detected image compares with the high and low threshold images and obtain a difference binary image. If the blob that is larger than the area threshold image in the binary map is considered as a defect.

- **Width/height Tolerance:** it is used to expand the initial character area and obtain the range of the character's accurate positioning.

6. Click Run Once to run the solution, and the defective graphic will be detected, as shown below.

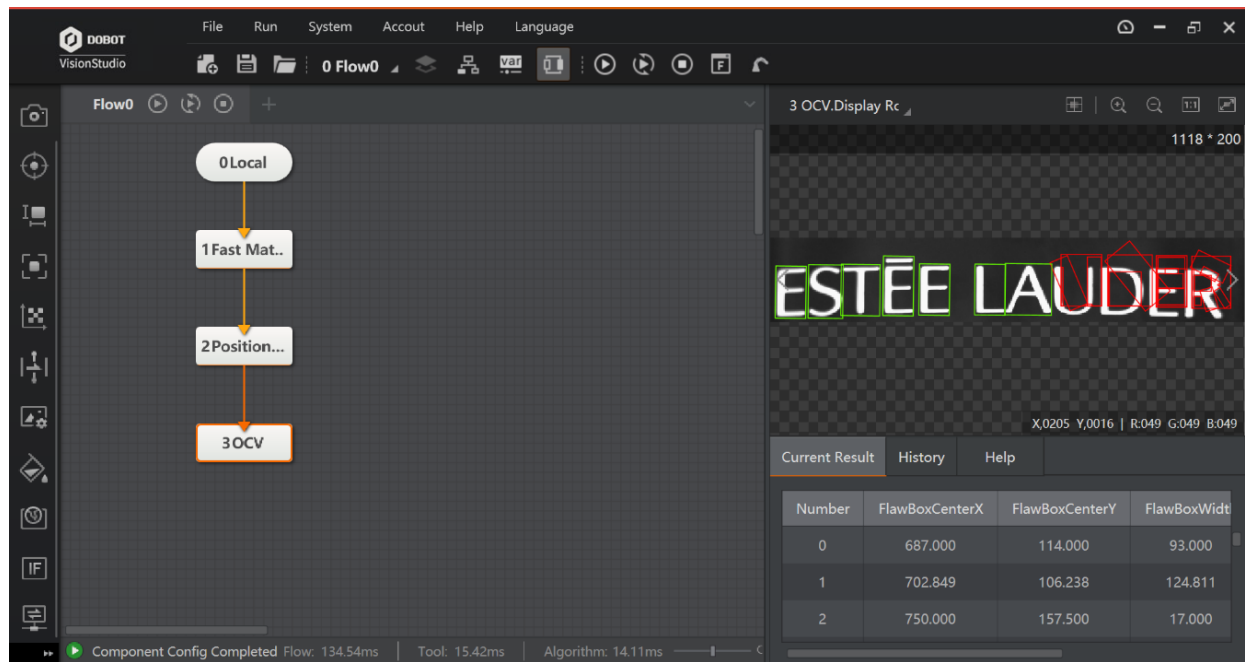


Figure 6.104 Run solution

Note: the foreground refers to the obvious target defects, and the areas that encircle defects are background.

6.10 Logical Tools

6.10.1 If Module

The if module determines whether the input data satisfies the condition. If it is satisfied, the operation interface will display OK character. Otherwise, the NG character is displayed, as shown below.

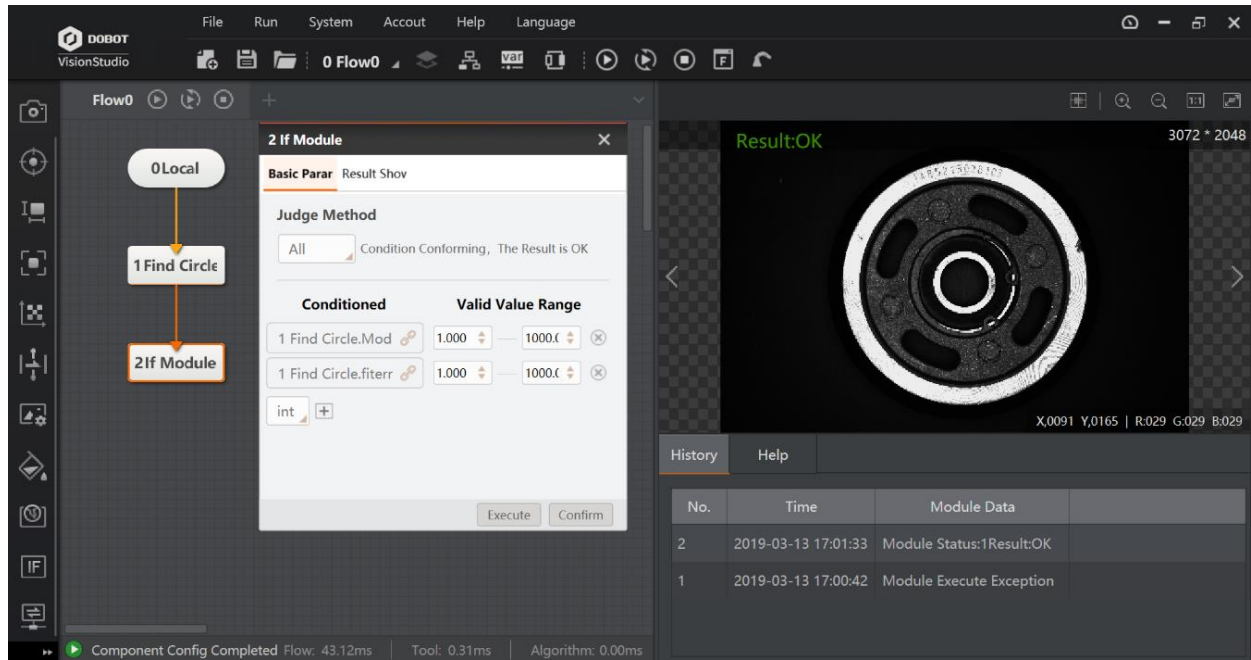


Figure 6.105 The detection result

- **Judgment Method:** you can select all or any, and the result is OK.
- **Condition:** you can add the judgment data of int type or float type, and set the valid value range. If the result selected meets the range from the min. value to the max. value, and the result is OK, otherwise is NG.

6.10.2 Branch

The tool of branch can set the input conditions for different branch modules according to the actual requirements. When the input condition is this value, the branch module is executed. The input value only supports integers and it does not support strings. If you need to enter string format, you need to use character branch, or use character recognition and branch modules. When you need to determine the subsequent branch work according to the template matching status, you can set the input condition as the template matching status, and set the condition value of the branch module, as shown below.

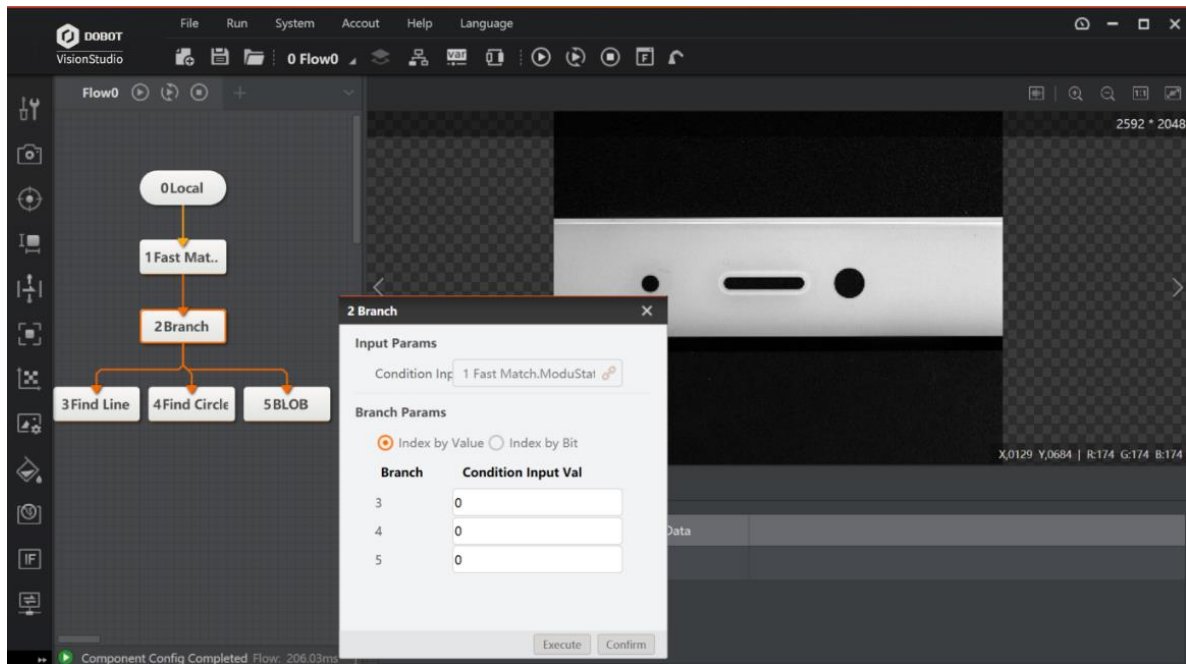


Figure 6.106 Branch

- **Condition Input:** you can select inputting parameters.

- **Branch Parameters:** you can set the index by value or by bit.

- Index by value is to compare the condition input value with the setting value after the module ID index. If these two values are the same, and then the branch is executed. Otherwise, the branch is not executed.

- Index by bit is to convert the condition input value into binary system in the background, and the binary sequence corresponds to the bit sequence of the module ID. When the bit is 1, the branch is executed (multiple is available at a time). Otherwise, the branch is not executed.

6.10.3 Branch String Comparison

The branch string comparison module detects the input characters. If the detection passes, and data transmission is executed, as shown below.

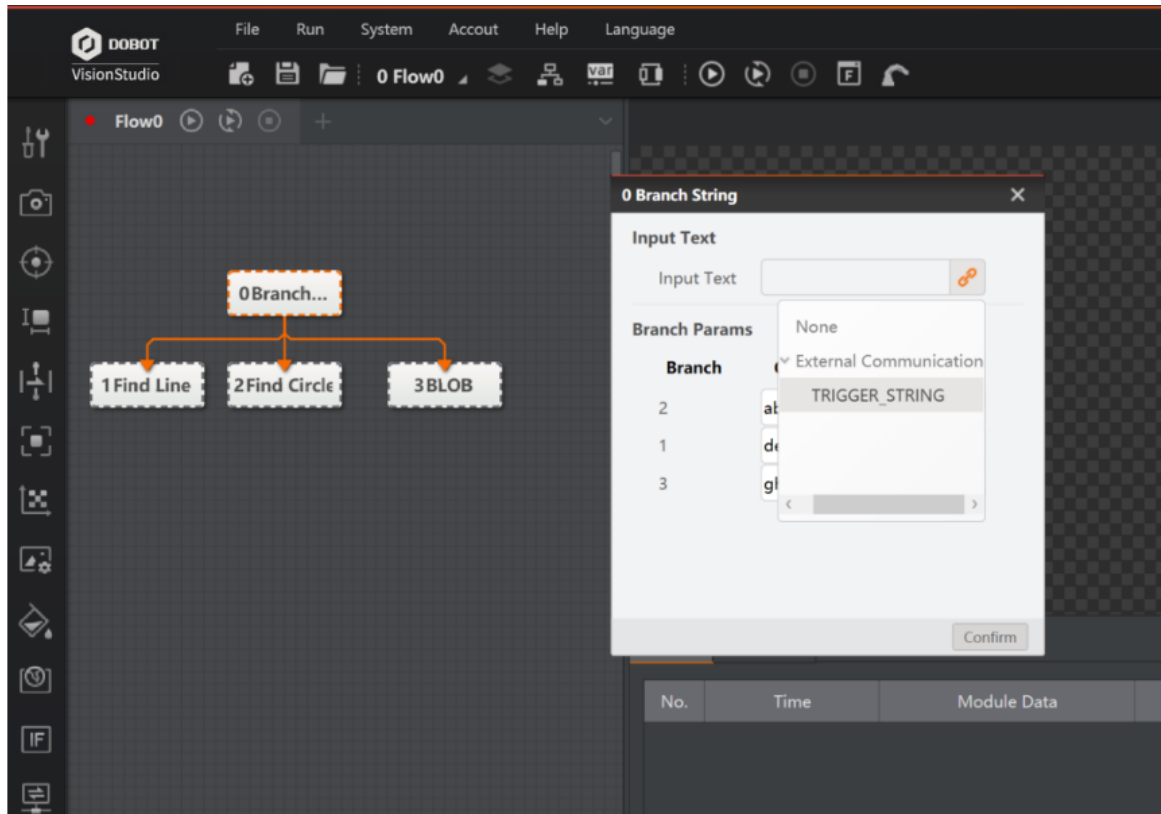


Figure 6.107 Branch string comparison

- **Input Text:** you can select the input text.
- **Branch Parameters:** it sets the condition input value, and it selects branch module in accordance with the comparison of input text and condition input value.

6.10.4 Save Text

The tool of saving text composes strings according to the text format edited by the user and saves them to a local file. The input variable is string type only, and the max. length is 4095 bytes. After running this tool, it creates cache file first. When the cache file reaches the file capacity, it then creates txt text, as shown below.

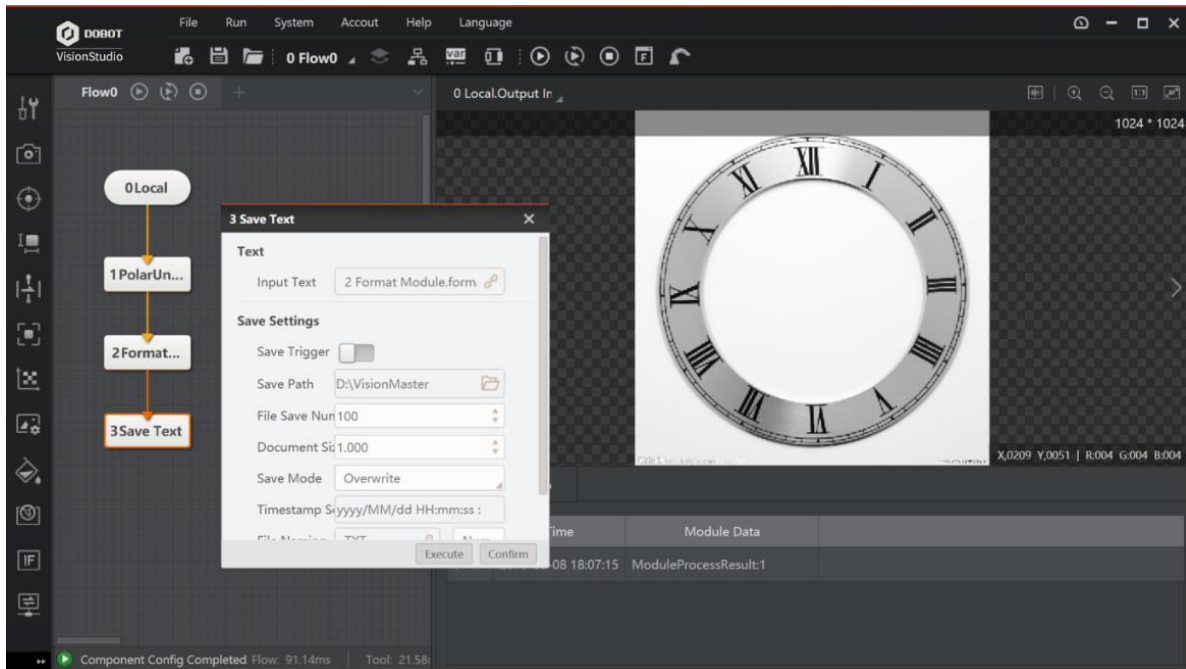


Figure 6.108 Save Text

- **Input Text:** you can select the input text.
- **Save Trigger:** when saving, it adds limitation condition. When the trigger variable reaches the save condition, it saves the text.
- **Save path:** you can self-defined the location where the text is saved.
- **File Save Number:** the max. number of files stored.
- **Document Size:** the size of each file.
- **Save Mode:** it sets the max. quantity of storage or the way the text is processed when the disk space is insufficient. It has two modes, including overwrite and stop saving.
- **Timestamp Setting:** the strings that are added to the text each time when strings are saved.
- **File Naming:** it sets the name of text file.

6.10.5 Logic Module

The logic module includes calculation type and calculation data, as shown below.

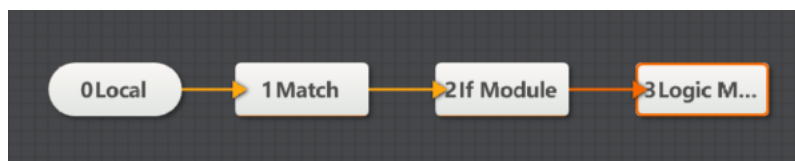


Figure 6.109 Logig module

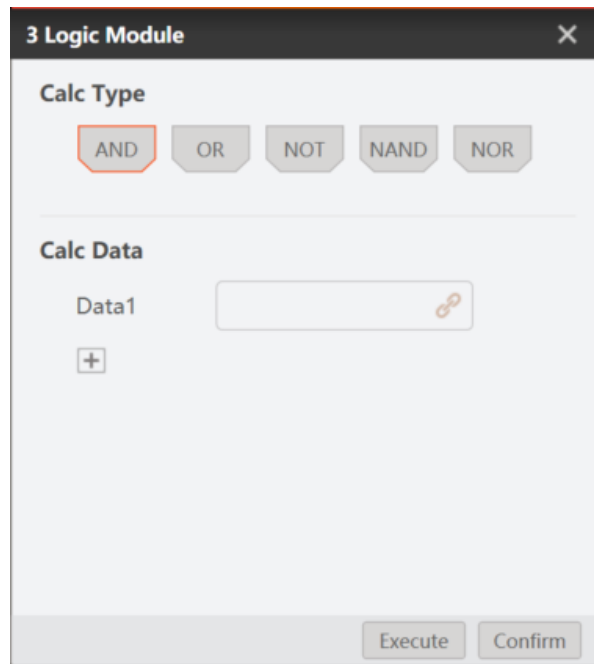



Figure 6.110 Set parameter

- **Calculation Type:** you can select and, or, not, nand and nor.
- **Calculation Data:** you can select data source to execute logic calculation.

6.10.6 Format Module

The tool of format module is used to integrate data, format the data into string, and output the string. The max. length of format is 256 bytes. Click  on the right side to select the data you want to format, and you can select multiple data. Set the appropriate interval between different data in the data frame. You can select the appropriate output end symbol as needed. After the setting is complete, you can click Format Check to verify whether the format meets the requirements, as shown below.

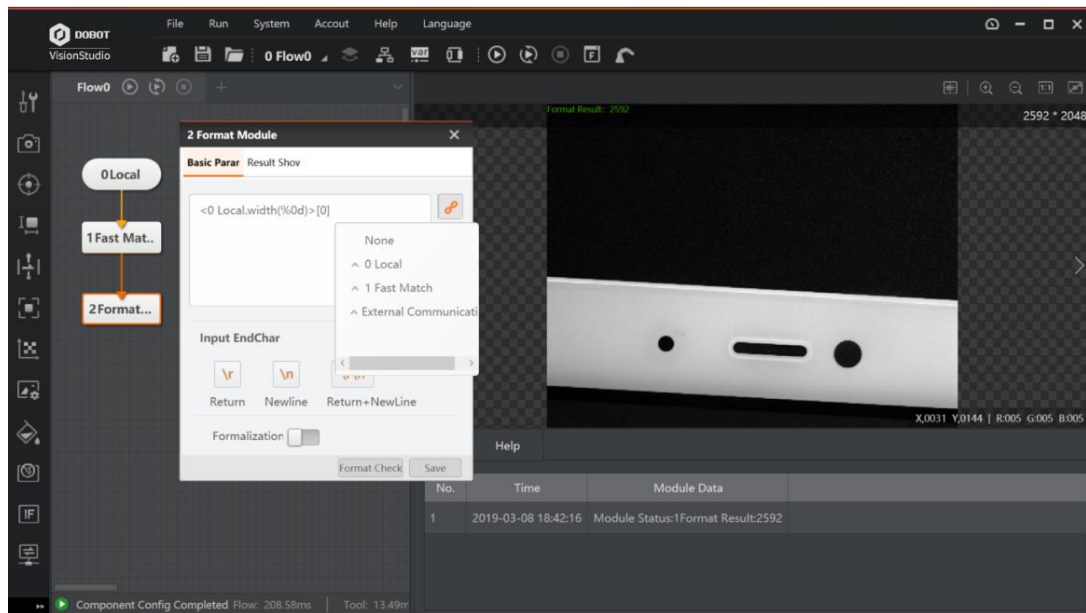


Figure 6.111 Format module

- **Input End Character:** \r return, \n newline, \r\n return and newline. Click the corresponding input end symbol, and then click the mouse at the corresponding position to select the data to be input.
- **Formalization:** in branch structure, it formats branch result and outputs. If one or more branches are not running, you can enable this parameter to output the value of the normal branch.
- **Format Check:** it is used to verify the setting format and ensure correct data format.
- **Save:** click it to save the settings.

Additional instructions for format module:

- **Format 1:** <filter(%s)>

It means that the input variable is not formatted and output as a string, and (%s) can be omitted.

- **Format 2:** <filter(%0d)>

It means that the input variable is rounded out by format, and then it is output. The 0 here is a number that can replace other positive integers, and it means the length of the output format. When the actual input value integer bit length is less than the given length, the insufficient bit will be complemented by 0. When the actual input digit integer bit length is longer than the given length, and then it is output as the actual integer length.

- **Format 3:** <filter (%0.0f)>

It means that the input variable is output as a floating point format. The 0.0 here means that the integer and decimal that need to be reserved, and replace them with other positive integers as

needed. The integer number is the same as the previous section. If the actual length of the decimal exceeds that of given decimal, and the actual length is rounded to the given length.

- **Format 4:** <filter >[*{,}]

* refers to output all input arrays by format. {,} is a separator meaning that each formatted value is separated by ",". The characters between "[]" are the actual separators, and the user can modify them to other strings as needed. Separators can be omitted, and then separated by "," by default.

- **Format 5:** <filter >[a,b,c{,}]

a, b, c refer to take the value of the subscript a, b, c in the array for outputting by formatting. a, b, c need to be replaced with the actual subscript value. If the user enters a subscript that exceeds the array range, and then format will automatically ignore the error subscript.

- **Format 6:** <filter > [s~e{,}]

s~e means that the starting and ending range of the value subscript. s or e can be omitted, that is [~e{,}] or [s~{,}], which respectively represent the value of the subscript 0 to the subscript e of the array and the value of the subscript s to the end of the array for formatting output. A null character will be output when the start index exceeds the actual input array range. If the end subscript exceeds the array range, and then the actual array should prevail.

6.10.7 Calculator

The tool of calculator supports mixed calculation of multiple inputs. You can customize parameters or select the module data for calculation, as shown below.

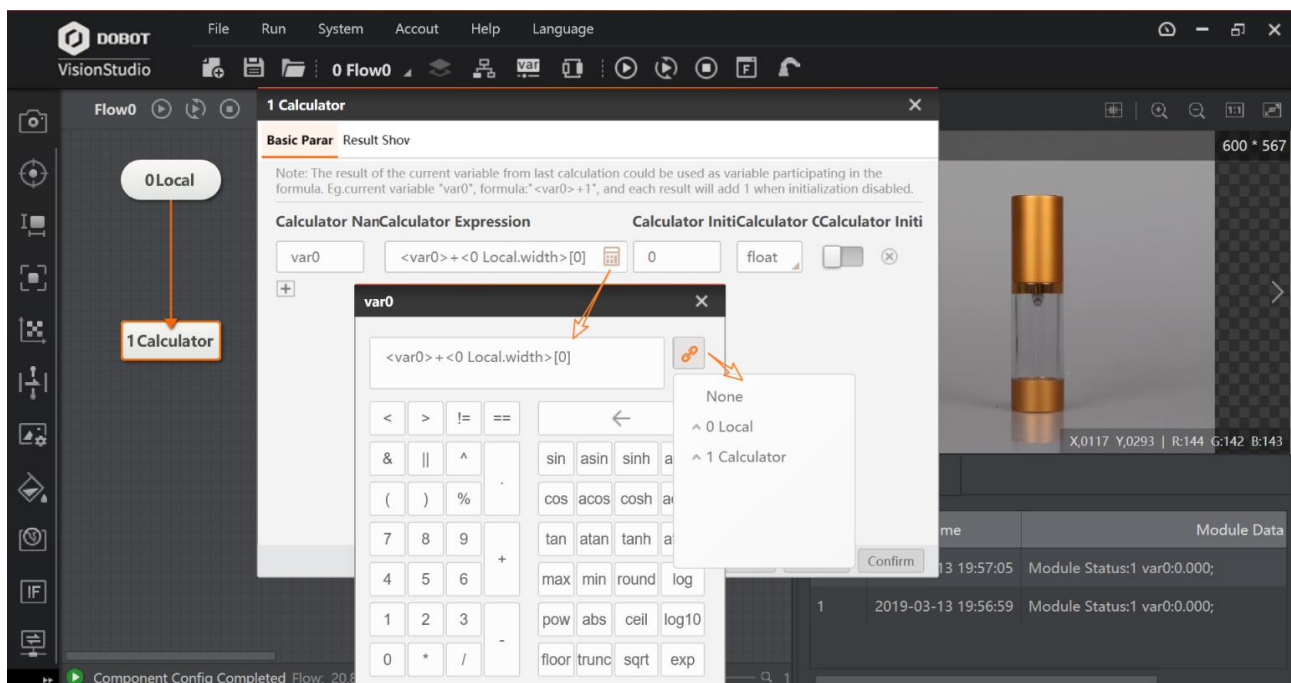


Figure 6.112 Calculator

Parameter Settings:

- **Reset:** it restores the variable calculation to its initial status and click it to take effect.
- **Execute:** it executes a variable calculation.
- **Calculator Initial Value/Calculator Initialization:** the variable calculates the initial value. When enabling calculator Initialization, each time the process execution resets the variable to the initial value of the setting.
- **Check Formula:** it verifies that the configured formula is correct or not.
- **Save:** it saves the variable calculation formula.
- **Calculator Name:** it sets the name of the variable.
- **Calculator Expression:** it edits the expression of variable calculation.
- **Calculator Type:** it sets two types, including int and flot.
- **Variable Calculation:** it adds the variables that need to be calculated and sets the formula as needed.
- **Array calculation:** the calculator module supports calculation of a single value in an array variable, and the format is <filter> [index]. The index needs to be replaced with a specific array subscript, which supports variable subscription. If the subscript exceeds the actual range of the array, it may lead to calculation failure.
- **Cumulative Calculation:** in the calculator module formula, the var0 represents the calculated value of the current system. When enabling calculator Initialization, each time the process execution resets the variable to the initial value of the setting. Otherwise, the var0 value is the previous operation result. The users can accumulate, multiply var0 value via enabling calculator Initialization.

6.10.8 String Comparison

The tool of string comparison is used to compare the input character with the configured character. if the comparison result is same, the system will output the corresponding index value. If the comparison result is not same, the system will output the first index item in the text list which exactly matches the input text. The index value can be manually modified, its range is from 0-2147483647 (long int), with max. 32 indexes. In general, this tool is used together with the branch module, as shown below.

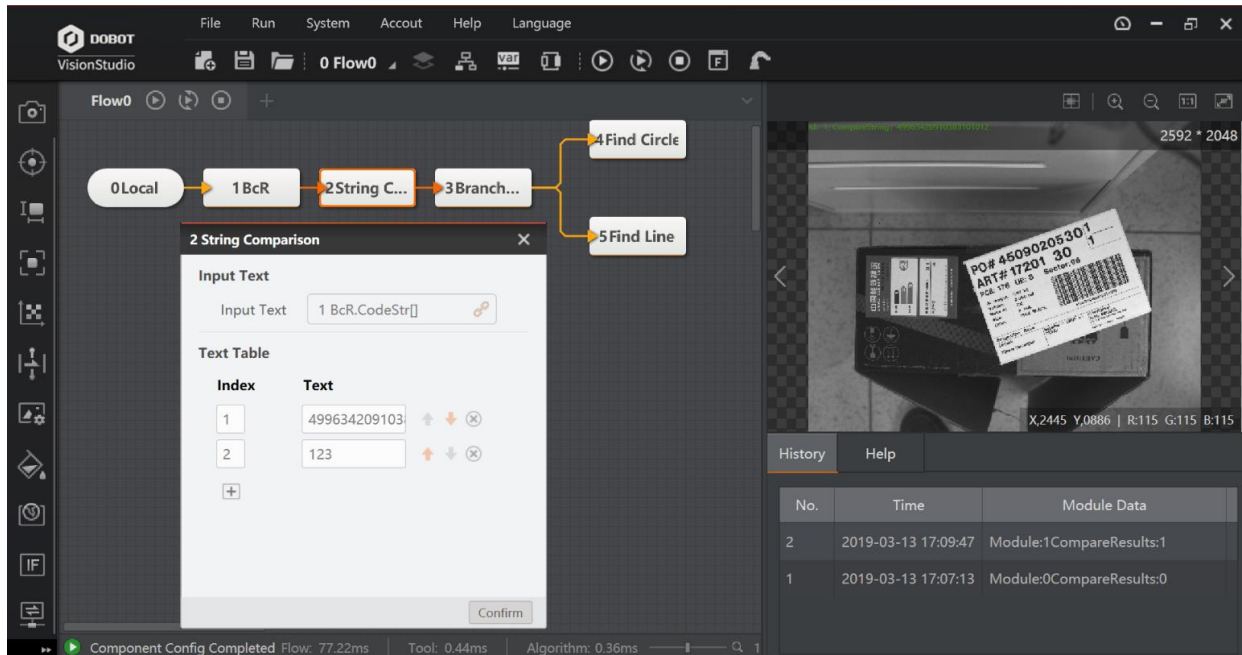


Figure 6.113 String comparison

6.10.9 Script

The tool of script is used to process complex data. It saves and loads the contents of the script that has been written, and the script file format is suffixed with cs. There is no limitation for the length of script code, and script code supports import and export. After the import is completed, the script tool will execute a compilation. If the compilation fails, the system will show exception. If the compilation is successful, the system will operate according to the new code. The parameter settings are shown below.

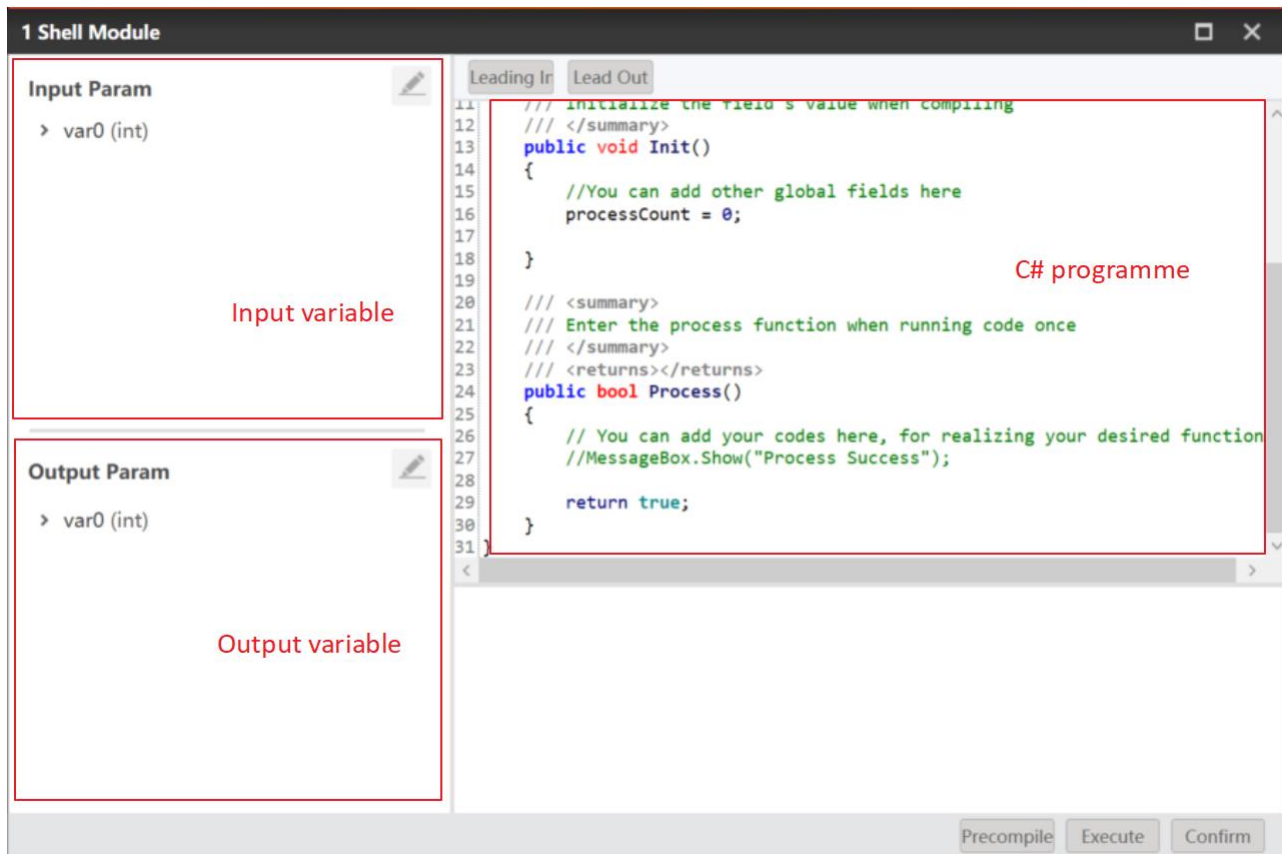


Figure 6.114 Script

The function interfaces include data acquisition interface, data output interface, and debugging related interface.

- **Data Acquisition Interface:**

– GetIntValue

Table 6.1 GetIntValue

Function Description	Obtain INT variable value		
Function Method	int GetIntValue(string paramName, ref int paramValue)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
Output	paramValue	int	Variable Value

Return Value	Success: 0 Exception: error code rather than 0.
--------------	--

– GetFloatValue

Table 6.2 GetFloatValue

Function Description	Obtain Float variable value		
Function Method	int GetFloatValue (string paramName, ref float paramValue)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	String	Variable Name
Output	paramValue	Float	Variable Value
Return Value	Success: 0 Exception: error code rather than 0.		

– GetStringValue

Table 6.3 GetStringValue

Function Description	Obtain String variable value		
Function Method	int GetStringValue (string paramName, ref string paramValue)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	String	Variable Name
Output	paramValue	String	Variable Value
Return Value	Success: 0 Exception: error code rather than 0.		

– GetIntArrayValue

Table 6.4 GetIntArrayValue

Function Description	Obtain INT array variable		
Function Method	int GetIntArrayValue(string paramName, ref int[] paramValue, out int arrayCount)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
Output	paramValue	int []	Variable Value
	arrayCount	int	Array Quantity
Return Value	Success: 0 Exception: error code rather than 0.		

– GetFloatArrayValue

Table 6.5 GetFloatArrayValue

Function Description	Obtain float array variable		
Function Method	int GetFloatArrayValue(string paramName, ref float[] paramValue, out int arrayCount)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
Output	paramValue	float []	Variable Value
	arrayCount	int	Array Quantity
Return Value	Success: 0 Exception: error code rather than 0.		

– GetStringArrayValue

Table 6.6 GetStringArrayValue

Function Description	Obtain string array variable		
Function Method	int GetStringArrayValue(string paramName, ref string[] paramValue, out int arrayCount)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
Output	paramValue	string []	Variable Value
	arrayCount	int	Array Quantity
Return Value	Success: 0 Exception: error code rather than 0.		

● Data Output interface:

– SetIntValue

Table 6.7 GetIntValue

Function Description	Set INT variable value		
Function Method	int SetIntValue(string key, int value)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
	paramValue	int	Variable Value
Output			
Return Value	Success: 0 Exception: error code rather than 0.		

– SetFloatValue

Table 6.8 SetFloatValue

Function Description	Set float variable value		
Function Method	int SetFloatValue (string key, int value)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
	paramValue	float	Variable Value
Output			
Return Value	Success: 0 Exception: error code rather than 0.		

– SetStringValue

Table 6.9 SetStringValue

Function Description	Set string variable value		
Function Method	int SetStringValue (string key, string value)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
	paramValue	string	Variable Value
Output			
Return Value	Success: 0 Exception: error code rather than 0.		

– SetStringValueByIndex

Table 6.10 SetStringValueByIndex

Function Description	Set string variable value		
Function Method	int SetStringValueByIndex(string key, string value, int index, int total)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	string	Variable Name
	value	string	Variable Value
	index	int	Array index
	total	int	Array element quantity
Output			
Return Value	Success: 0 Exception: error code rather than 0.		

– SetIntValueByIndex

Table 6.11 SetIntValueByIndex

Function Description	Set string variable value		
Function Method	int SetStringValueByIndex(string key, string value, int index, int total)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	String	Variable Name
	value	Int	Variable Value
	index	Int	Array index
	total	Int	Array element quantity

Output			
Return Value	Success: 0 Exception: error code rather than 0.		

– SetFloatValueByIndex

Table 6.12 SetFloatValueByIndex

Function Description	Set string variable value		
Function Method	int SetFloatValueByIndex (string key, int value, int index, int total)		
	Parameter Name	Data Type	Parameter Description
Input	paramName	String	Variable Name
	value	Float	Variable Value
	index	Int	Array index
	total	Int	Array element quantity
Output			
Return Value	Success: 0 Exception: error code rather than 0.		

● Debugging Related Interface:

– ConsoleWrite

Table 6.13 ConsoleWrite

Function Description	Print information into Debug View
Function Method	void ConsoleWrite(string content)

	Parameter Name	Data Type	Parameter Description
Input	Content	String	Print content
Output			
Return Value	None		

The Init() function is the initialization function and the Process() is processing function. The initialization function will run during the first time only, as shown below.

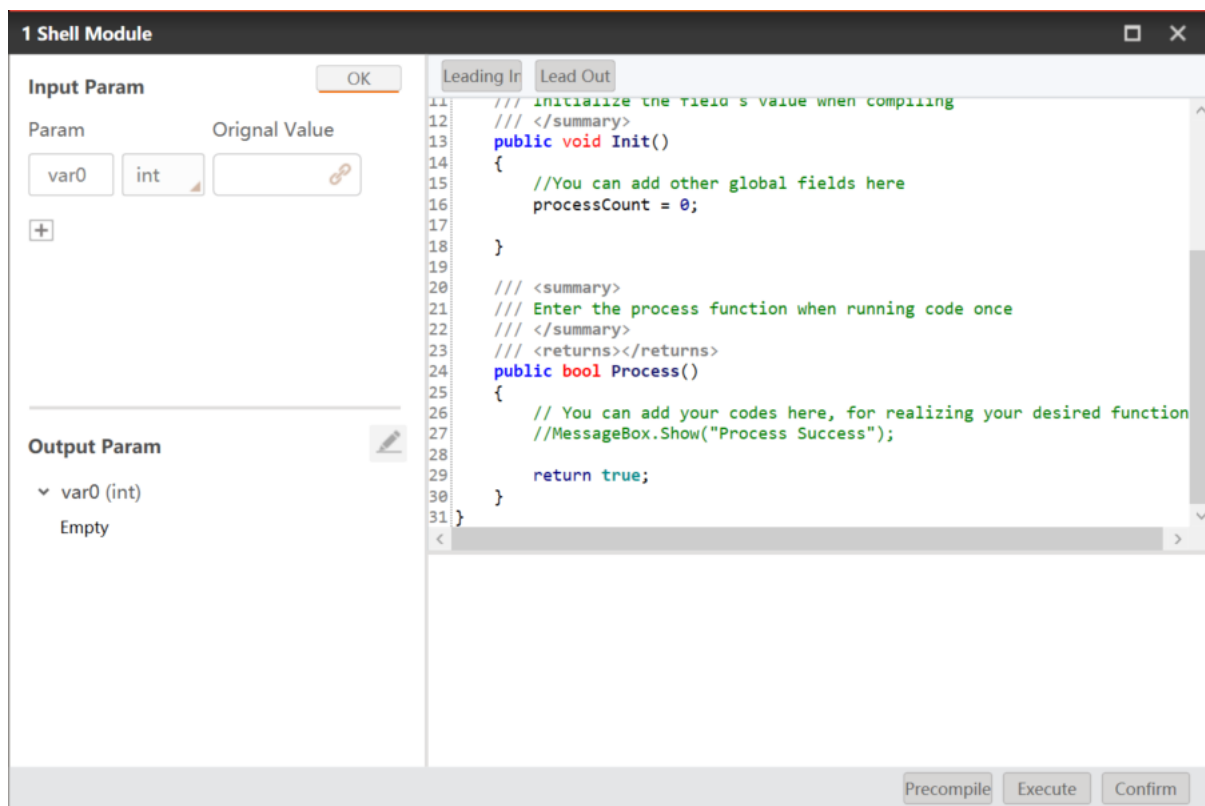


Figure 6.115 The Init function

6.10.10 Loop Control

The tool of loop control is used to run the module required by the user in loop. User need to drag the loop frame into the edit area first, then drag loop tool into loop frame, and set

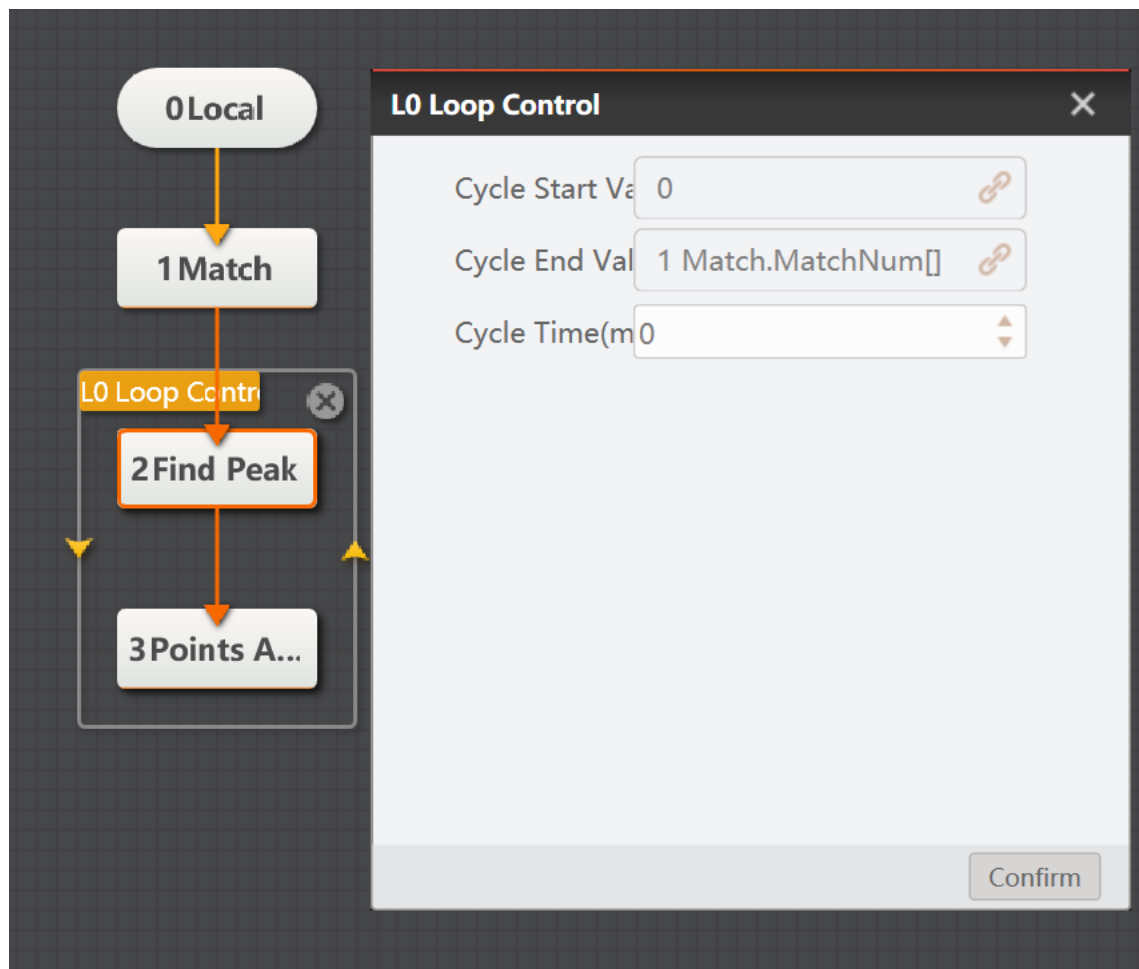


Figure 6.116 Loop control

When using this tool, you need to set loop variable in basic parameters interface, as shown below.

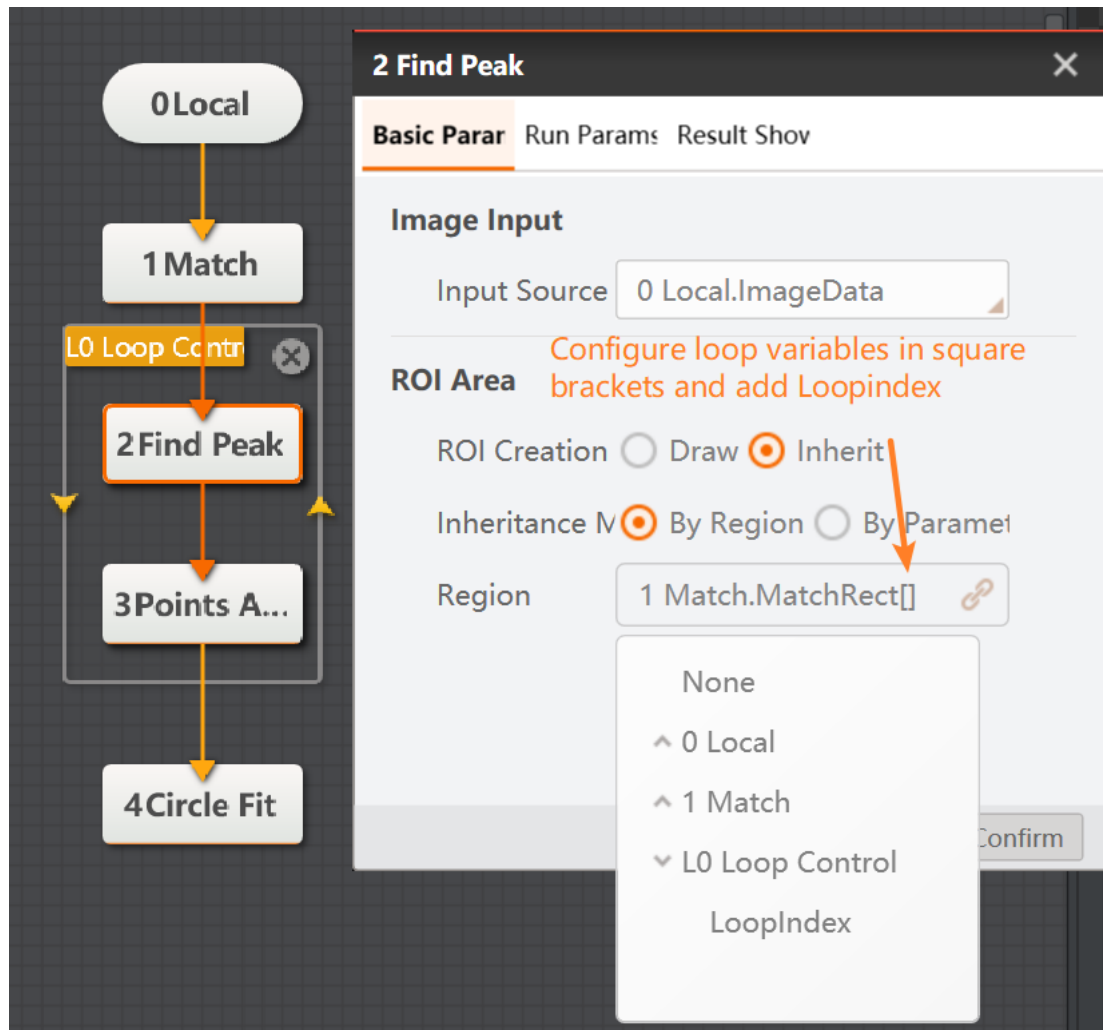


Figure 6.117 Find peak

6.10.11 Points Assemble

The tool of points assemble is used to combine the related data of other modules into points assemble with max. 16 point assemble to facilitate subsequent operation. This tool supports point or coordinate inputting, and it can be dragged into loop control module after enabling loop enable to assemble points.

- **Point Input:** you can input the data of other modules by selecting by point or by coordinate.
- **Loop Enable:** it makes points assemble in loop, and add points assemble obtained from each loop into output point assemble.

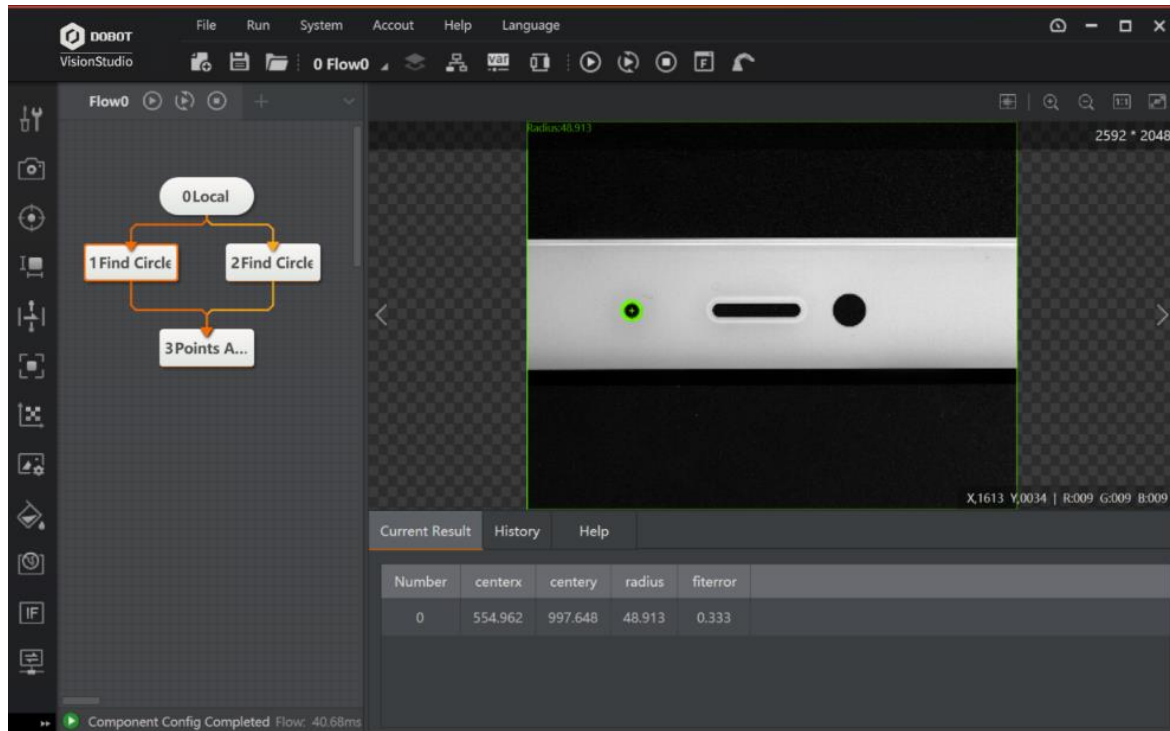


Figure 6.118 Loop enable

6.10.12 Time Consuming

The tool of time consuming is used to calculate the time from flow entering the start module to flow leaving the end module. The end time should be set later then the start time in basic parameters interface of time consuming. Otherwise, this tool cannot calculate the time consuming. Parameter settings and operation results are shown below.

- **Input Configuration:** select the start and end time.

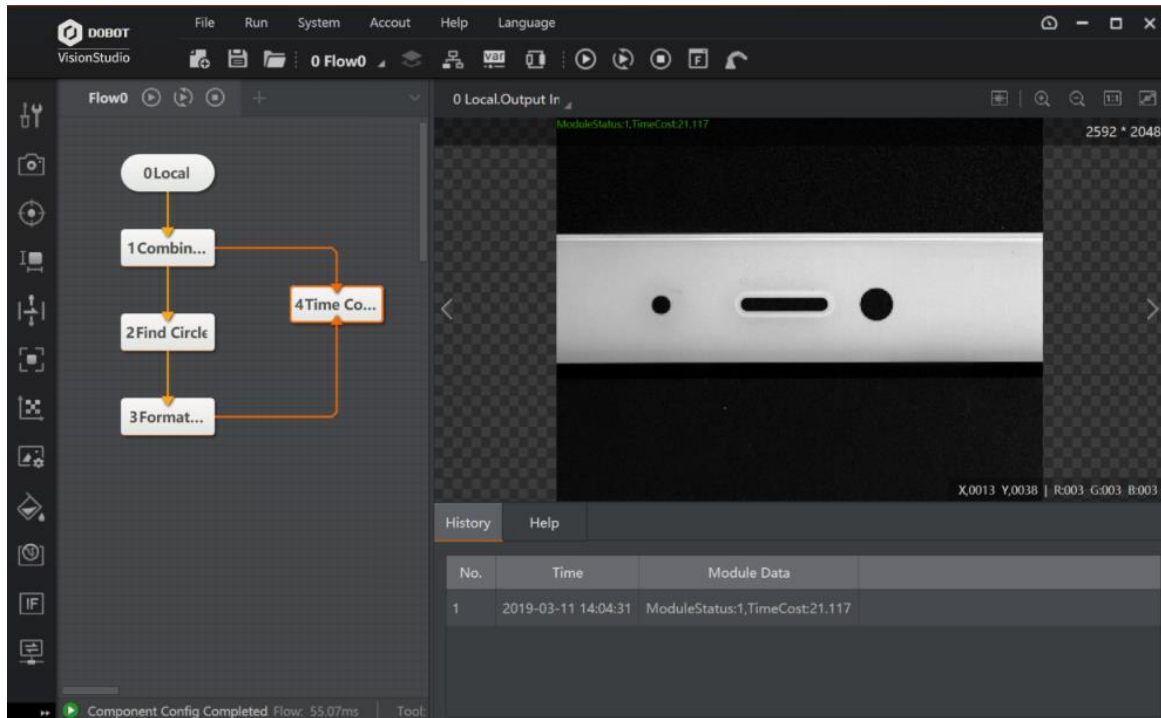


Figure 6.119 Time consuming

6.10.13 Wait Time

The tool of the wait time is used to control the wait time of program. You can set the wait time (ms).

- **Wait time Configuration:** For example, the robot moves from a point 1 to a point 2. If you set the wait time to 2000ms, it will move to point 2 after 2000ms. As show below.

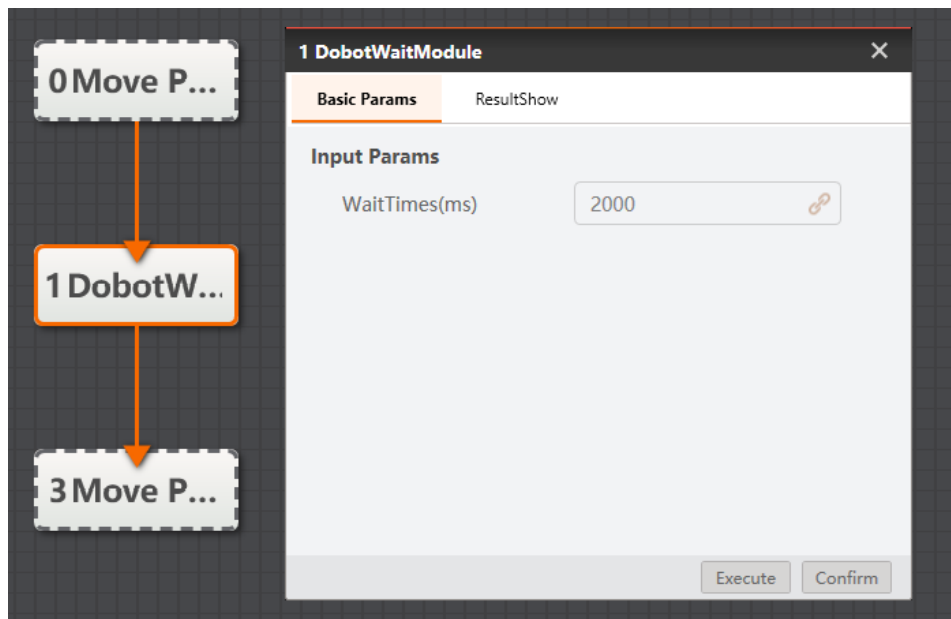


Figure 6.120 Set wait time

Chapter 7 Communication

Common communication protocols are encapsulated in modules for the output of algorithm results. Currently, receiving data, sending data, PLC communication, IO communication, and Modbus communication are supported.

7.1 Receive Data

Receiving data module is mainly used for data transmission among different flows via different media.

- **Input Configuration:** select the source of the input data.
 - **Data Source:** data can be received from data queues, communication devices or global variable.
- When the received data source is from data queues or global variables, you can set at most 16 inputs, as shown below.

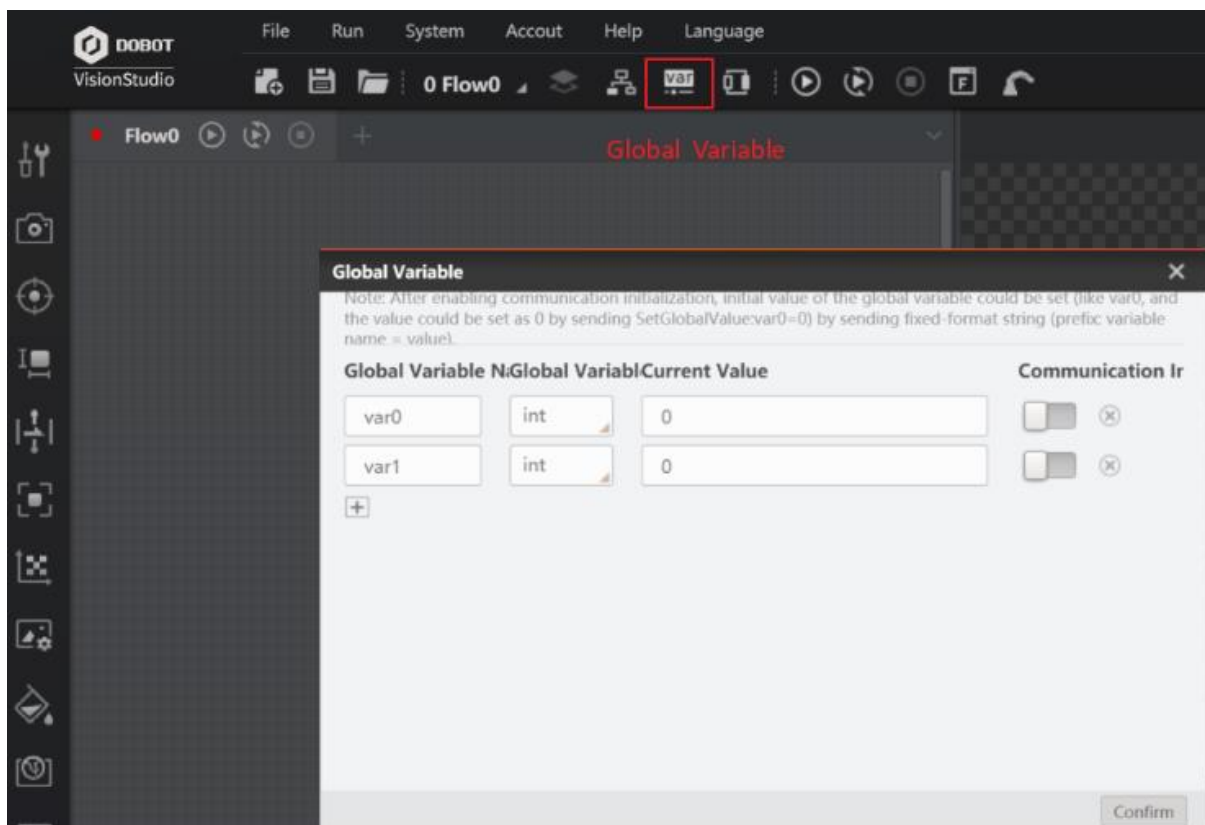


Figure 7.1 Set global variables

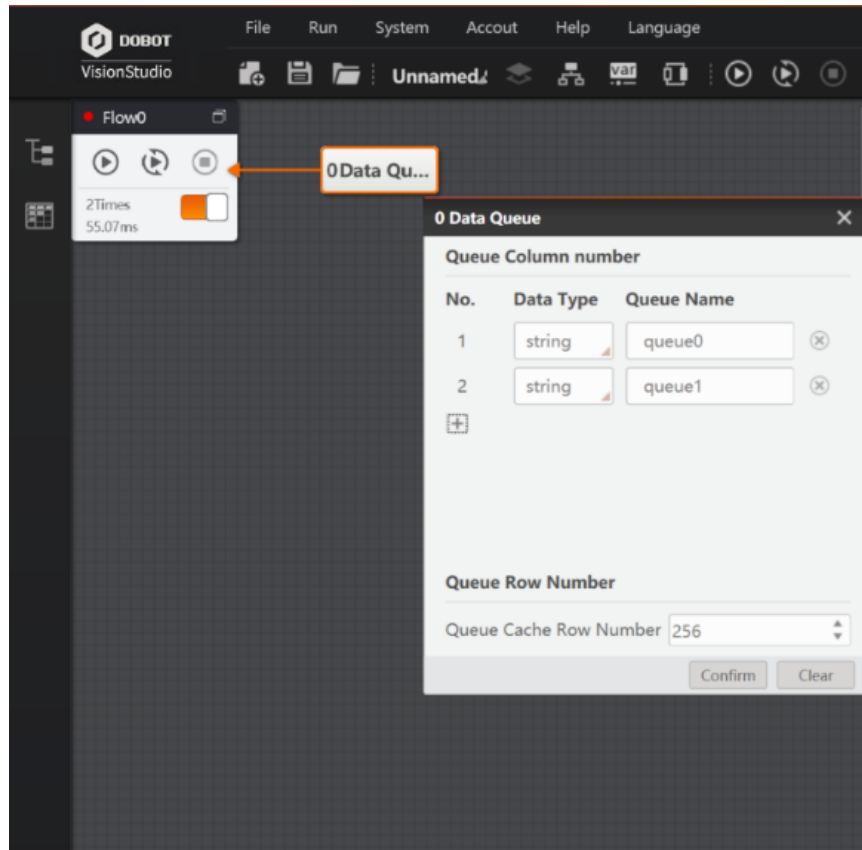


Figure 7.2 Set data queues

– TCP client, TCP server, UDP and serial port can be set in the communication management interface. When the received data is the communication device, only one input can be set. The setting of communication device is shown below.

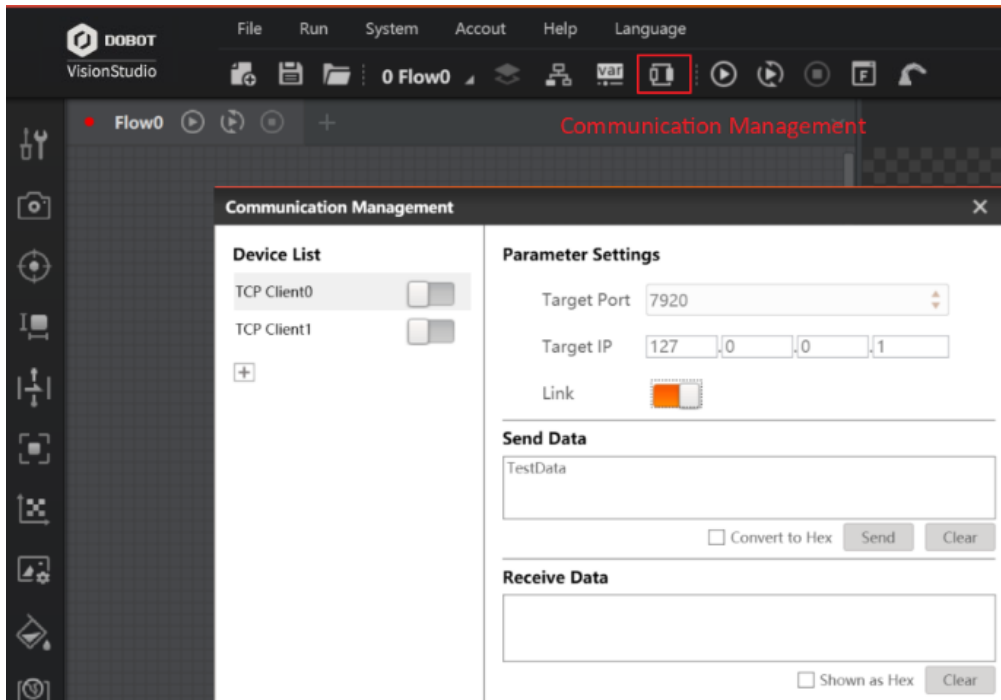


Figure 7.3 Set communication management

- **Row Number Acquisition:** you can select the row number of receiving data, and each queue can have 256 rows at most.
- **Input Data:** set the variable name to store the received data, as shown below.

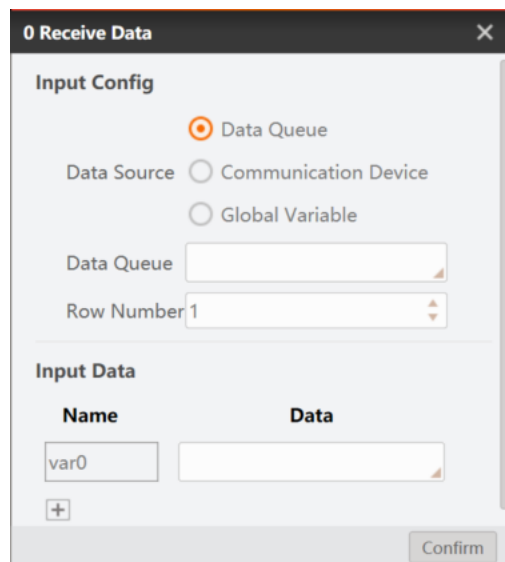


Figure 7.4 Receive data

7.2 Send Data

Data in the process can be sent to data queues, communication devices, or global variable, as shown below. At most 16 outputs can be set when setting output to data queues or global variable. Only one output can be set when setting output to communication devices.

The screenshot shows a dialog box titled "2 Send Data". It has two tabs: "Basic Param" and "Result Show". The "Basic Param" tab is active. Under the "Output Settings" section, there are three radio buttons: "Data Queue", "Communication Device" (which is selected), and "Global Variable". Below these radio buttons is a text field labeled "Communication Device" containing the text "TCP Client0". Under the "Output Data" section, there is a text field labeled "Send Data" containing the text "TRIGGER_STRING[]" with a key icon to its right. A "Confirm" button is located at the bottom right of the dialog.

Figure 7.5 Set sending data

- **Output Settings:** select output to data queue, communication device, or global variable.
- **Send Data:** select the data to be sent.

In the example, the radius obtained by find circle is sent to the global variable via sending data module. The parameter settings and running results are shown below.

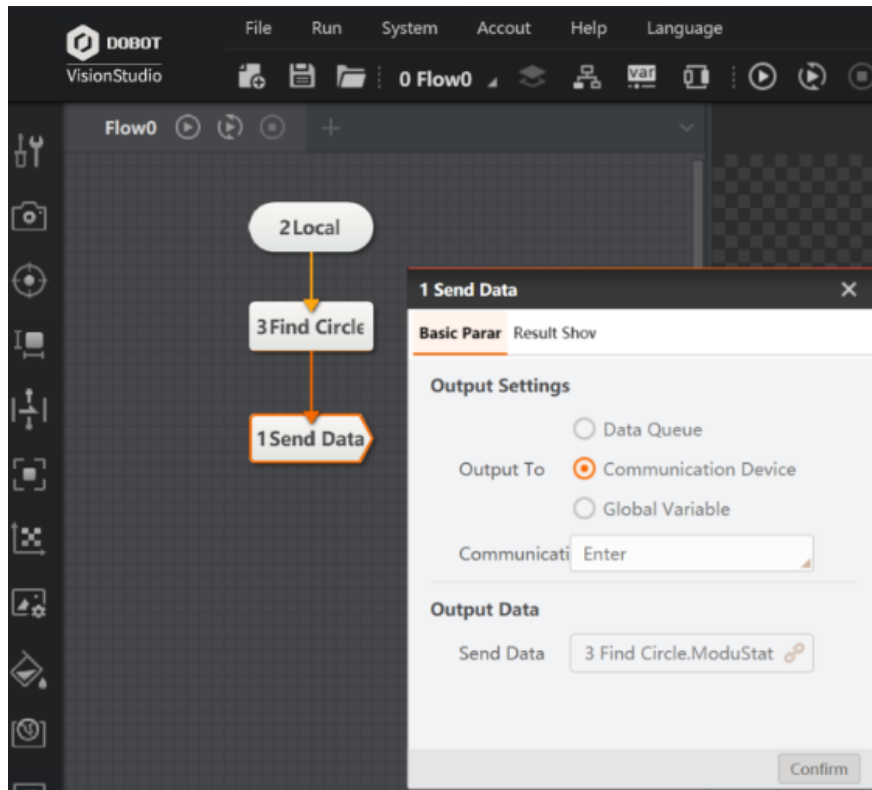


Figure 7.6 Send data

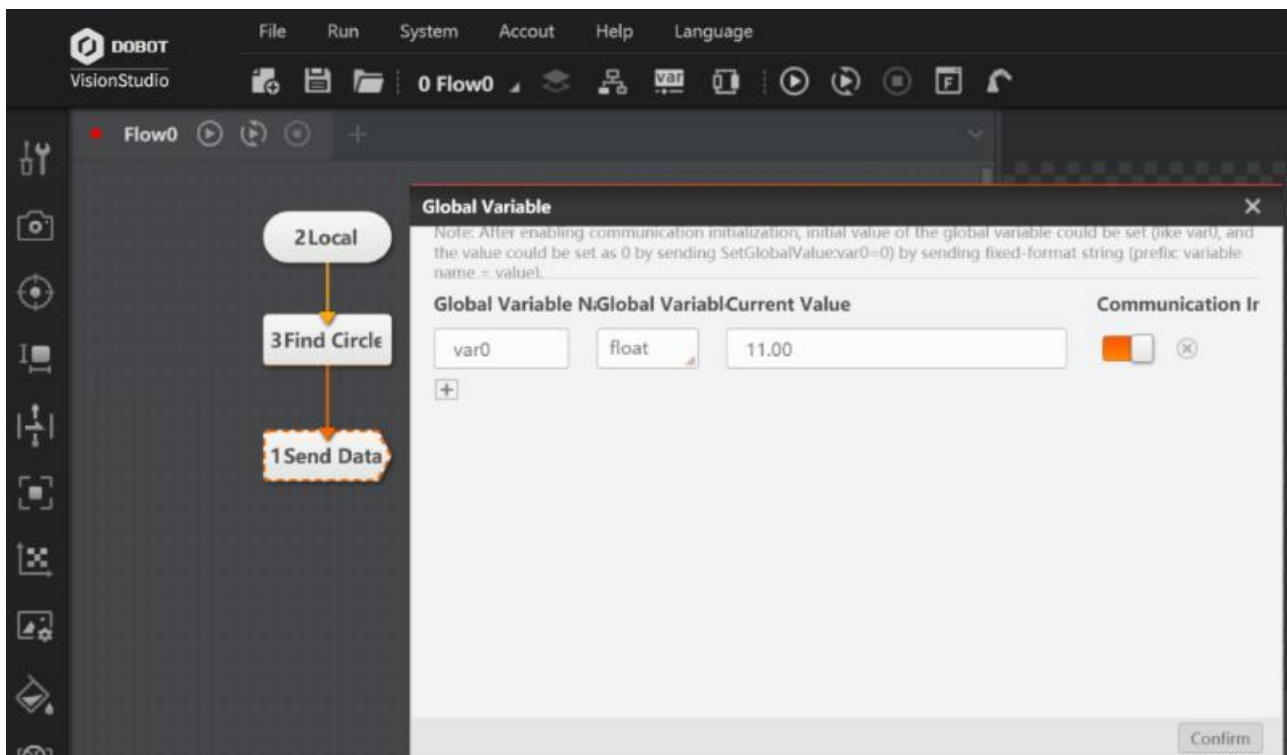


Figure 7.7 The result

7.3 TCP Communication

TCP is a connection-oriented, reliable, and byte-stream based transport layer communication. When selecting a communication device in sending or receiving data, TCP can be set. Taking sending data as an example, the specific steps are as follows:

Steps:

1. Load the TCP server or client in the communication management.
2. Set the target port and target IP as shown below. At the same time, set the corresponding IP and port at the other end of the corresponding TCP.

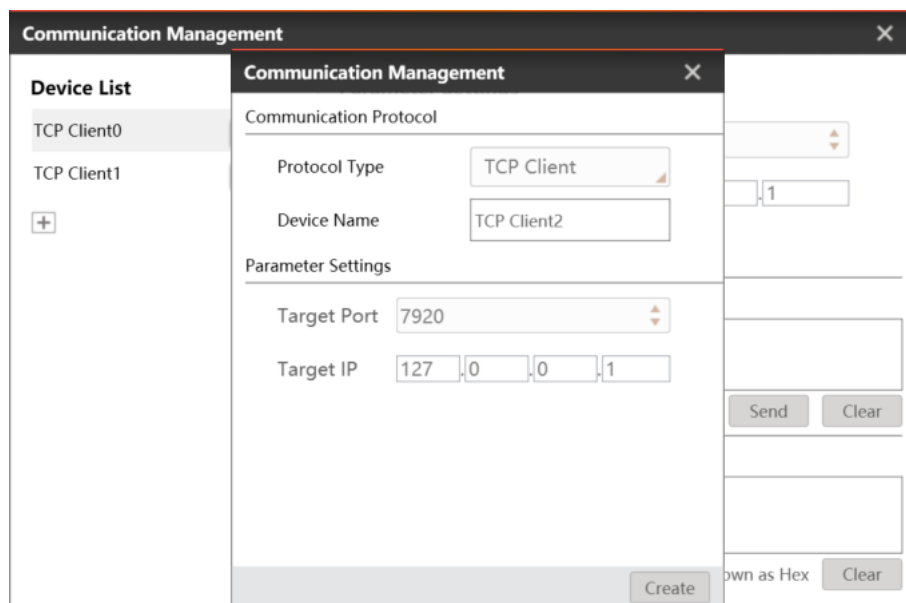



Figure 7.8 Communication management

3. Select **Communication Device** as Output to, and select corresponding device.
4. Click  in Send Data module, as shown below.

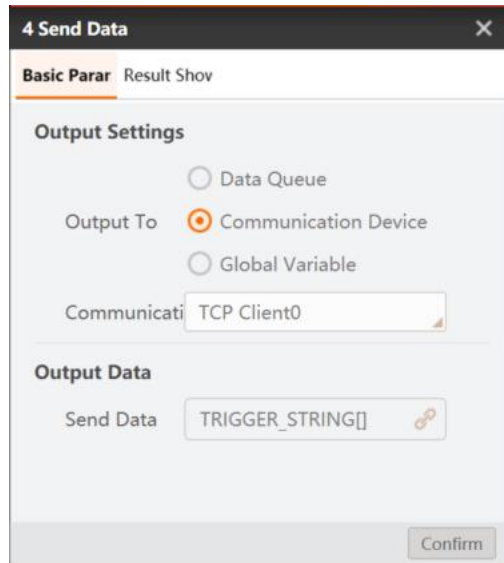


Figure 7.9 Send data

7.4 IO Communication

At present, IO communication module only supports the vision controller of Dobot. When the vision controller is not connected, the module cannot be used. Serial COM2 is used to communicate internally. If the computer does not have COM2 port or the COM2 port is occupied, the module cannot be output. Before using this module, the IO port needs to be connected correctly according to the user manual of the visual controller. When using this module, you need to set IO port condition, output level and output type. The visual controller can set the active level, as shown below.

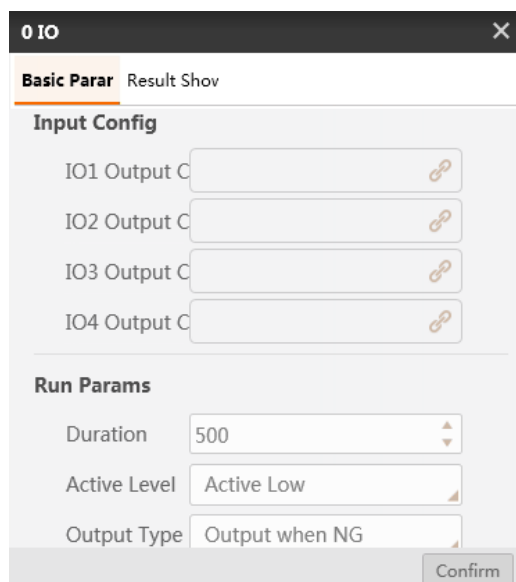


Figure 7.10 IO communication

- **Input Configuration:** it sets IO1, IO2, IO3 and IO4 output conditions. IO signal will be output only when the conditions are met, and you can subscribe to string-type data. OK or NG are valid. In general, it is set as results of condition detection or the output of scripts.
- **Running Parameters:** it sets output type, active level and duration.
- **Duration:** the time to output the active level.
- **Active Level:** it is the output level of IO interface. If the output level of IO port is set as active low, the default level of IO port is active high.
- **Output Type:** it has two types, including Output when NG and Output when OK. If the duration is 500, active level is active low, output when NG, and the module of IO1 output condition is NG, and then the output duration is 500 ms and active low. If the output type is Output when OK, the active level is the default one.

7.5 ModBus Communication

Configure standard ModBus protocol communication as needed, as shown in the figure below.

The screenshot shows a software window titled "0 Modbus" with a close button (X) in the top right corner. Below the title bar is a tabbed interface with three tabs: "Basic Parar", "Run Params", and "Result Shov". The "Basic Parar" tab is selected and highlighted with an orange underline. Under this tab, the section "Transformation Settings" is visible. It contains four configuration fields, each with a text input and a small up/down arrow icon on the right:

- Device Address: 1
- Function Code: 0x03:Read Holding Regist
- Register Address: 0
- Register Number: 1

At the bottom of the window, there are two buttons: "Execute" and "Confirm".

Figure 7.11 ModBus communication

The screenshot shows a software window titled '0 Modbus' with a close button in the top right. It has three tabs: 'Basic Param', 'Run Param' (which is selected and highlighted in orange), and 'Result Shov'. Under the 'Run Param' tab, there are two main sections. The 'Communication' section contains a 'Master-Slave' dropdown menu currently showing 'Host' and an empty 'Communication' dropdown menu. The 'Protocol Settings' section contains a 'Protocol' dropdown menu showing 'RTU' and a 'Timeout' numeric input field showing '3000'. At the bottom right of the window are two buttons: 'Execute' and 'Confirm'.

Figure 7.12 Communicaiton method

- **Device Address:** It ranges from 0 to 247 and the master-slave address is default as 0 for broadcast function.
- **Function Code:** 0x03: read holding register, 0x04: read input register, 0x06: write single register and 0x10: write multi register.
- **Register Address:** Register Address to be accessed ranges from 0 to 65535.
- **Register Number:** [1,125].
- **Master-Slave Mode:** It only supports host mode and sets Modbus module as the client.
- **Communication Device:** Serial port or TCP client can be selected. Serial port and TCP client need to be created in the communication management.
- **Protocol:** RTU or ASCII can be selected.
- **Timeout:** [1,10000], the time between sending out data and receiving the returned confirmation signal (in MS).

7.6 PLC Communication

Input Configuration: it selects the command type as read data or write data.

PLC Type: it selects the type of PLC, and it is Mitsubishi by default.

Communication Protocol Type: it has three types, including 3E, 3C- 3, 4C-5.

Message Type: it has two types, including ASCII or binary.

Soft Component Type: it has four types, including X, Y, M or D.

Soft Component Address: it sets the range from 0 to 2047.

Soft Component Points: it sets the range from 0 to 64.

Communication: it selects serial, TCPClient or TCPServer. If serial is selected, you can set port number, baudrate, data bit, parity bit and stop bit. If TCPClient is selected, you can set target IP and target port. If TCPServer is selected, you can set local IP and local port.

Communication Parameters: it sets read timeout and write timeout.

Chapter 8 Dobot Tools

8.1 Overview

Common Dobot API are encapsulated in modules for the output of algorithm results. It is convenient for users to use. Dobot tools contain the tool group of Dobot Magician and the tool group of Dobot M1.

8.2 Dobot Magician Tools

8.2.1 Conveyor Belt

Connect conveyor belt to the extended motor interface of Dobot Magician to make Dobot Magician control motor speed. You can configure based on site requirements.as shown below.

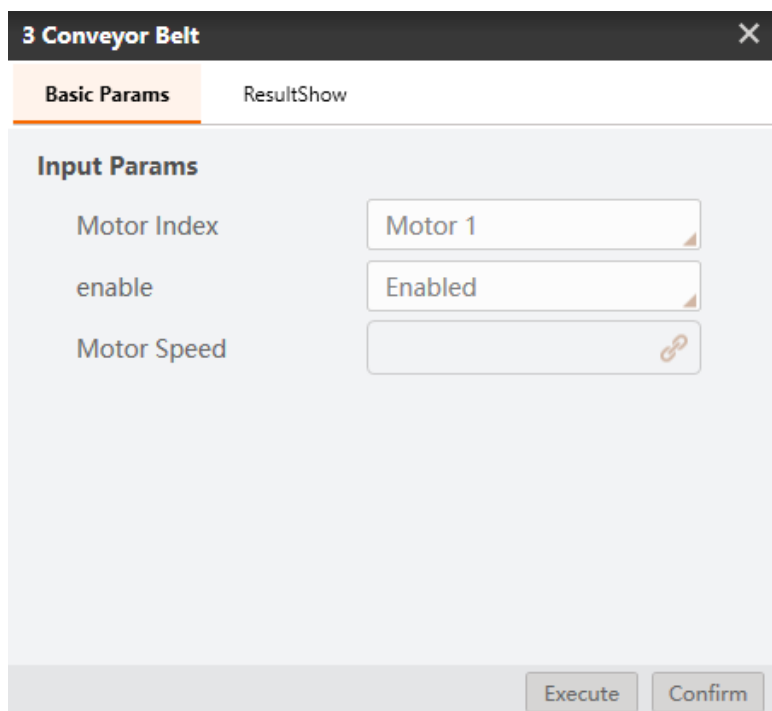


Figure 8.1 Set motor parameters

- Motor Index: it means the extended motor interface of Dobot Magician. Value range: Motor 1 or motor 2. Motor1 is that conveyor belt is connected to Stepper1 interface. Motor2 is that conveyor belt is connected to Stepper2 interface.
- Enable: enable motor.
- Motor speed: Set motor speed. Value range: -20000~20000 pulse/s.

8.2.2 Sliding Rail

After connecting sliding rail to Dobot Magician, Dobot Magician will run with Sliding rail which can extend the workspace of Dobot Magician. As shown below.

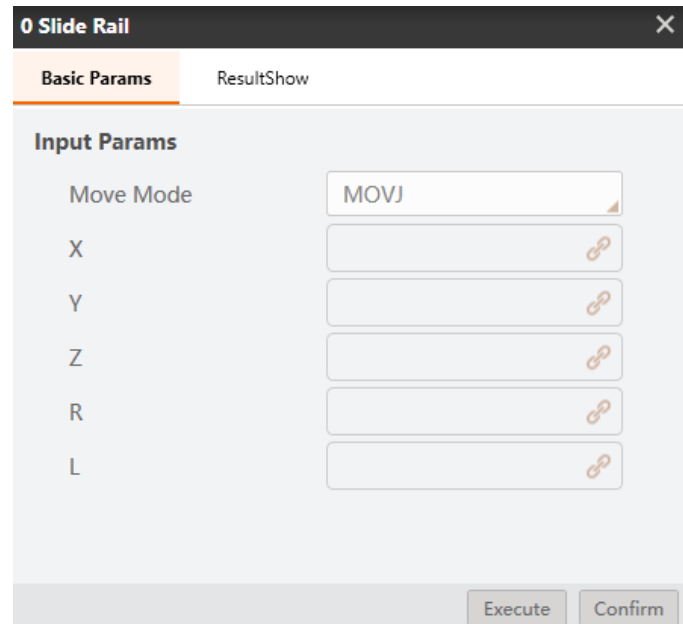


Figure 8.2 Set Sliding rail parameter

- Move mode: Support MOVJ, MOVL, and JUMP, which means point to point movement.
 - MOVJ: Joint movement. From point A to point B, each joint will run from initial angle to its target angle, regardless of the trajectory is shown below.

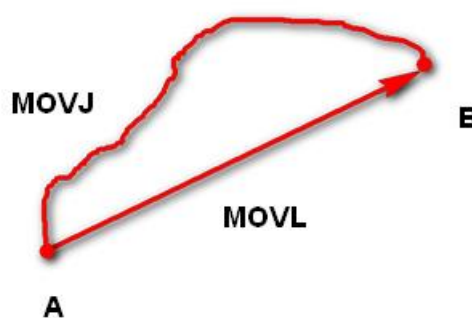


Figure 8.3 MOVL MOVJ mode

- MOVL: Rectilinear movement. The joints will perform a straight line trajectory from point A to point B.

– JUMP: From point A to point B, The joints will move in MOVJ mode, of which the trajectory looks like a door. As shown below.

1. Move up to the lifting Height (**Height**) in MOVJ mode.
2. Move up to the maximum lifting height (**Limit**).
3. Move horizontally to a point that is above B by height.
4. Move down to a point that is above B by height, which the height of the point is that of point B plus **Height**.
5. Move down to Point B.

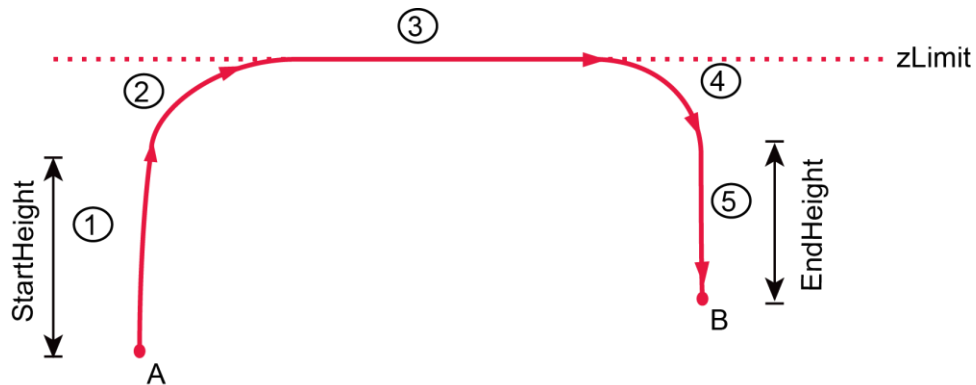


Figure 8.4 JUMP mode

- X: The value of coordinate X.
- Y: The value of coordinate Y.
- Z: The value of coordinate Z.
- R: The value of coordinate R.
- L: The distance the Sliding rail moves

8.2.3 Moving To a Point

Moving to a point means that Dobot Magician moves to a specified point according to the set motion mode. You can select motion mode based on site requirements. As shown below.

Figure 8.5 Set point parameters

- Move mode: Support MOVJ, MOVL, and JUMP, which means point to point movement.
- X: The value of coordinate X.
- Y: The value of coordinate Y.
- Z: The value of coordinate Z.
- R: The value of coordinate R.

8.2.4 Suction Cup Switch

Suction cup is used to suck up or release object. After connecting suction cup to Dobot Magician, you can make it suck up or release object by controlling switch, as shown below.

ON/OFF: Control the intake and outtake of air pump. Value range: **On** means intake, **Off** means outtake.

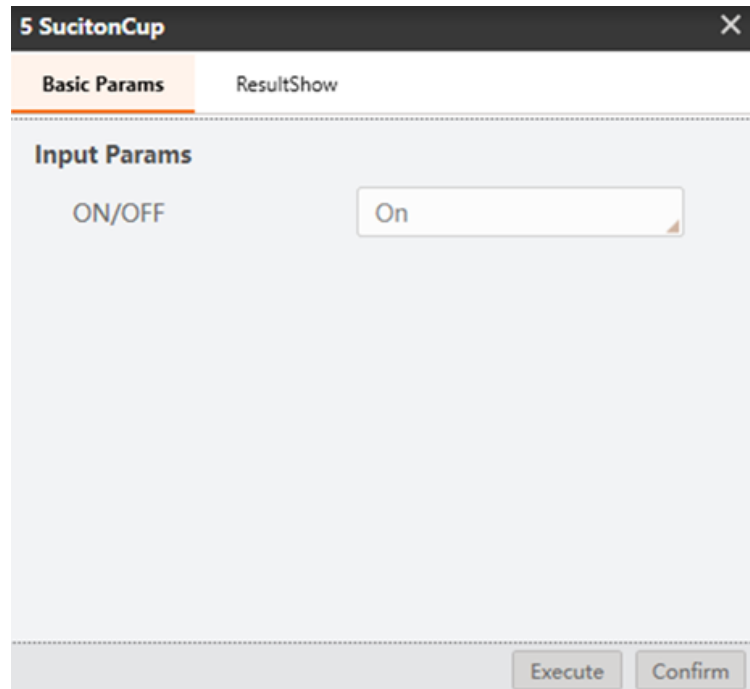


Figure 8.6 Suction cup switch parameter

8.2.5 Gripper Switch

Gripper is used to grab or release object. After connecting gripper to Dobot Magician, you can make it grab or release object by controlling switch, as shown below.

ON/OFF: Set gripper to grab or release object. Value range: **On** means grab, **Off** means release.

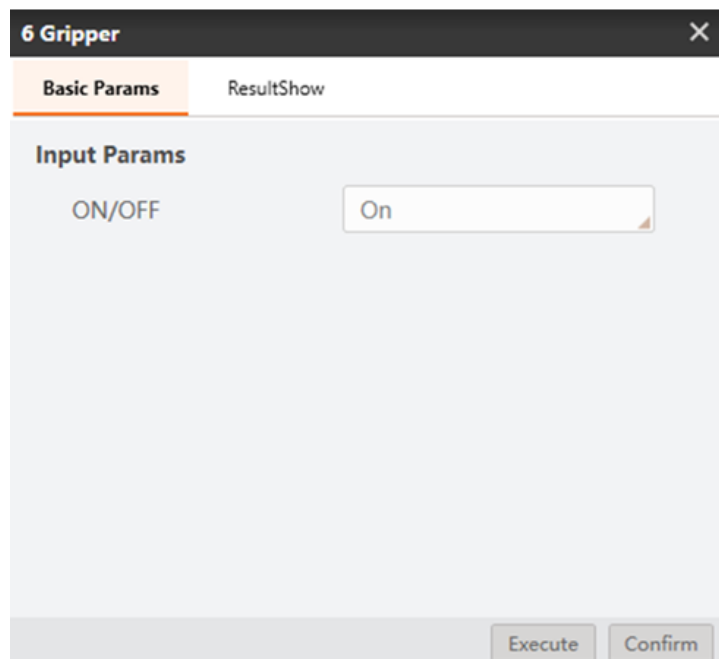


Figure 8.7 Set gripper switch parameter

8.2.6 Laser Switch

Laser switch is used to laser engraving. After connecting laser engraving kit to Dobot Magician, you can make it work by controlling switch, as shown below.

Laser: Open or close laser. Value range: **On** and **Off**.

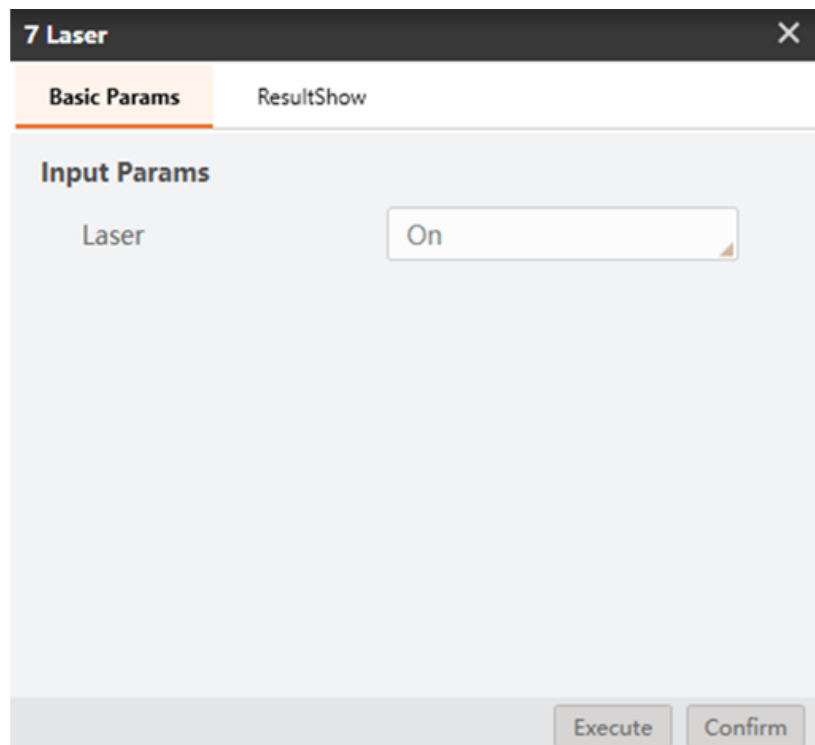


Figure 8.8 Set laser switch parameter

8.2.7 Home Calibration

If Dobot Magician loses step during working because of crash, you need to execute homing operation to ensure to improve accuracy. The calibration tool is shown below.

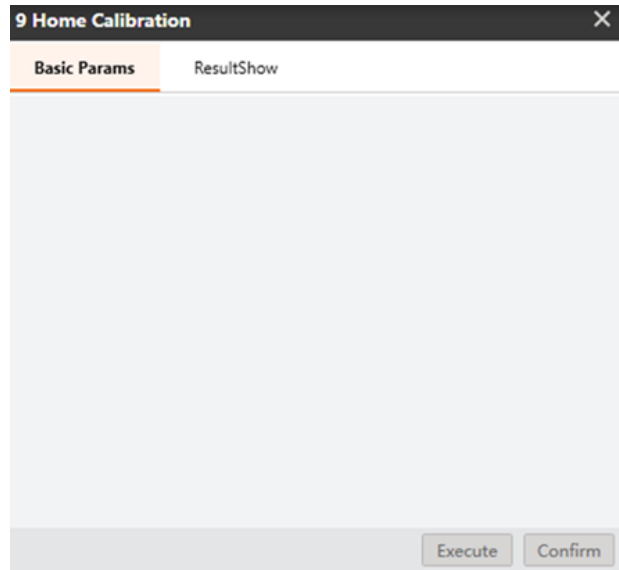


Figure 8.9 Calibration tool

8.2.8 I/O Input

The addresses of the I/O interfaces in Dobot Magician are unified. You can get specified input status by I/O input tool. As shown below.

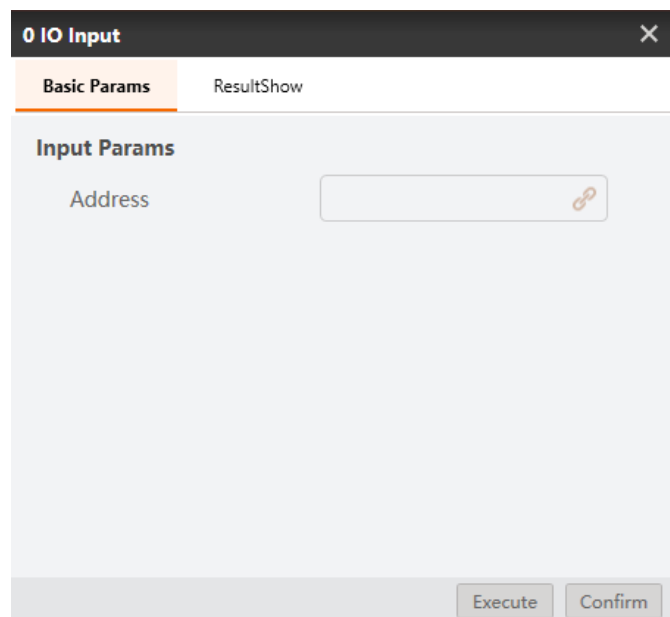


Figure 8.10 IO input tool

Address: Set the address of input interface, for the definition of I/O interface, please refer to ***Dobot Magician User Guide***.

8.2.9 I/O output

The addresses of the I/O interfaces in Dobot Magician are unified. You can Set specified output status by I/O output tool. As shown below.

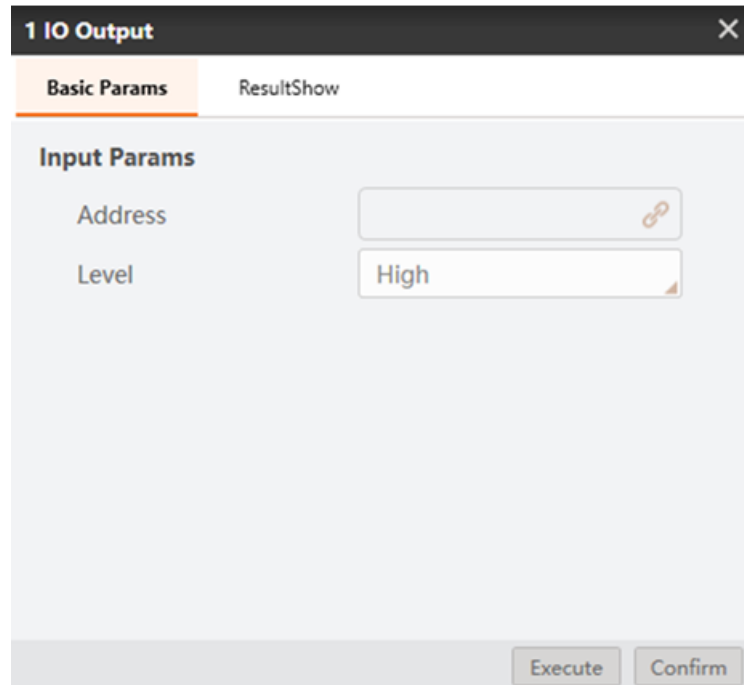


Figure 8.11 IO output tool

- Address: Set the address of output interface, for the definition of I/O interface, please refer to ***Dobot Magician User Guide***.
- Level: Set I/O output status. Value range: **High** means high level, **Low** means low level.

8.2.10 Speed Ratio

The speed ratio is used to set the speed ratio of the Dobot Magician to control the Dobot Magician's speed. As shown below, set the speed ratio to 50%, the Dobot Magician will move at the 50% speed ratio.



Figure 8.12 Speed ratio

8.3 Dobot M1 Tools

8.3.1 I/O Input

The addresses of the I/O interfaces in Dobot Magician are unified. You can get specified input status by I/O input tool. As shown below.

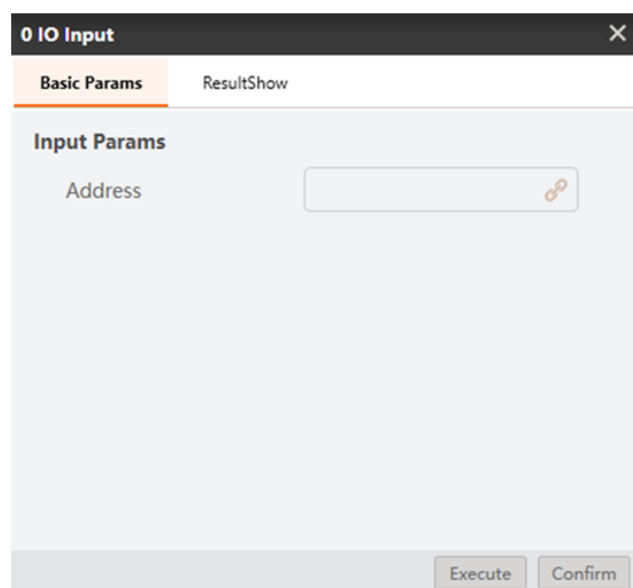


Figure 8.13 I/O input tool

Address: Set the address of I/O input interface, for the definition of I/O interface, please refer to ***Dobot M1 User Guide***

8.3.2 I/O Output

The addresses of the I/O interfaces in Dobot Magician are unified. You can Set specified output status by I/O output tool. As shown below.

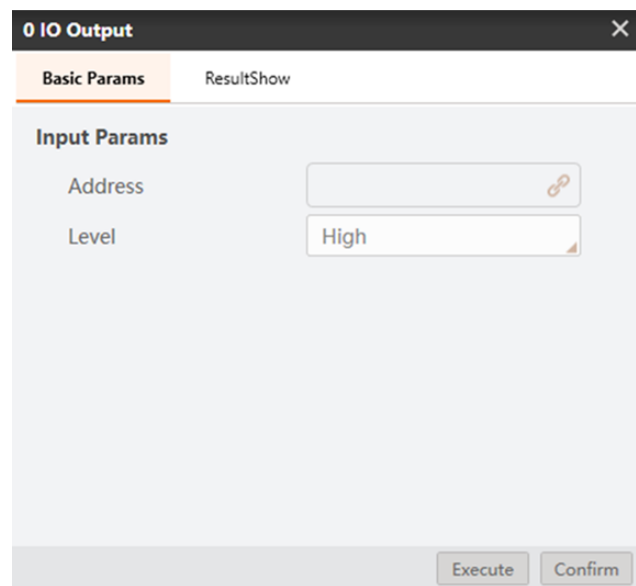


Figure 8.14 I/O output tool

- Address: Set the address of I/O output interface, for the definition of I/O interface, please refer to ***Dobot M1 User Guide***
- Level: Set I/O output status. Value range: **High** means high level, **Low** means low level.

8.3.3 Moving to a point

Moving to a point means that Dobot M1 moves to a specified point according to the set motion mode. You can select motion mode based on site requirements. As shown below.

Figure 8.15 Set point parameters

- Move mode: Support MOVJ, MOVL, and JUMP, which means point to point movement.
 - MOVJ: Joint movement. From point A to point B, each joint will run from initial angle to its target angle, regardless of the trajectory is shown below.

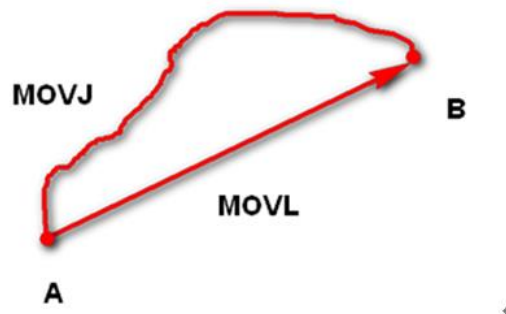


Figure 8.16 MOVL/MOVJ mode

- MOVL: Rectilinear movement. The joints will perform a straight line trajectory from point A to point B.
- JUMP: From point A to point B, The joints will move in MOVJ mode, of which the trajectory looks like a door.
 1. Move up to the lifting Height (**Height**) in MOVJ mode.
 2. Move up to the maximum lifting height (**Limit**).
 3. Move horizontally to a point that is above B by height.

4. Move down to a point that is above B by height, which the height of the point is that of point B plus **Height**.
5. Move down to Point B.

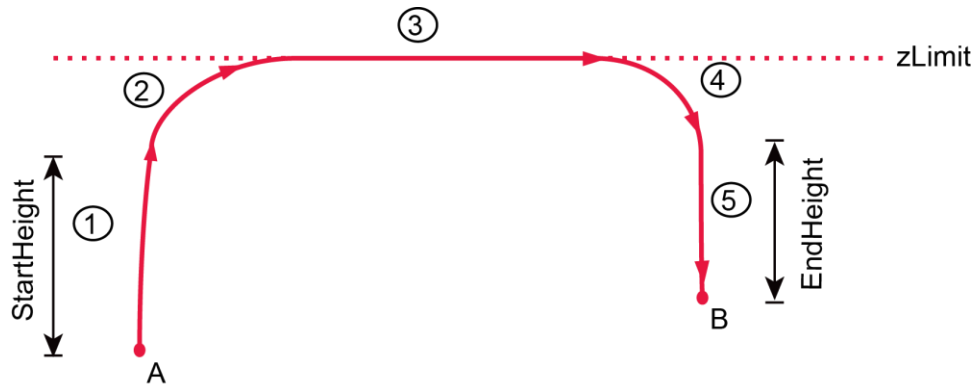


Figure 8.17 JUMP mode

- X: The value of coordinate X.
- Y: The value of coordinate Y.
- Z: The value of coordinate Z.
- R: The value of coordinate R.

8.3.4 Dobot Arm Orientation

With two types of arm orientation (the left-hand orientation and the right-hand orientation), the Dobot M1 can move to nearly any position and orientation within a given work envelope. You need to specify the arm orientation when the Dobot M1 is moving. If you fail to do so, the Dobot M1 may move following an unexpected path, resulting in some interferences with a peripheral equipment.

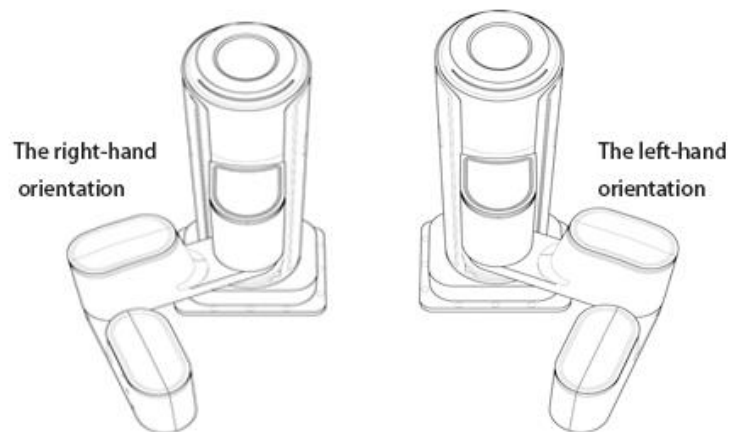


Figure 8.18 Arm orientation

You can set the arm orientation by the Dobot arm orientation tool, as shown below.

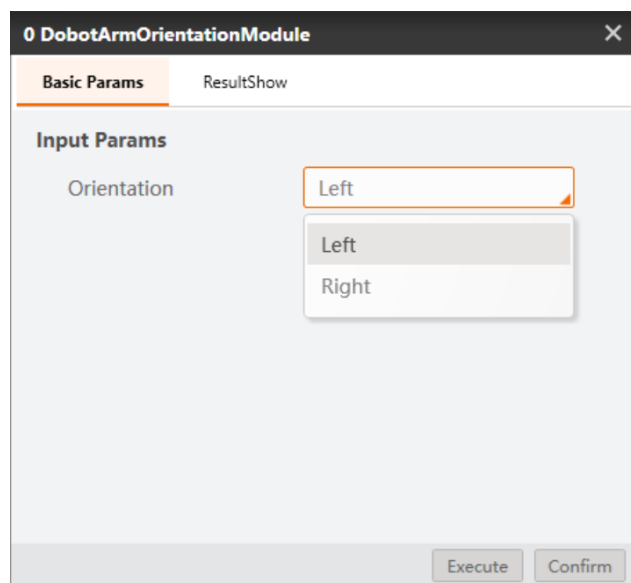


Figure 8.19 Set arm orientation

8.3.5 Speed Ratio

The speed ratio is used to set the speed ratio of the Dobot M1 to control the Dobot M1's speed. As shown below, set the speed ratio to 50%, the Dobot M1 will move at the 50% speed ratio.

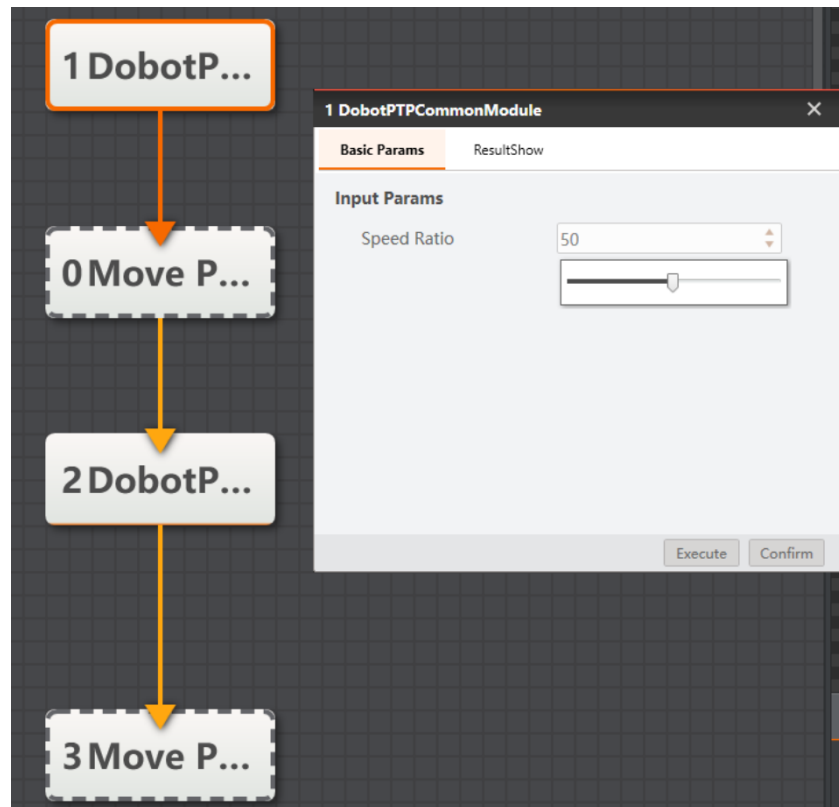



Figure 8.20 Set speed ratio

Chapter 9 Operation Interface

9.1 Editing Interface

Click  to open the front running interface, and enter the editing interface. Editing interface is divided into control area, editing area, parameter setting area, as shown below.

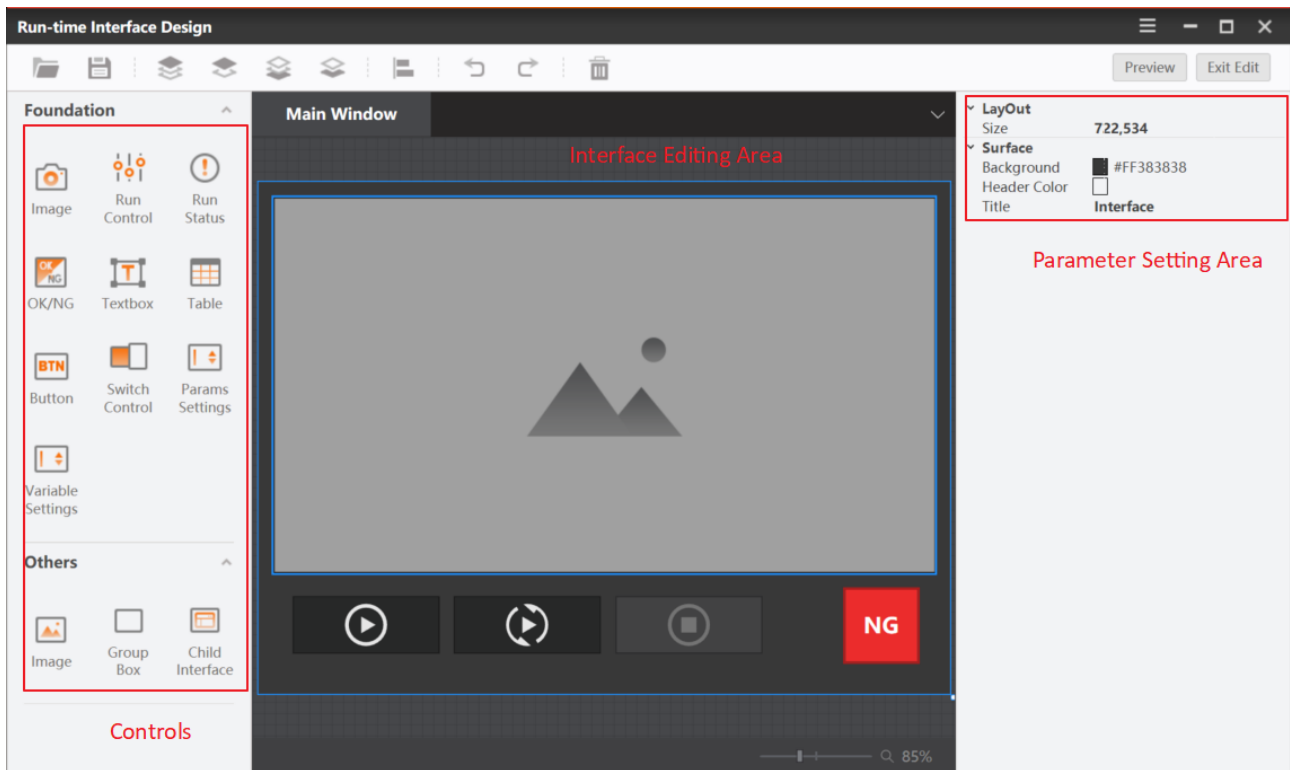


Figure 9.1 Editing Interface

9.2 Controls

In this module, it provides various controls to allow users to customize the editing interface. Controls include Image, Run Control, Run Status, OK/NG, Textbox, Table, Button, Switch Control, Parameter Settings, Variable Settings, Group Box and Sub Interface. After clicking the different control icons, click the mouse again in the editing area to add the control. You can right-click the control to delete, redo and execute other operations.

9.3 Editing Area

It provides the area to edit the interface. You can add various controls to this area to achieve the desired results.

9.4 Parameter Settings

Select a control to set its associated parameters.

- **Image:** it associates the running image in the solution.
- **Run Control:** it controls the run status, including once, continuously, stop and other functions.
- **Run Status:** it displays the current running status.
- **OK/NG:** it displays the status of the current running results.
- **Textbox:** it edits the displayed texts.
- **Table:** it associates the data you want to display with the table.
- **Button:** it associates the parameter page of certain tool with this button. You can click this button directly on the interface to call the parameter page of the related tool to adjust the parameters.
- **Switch Control:** it sets the basic parameters, run parameters and the parameters of the result display page in the associated process. For example, when the control is associated with the save image in the local image module, it can switch the function of save image of the process in the running interface.
- **Parameters Settings:** it adds parameter settings box, links int or float data, and input data or adjust data size in the parameter settings box of the running interface.
- **Variable Settings:** in data linking, only variables in calculator module can be linked, and the value of bound variables can be directly modified in the running interface.
- **Image:** it associates images in your local computer and display them on the interface.
- **Group Box:** It is a combination function that can put some buttons into this combination.
- **Sub Interface:** the sub interface is an extension of the main interface function. After adding sub interface control, the parameters of the sub interface can be set in the parameter setting area and the sub interface can be edited. After entering the sub interface, any control can be added to the sub interface. The editing mode of the interface is the same as that of the main interface.

The editing and preview effects of the running interface are shown below.

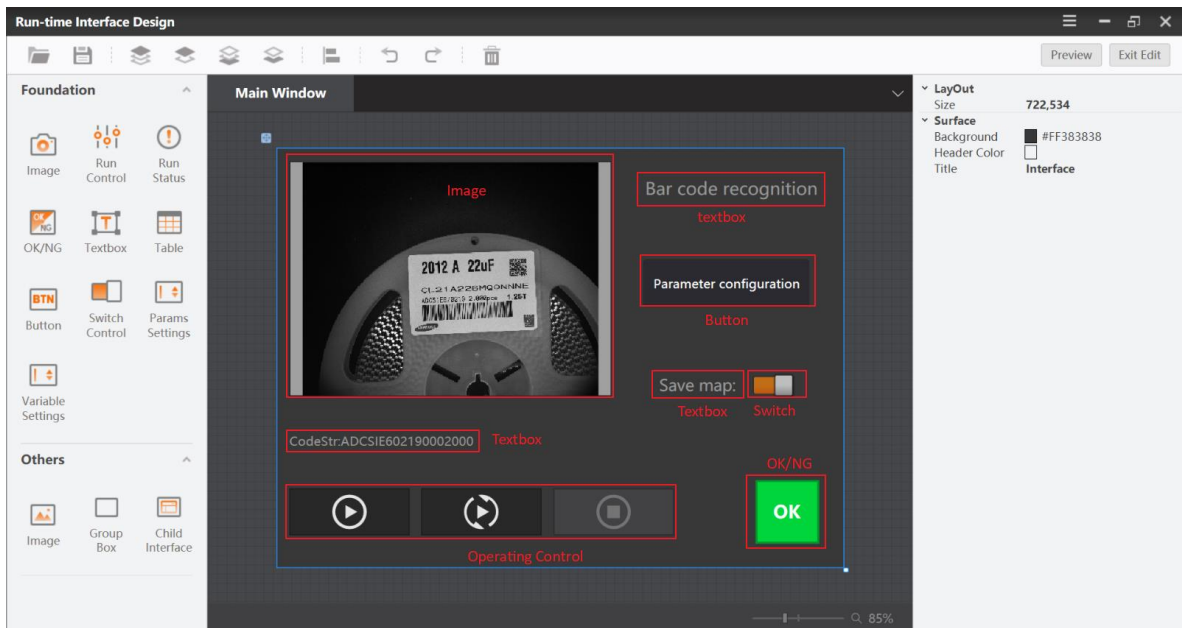


Figure 9.2 Interface edit

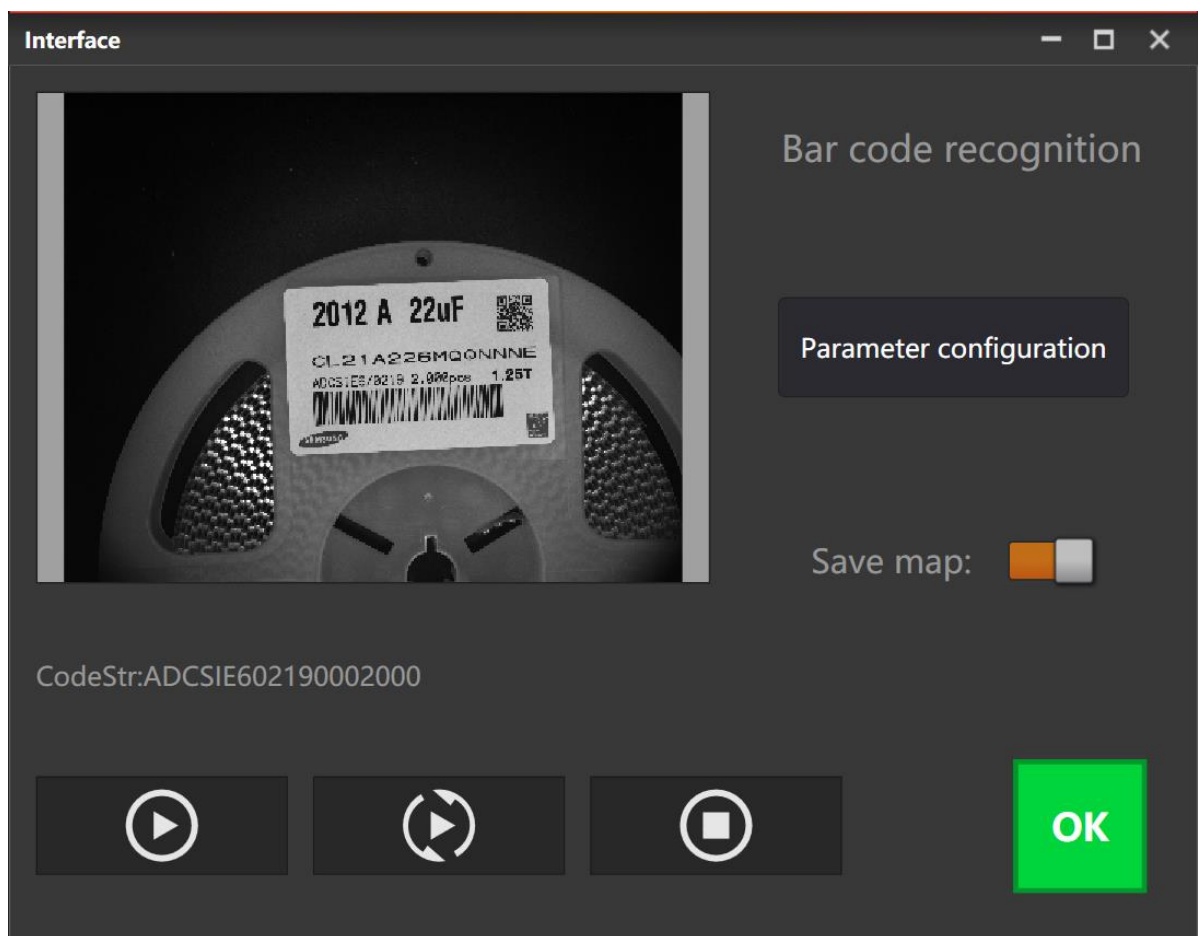


Figure 9.3 Result preview

9.5 Role Interface Settings

Software control includes admin control, technician control, operator control and no control.

- When the actual controller of the software is the admin, enter the start-up interface and select any solution type to jump to the running interface. Admin can control the operating and stopping of the running interface, as shown below.

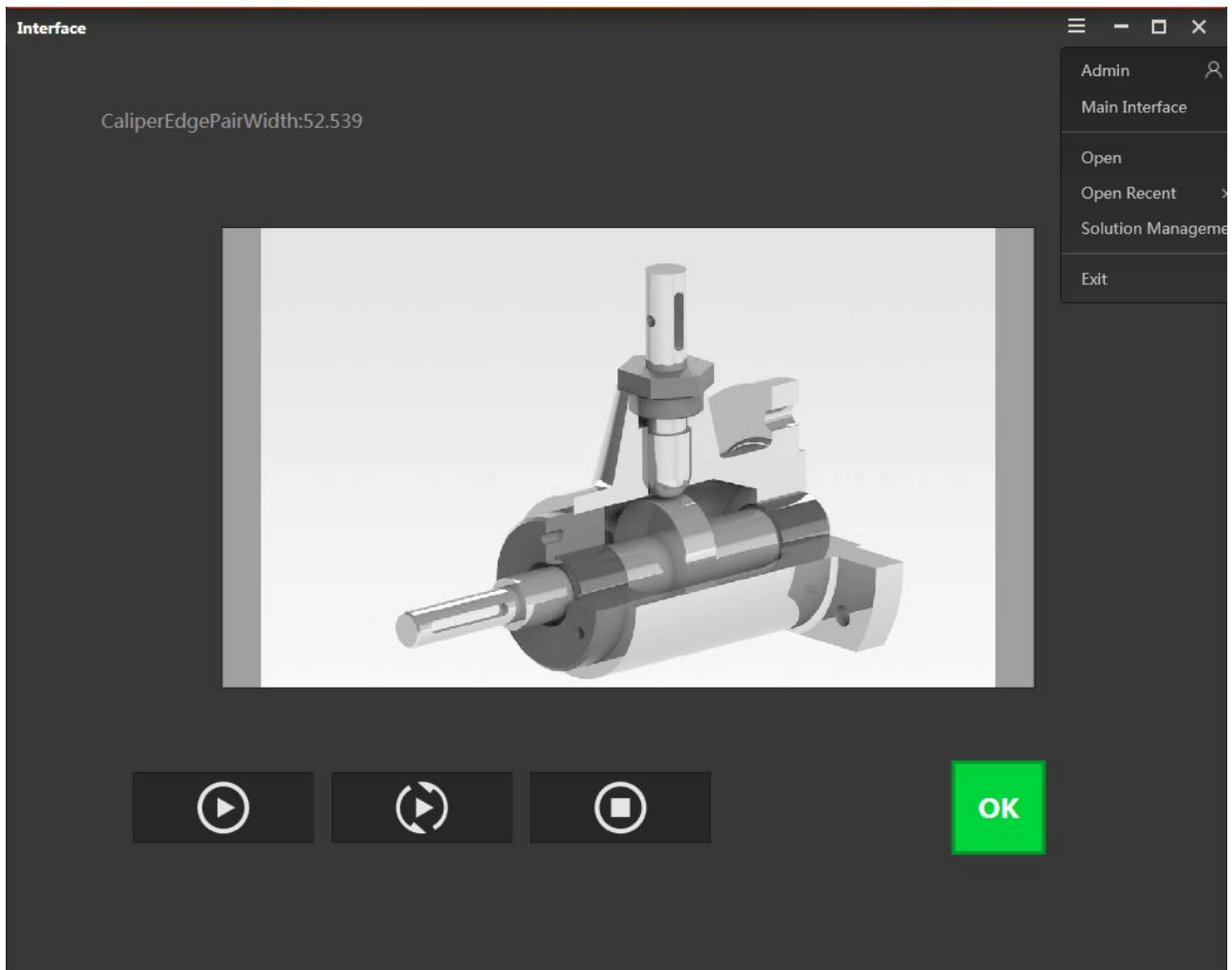


Figure 9.4 Admin interface

Steps:

1. Click **Admin** to switch the role of the software controller.
2. Click **Main Interface** and input the correct password to enter the main interface.
3. Click **Open** to enter the running interface of solutions.
4. (Optional) Click **Open Recent** to open the solutions that you have opened recently.

5. Click **Solution Management** to set the path, string, and etc.
 6. Click **Exit** to exit the software.
- When the actual controller of the software is the technician, enter the start-up interface and select any solution type to jump to the running interface. Technician can control the operating and stopping of the running interface, as shown below.

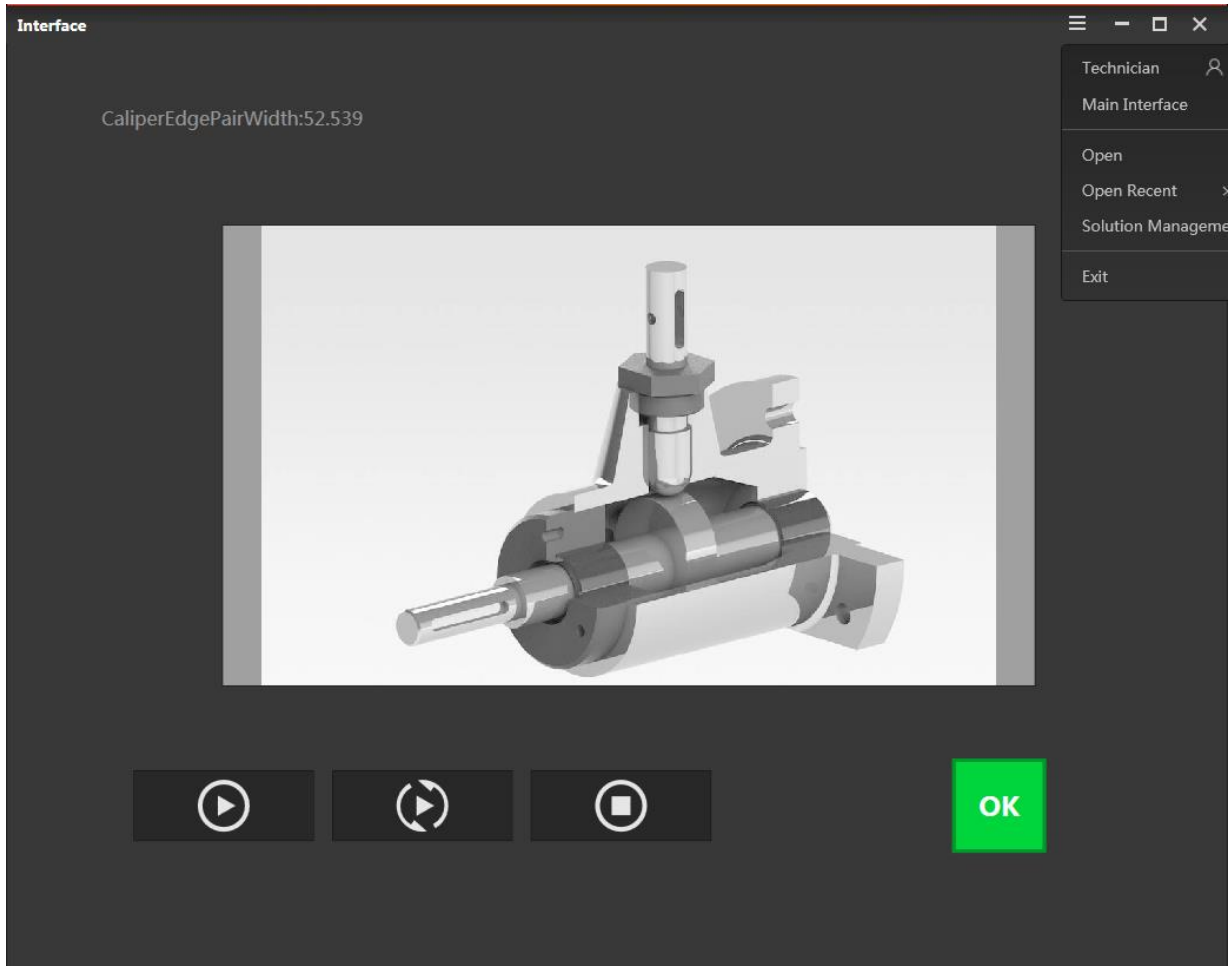


Figure 9.5 Technician interface

Please refer to the admin section for the same settings. The difference between admin and technician is that the technician can only use the permissions assigned by the admin after entering the main interface.

- When the actual controller of the software is the operator, enter the start-up interface and select any solution type to jump to the running interface. At this time, operator can only control the operating and stopping of the running interface, but have no permission to enter the main interface, as shown below.

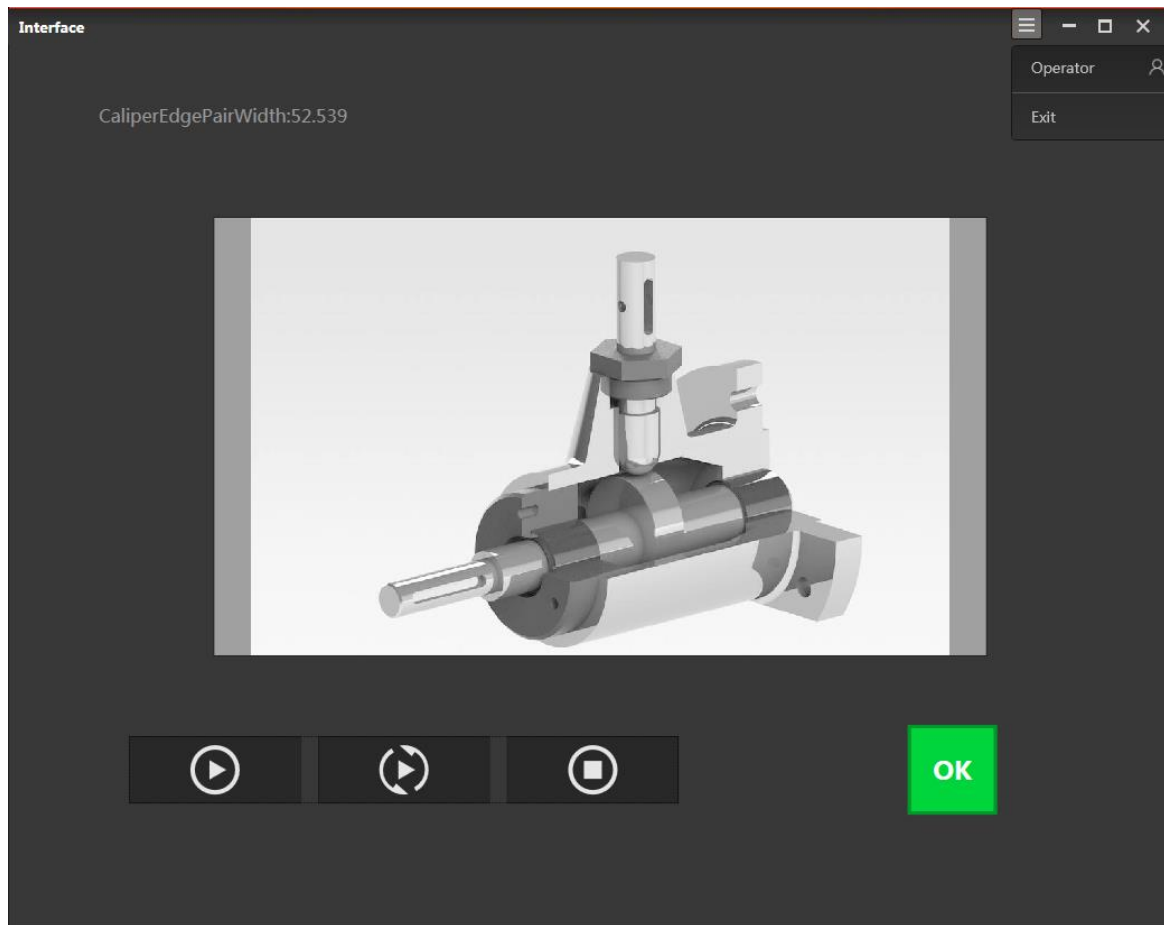


Figure 9.6 Operator interface

When the software controller is not set, enter the start-up interface and select any solution type to directly jump to the main interface.

Chapter 10 Examples

10.1 Location for USB Hole

In this example, measure the position information of the CNC USB interface, and determine the center point position, as shown below.



Figure 10.1 CNC USB interface

Steps:

Step 1. Solution establishment idea:

- Find the position of the USB interface in the image by matching template.
- Find the four lines of the trapezoidal interface, including upper line, left line, lower line, right line.
- Measure the intersection of two adjacent lines: the intersection point on the upper left line, the intersection point on the upper right line, the intersection point on the lower left line, and the intersection point on the lower right line.
- Measure the distance between two diagonal points: the distance between the upper left line intersection point and the lower right line intersection point, and the distance between the upper right line intersection point and the lower left line intersection point.

The overall solution flow is shown below.

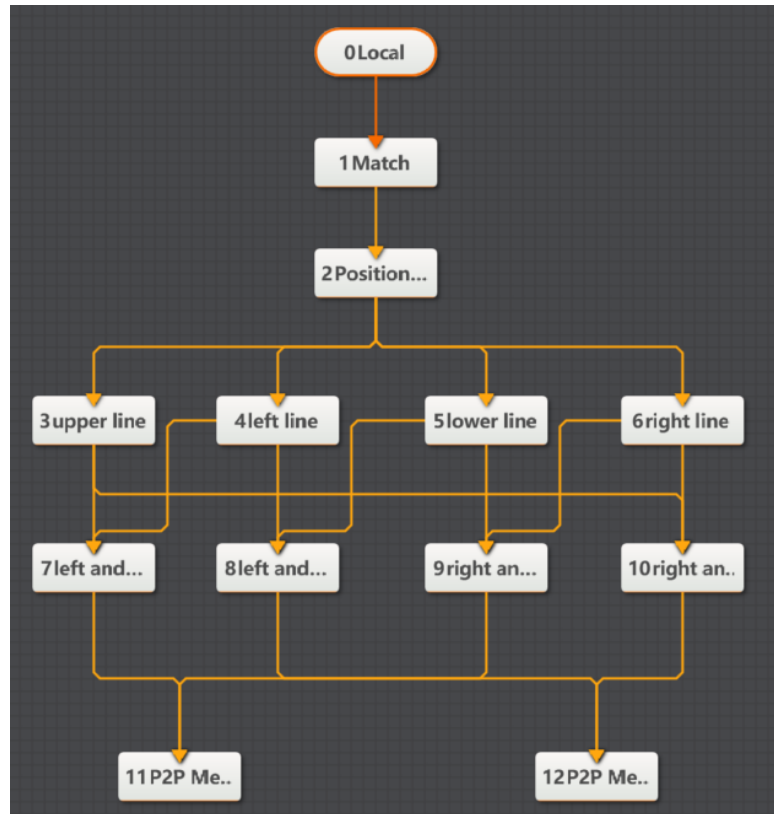


Figure 10.2 Flow

Step 2. Save the solution, as shown below.

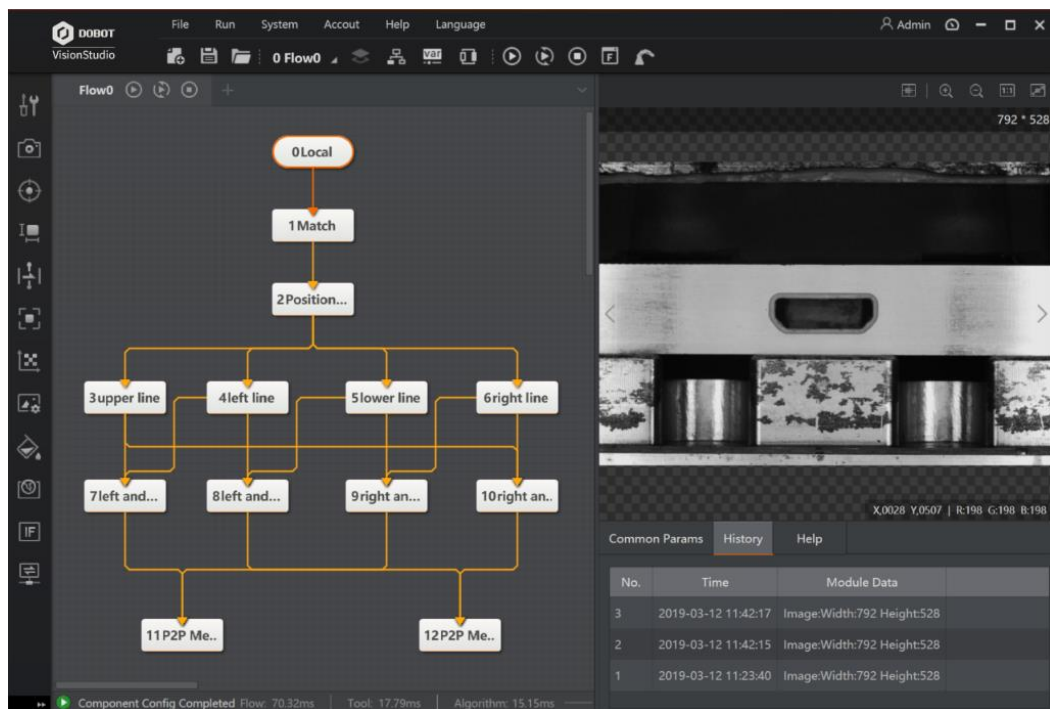


Figure 10.3 Save solution

Step 3. Load solution, as shown below.

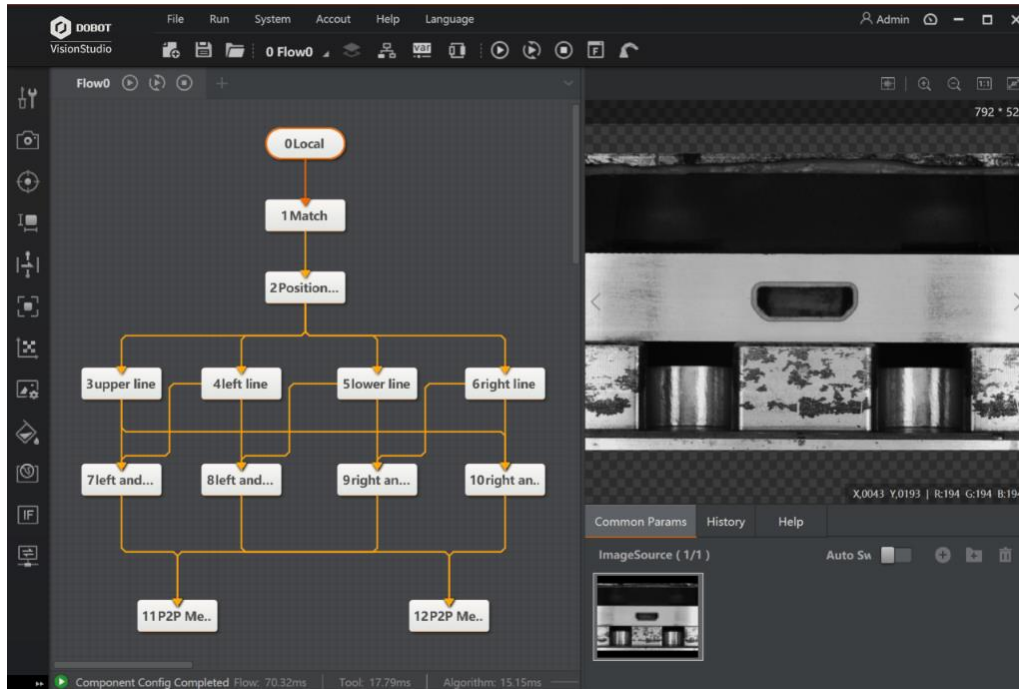


Figure 10.4 Load solution

Step 4. Train model, as shown below.

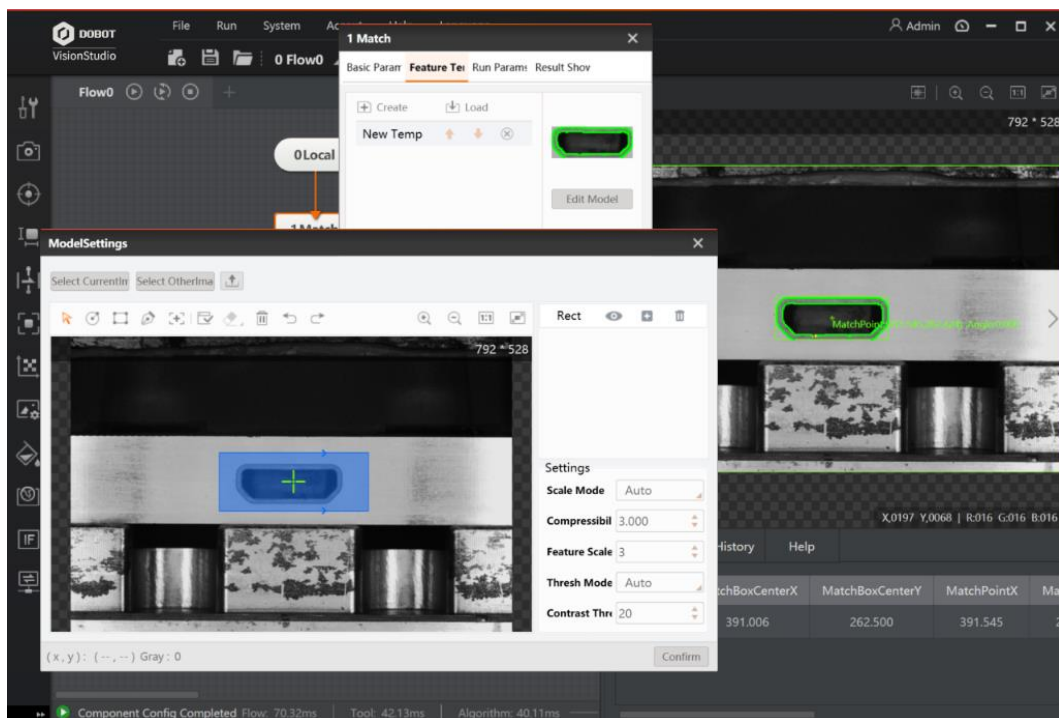


Figure 10.5 Train model

Locate the trapezoidal interface by matching template, as shown below.



Figure 10.6 Result

Step 5. Find lines. Select ROI according to actual features of lines in the image, and set parameters.

- **Edge Polarity:** it includes dark to light, light to dark and any.
- **Search Mode:** it includes best line, first line and last line.
- **Search Direction:** it includes top to bottom and left to right.

In this example, find the upper line of the trapezoidal interface is shown below.

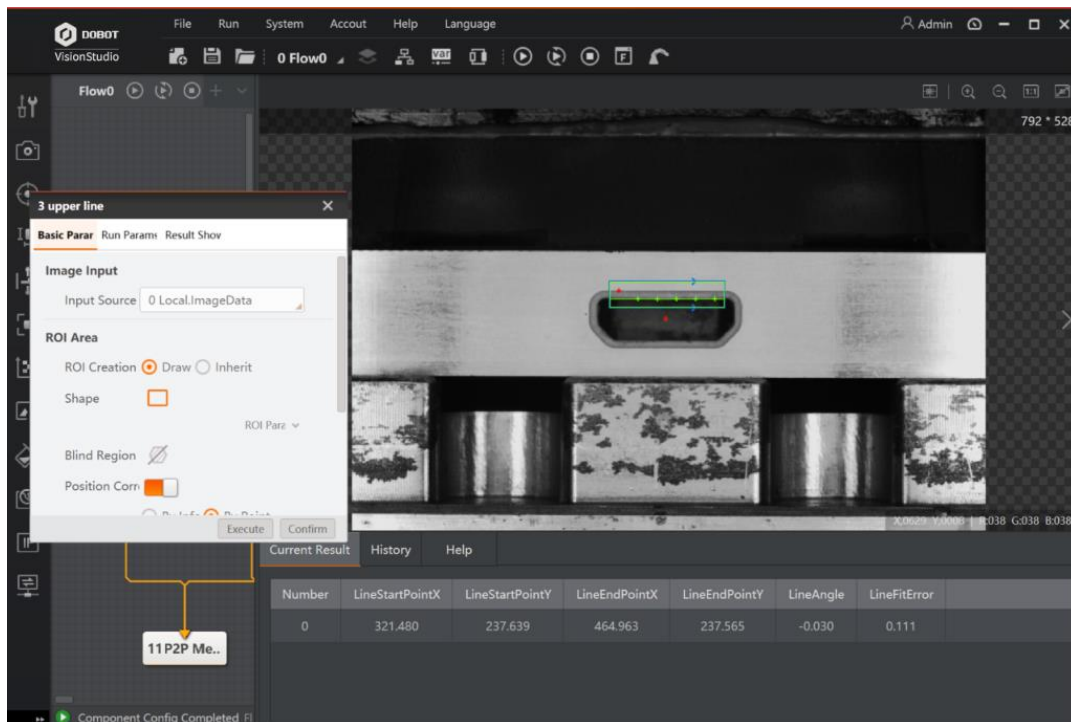


Figure 10.7 The upper line

In this example, find the lower line of the trapezoidal interface is shown below.

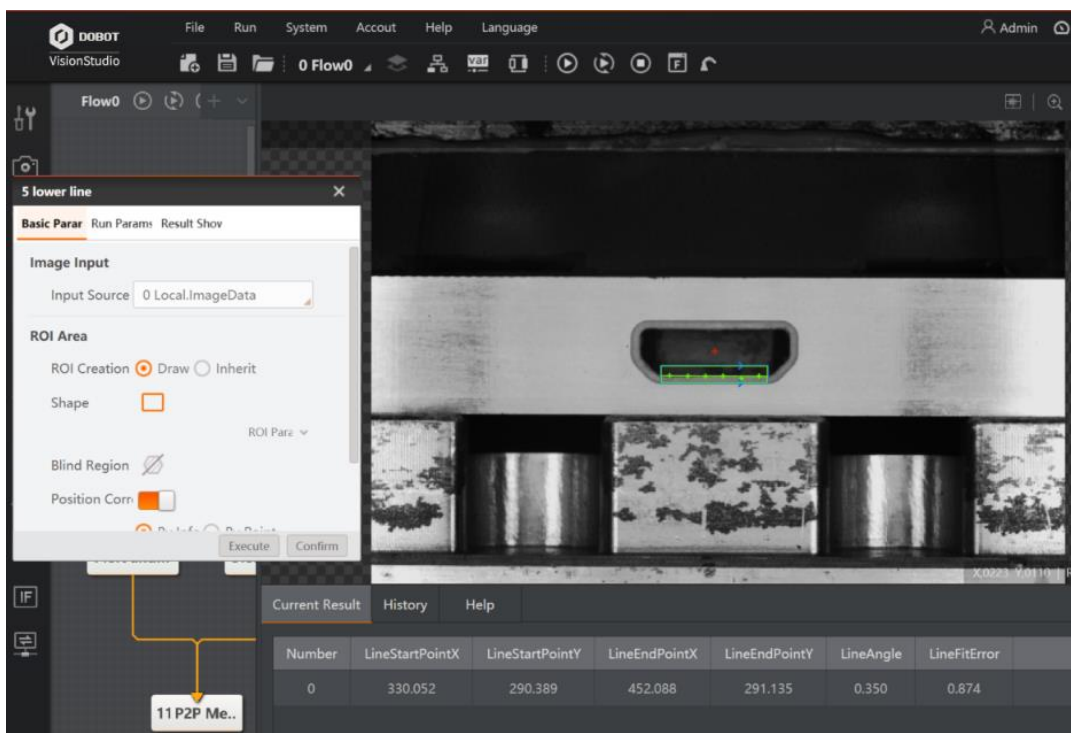


Figure 10.8 The lower line

In this example, find the left line of the trapezoidal interface is shown below.

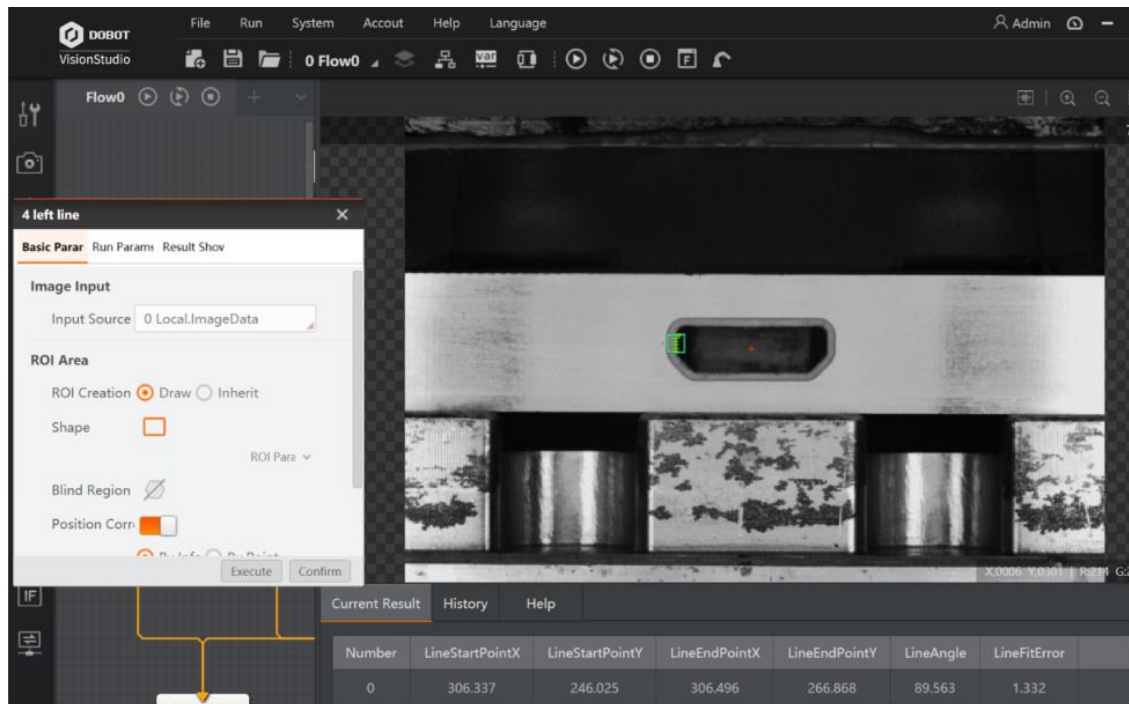


Figure 10.9 The left line

In this example, find the right line of the trapezoidal interface is shown below.

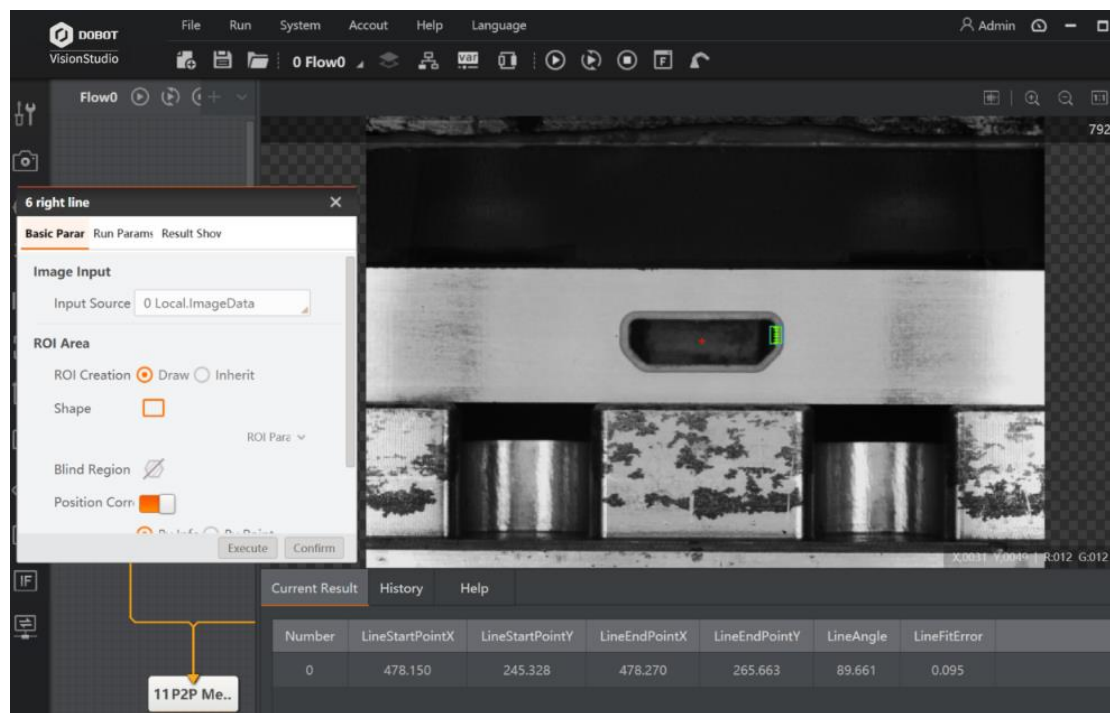


Figure 10.10 The right line

Step 6. Determine the left and upper intersection point of lines, as shown below.

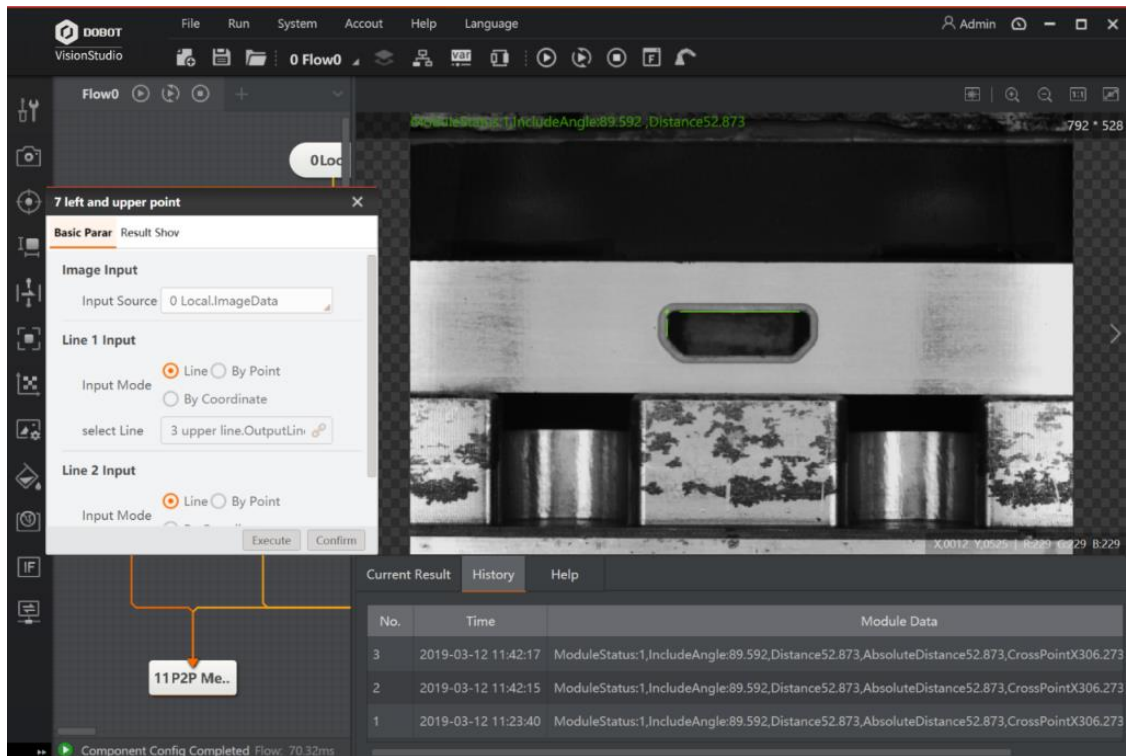


Figure 10.11 The left and upper intersection point of lines

Step 7. After obtaining the four intersection points, execute point to point (P2P) measurement, as shown below.

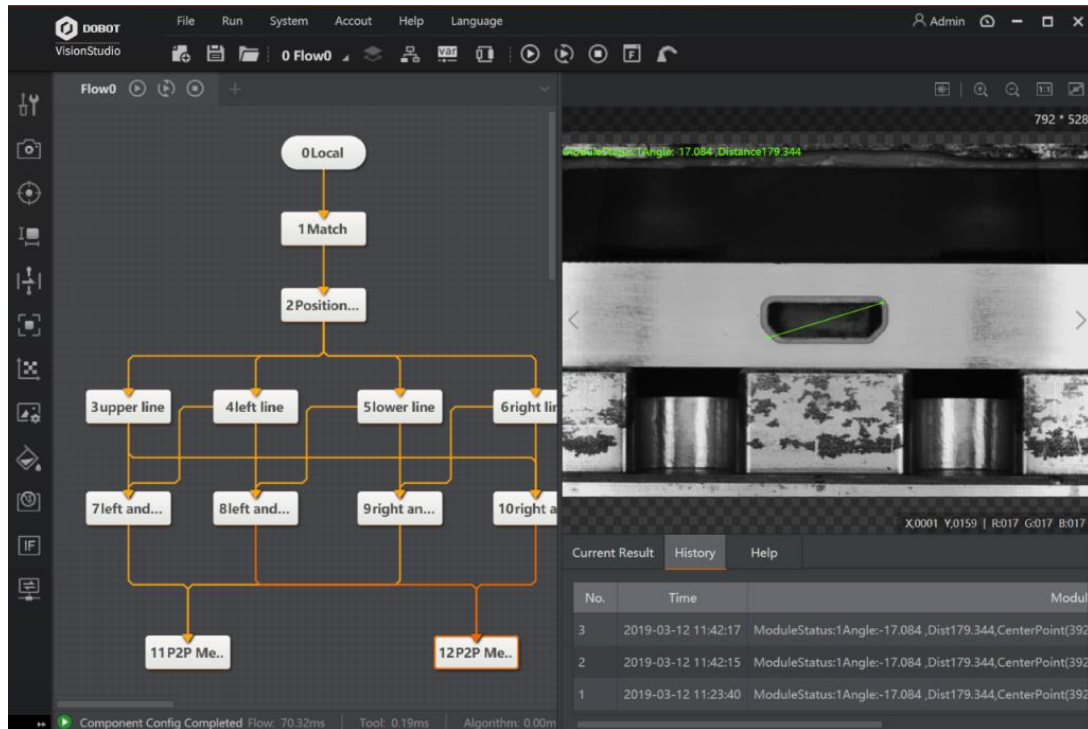


Figure 10.12 Point distance

10.2 Detection for Detective Metal

In this example, it detects whether there is a metal cover and its position is right, and the sample image is shown below.

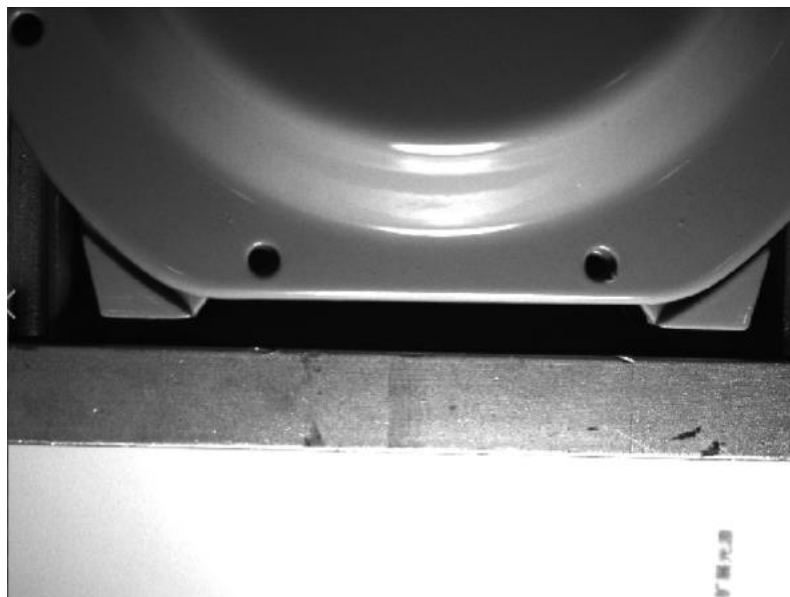


Figure 10.13 Examples

Steps:

Step 1. Solution establishment idea:

- The surface of the metal cover is reflective, and whether there is a metal cover can be detected by intensity analysis.
- The bottom triangle can determine the installation position of the metal cover. Through feature matching tool, you can set the center of the base as a matching point by the training template.
- Find the center of the installation hole on the metal cover by the tool of finding circle.
- Measure the distance between the matching point and the center of the circle by the tool of point to point measurement (P2P measure).
- Determine whether the metal cover is installed correctly by the center distance and the tool of if module.

Step 2. The overall solution flow is shown below.

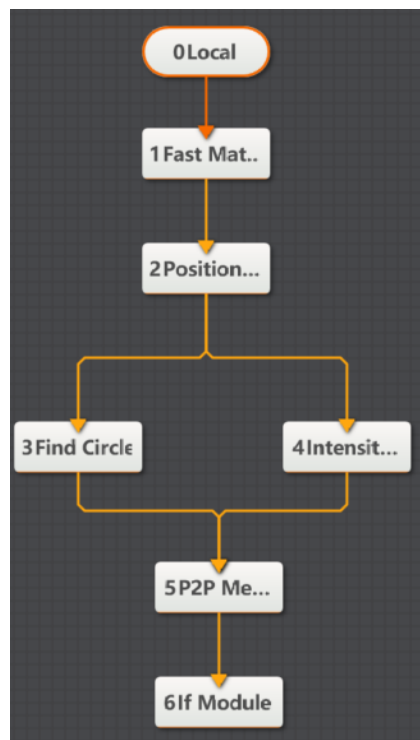


Figure 10.14 Flow

Step 3. Train model, as shown below.

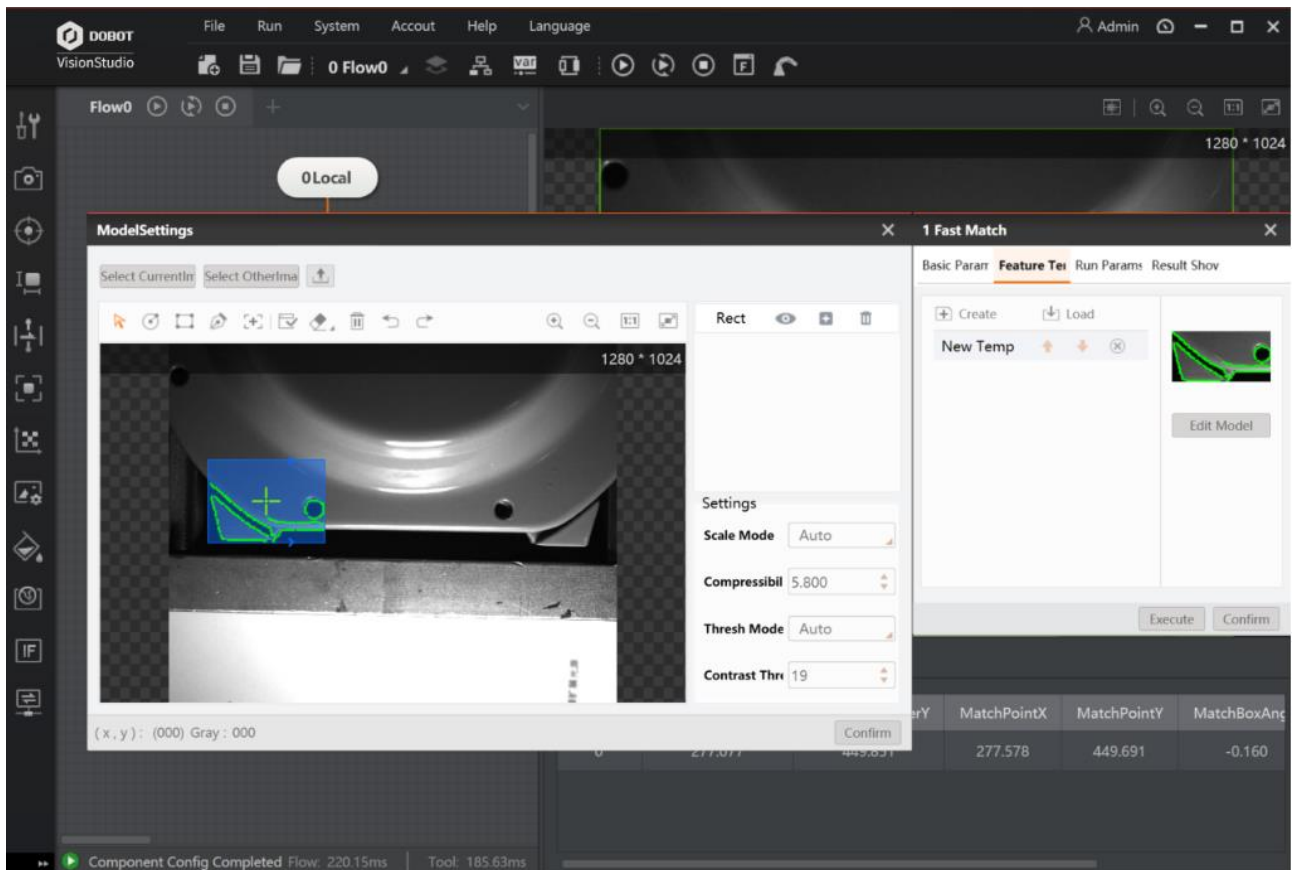


Figure 10.15 Train model

Step 4. Find circle. Select the search range of circle, and determine the the location of metal cover, as shown below.

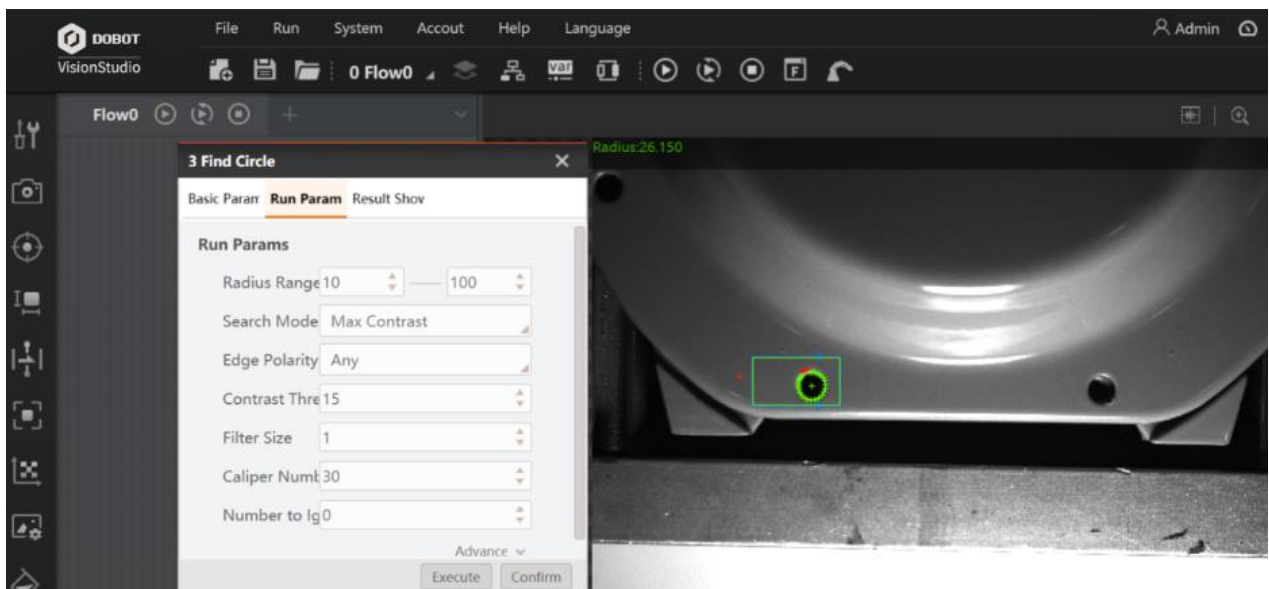


Figure 10.16 Find circle

Step 5. Measure. The distance between circle center and the matching point is shown below.

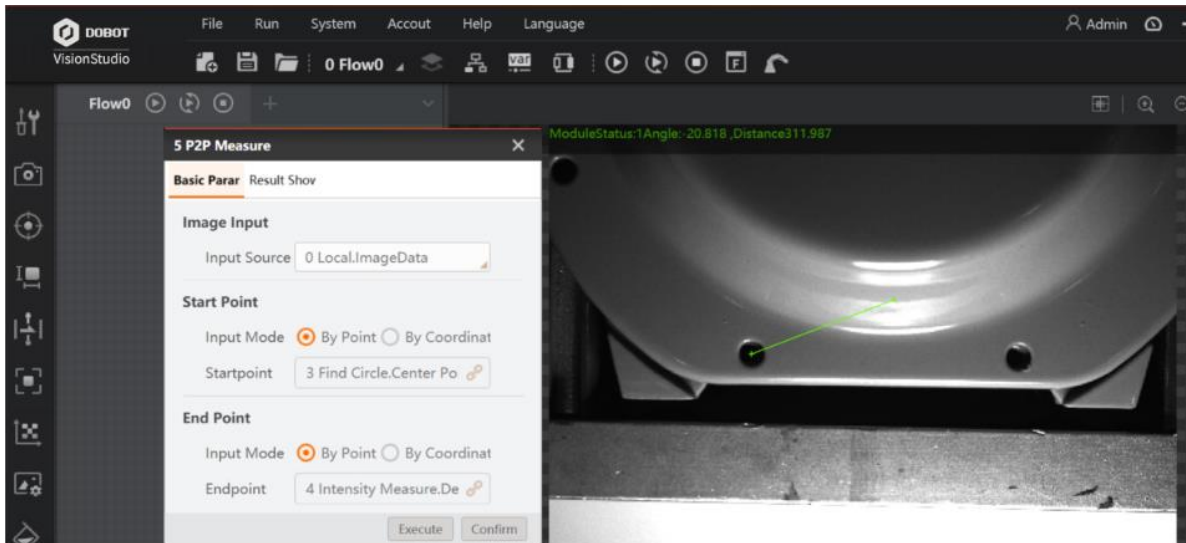


Figure 10.17 Measure

According to valid value range to determine whether the installation position is right or not.

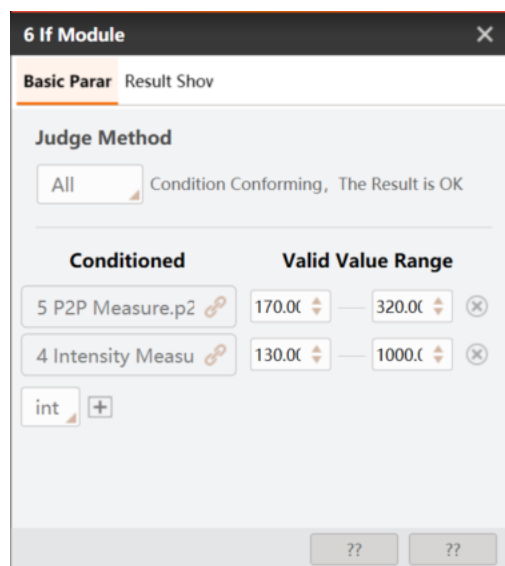


Figure 10.18 If module

Step 6. Intensity measurement analysis. Determine whether there is a metal cover by intensity measure and if module, as shown below.

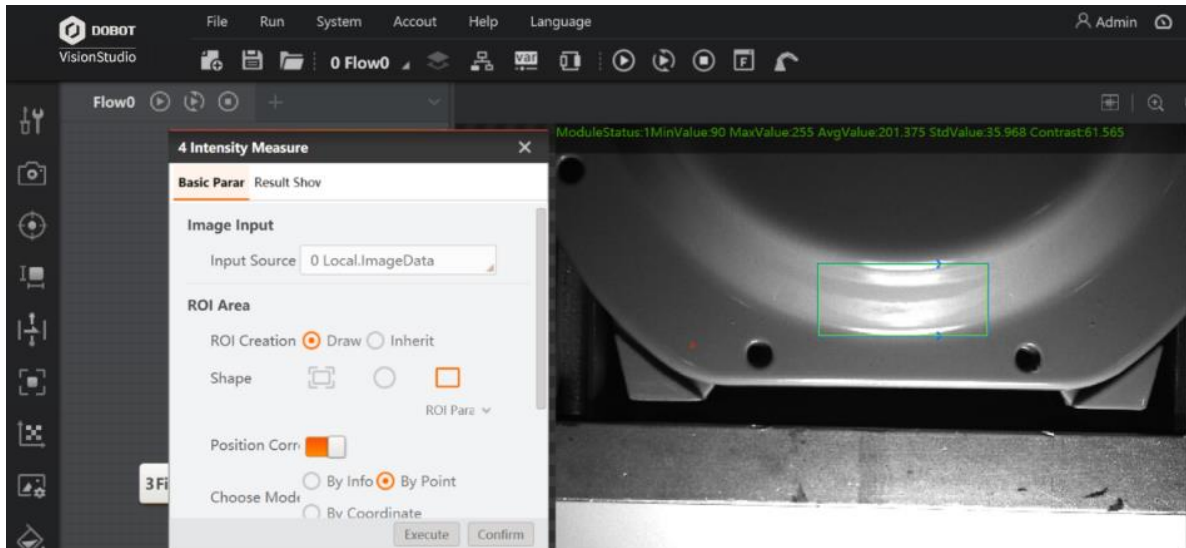


Figure 10.19 Intensity measurement analysis

Step 7. Detection result is shown below.

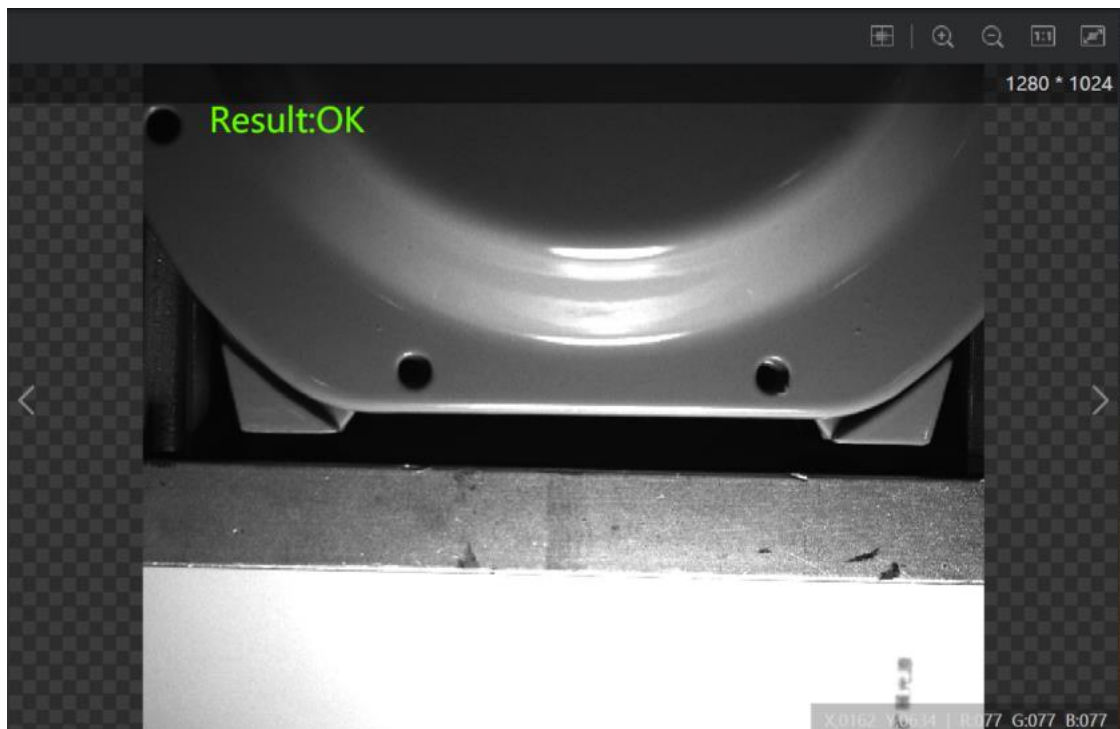


Figure 10.20 Result

10.3 Calibration and Detection

In this example, detect the distance between the far left and far right circle holes on the electronics, as shown below.

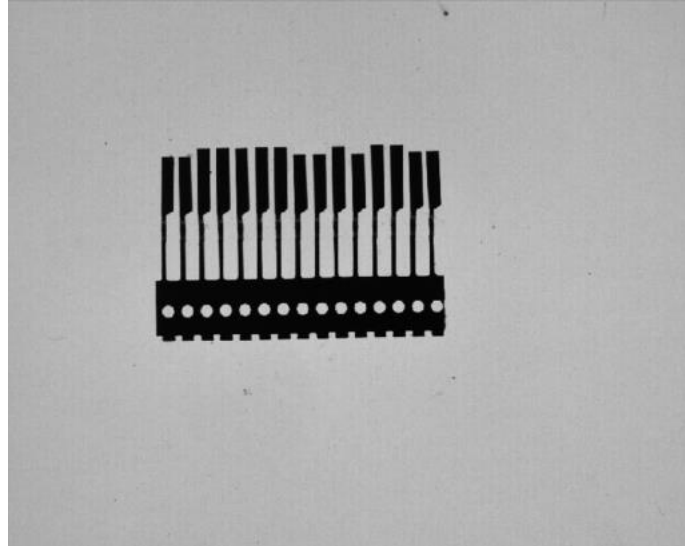


Figure 10.21 Example

Steps:

Step 1. Solution establishment idea:

- Calibration can be calibrated by checkerboard.
- There are a lot of interference for many circles among far left and far right circles. Thus, template matching and location is needed for capture the circle accurately.
- Transform the two circle centers to the post-calibration coordinates via calibration transformation.
- Measure the circle center distance by using point to point measure (P2P measure).

Step 2. The overall solution flow is shown below.

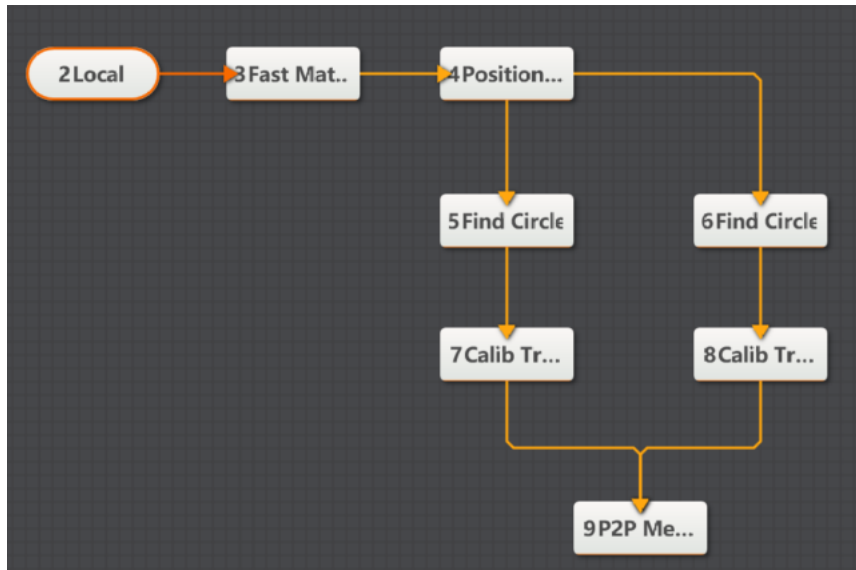


Figure 10.22 The overall solution flow

Step 3. Calibration.

Load the calibration image into local image, connect calibration tool, set parameters of calibration tool, and select save path.

Step 4. Find circle.

Select the ROI area of circle, and find the circle centers of two circles, as shown below.

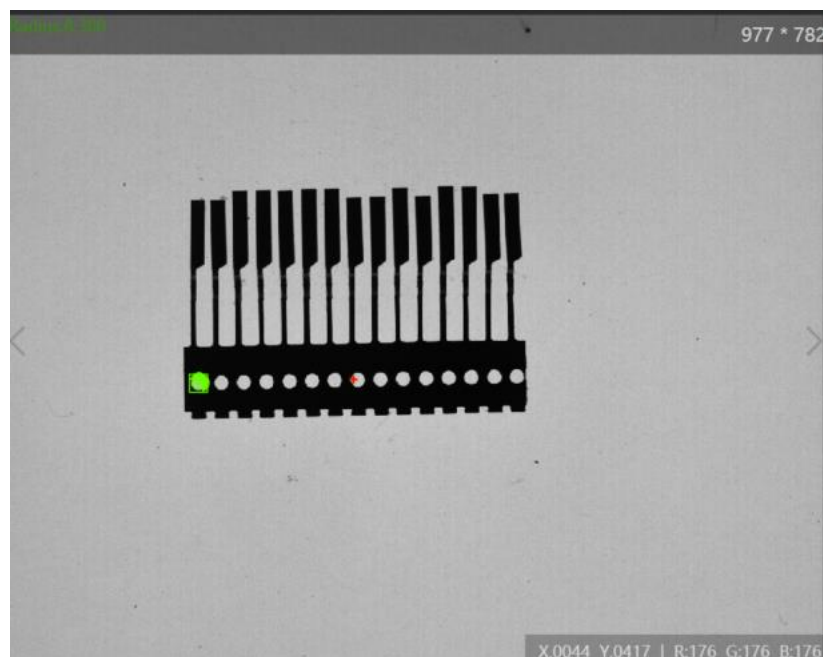


Figure 10.23 Find circle

Step 5. Calibration transformation

Transform the two circle centers to the post-calibration coordinates via calibration transformation.

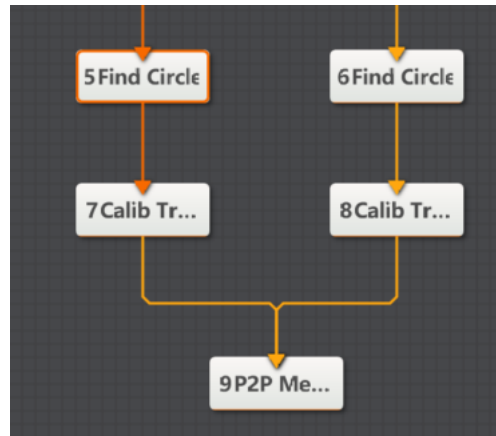


Figure 10.24 Calibration transformation solution flow

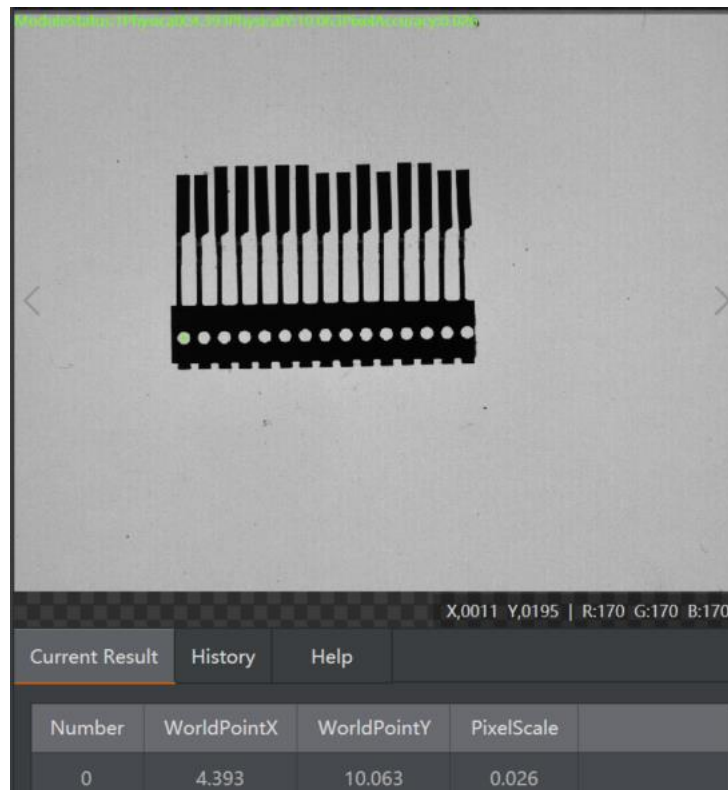


Figure 10.25 Calibration transformation

Step 6. P2P measure

In the basic parameter interface of P2P measure, enter the coordinates of two circle centers as start point and end point. The result is shown below after running the solution.

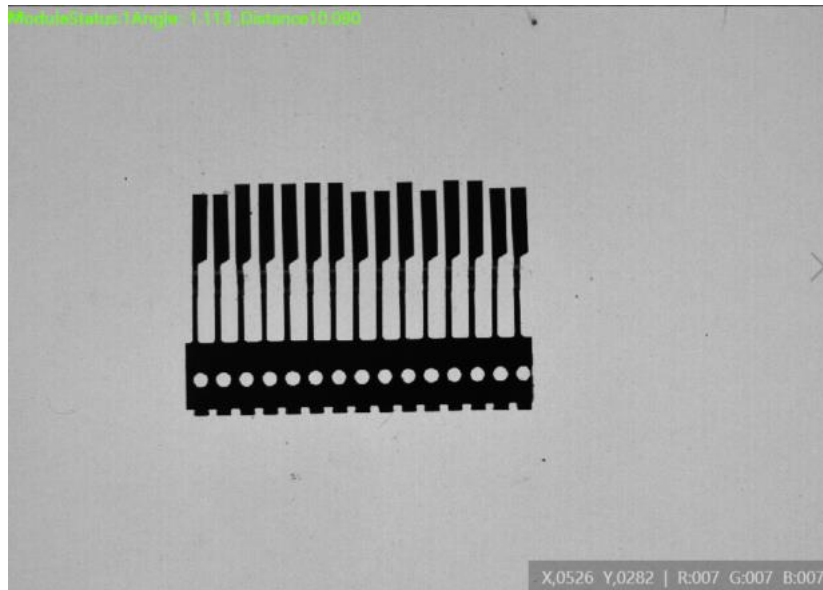


Figure 10.26 P2P measure

10.4 Loop Function

Steps:

Step 1. Set up solution flow.

The overall solution flow is shown below.

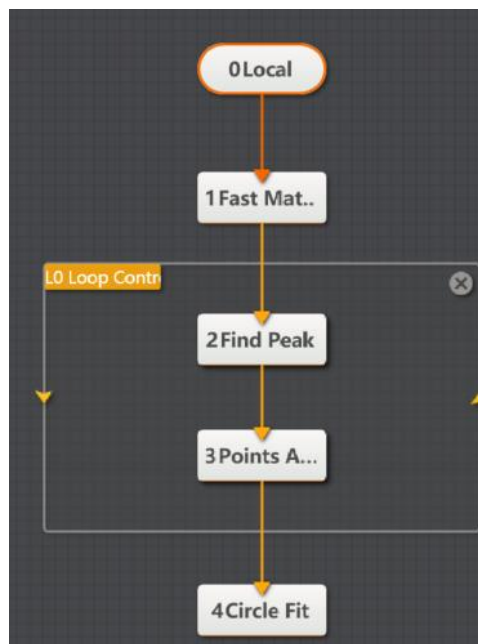


Figure 10.27 The overall solution flow

Step 2. Set loop parameters.

The settings of loop parameters must have circle start value and circle end value, as shown below.

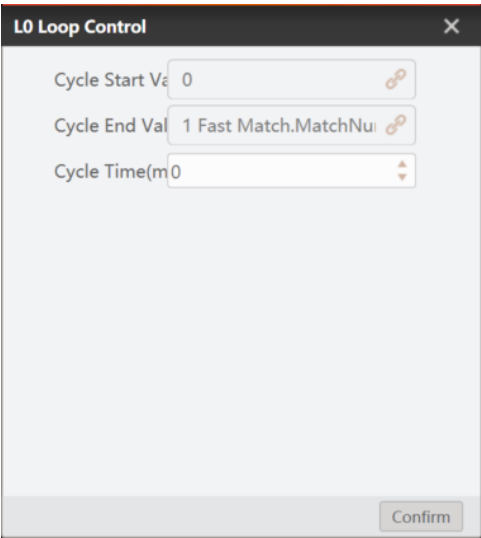


Figure 10.28 Loop control

Step 3. Set the parameters of finding peak.

The tool of finding peak should set up standards according to loop times, as shown below.

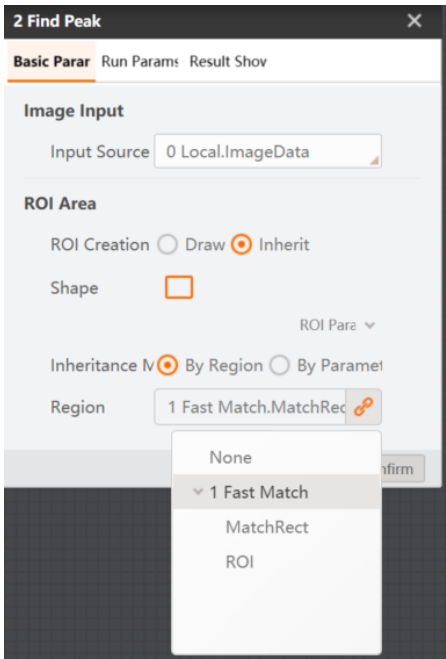


Figure 10.29 Set the parameters of finding peak

Step 4. Display result.

The result is shown below after running the loop control tool.

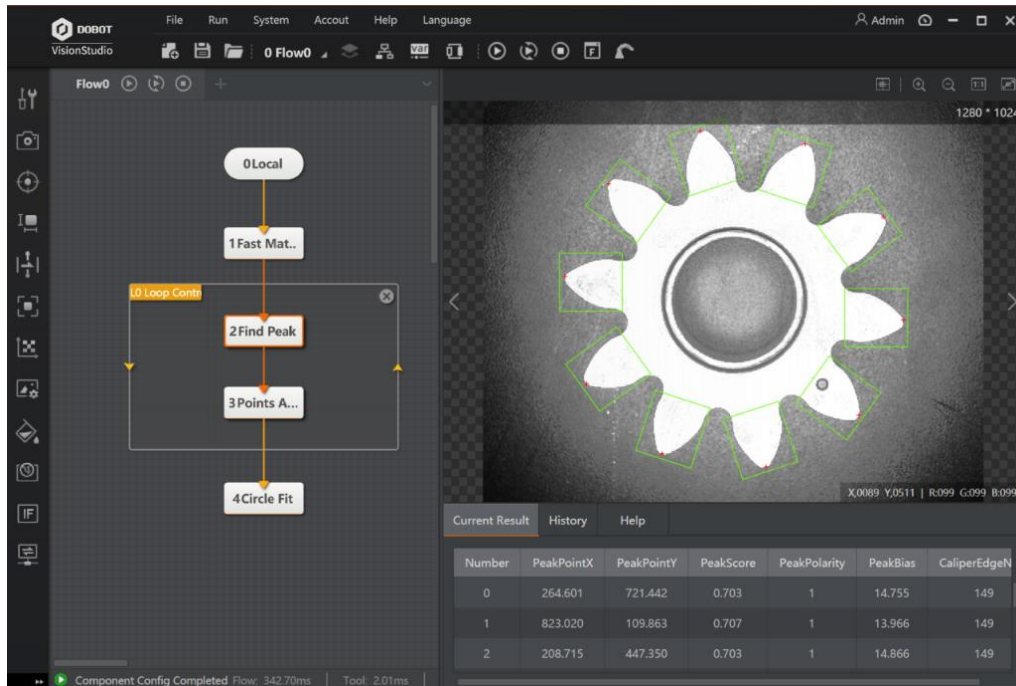


Figure 10.30 The result

10.5 Script Function

The script function provides input interface, and it transmits to output via C# programmer process.

Steps:

Step 1. Set up solution flow.

Find circle output diameter by template matching, as shown below.

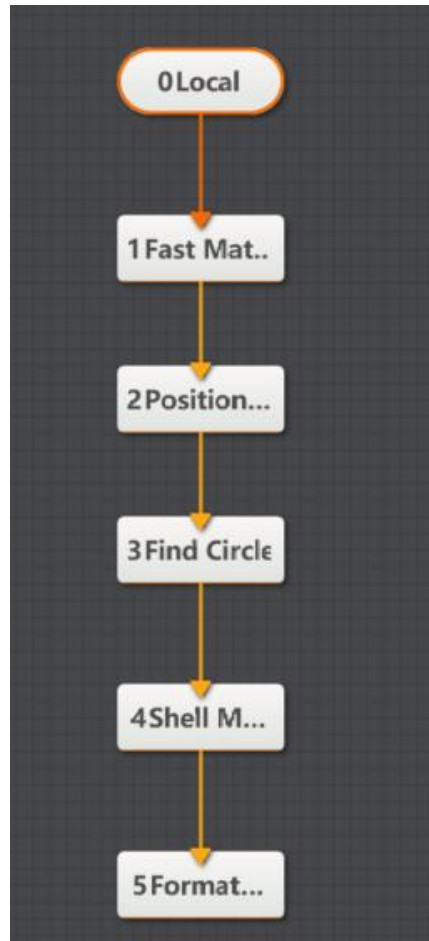


Figure 10.31 Solution flow

Step 2. Edit script.

Double-click the script module and enter editing page, as shown below.

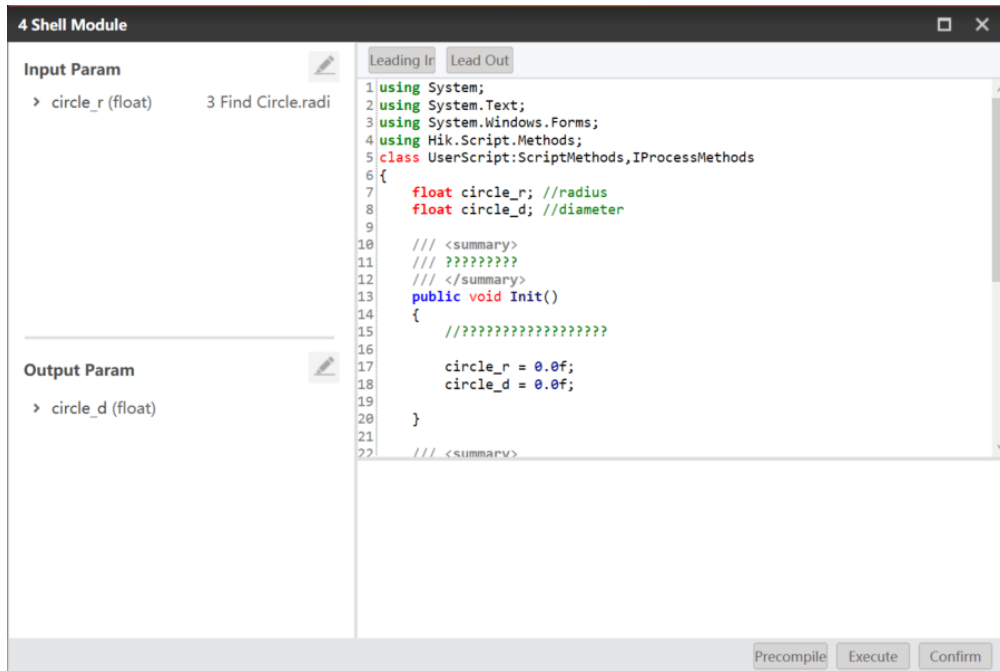


Figure 10.32 Edit script

You can add new variable by clicking +. The variable has two types, including int and float. You can set value in original value, as shown below.

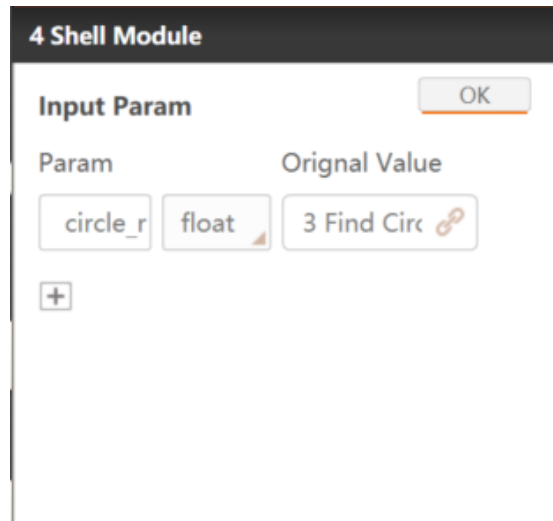


Figure 10.33 Shell module

The main body of the script is divided into three parts, as shown below.

- The first part defines the global variable, and it only runs during first time. This solution should define an integer variable and a diameter variable, as shown below.

```

using System;
using System.Text;
using System.Windows.Forms;
using Hik.Script.Methods;
class UserScript:ScriptMethods,IProcessMethods
{
    float circle_r; //radius
    float circle_d; //diameter

    /// <summary>
    /// ????????
    /// </summary>
    public void Init()
    {
        //????????????????????

        circle_r = 0.0f;
        circle_d = 0.0f;
    }
}

```

Figure 10.34 Global variable

- The second part defines the initial variable, and it only runs during first time. This solution should define an initial integer variable and a diameter variable, as shown below.

```

/// <summary>
/// ????????Process??
/// </summary>
/// <returns></returns>
public bool Process()
{
    //????????????????????????????
    // MessageBox.Show("Process Success");
}

```

Figure 10.35 initial variable

- The third part defines the main function area, and user can edit according to its demands. The detailed code of this solution is shown below.

```

GetFloatValue("circle_r",ref circle_r);
circle_d = circle_r*2;

//Gained the diameter
SetFloatValue("circle_d",circle_d);

return true;
}

```

Figure 10.36 Main function

10.6 Multi-Flow Communication

Data transmission communication between multiple processes is realized via data queue.

Steps:

- Step 1.** Create two new processes and set the data queue on the main flow interface, as shown below.

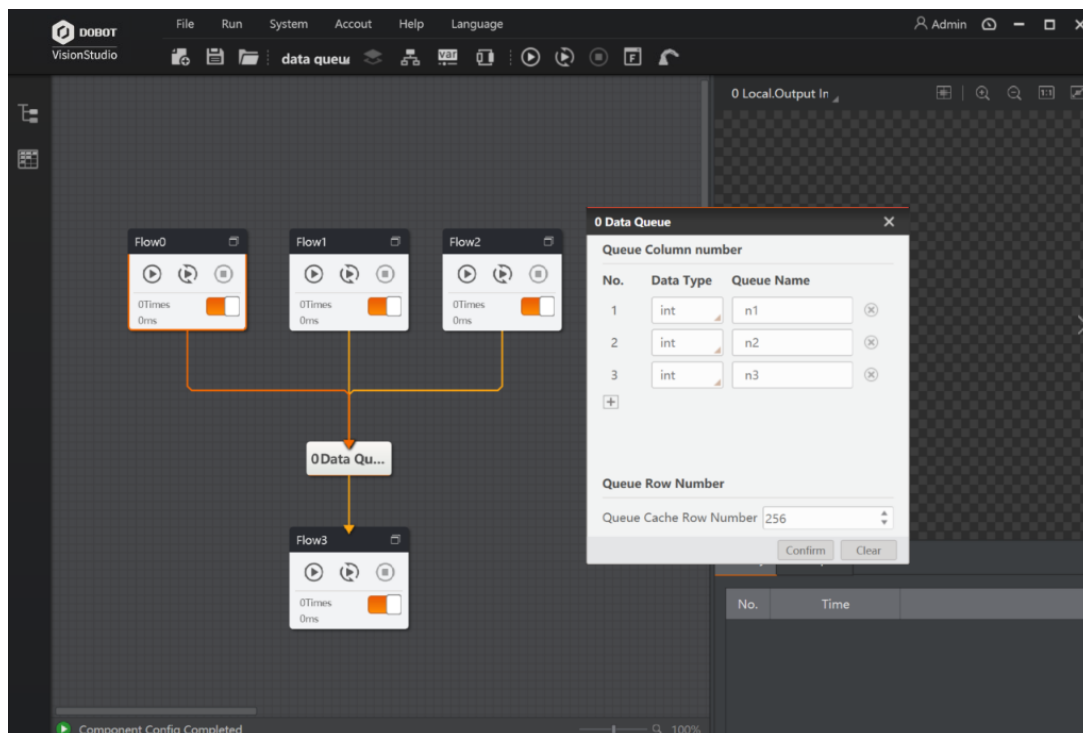


Figure 10.37 Create two new processes and set the data queue

Step 2. Set up the solution of finding circle, set the parameters of sending data module, and send the radius data obtained by finding circle to the queue 0 of the 0 data queue. The parameter settings are shown below.

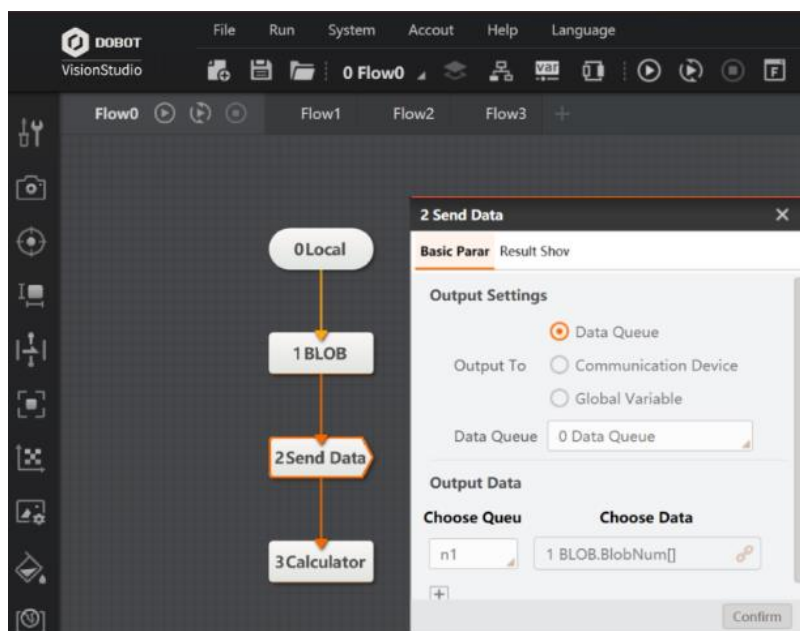


Figure 10.38 Set parameter

Step 3. Set up the solution of receiving data and set its parameters, as shown below.

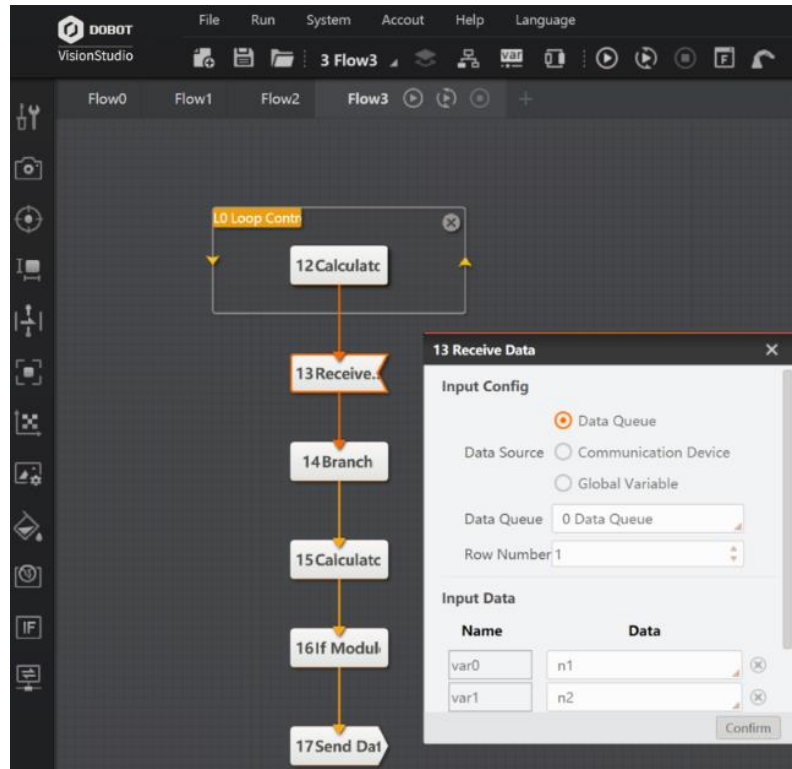


Figure 10.39 Set up the solution of receiving data and set its parameters

Step 4. The running results are shown below.

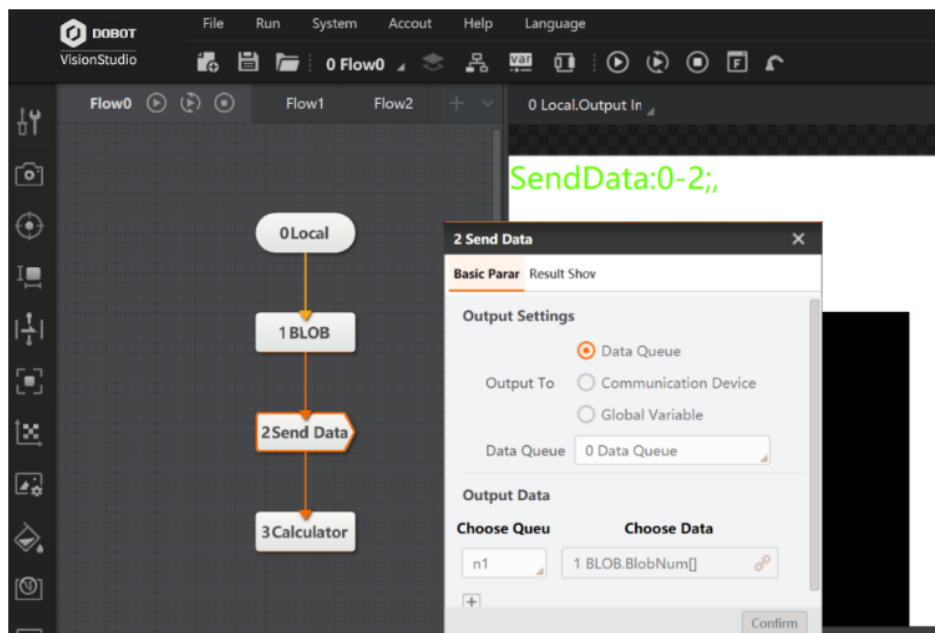


Figure 10.40 Result

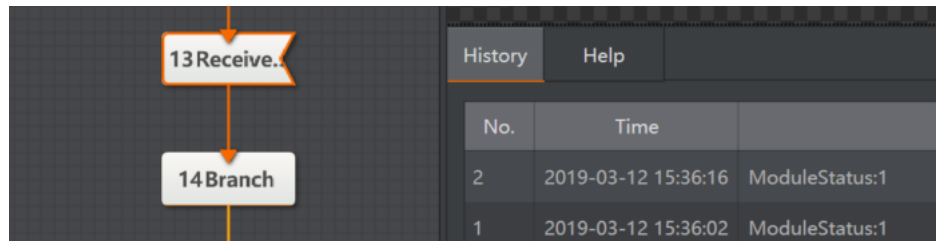


Figure 10.41 Result

Chapter 11 Dobot Magician Examples

The examples in this topic need to use Dobot Magician, for Dobot Magician details, please refer to *Dobot Magician User Guide*.

11.1 Robot Calibration

In this example, make calibration files by calibration tool.

Step 1. Overall solution flow.

The overall solution flow is shown below.

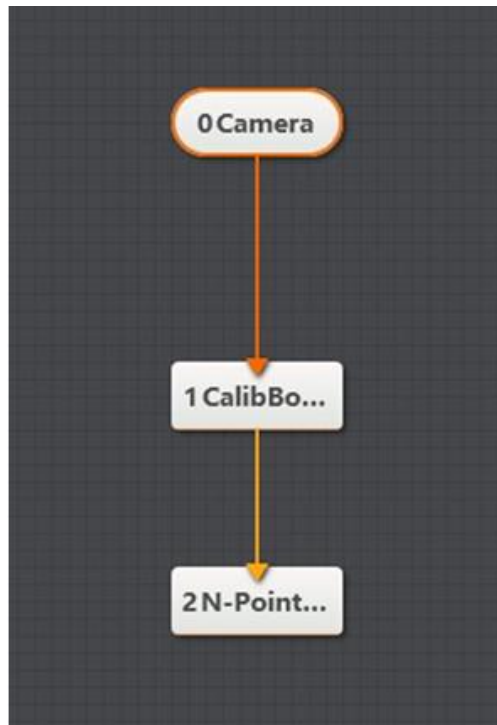


Figure 11.1 overall solution flow

Step 2. Select camera.

Select the camera that you want to use, as shown below.

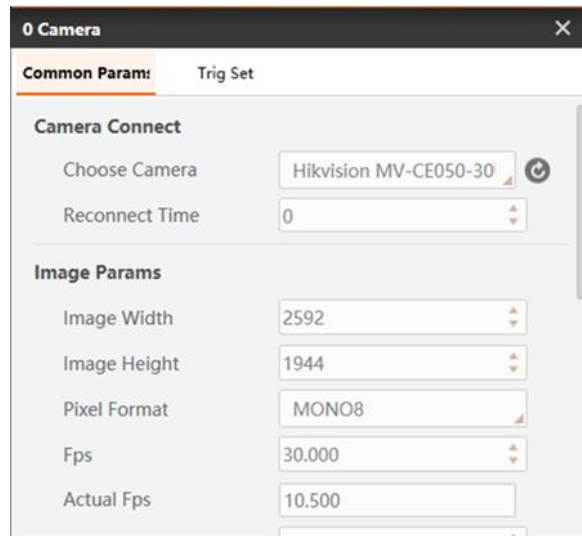


Figure 11.2 Select camera

Step 3. N-Point Calibration.

This example shows 9-points calibration.

1. Please select **pen** to be the end-effector of Dobot Magician, jog X-axis and Y-axis, as shown below.

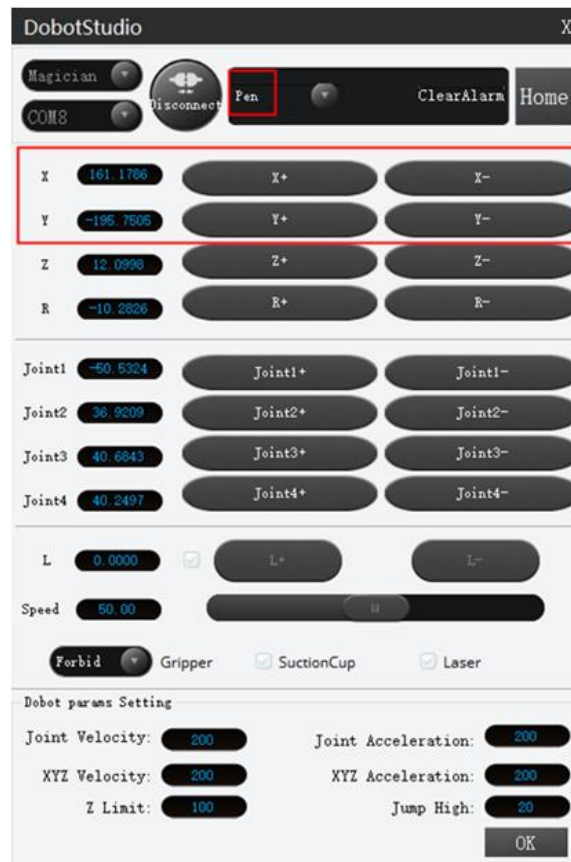


Figure 11.3 DobotStudio interface

2. Jog X-axis and Y-axis on DobotStudio to make the pen move to the 9 points in the order of the green arrow, and write down every point value according to the corresponding ID, and click **Confirm**.



Figure 11.4 N-Point calibration

3. Click **Create Calibration File** and set saving path.

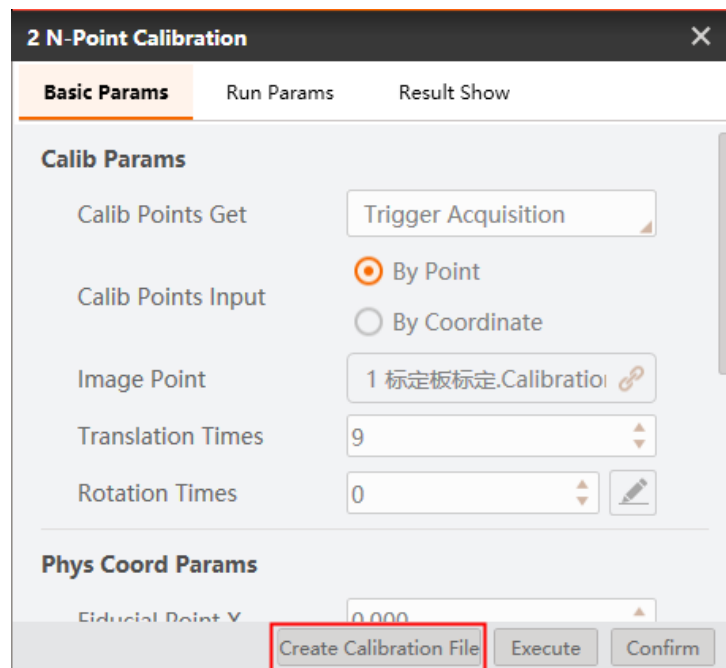


Figure 11.5 Create calibration file

11.2 Color Sort

Note: Calibration file needs to be generated beforehand.

In this example, detect the color of blocks and sort them, as shown below.



Figure 11.6 Detection example

Step 1. Solution establishment idea.

- Extract the sorted color in the camera image.
- Convert image coordinate to physical coordinate.
- Sort the specified color blocks by Dobot Magician.

The overall solution flow is shown below.

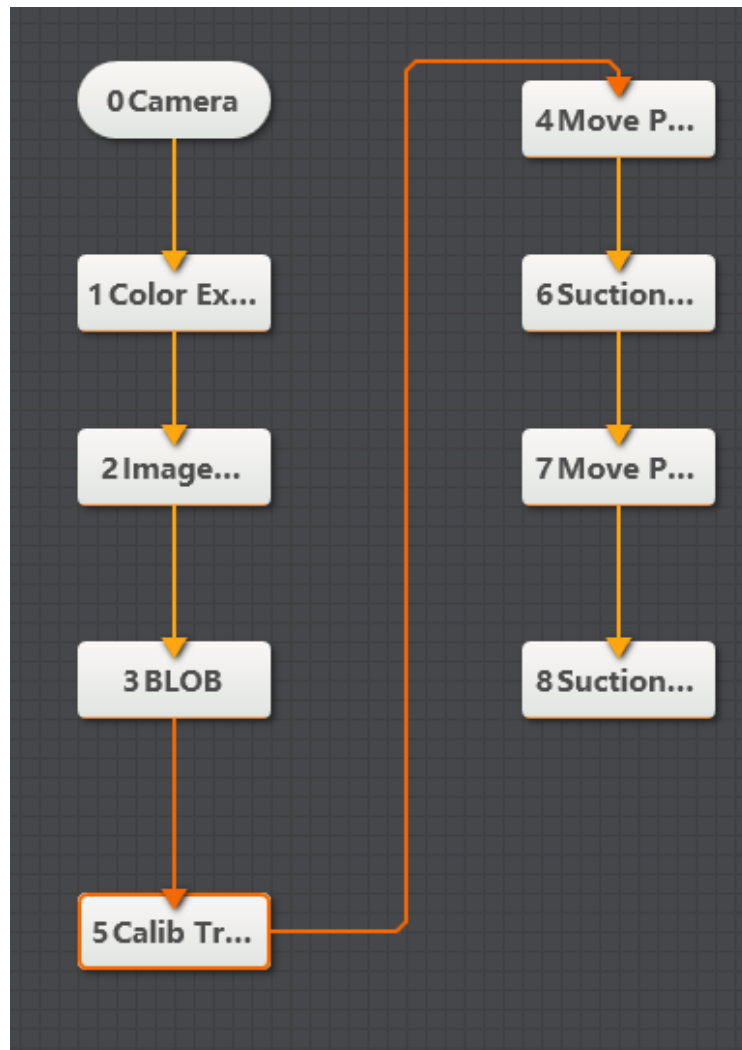


Figure 11.7 The overall solution flow

Step 2. Extract color.

Select extraction range of three channels in HSV space. After image processing, the image after color extraction is created, as shown below.

1 Color Extraction

Basic Params

Run Params

Result Show

Run Params

Color Space	HSV
Channel 1 Lower Limit	98
Channel 1 Upper Limit	204
Channel 2 Lower Limit	47
Channel 2 Upper Limit	255
Channel 3 Lower Limit	0
Channel 3 Upper Limit	255

Execute

Confirm

Figure 11.8 Set color

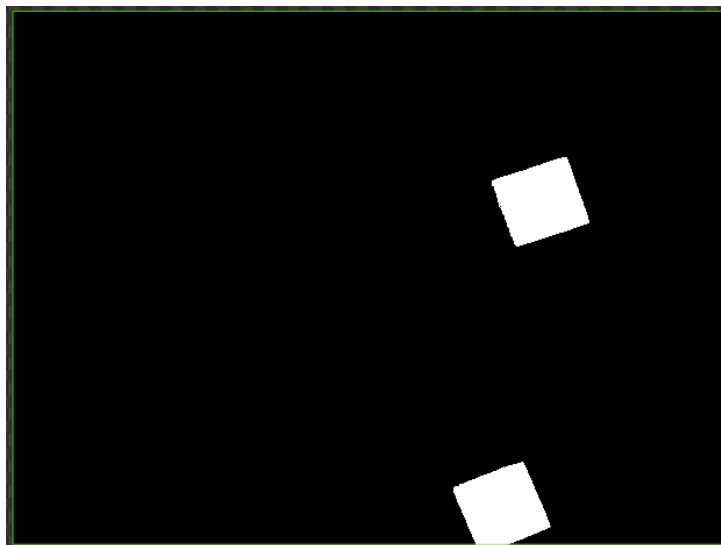


Figure 11.9 Result show

Step 3. Convert image coordinate to physical coordinate

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

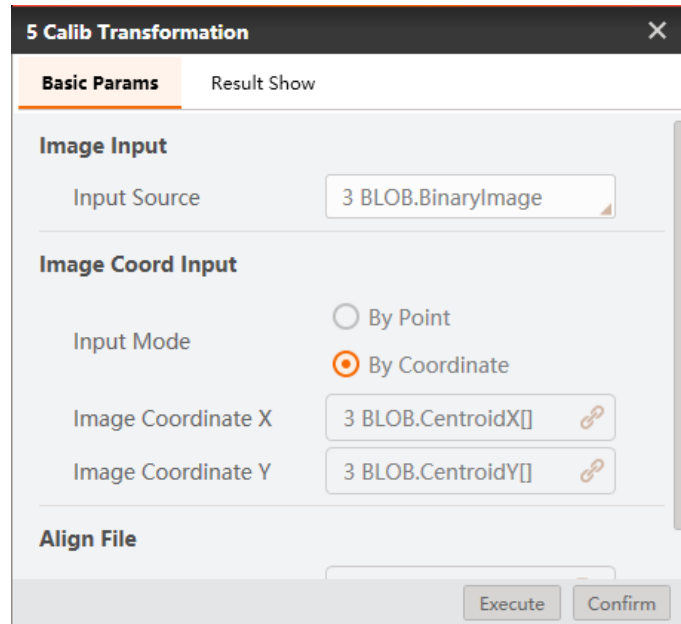


Figure 11.10 Calibration transformation

Step 4. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates, and set the Z-axis value based on site requirements.

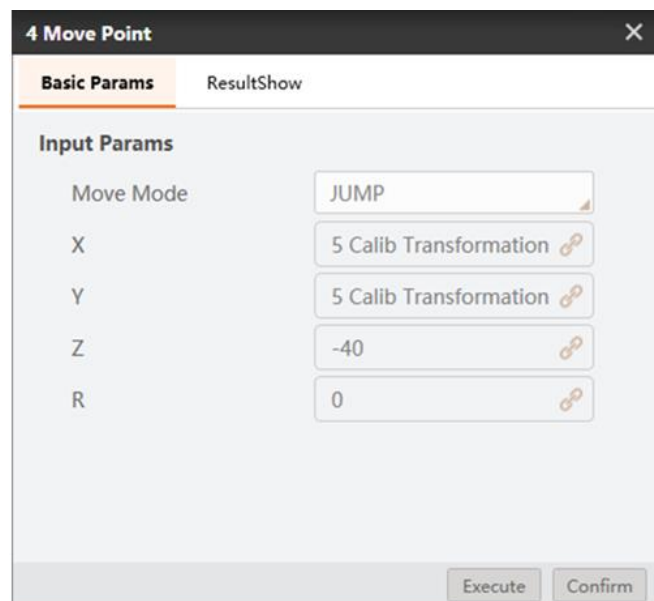


Figure 11.11 Move to a point

Step 5. Set suction cup

Set suction cup to suck up or release based on site requirements.

6 SuctionCup [X]

Basic Params ResultShow

Input Params

ON/CLOSE On

Execute Confirm

Figure 11.12 set suction cup

Step 6. Set placing positions.

Set block placing positions according to different color.

7 Move Point [X]

Basic Params ResultShow

Input Params

Move Mode JUMP

X 120 [Link]

Y -170 [Link]

Z -10 [Link]

R 0 [Link]

Execute Confirm

Figure 11.13 set placing positions

11.3 Character Defect Detection

In this example, detect whether characters have defects like characters missing and redundant.

Note: Calibration file needs to be generated beforehand.



Figure 11.14 Detection example

Step 1. Solution establishment idea.

- Create the template of the standard characters and train them. Compare the trained characters with the target characters to find detects.
- Analyze the result to sort characters with Dobot Magician.

The overall solution flow is shown below.

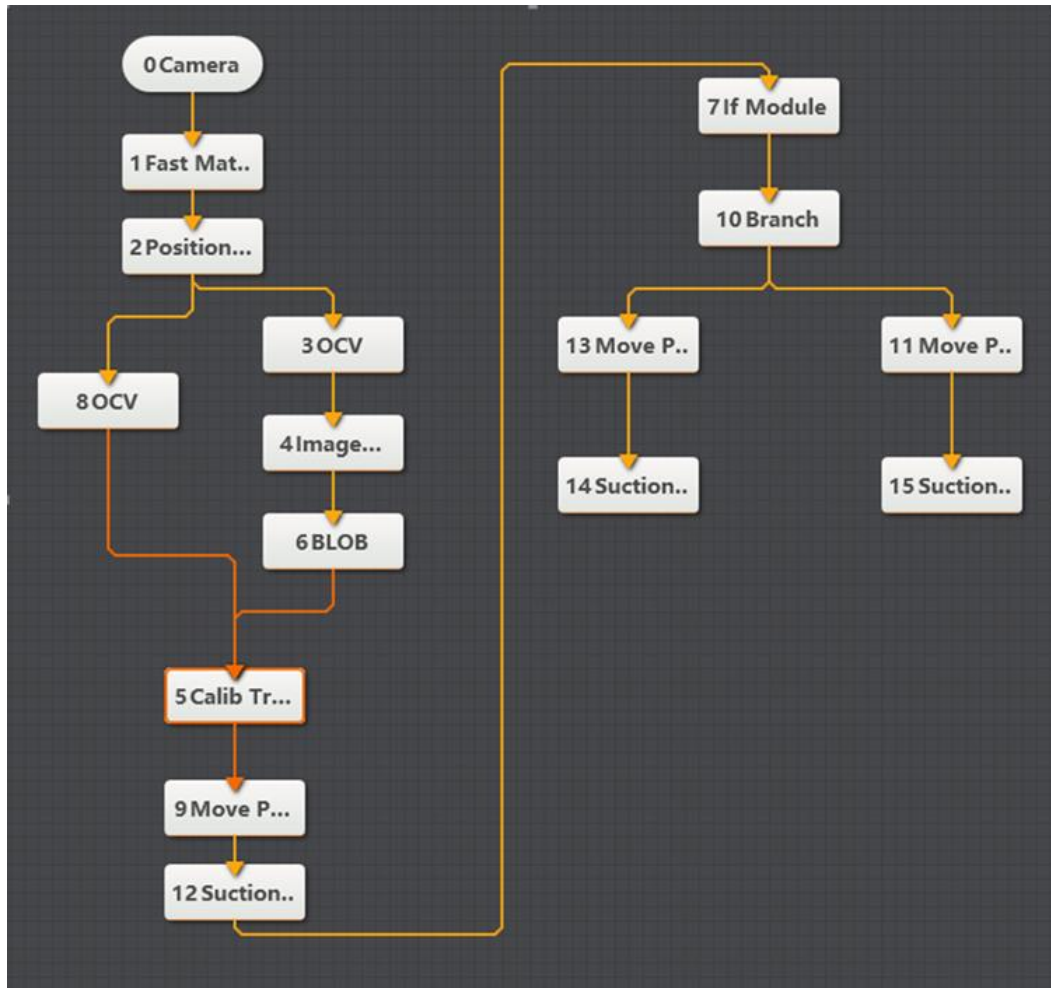


Figure 11.15 The overall solution flow

Step 2. Feature match.

Create feature template by feature matching module, as shown below .You need to select all characters completely, you can set **Scale Mode** to **Manual** to select manually, and then set the **Compressibility** and the **Contrast Threshold** based on site requirements.

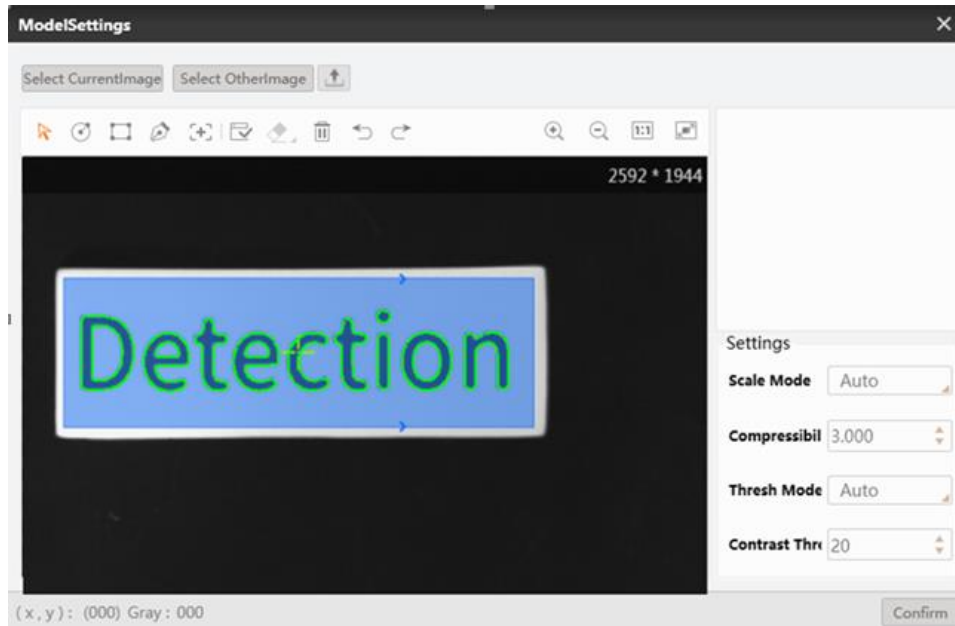


Figure 11.16 Feature match

Step 3. Position correction.

Create the reference of position deviation according to the match points and match frame angle in the template matching result. Click **Create reference**, as shown below.

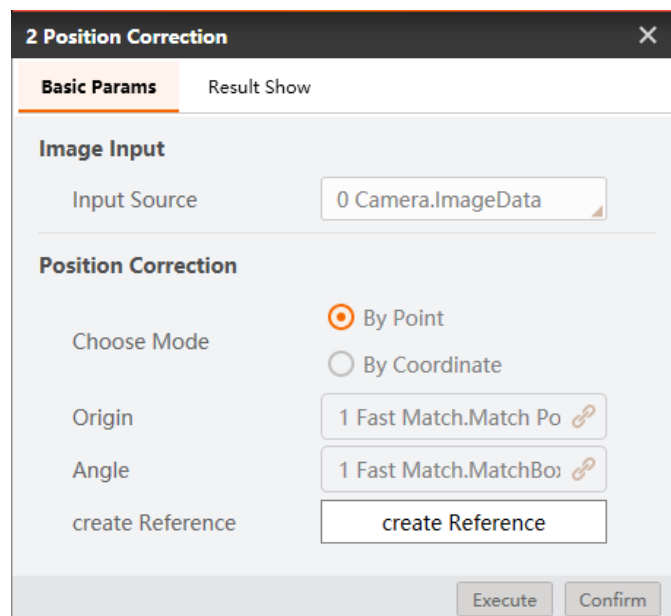


Figure 11.17 Position correction

Step 4. Train model.

Click **Create model** to create character template. Click **Character Extraction**. If characters are not extracted completely, please extract them manually, as shown below. The extraction is shown below.

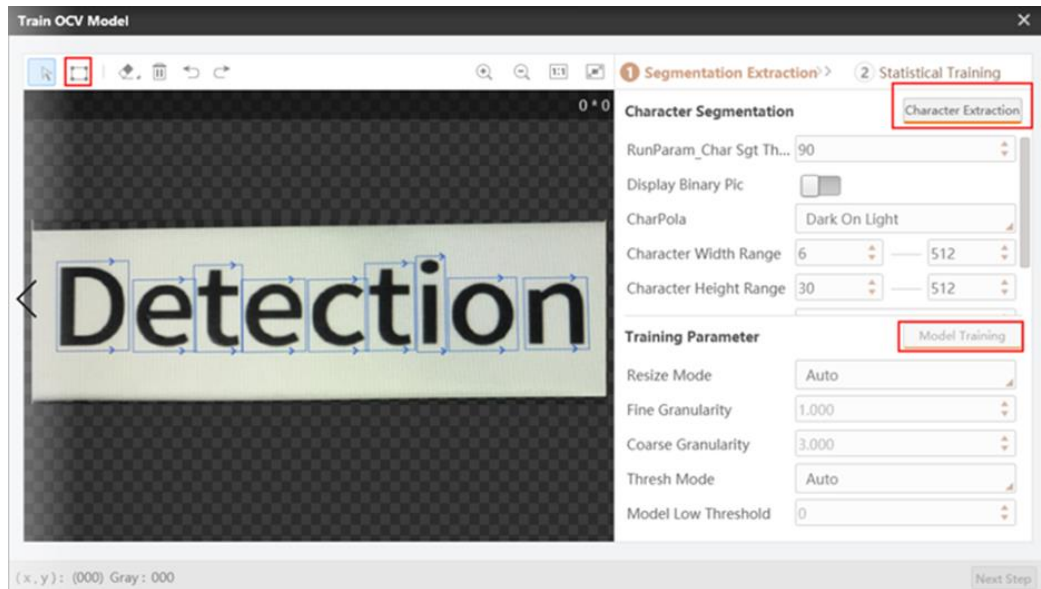


Figure 11.18 Train model

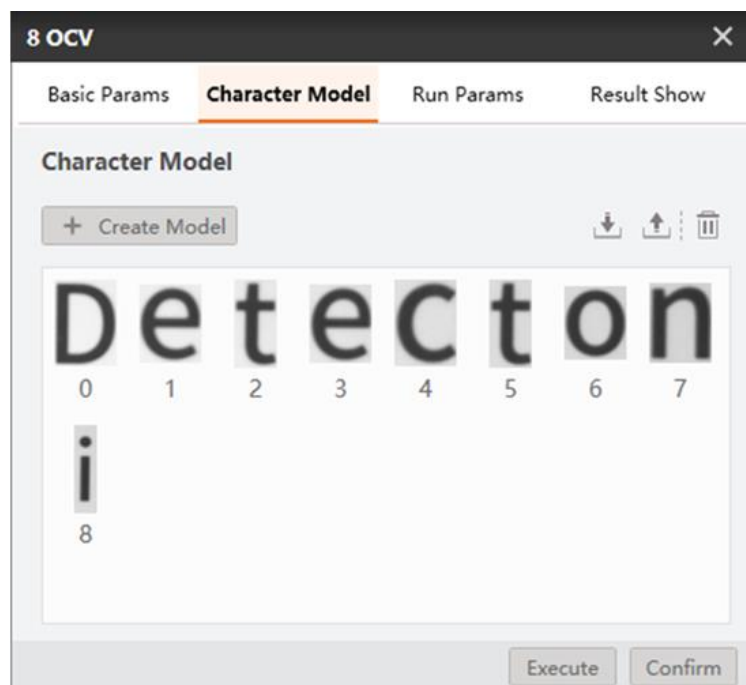


Figure 11.19 The result model

Step 5. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**, as shown below.

5 Calib Transformation

Basic Params Result Show

Image Input

Input Source: 0 Camera.ImageData

Image Coord Input

Input Mode: ☒ By Point ☐ By Coordinate

Image Point: 2 Position Correction.L

Align File

Load Calibration File: [Empty]

Execute Confirm

Figure 11.20 Calibration convert

Step 6. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates, and set the Z-axis value based on site requirements.

9 Move Point

Basic Params ResultShow

Input Params

Move Mode: JUMP

X: 5 Calib Transformation

Y: 5 Calib Transformation

Z: -65

R: 0

Execute Confirm

Figure 11.21 Move to a point

Step 7. Set suction cup.

Set suction cup to suck up or release based on site requirements.

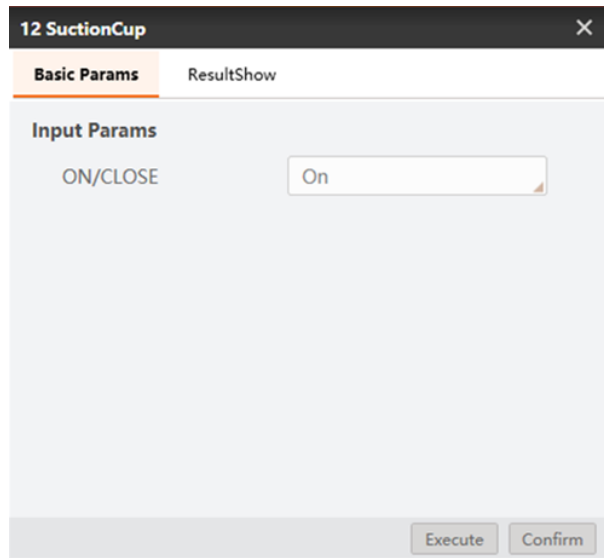


Figure 11.22 Set suction cup

Step 8. Condition detection.

Return the result value 0 or 1 by judging the number of defective characters. 0 means that there are character defects, 1 means no character defects.

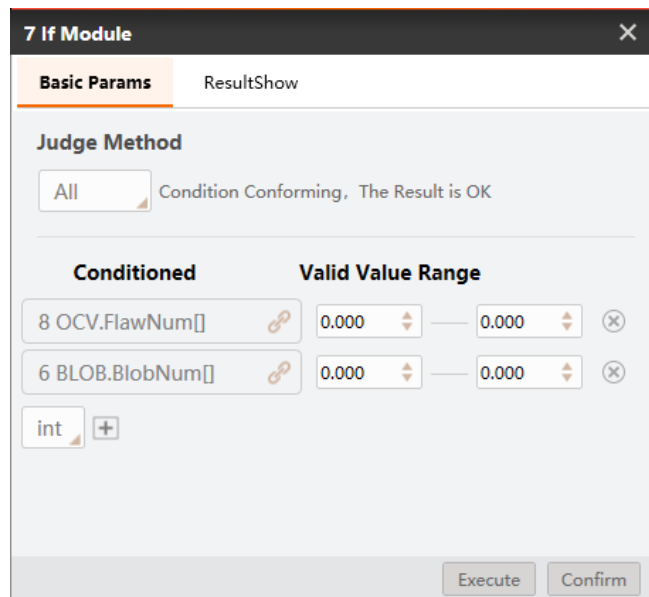


Figure 11.23 Condition detection

Step 9. Branch module.

Execute the different branch module according to the detection results. If the result is 1, execute branch 13, otherwise, execute branch 11.

10 Branch

Input Params

Condition Input: 7 If Module.nIfResult[]

Branch Params

☒ Index by Value ☐ Index by Bit

Branch	Condition Input Value
13	1
11	0

Execute Confirm

Figure 11.24 Branch module

Step 10. Set placing positions.

Set two sort positions according to detection results.

13 Move Point

Basic Params ResultShow

Input Params

Move Mode: JUMP

X	180
Y	-180
Z	0
R	0

Execute Confirm

Figure 11.25 Set sort position

Step 11. Detection results.

The detection results are shown below.



Figure 11.26 NG



Figure 11.27 OK

11.4 Measure and Sort

In this example, measure the diameter of Round-shaped objects and sort them, as shown below.

Note: Calibration file needs to be generated beforehand.



Figure 11.28 Example

Step 1. Solution establishment idea

- To find the measured Round-shaped objects by find circle module.
- Convert image size to physical size by scale transformation.
- Convert image coordinate to physical coordinate by calibration transformation.
- Sort Round-shaped objects by Dobot Magician.

The overall solution flow is shown below.

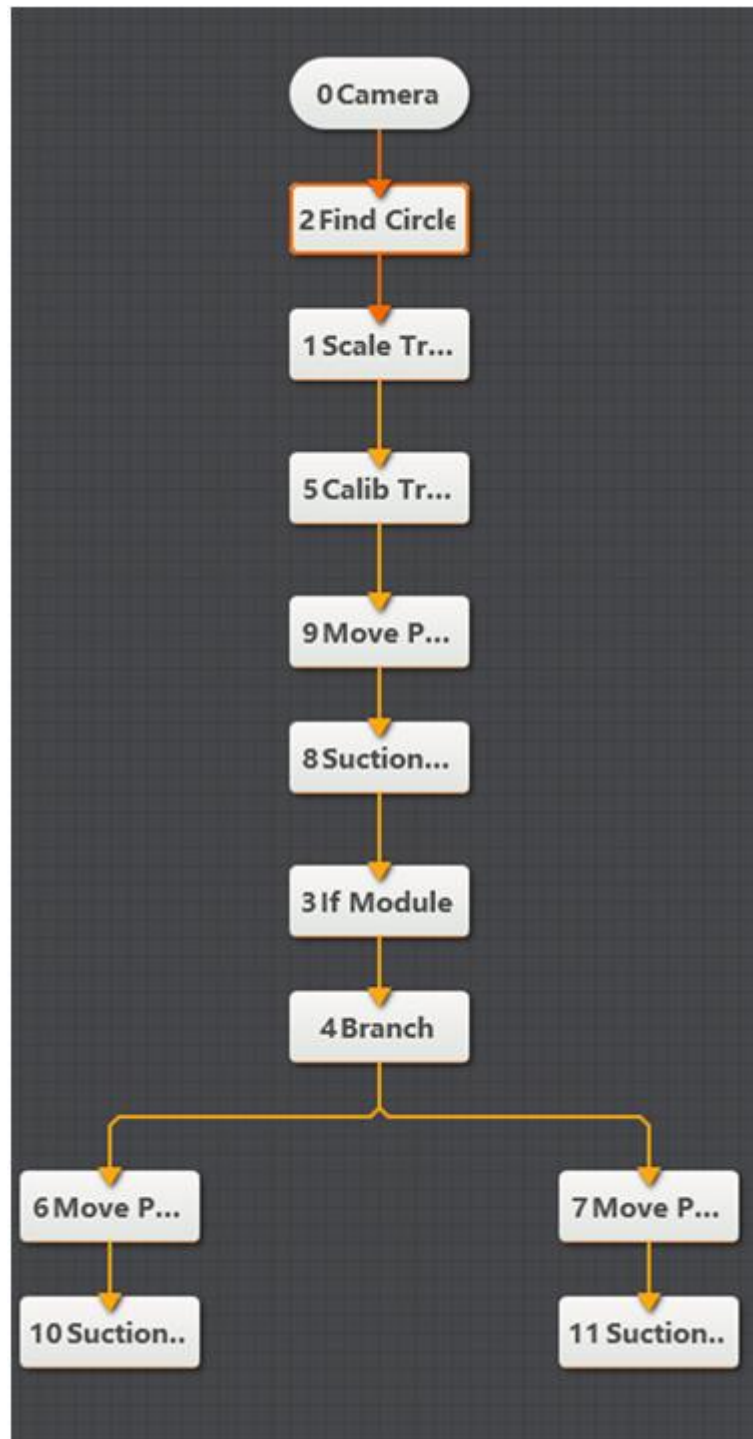


Figure 11.29 The overall solution flow

Step 2. Find circle

Set radius range to find circle.

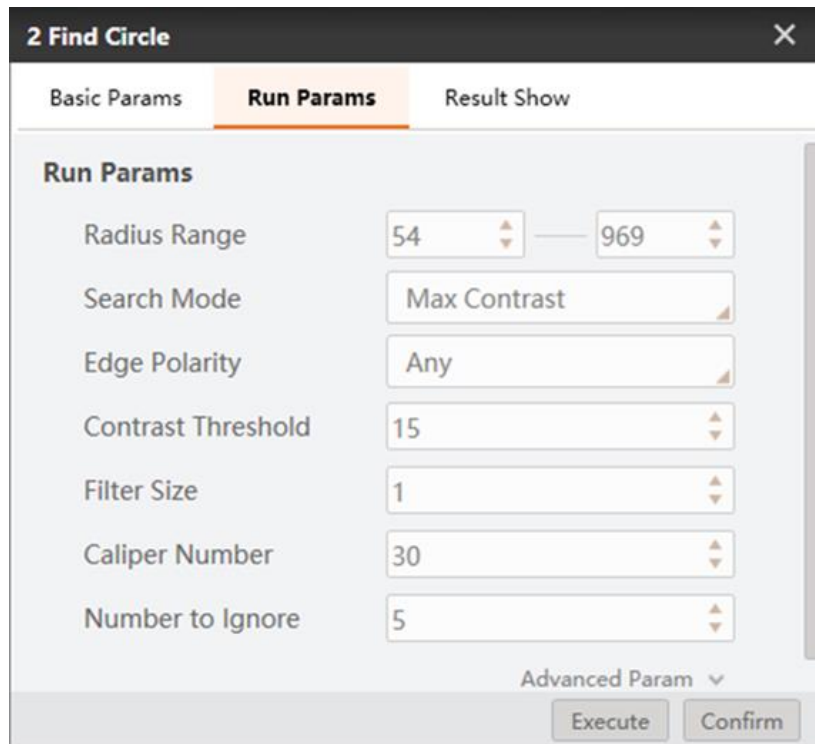


Figure 11.30 Find circle

Step 3. Scale transformation

Load calibration file to convert pixel unit to physical unit by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

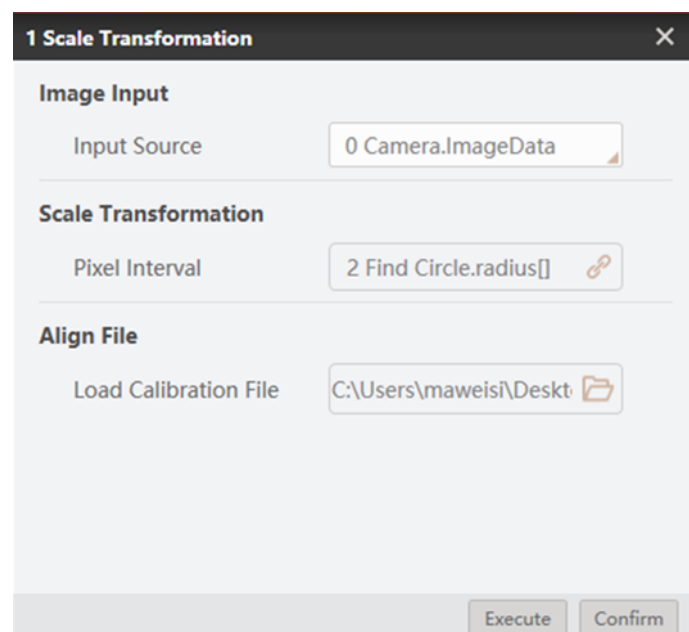


Figure 11.31 Convert unit

Step 4. Convert image coordinate to physical coordinate

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

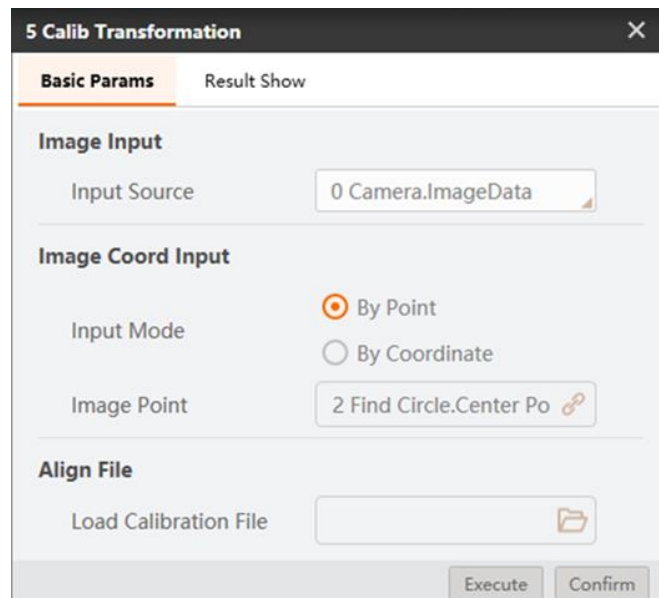


Figure 11.32 Calibration convert

Step 5. Condition detection

Set radius range. As shown below.

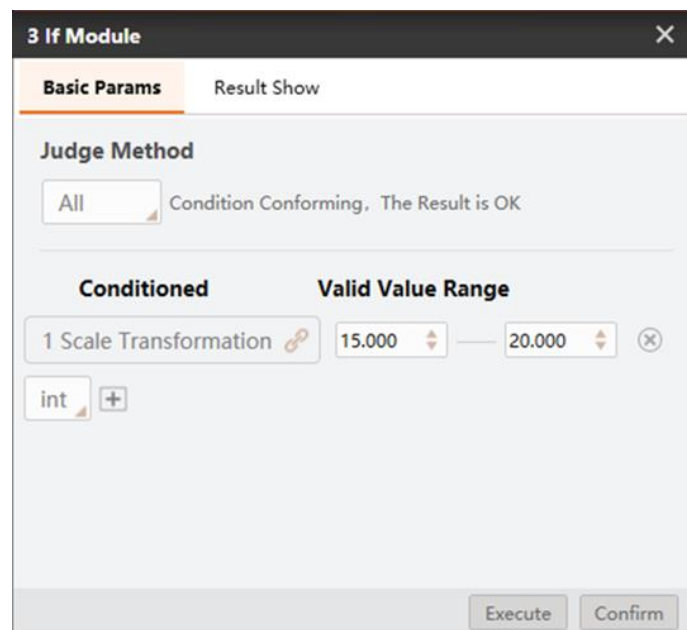


Figure 11.33 Condition detection

Step 6. Branch module.

Execute the different branch modules according to the measured results. If result is 1, execute branch 6, otherwise execute branch 7.

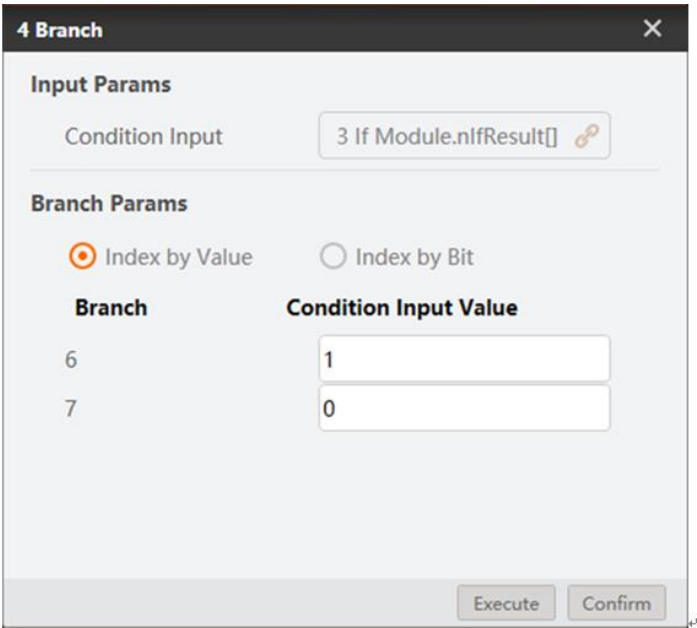


Figure 11.34 Branch module

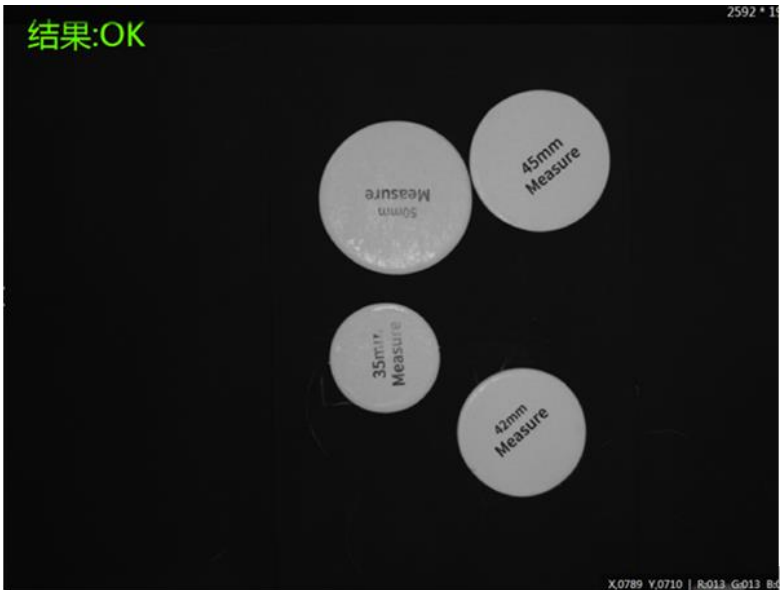


Figure 11.35 OK

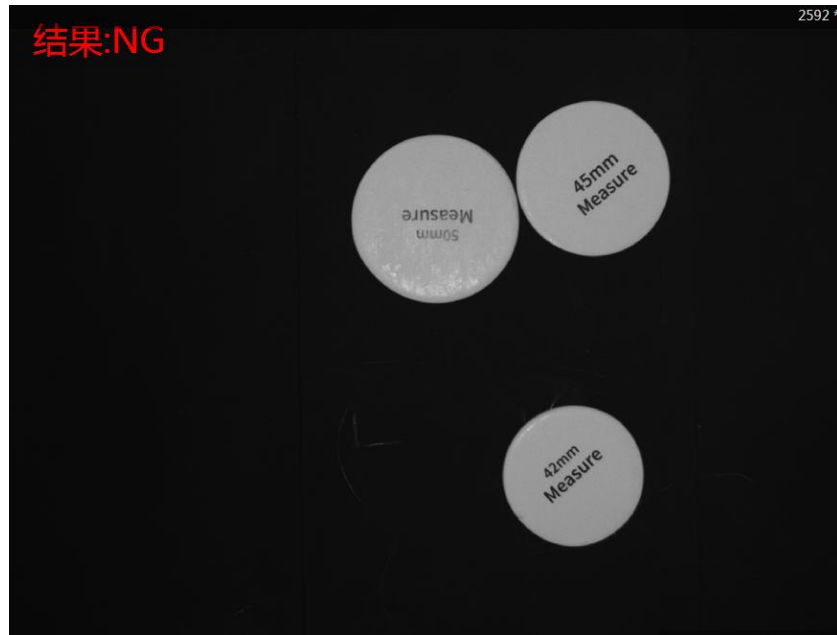


Figure 11.36 NG

Step 7. Set suction cup.

Set suction cup to suck up or release based on site requirements.

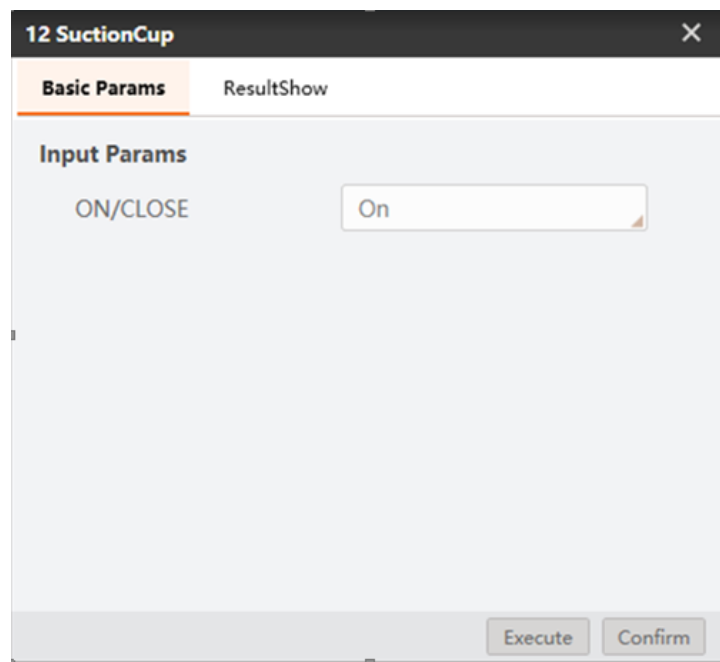
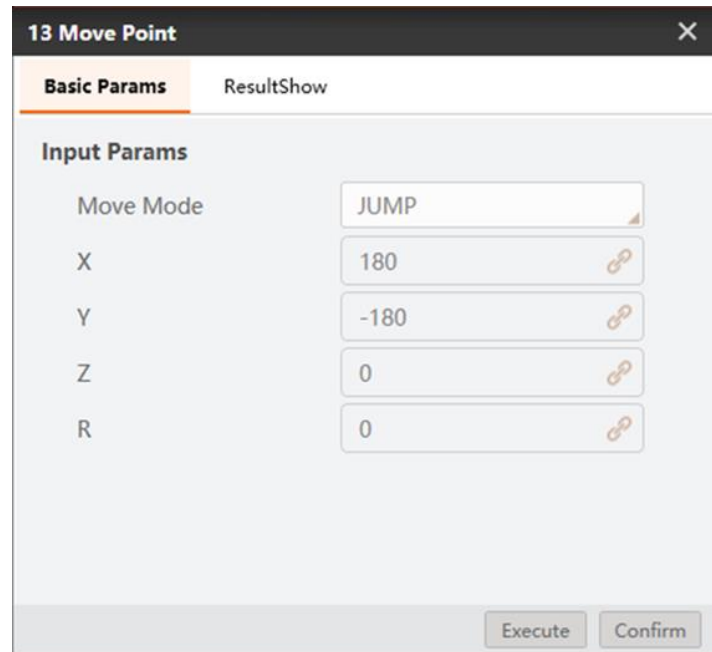


Figure 11.37 Set suction cup

Step 8. Set placing positions.

Set round-shaped objects placing positions according to the measured results.



The screenshot shows a software interface titled "13 Move Point" with a close button (X) in the top right corner. Below the title bar, there are two tabs: "Basic Params" (which is selected and highlighted with an orange underline) and "ResultShow". Under the "Basic Params" tab, the section "Input Params" contains five rows of input fields:

Parameter	Value	Link Icon
Move Mode	JUMP	
X	180	⛶
Y	-180	⛶
Z	0	⛶
R	0	⛶

At the bottom right of the dialog, there are two buttons: "Execute" and "Confirm".

Figure 11.38 Set sort position

11.5 RectangleTemplate Match

In this example, find rectangles and put them to the corresponding template, as shown below.

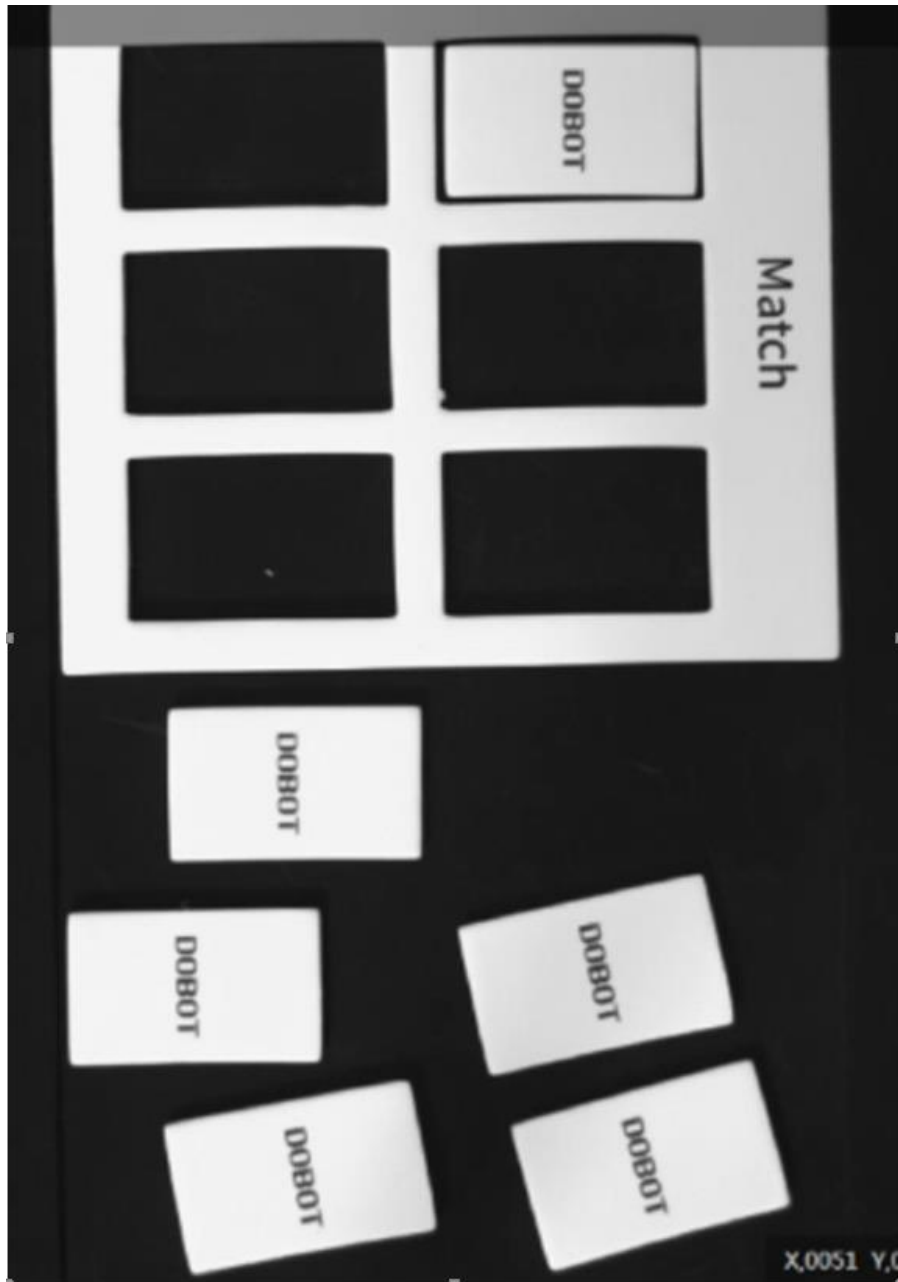


Figure 11.39 Rectangle match

Step 1. Solution establishment idea

- Create feature template for rectangle and rectangle model by feature matching module.
- Find rectangle and rectangle model by finding line module to match them.

The overall solution flow is shown below.

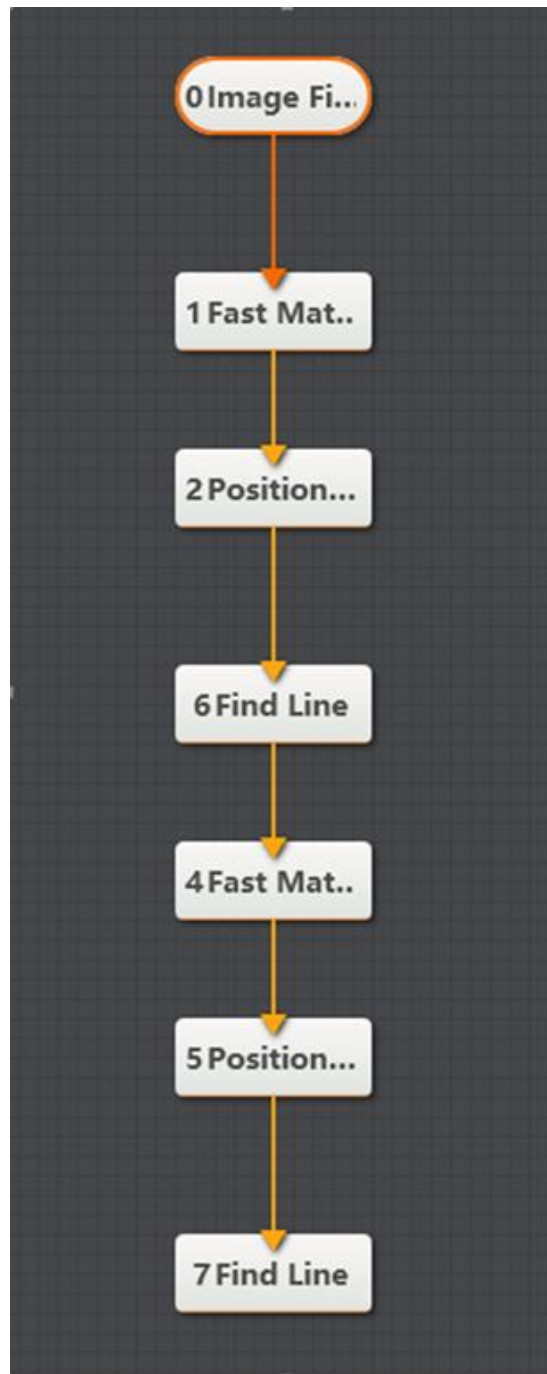


Figure 11.40 The overall solution flow

Step 2. Feature match.

Create feature template, as shown below .you need to select all characters completely, you can set **Scale Mode** to **Manual** to select manually. And then set the **compressibility** and **contrast threshold** based on site requirements.

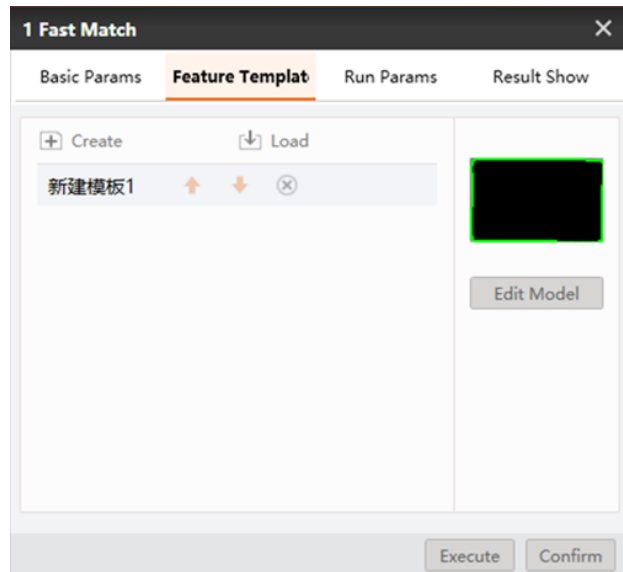


Figure 11.41 Rectangle model feature match

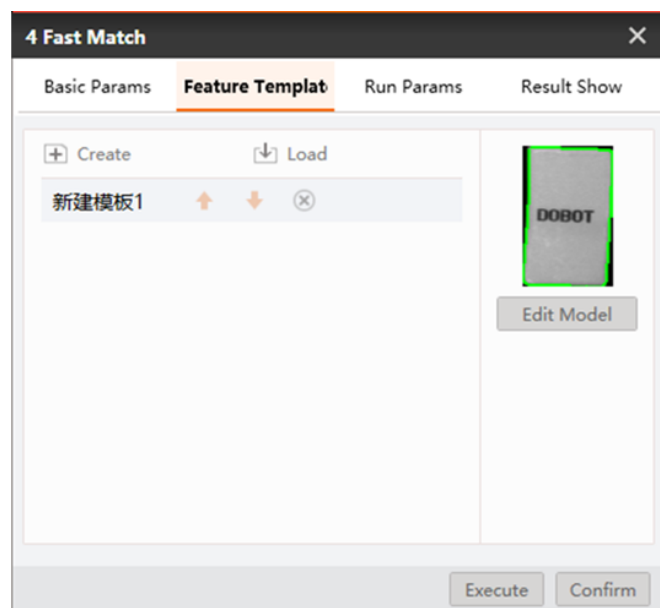


Figure 11.42 Rectangle feature match

Step 3. Position correction.

Create the reference of position deviation according to the match points and match frame angle in the template matching result. Click Create reference as shown below.

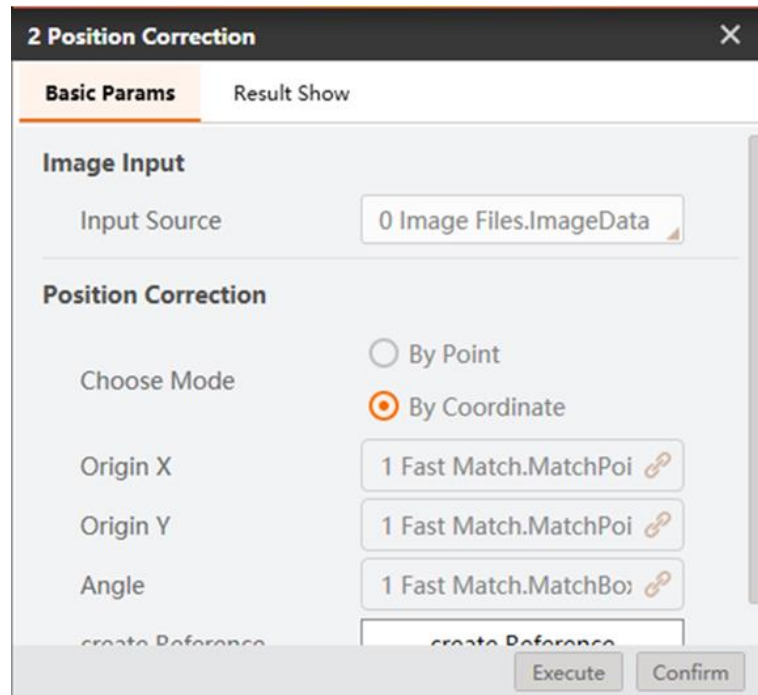


Figure 11.43 Rectangle model correction

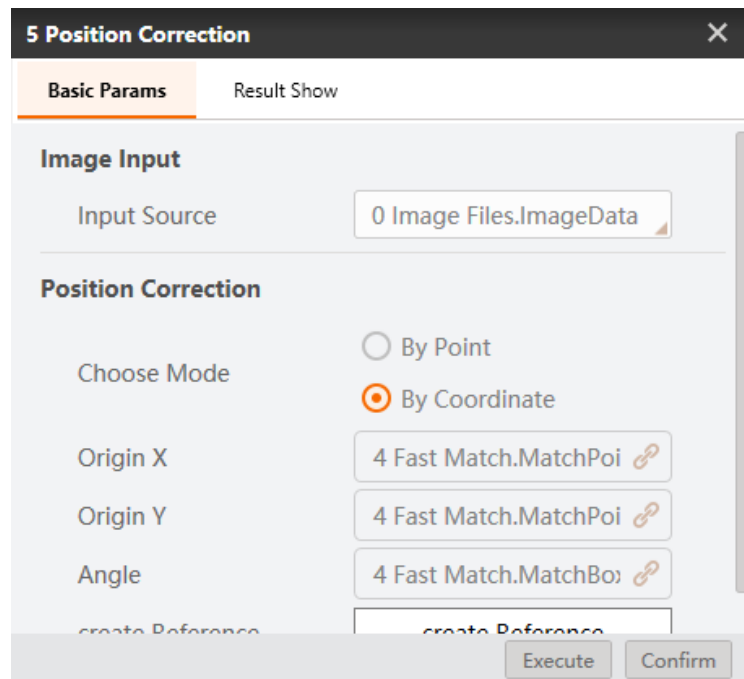


Figure 11.44 Rectangle correction

Step 4. Find line.

Set **Run Params** based on site requirements to find rectangle and rectangle model.

The image shows a software window titled "6 Find Line" with a close button (X) in the top right corner. The window has three tabs: "Basic Params", "Run Params" (which is selected and highlighted with an orange underline), and "Result Show". Below the tabs, the "Run Params" section contains several settings:

- Search Mode: A dropdown menu set to "Best Line".
- Search Direction: A dropdown menu set to "Top to Bottom".
- Edge Polarity: A dropdown menu set to "Any".
- Edge Strength: A numeric input field set to "5" with up and down arrow buttons.
- Edge Width: A numeric input field set to "1" with up and down arrow buttons.
- Caliper Number: A numeric input field set to "6" with up and down arrow buttons.
- Points to Ignore: A numeric input field set to "0" with up and down arrow buttons.

At the bottom right of the "Run Params" section, there is a label "Advanced Param" followed by a downward-pointing arrow. Below this, there are two buttons: "Execute" and "Confirm".

Figure 11.45 Find line

11.6 Circle Template Match

In this example, measure the diameter of circles and put them into the corresponding model as shown below.

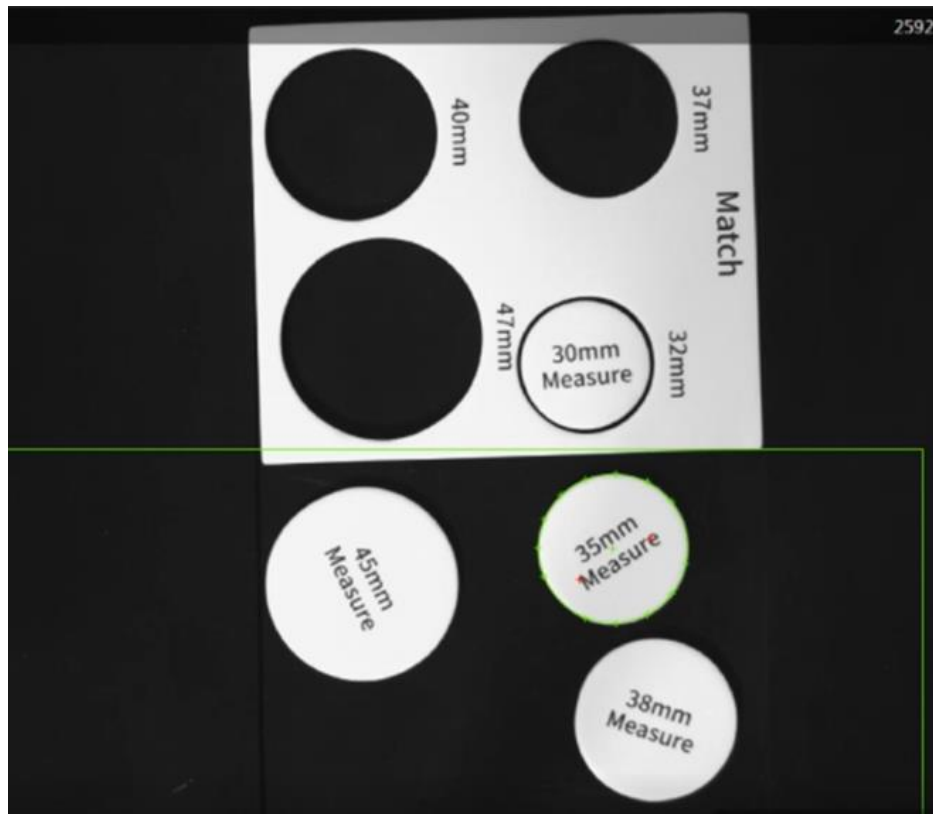


Figure 11.46 Circle match

Step 1. Solution establishment idea.

- Find the measured circle by finding circle module.
- Convert image coordinate to physical coordinate by calibration transformation.
- Put circles to corresponding circle models by Dobot Magician.

The overall solution flow is shown below.

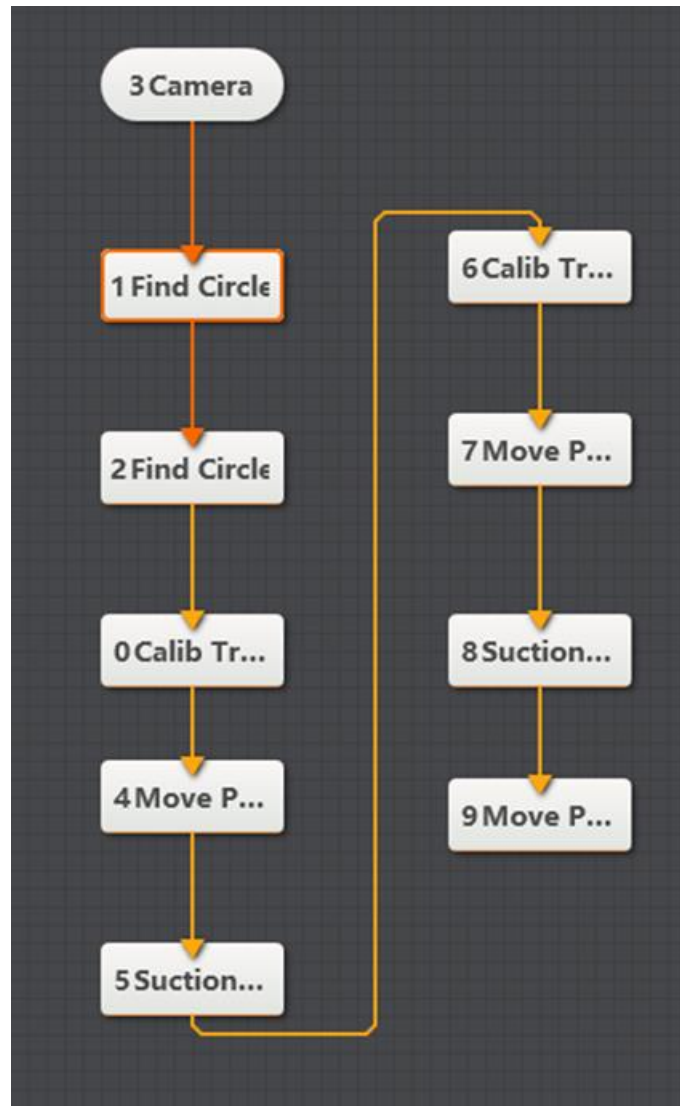


Figure 11.47 The overall solution flow

Step 2. Find circle.

Set finding circle modules for circle and circle model to set **Run Params**, such as **Radius Range**, **Search Mode**, and **Contrast Threshold**. As shown below.

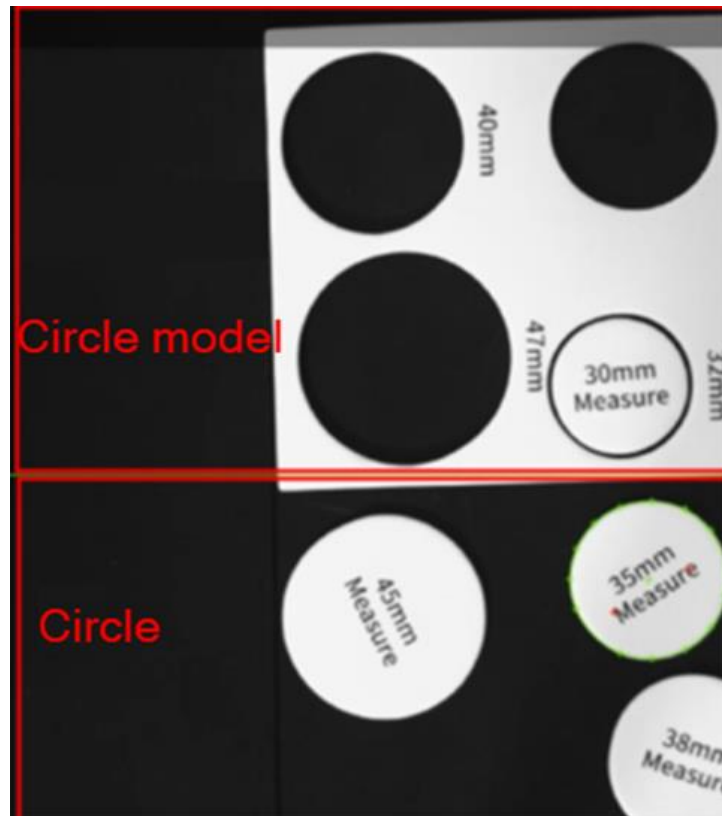


Figure 11.48 Find circle

Figure 11.49 Find circle module

Step 3. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

0 Calib Transformation

Basic Params Result Show

Image Input

Input Source: 3 Camera.ImageData

Image Coord Input

Input Mode: ☒ By Point ☐ By Coordinate

Image Point: 2 Find Circle.Center Po

Align File

Load Calibration File: [Empty field]

Execute Confirm

Figure 11.50 Calibration transformation

Step 4. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates according to different point.

4 Move Point

Basic Params ResultShow

Input Params

Move Mode: JUMP

X: 0 Calib Transformation

Y: 0 Calib Transformation

Z: -62

R: 0

Execute Confirm

Figure 11.51 Move to a point

Step 5. Set suction cup.

Set suction cup to suck up or release based on site requirements.

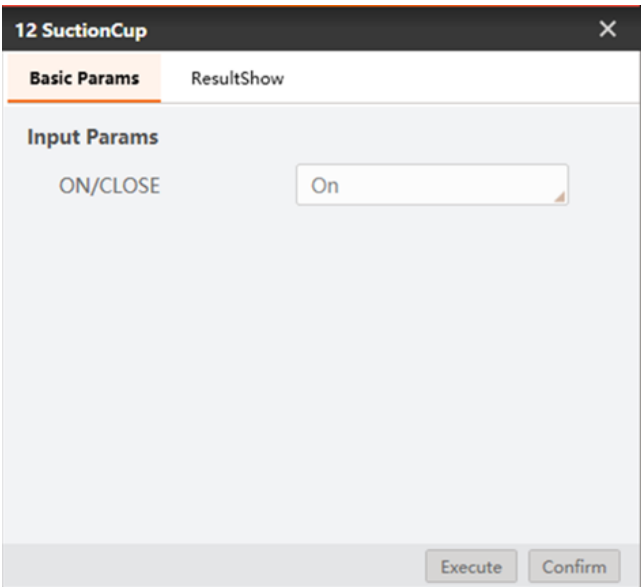


Figure 11.52 Set suction cup

Step 6. The matching result.

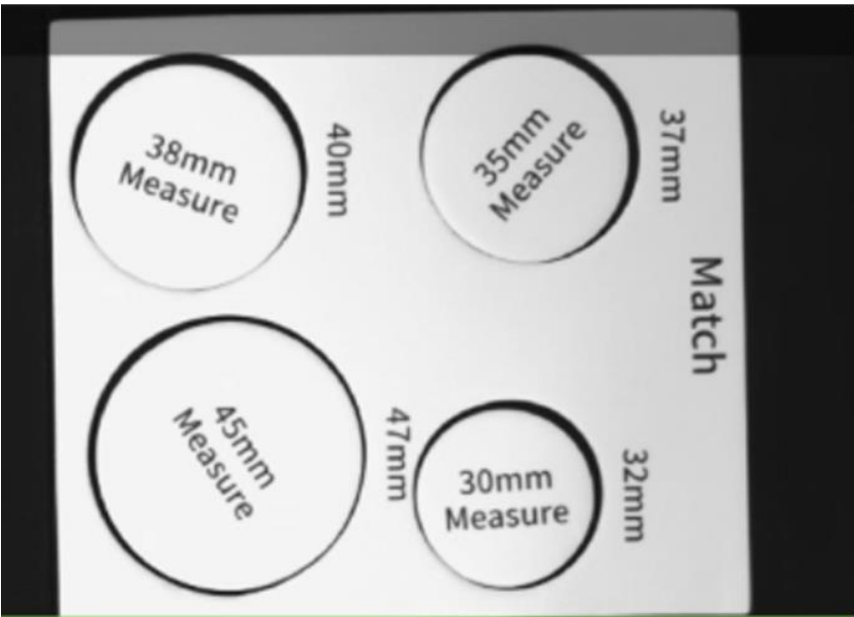


Figure 11.53 The matching result

Chapter 12 Dobot M1 Examples

The examples in this topic need to use Dobot M1, for Dobot M1 details, please refer to ***Dobot M1 User Guide***.

12.1 Robot Calibration

In this example, make calibration files by calibration tool.

Step 1. Overall solution flow.

The overall solution flow is shown below.

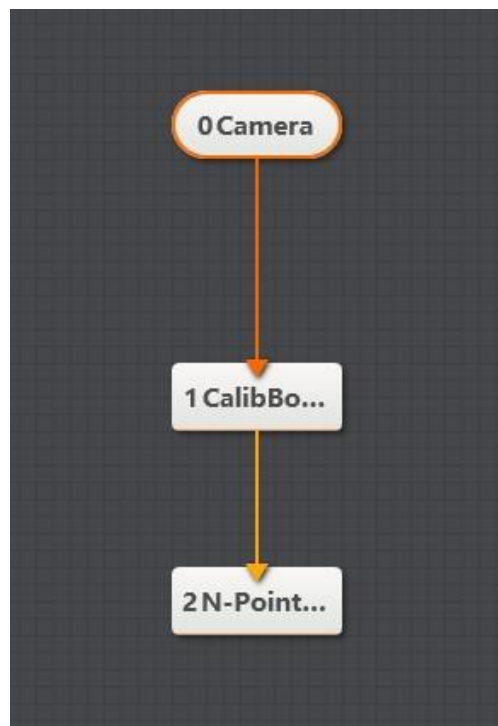


Figure 12.1 overall solution flow

Step 2. Select camera.

Select the camera that you want to use, as shown below.

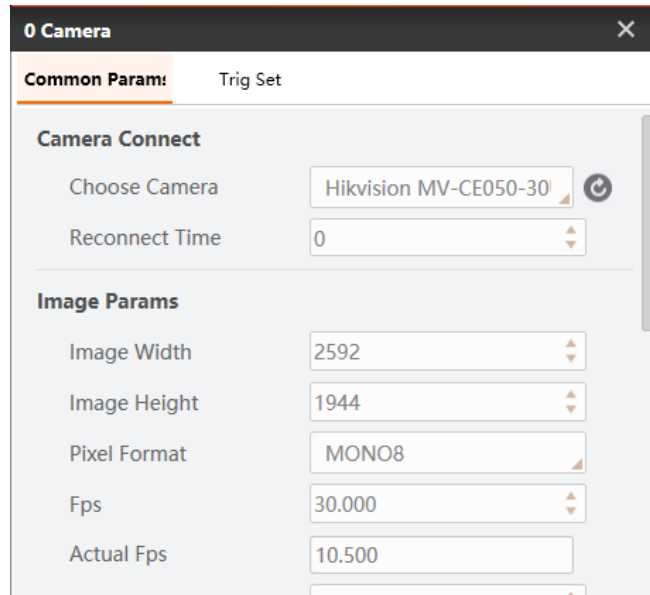


Figure 12.2 Select camera

Step 3. N-Point Calibration.

This example shows 9-points calibration.

1. Please select **pen** to be the end-effector of Dobot M1, jog X-axis and Y-axis, as shown below.

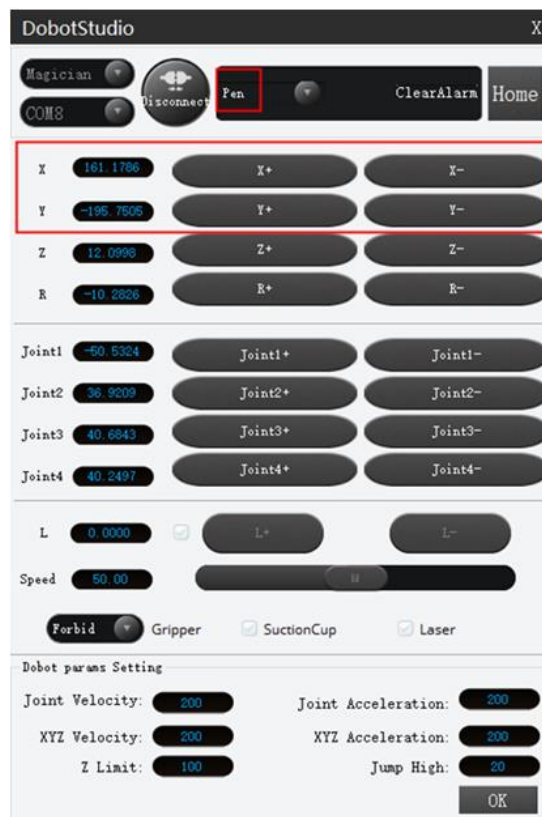


Figure 12.3 DobotStudio interface

2. Jog X-axis and Y-axis on DobotStudio to make the pen move to the 9 points in the order of the green arrow, and write down every point value according to the corresponding ID, and click **Confirm**.

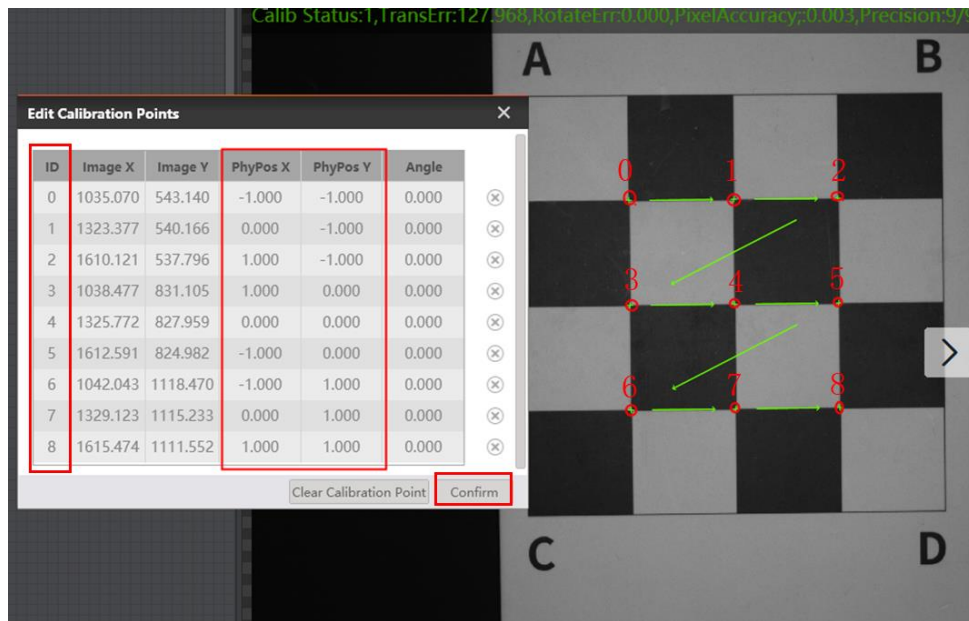


Figure 12.4 N-Point calibration

3. Click **Create Calibration File** and set saving path.

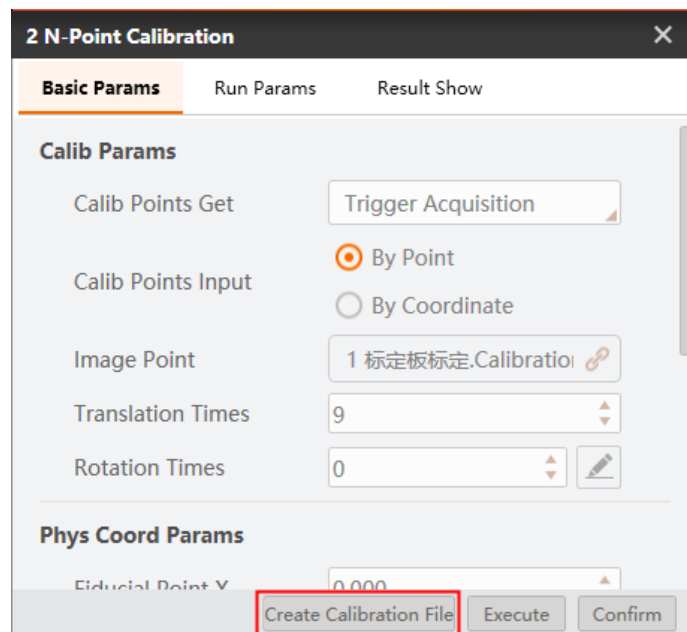


Figure 12.5 Create Calibration file

12.2 Color Sort

Note: Calibration file needs to be generated beforehand.

In this example, detect the color of blocks and sort them, as shown below.

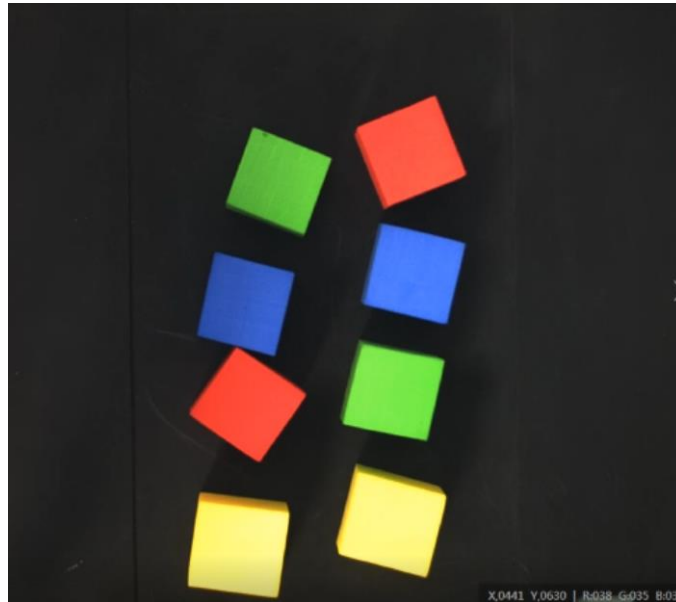


Figure 12.6 Detection example

Step 1. Solution establishment idea.

- Extract the sorted color in the camera image.
- Convert image coordinate to physical coordinate.
- Sort the specified color blocks by Dobot M1.

The overall solution flow is shown below.

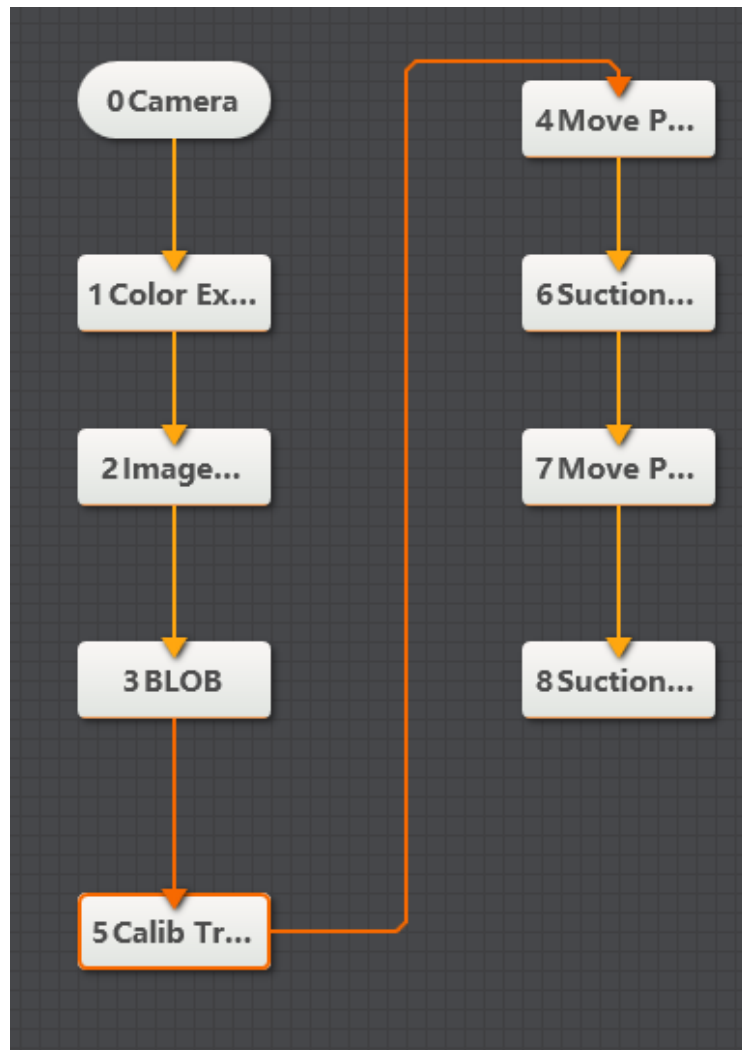


Figure 12.7 The overall solution flow

Step 2. Extract color.

Select extraction range of three channels in HSV space. After image processing, the image after color extraction is created, as shown below.

1 Color Extraction

Basic Params

Run Params

Result Show

Run Params

Color Space

HSV

Channel 1 Lower Limit

98

Channel 1 Upper Limit

204

Channel 2 Lower Limit

47

Channel 2 Upper Limit

255

Channel 3 Lower Limit

0

Channel 3 Upper Limit

255

Execute

Confirm

Figure 12.8 Set color

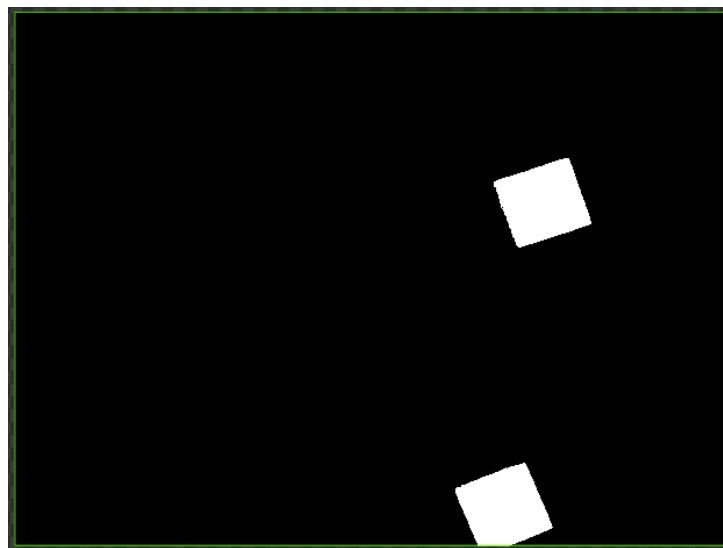


Figure 12.9 Result show

Step 3. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

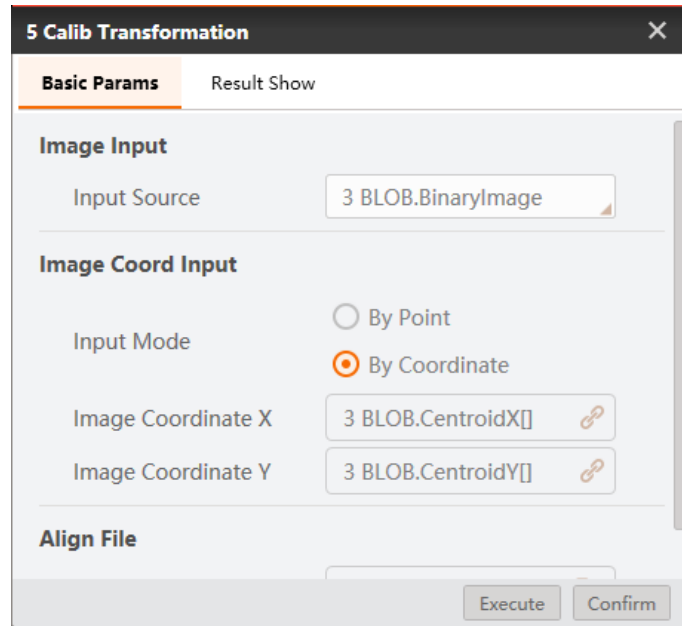


Figure 12.10 Calibration transformation

Step 4. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates, and set the Z-axis value based on site requirements.

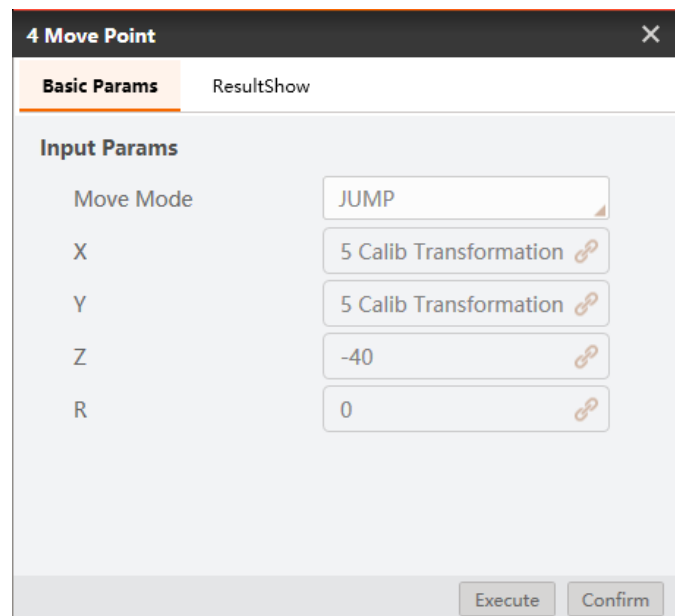


Figure 12.11 Move to a point

Step 5. Enable air pump.

Set I/O level to enable air pump.

Address: 18

Low: Enable

High: Disabled

The screenshot shows a software window titled "8 IO Output" with a close button (X) in the top right corner. Below the title bar, there are two tabs: "Basic Params" (which is selected and highlighted with an orange underline) and "ResultShow". Under the "Basic Params" tab, there is a section labeled "Input Params". This section contains two fields: "Address" with a text input box containing the value "18" and a chain-link icon to its right, and "Level" with a dropdown menu currently showing "Low". At the bottom right of the window, there are two buttons: "Execute" and "Confirm".

Figure 12.12 Enable air pump

Step 6. Set suction cup.

Set I/O level to control suction cup.

Address: 17

Low: Intake

High: Outtake

6 IO Output [X]

Basic Params ResultShow

Input Params

Address: 17 [Link]

Level: Low [Dropdown]

[Execute] [Confirm]

Figure 12.13 Control suction cup

Note: Use address 18 to enable air pump and address 17 to control suction cup in this example. You should select address based on site requirements.

Step 7. Set placing positions.

Set block placing positions according to different color.

7 Move Point [X]

Basic Params ResultShow

Input Params

Move Mode: JUMP [Dropdown]

X: 120 [Link]

Y: -170 [Link]

Z: -10 [Link]

R: 0 [Link]

[Execute] [Confirm]

Figure 12.14 set placing positions

12.3 Character Defect Detection

In this example, detect whether characters have defects like characters missing and redundant.

Note: Calibration file needs to be generated beforehand.



Figure 12.15 Detection example

Step 1. Solution establishment idea.

- Create the template of the standard characters and train them. Compare the trained characters with the target characters to find defects.
- Analyze the result to sort characters with Dobot M1.

The overall solution flow is shown below.

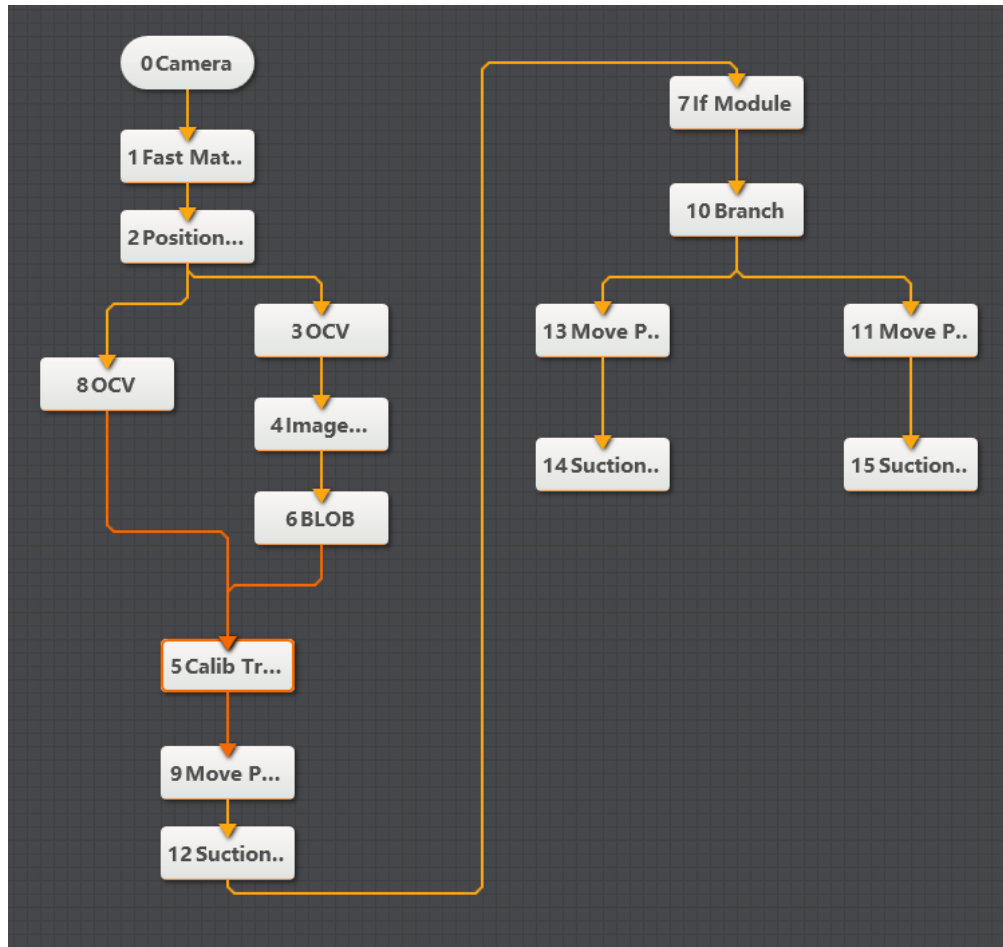


Figure 12.16 The overall solution flow

Step 2. Feature match.

Create feature template by feature matching module, as shown below .You need to select all characters completely, you can set **Scale Mode** to **Manual** to select manually, and then set the **Compressibility** and the **Contrast Threshold** based on site requirements.

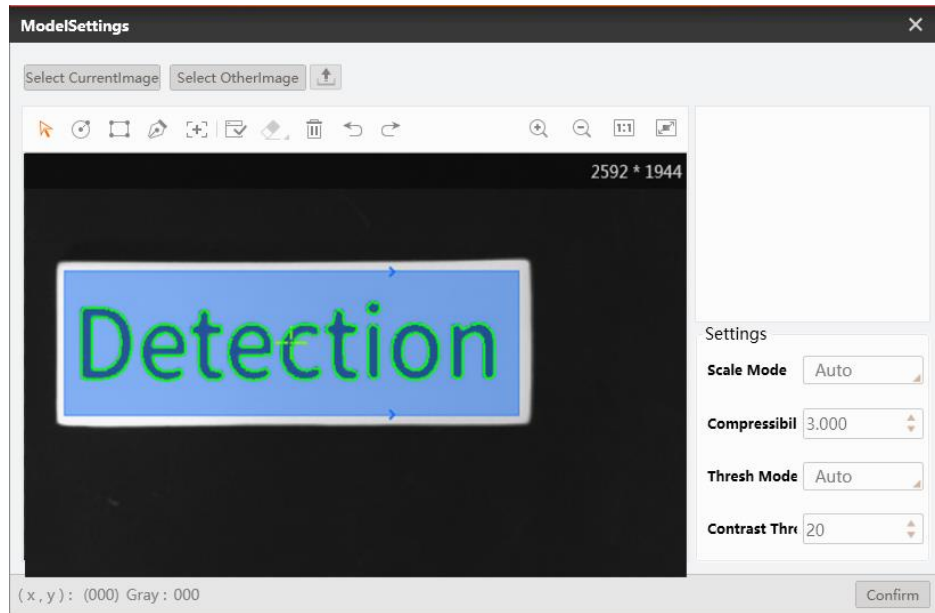


Figure 12.17 Feature match

Step 3. Position correction.

Create the reference of position deviation according to the match points and match frame angle in the template matching result. Click **Create reference**, as shown below.

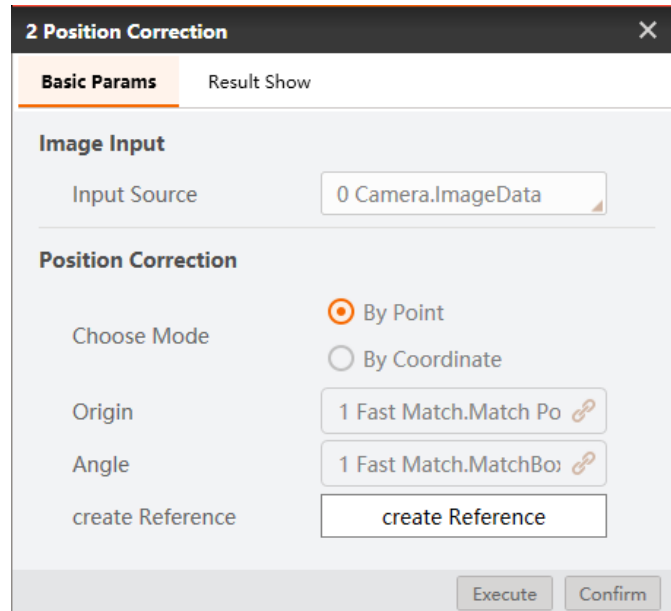


Figure 12.18 Position correction

Step 4. Train model.

Click **Create model** to create character template. Click **Character Extraction**. If characters are not extracted completely, please extract them manually, as shown below. The extraction is shown below.

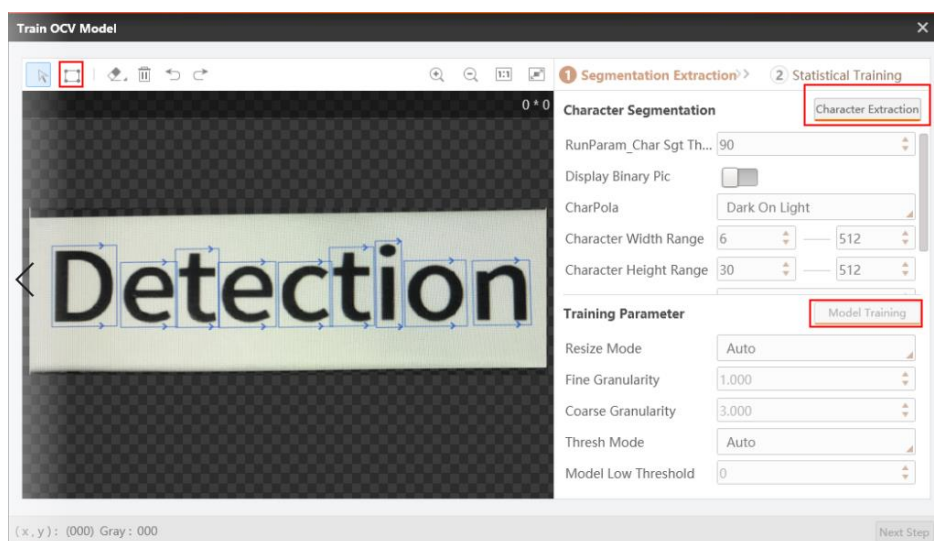


Figure 12.19 Train model

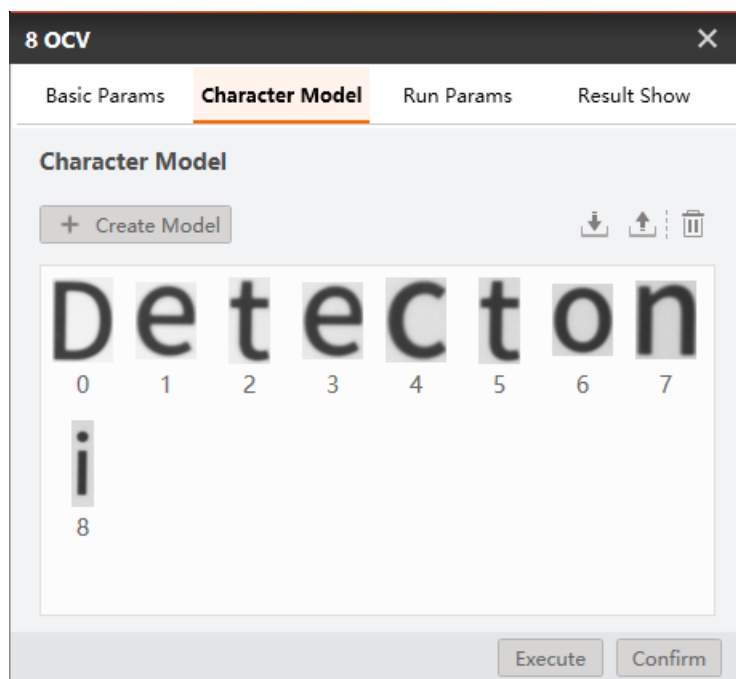


Figure 12.20 The result model

Step 5. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**, as shown below.

5 Calib Transformation

Basic Params Result Show

Image Input

Input Source: 0 Camera.ImageData

Image Coord Input

Input Mode: ☒ By Point ☐ By Coordinate

Image Point: 2 Position Correction.U

Align File

Load Calibration File: [Empty]

Execute Confirm

Figure 12.21 Calibration convert

Step 6. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates, and set the Z-axis value based on site requirements.

9 Move Point

Basic Params ResultShow

Input Params

Move Mode: JUMP

X: 5 Calib Transformation

Y: 5 Calib Transformation

Z: -65

R: 0

Execute Confirm

Figure 12.22 Move to a point

Step 7. Enable air pump.

Set I/O level to enable air pump.

Address: 18

Low: Enable

High: Disabled

8 IO Output

Basic Params ResultShow

Input Params

Address 18

Level Low

Execute Confirm

Figure 12.23 Enable air pump

Step 8. Set suction cup.

Set I/O level to control suction cup.

Address: 17

Low: Intake

High: Outtake

6 IO Output

Basic Params ResultShow

Input Params

Address 17

Level Low

Execute Confirm

Figure 12.24 control suction cup

Note: Use address 18 to enable air pump and address 17 to control suction cup in this example. You should select address based on site requirements.

Step 9. Condition detection.

Return the result value 0 or 1 by judging the number of defective characters. 0 means that there are character defects, 1 means no character defects.

7 If Module

Basic Params ResultShow

Judge Method

All Condition Conforming, The Result is OK

Conditioned	Valid Value Range
8 OCV.FlawNum[]	0.000 — 0.000
6 BLOB.BlobNum[]	0.000 — 0.000

int +

Execute Confirm

Figure 12.25 Condition detection

Step 10. Branch module.

Execute the different branch module according to the detection results. If the result is 1, execute branch 13, otherwise, execute branch 11.

The screenshot shows the '10 Branch' module configuration window. It has a title bar with '10 Branch' and a close button. The window is divided into two main sections: 'Input Params' and 'Branch Params'. In the 'Input Params' section, there is a 'Condition Input' field with the text '7 If Module.nIfResult[]' and a link icon. In the 'Branch Params' section, there are two radio buttons: 'Index by Value' (selected) and 'Index by Bit'. Below these, there is a table with two columns: 'Branch' and 'Condition Input Value'. The table has two rows: one for branch '13' with a value of '1', and one for branch '11' with a value of '0'. At the bottom right, there are two buttons: 'Execute' and 'Confirm'.

Branch	Condition Input Value
13	1
11	0

Figure 12.26 Branch module

Step 11. Set placing positions.

Set two sort positions according to detection results.

The screenshot shows the '13 Move Point' module configuration window. It has a title bar with '13 Move Point' and a close button. The window has two tabs: 'Basic Params' (selected) and 'ResultShow'. In the 'Basic Params' section, there is an 'Input Params' section. It contains a 'Move Mode' dropdown menu set to 'JUMP'. Below this, there are four rows for 'X', 'Y', 'Z', and 'R' coordinates. Each row has a text input field and a link icon. The values are: X=180, Y=-180, Z=0, and R=0. At the bottom right, there are two buttons: 'Execute' and 'Confirm'.

Coordinate	Value
Move Mode	JUMP
X	180
Y	-180
Z	0
R	0

Figure 12.27 Set sort position

Step 12. Detection results.

The detection results are shown below.



Figure 12.28 NG



Figure 12.29 OK

12.4 Measure and Sort

In this example, measure the diameter of Round-shaped objects and sort them, as shown below.

Note: Calibration file needs to be generated beforehand.



Figure 12.30 Example

Step 13. Solution establishment idea.

- To find the measured Round-shaped objects by find circle module.
- Convert image size to physical size by scale transformation.
- Convert image coordinate to physical coordinate by calibration transformation.
- Sort Round-shaped objects by Dobot M1.

The overall solution flow is shown below.

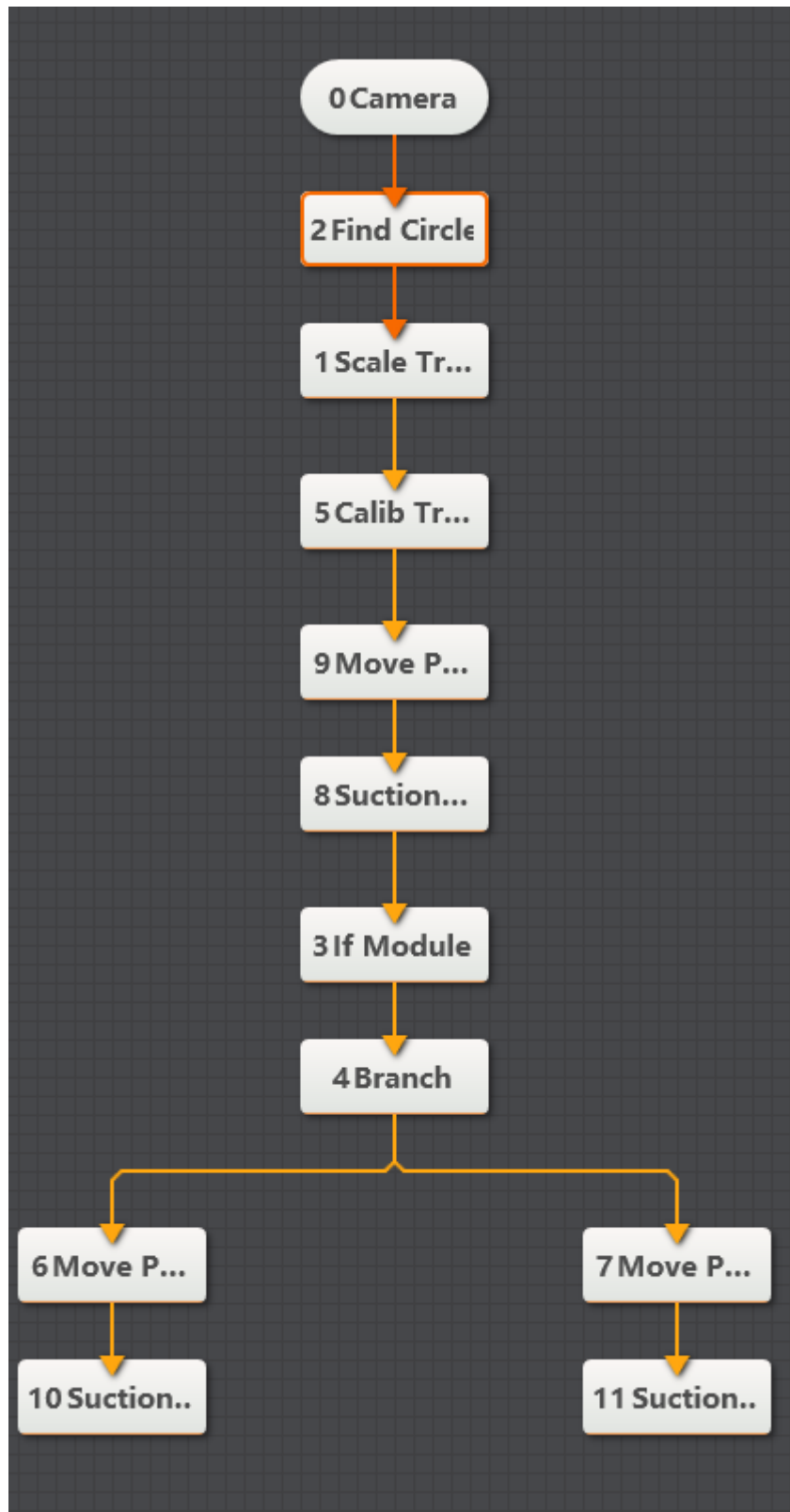


Figure 12.31 The overall solution flow

Step 14. Find circle

Set radius range to find circle.

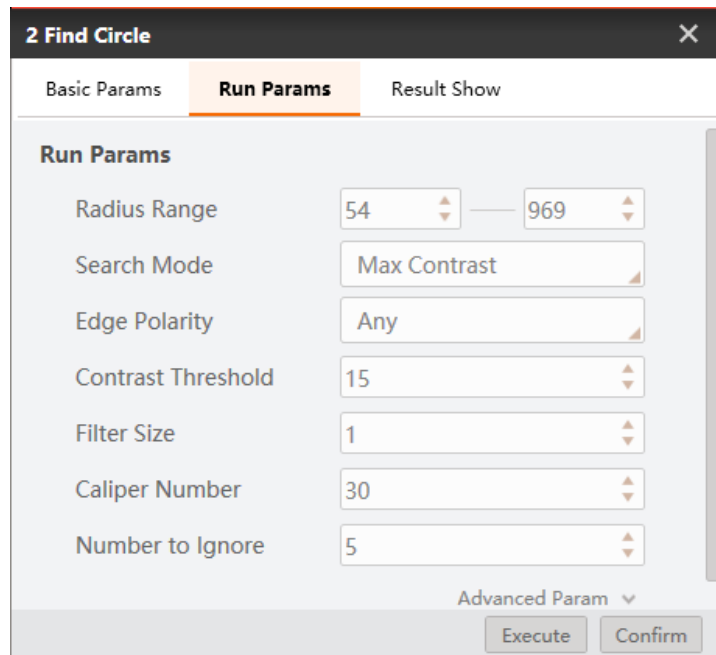


Figure 12.32 Find circle

Step 15. Scale transformation

Load calibration file to convert pixel unit to physical unit by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

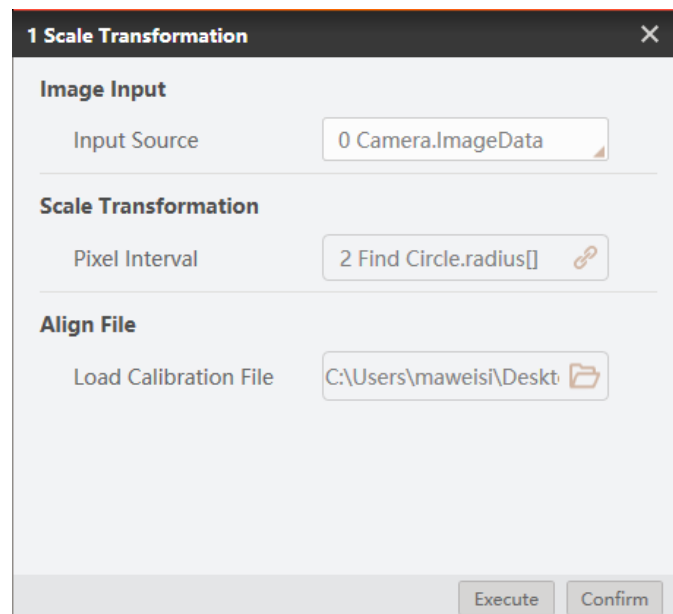


Figure 12.33 Convert unit

Step 16. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

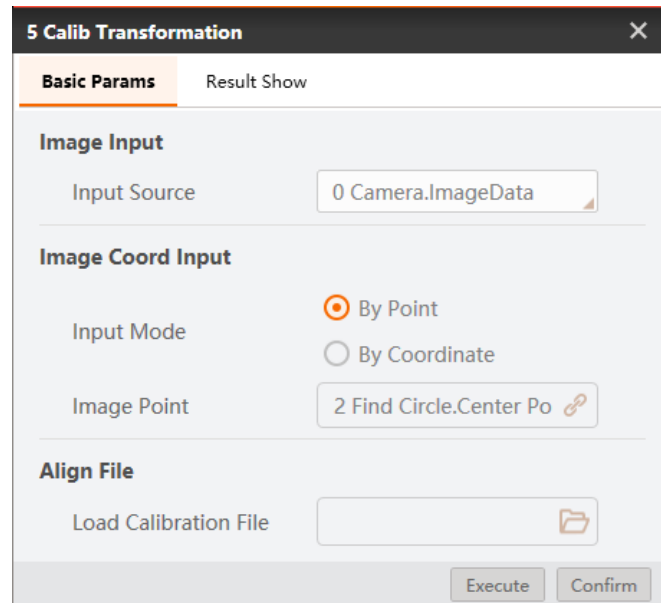


Figure 12.34 Calibration convert

Step 17.Condition detection.

Set radius range. As shown below.

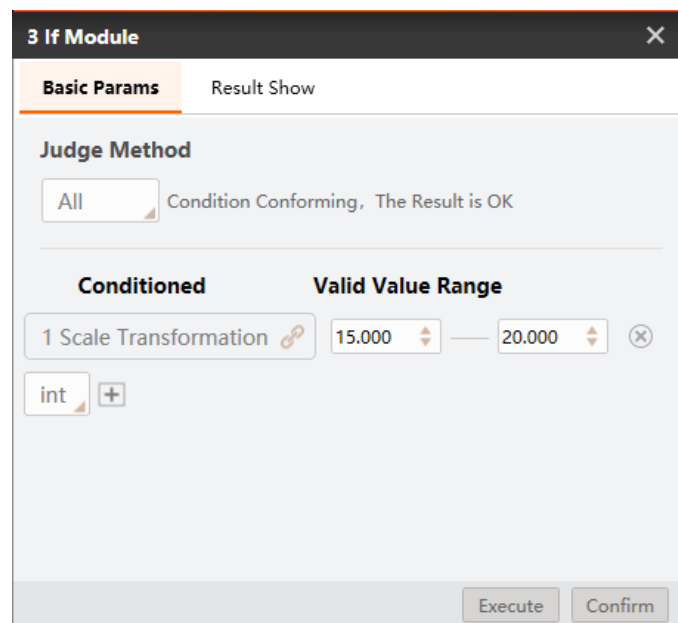


Figure 12.35 Condition detection

Step 18.Branch module.

Execute the different branch modules according to the measured results. If result is 1, execute branch 6, otherwise execute branch 7.

4 Branch

Input Params

Condition Input

3 If Module.nIfResult[]

Branch Params

Index by Value

Index by Bit

Branch

Condition Input Value

6

1

7

0

Execute

Confirm

Figure 12.36 Branch module

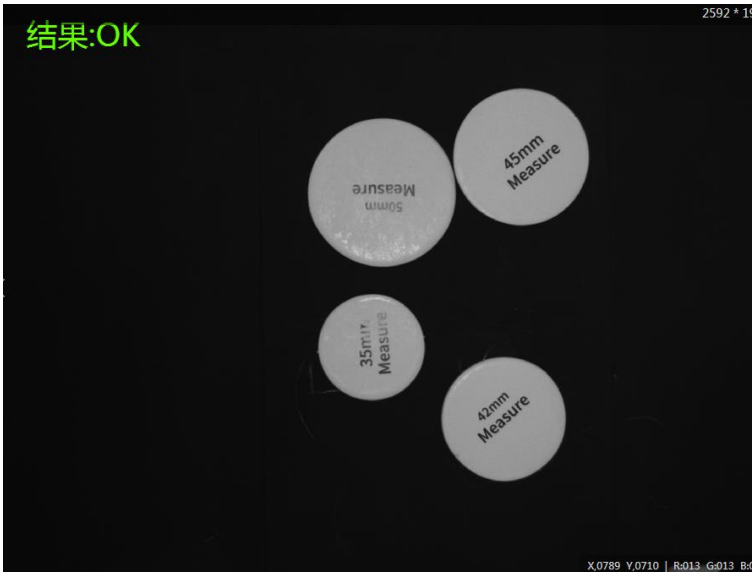


Figure 12.37 OK

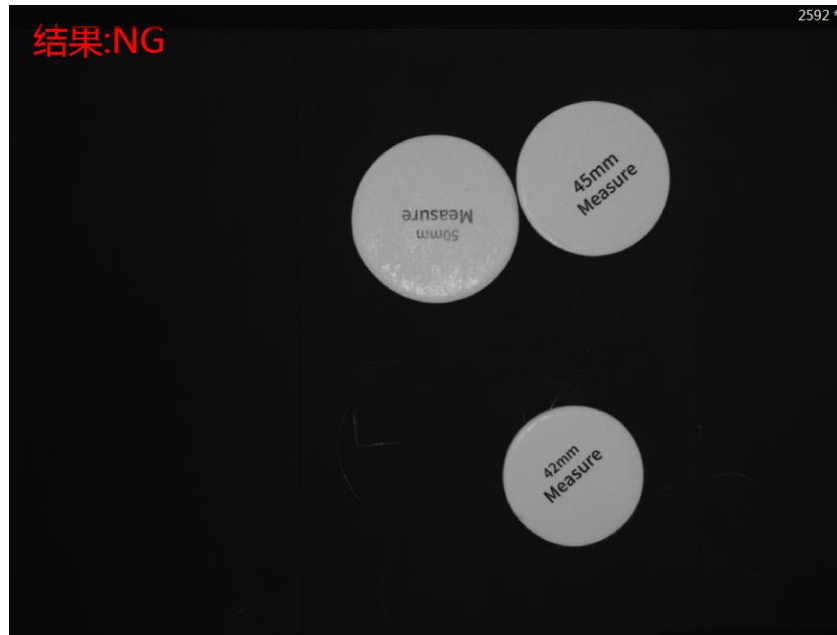


Figure 12.38 NG

Step 19.Enable air pump.

Set I/O level to enable air pump.

Address: 18

Low: Enable

High: Disabled

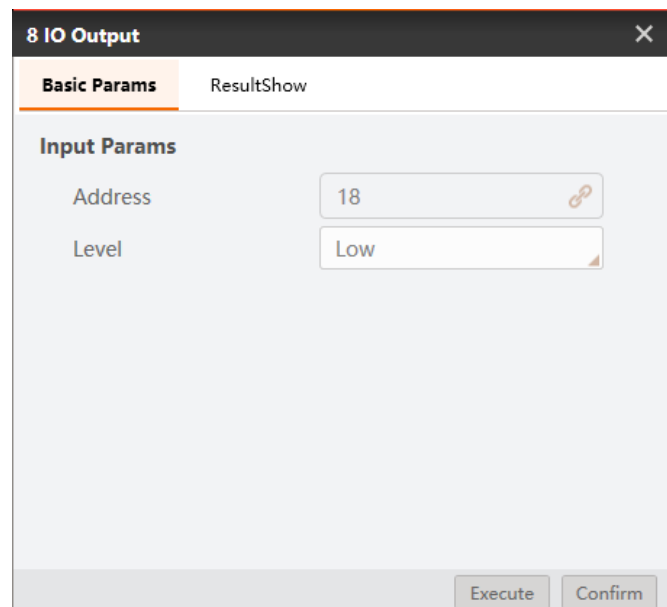


Figure 12.39 Enable air pump

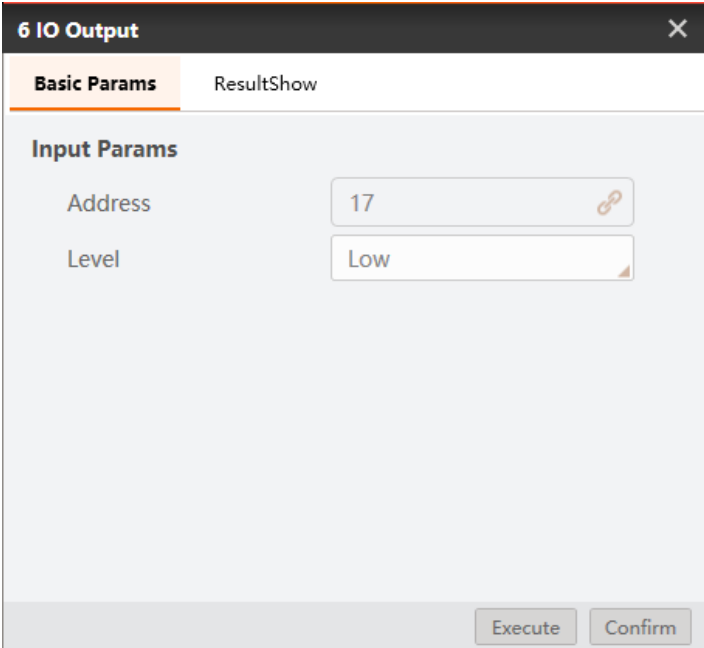
Step 20.Set suction cup.

Set I/O level to control suction cup.

Address: 17

Low: Intake

High: Outtake



The screenshot shows a software window titled "6 IO Output" with a close button (X) in the top right corner. Below the title bar, there are two tabs: "Basic Params" (which is selected and highlighted with an orange underline) and "ResultShow". Under the "Basic Params" tab, there is a section titled "Input Params". This section contains two input fields: "Address" with the value "17" and a chain-link icon to its right, and "Level" with the value "Low" and a small downward arrow icon to its right. At the bottom of the window, there are two buttons: "Execute" and "Confirm".

Figure 12.40 Control suction cup

Note: Use address 18 to enable air pump and address 17 to control suction cup in this example. You should select address based on site requirements.

Step 21. Set placing positions.

Set round-shaped objects placing positions according to the measured results.

13 Move Point [X]

Basic Params ResultShow

Input Params

Move Mode	JUMP
X	180 [Chain Icon]
Y	-180 [Chain Icon]
Z	0 [Chain Icon]
R	0 [Chain Icon]

[Execute] [Confirm]

Figure 12.41 Set sort position

12.5 RectangleTemplate Match

In this example, measure the diameter of circles and put them into the corresponding model as shown below.

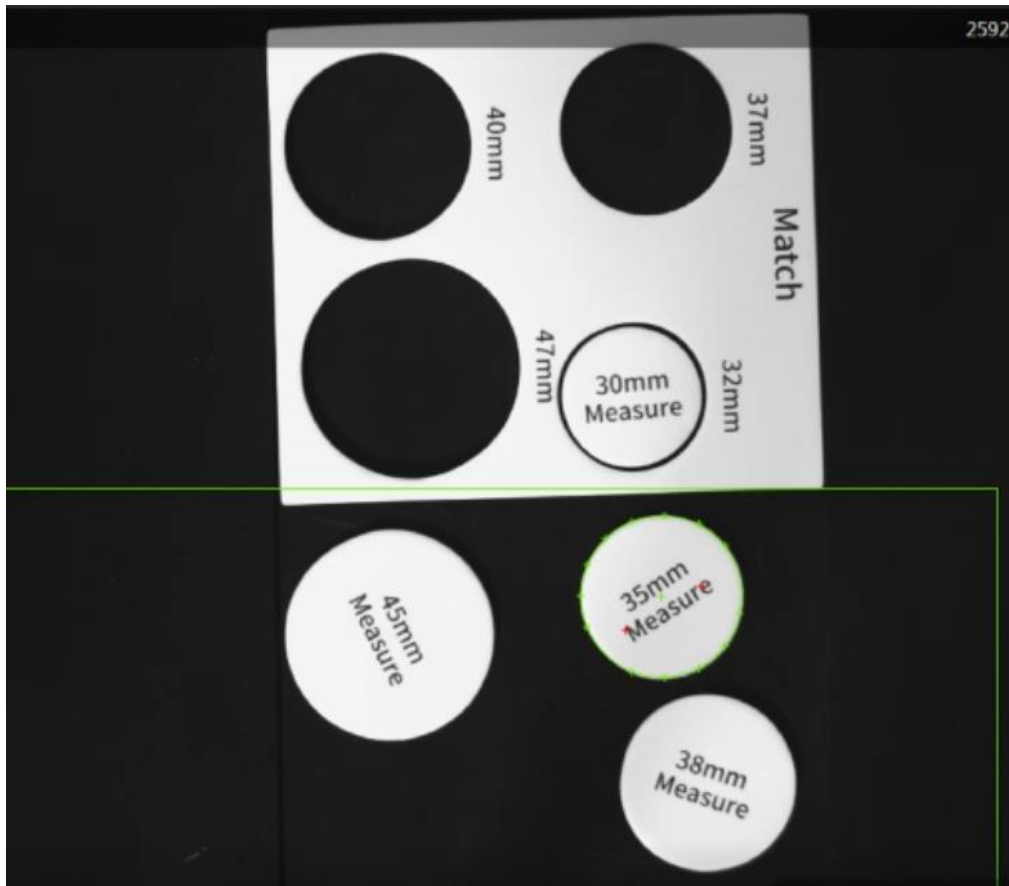


Figure 12.42 Circle match

Step 1. Solution establishment idea.

- Find the measured circle by finding circle module.
- Convert image coordinate to physical coordinate by calibration transformation.
- Put circles to corresponding circle models by Dobot M1.

The overall solution flow is shown below.

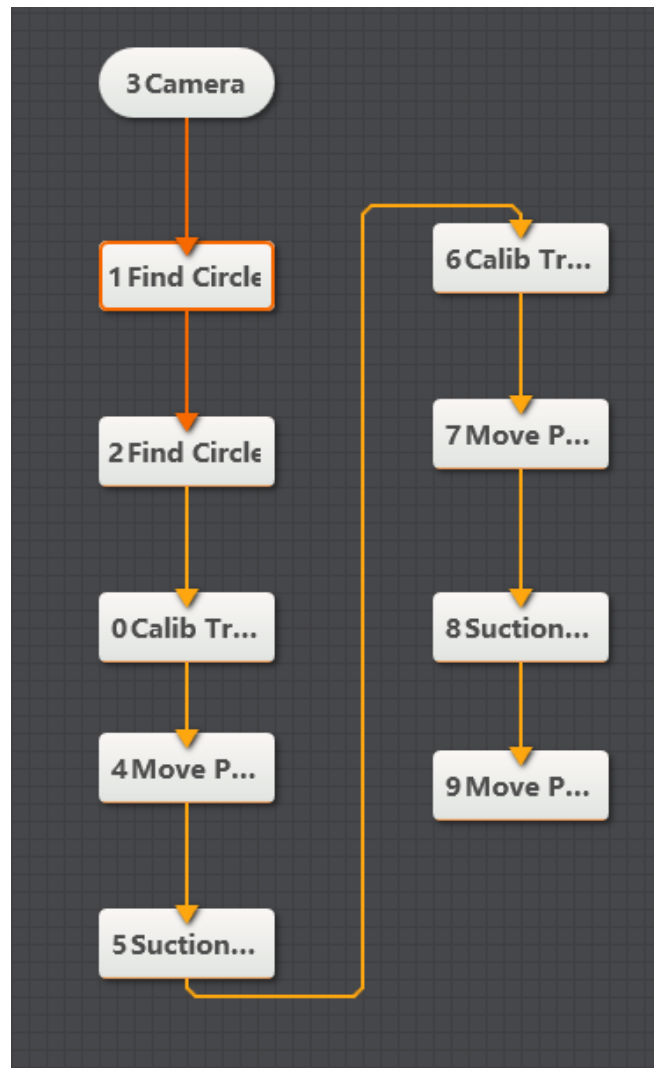


Figure 12.43 The overall solution flow

Step 2. Find circle.

Set finding circle modules for circle and circle model to set **Run Params**, such as **Radius Range**, **Search Mode**, and **Contrast Threshold**. As shown below.

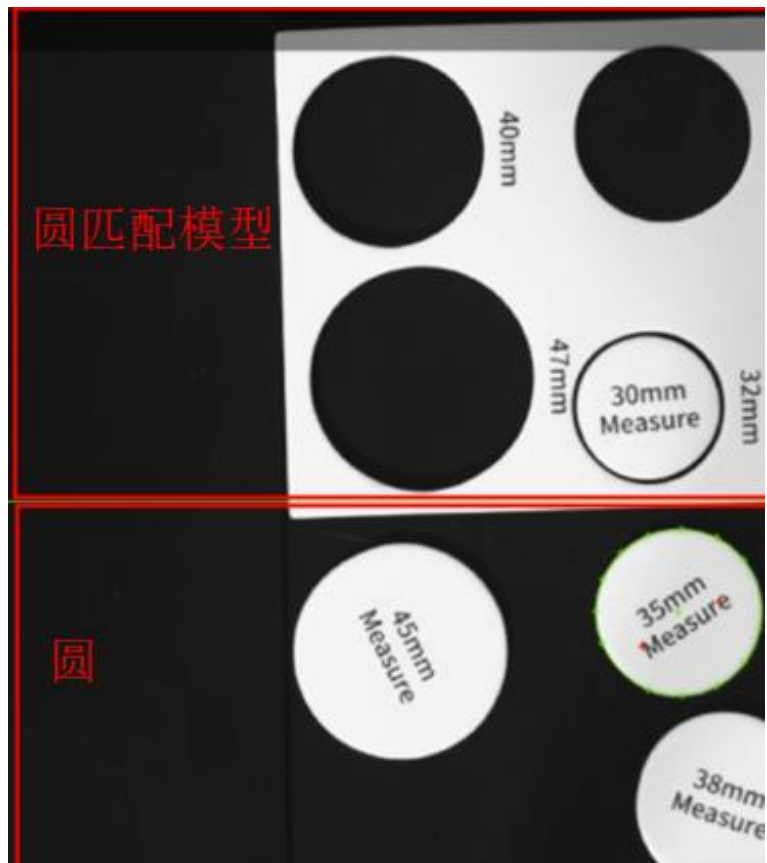


Figure 12.44 find circle

1 Find Circle

Basic Params
Run Params
Result Show

Run Params

Radius Range
100
500

Search Mode
Min Radius

Edge Polarity
Any

Contrast Threshold
15

Filter Size
1

Caliper Number
30

Number to Ignore
0

Advanced Param

Execute
Confirm

Figure 12.45 Find circle model

Step 3. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

0 Calib Transformation

Basic Params Result Show

Image Input

Input Source: 3 Camera.ImageData

Image Coord Input

Input Mode: ☒ By Point ☐ By Coordinate

Image Point: 2 Find Circle.Center Po

Align File

Load Calibration File: [Empty]

Execute Confirm

Figure 12.46 Calibration transformation

Step 4. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates according to different point.

4 Move Point

Basic Params ResultShow

Input Params

Move Mode: JUMP

X: 0 Calib Transformation

Y: 0 Calib Transformation

Z: -62

R: 0

Execute Confirm

Figure 12.47 Move to a point

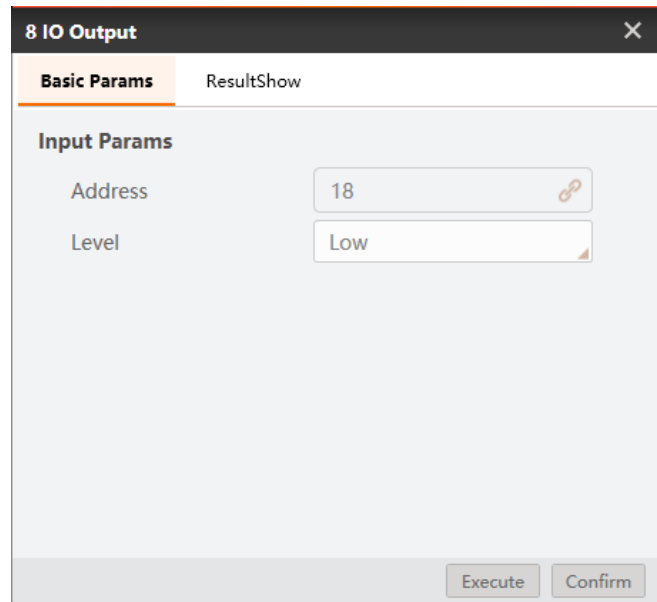
Step 5. Enable air pump.

Set I/O level to enable air pump.

Address: 18

Low: Enable

High: Disabled



The screenshot shows a software window titled "8 IO Output" with a close button (X) in the top right corner. Below the title bar, there are two tabs: "Basic Params" (which is selected and highlighted with an orange underline) and "ResultShow". Under the "Basic Params" tab, there is a section labeled "Input Params". This section contains two input fields: "Address" with the value "18" and a link icon to its right, and "Level" with a dropdown menu showing "Low". At the bottom right of the window, there are two buttons: "Execute" and "Confirm".

Figure 12.48 Enable air pump

Step 6. Set suction cup.

Set I/O level to control suction cup.

Address: 17

Low: Intake

High: Outtake

The screenshot shows a software window titled "6 IO Output" with a close button (X) in the top right corner. Below the title bar, there are two tabs: "Basic Params" (which is selected and highlighted with an orange underline) and "ResultShow". Under the "Basic Params" tab, there is a section labeled "Input Params". This section contains two input fields: "Address" with the value "17" and a red chain-link icon to its right, and "Level" with the value "Low" and a small downward arrow icon to its right. At the bottom right of the window, there are two buttons: "Execute" and "Confirm".

Figure 12.49 Control suction cup

Note: Use address 18 to enable air pump and address 17 to control suction cup in this example. You should select address based on site requirements.

Step 7. The matching result.

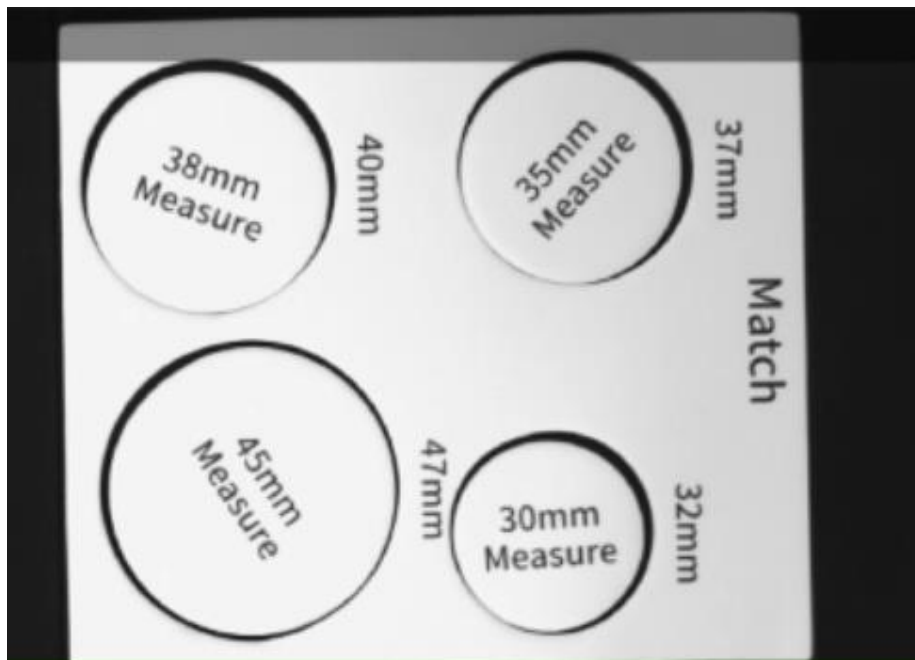


Figure 12.50 The matching result

12.6 Circle Template Match

In this example, measure the diameter of circles and put them into the corresponding model as shown below.

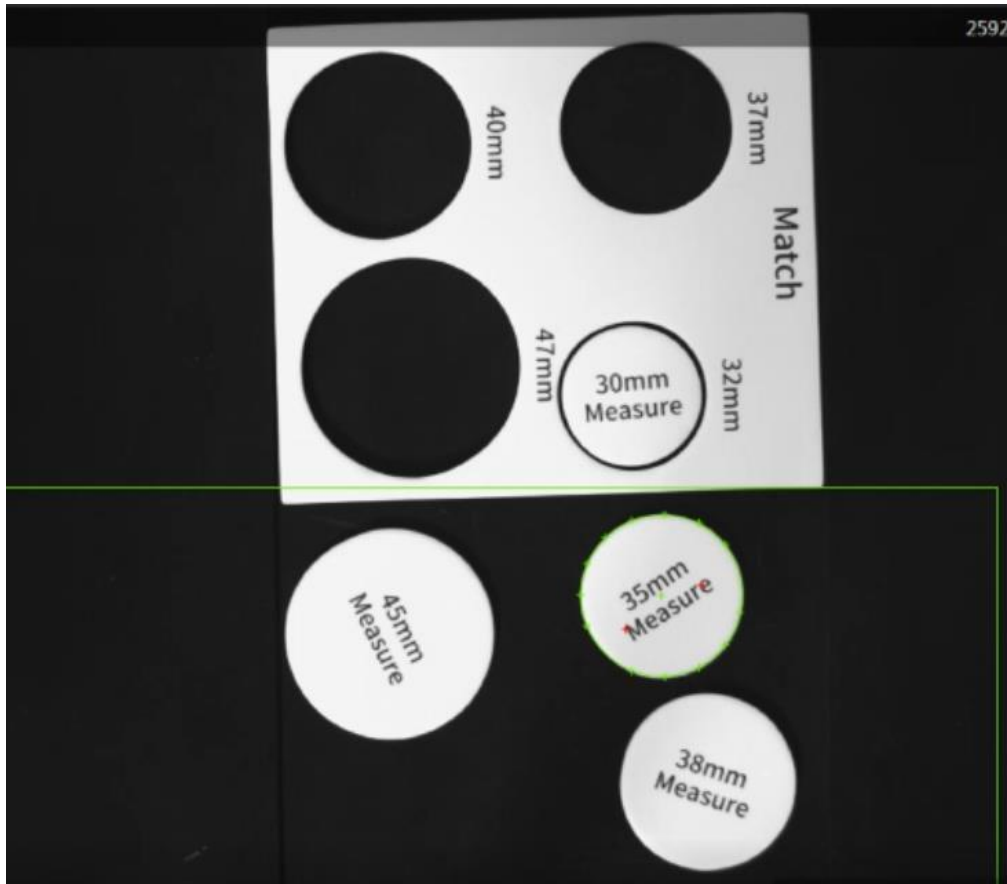


Figure 12.51 Circle match

Step 1. Solution establishment idea.

- Find the measured circle by finding circle module.
- Convert image coordinate to physical coordinate by calibration transformation.
- Put circles to corresponding circle models by Dobot M1.

The overall solution flow is shown below.

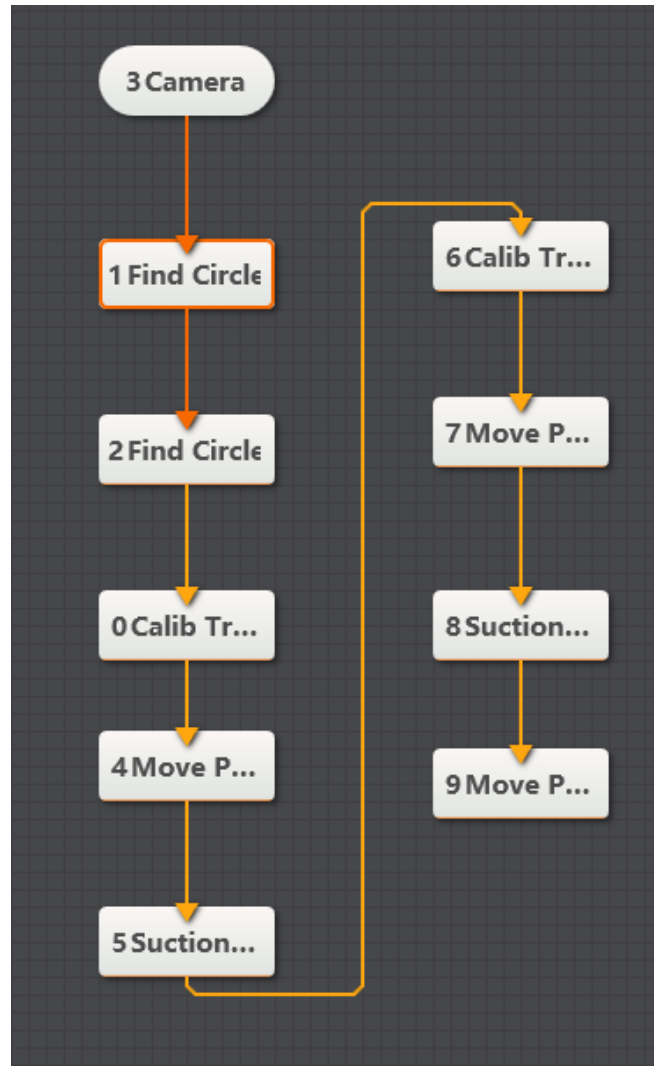


Figure 12.52 The overall solution flow

Step 2. Find circle.

Set finding circle modules for circle and circle model to set **Run Params**, such as **Radius Range**, **Search Mode**, and **Contrast Threshold**. As shown below.

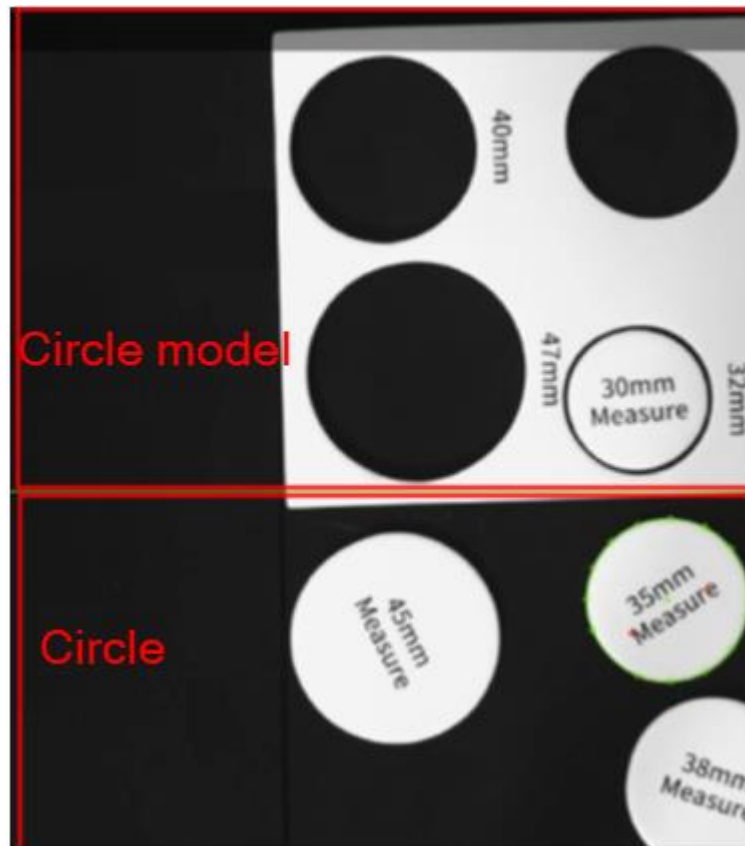


Figure 12.53 Find circle

1 Find Circle

Basic Params

Run Params

Result Show

Run Params

Radius Range

100 — 500

Search Mode

Min Radius

Edge Polarity

Any

Contrast Threshold

15

Filter Size

1

Caliper Number

30

Number to Ignore

0

Advanced Param

Execute

Confirm

Figure 12.54 Find circle model

Step 3. Convert image coordinate to physical coordinate.

Load calibration file to convert image coordinate to physical coordinate by calibration file, and click **Confirm**. Calibration file is generated by Robot Calibration.

0 Calib Transformation

Basic Params Result Show

Image Input

Input Source: 3 Camera.ImageData

Image Coord Input

Input Mode: ☒ By Point ☐ By Coordinate

Image Point: 2 Find Circle.Center Po

Align File

Load Calibration File: [Empty]

Execute Confirm

Figure 12.55 Calibration transformation

Step 4. Move to a point.

Set motion mode. For the motion mode details, please refer to 7.2.3 [Move To a Point](#). Select the corresponding physical coordinates according to different point.

4 Move Point

Basic Params ResultShow

Input Params

Move Mode: JUMP

X: 0 Calib Transformation

Y: 0 Calib Transformation

Z: -62

R: 0

Execute Confirm

Figure 12.56 Move to a point

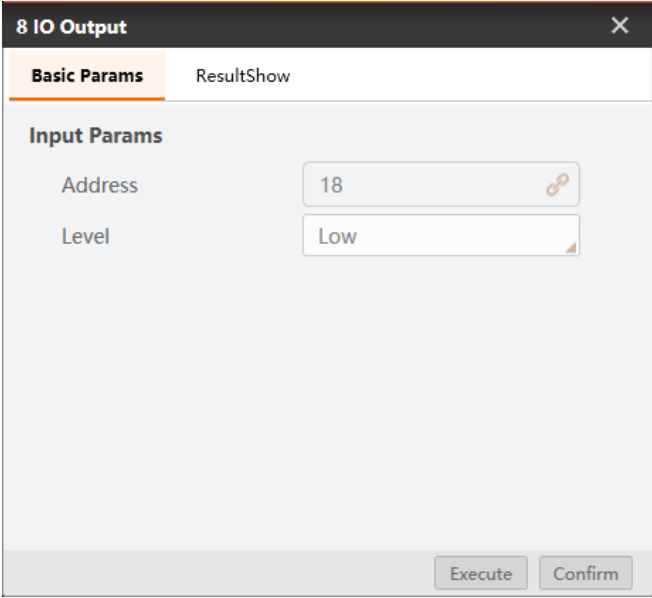
Step 5. Enable air pump.

Set I/O level to enable air pump.

Address: 18

Low: Enable

High: Disabled



The screenshot shows a software window titled "8 IO Output" with a close button (X) in the top right corner. Inside the window, there are two tabs: "Basic Params" (which is selected and highlighted with an orange underline) and "ResultShow". Under the "Basic Params" tab, there is a section titled "Input Params". This section contains two input fields: "Address" with the value "18" and a link icon to its right, and "Level" with a dropdown menu showing "Low". At the bottom right of the window, there are two buttons: "Execute" and "Confirm".

Figure 12.57 Enable air pump

Step 6. Set suction cup.

Set I/O level to control suction cup.

Address: 17

Low: Intake

High: Outtake

6 IO Output [X]

Basic Params ResultShow

Input Params

Address: 17 [Link Icon]

Level: Low [Dropdown Arrow]

[Execute] [Confirm]

Figure 12.58 Control suction cup

Note: Use address 18 to enable air pump and address 17 to control suction cup in this example. You should select address based on site requirements.

Step 7. The matching result.

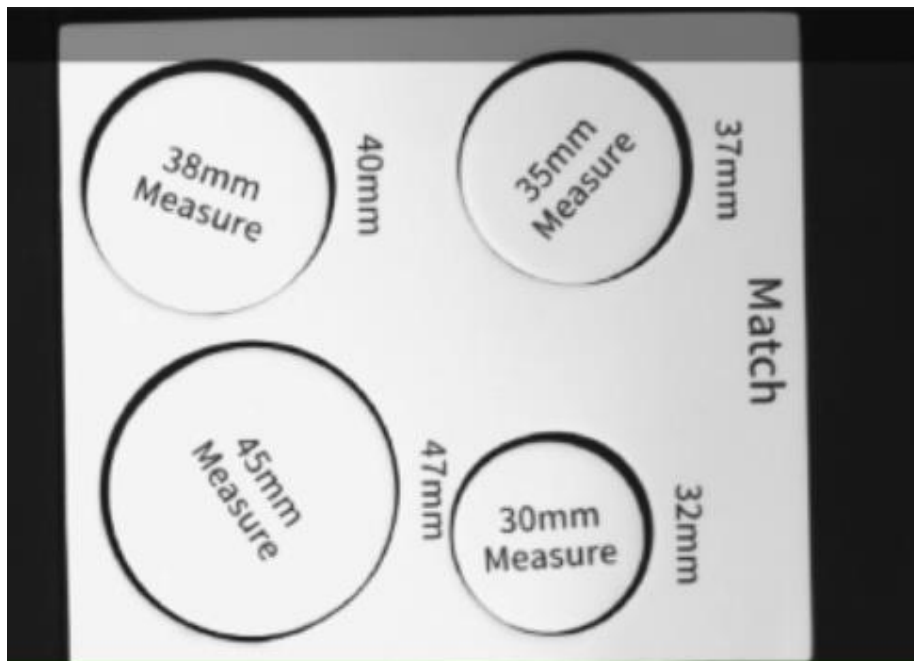


Figure 12.59 The matching result