

Yahboom DOFBOT AI Vision Robotic Arm with ROS Python programming for Raspberry Pi 4B 8GB/4GB

★★★★☆ 26 reviews

YAHBOOM

This Raspberry Pi robotic arm is mainly composed of an aluminum alloy bracket, a sturdy chassis, a multi-function expansion board, 6 bus servos and a 300,000 pixel camera.

We built an operating system specifically for it based on Ubuntu20.04/ROS noetic, using Open Source CV and Python3 programming to create a series of AI vision functions. It can be controlled by APP, PC software, handle, JupyterLab. We provide detailed tutorials, open source codes and technical support services for Raspberry Pi players and AI beginners.

Features

Rich list of materials

- Assembled Robotic arm.
- Matching color-printed map and 4 different colors blocks.
- PS2 gamepad.
- 32G TF card with image system.
- Yahboom special cooling HAT to help heat dissipation.

User-friendly detailed design

- Users can connect network by the mobile APP QR code on APP to get started quickly.
- OLED displays IP address and Raspberry Pi CPU related information in real time.
- 12V 5A power adapter power supply, no battery life limit.
- The chassis with suction cups is more stable and can be stably placed in any experimental environment at any time.

Exciting AI features

- It can be controlled by Android/iOS APP, PC upper computer, handle, Jupyter Lab webpage online programming.
- DOFBOT can realize Gesture recognition, color recognition, visual positioning, garbage classification, catch game, face tracking, building blocks stacking and other AI games.
- User can make it learn to customize action groups, and complete the synchronous action of the dual robotic arm.

Excellent extensibility

- The expansion board is compatible with Jetson NANO, Raspberry Pi, Arduino, Micro:bit four development boards.
- Aluminum alloy structure is detachable, users can extend or shorten the length of the robotic arm.
- We reserve 6 bus servo (DOFBOT only uses one) + 6 PWM servo, wireless controller receiver, WiFi/Bluetooth module, I2C and ultrasonic module interfaces.

ROS-CONTROLLER & ACCESSORIES

VIEW MORE



DOFBOT-Raspberry Pi version

AI Vision Robotic Arm

AI visual identity | ROS system | Inverse kinematics



The trash is not included in the shipping list

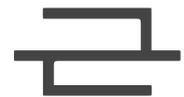
Product Description

DOFBOTAL visual robotic arm uses Raspberry Pi as the main control board, Open Source CV as the image processing library, Jupyter Lab as the development tool, and Python as programming language. It can not only realize color tracking and grabbing, but also recognize and interact with human body features, and even train garbage classification models. Through the ROS system, the complex motion control of the 6DOF serial bus servos is simplified. Yahboom will provide tutorials and codes to help you easily control this robotic arm.

AI



ROS



Artificial Intelligence

Raspberry Pi board

ROS robot operating system

Machine vision

Kinematic analysis

Movelt

RVIZ



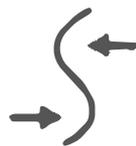
Movelt motion planning

Rviz simulation

Voice recognition interaction

Camera and robotic arm 2 in 1

Intelligent garbage sorting



Intelligent serial bus servo

6DOF

PID fitting

Cross-platform interconnection control

Python open source tutorial

Based on the Raspberry Pi 4B

With powerful AI computing power

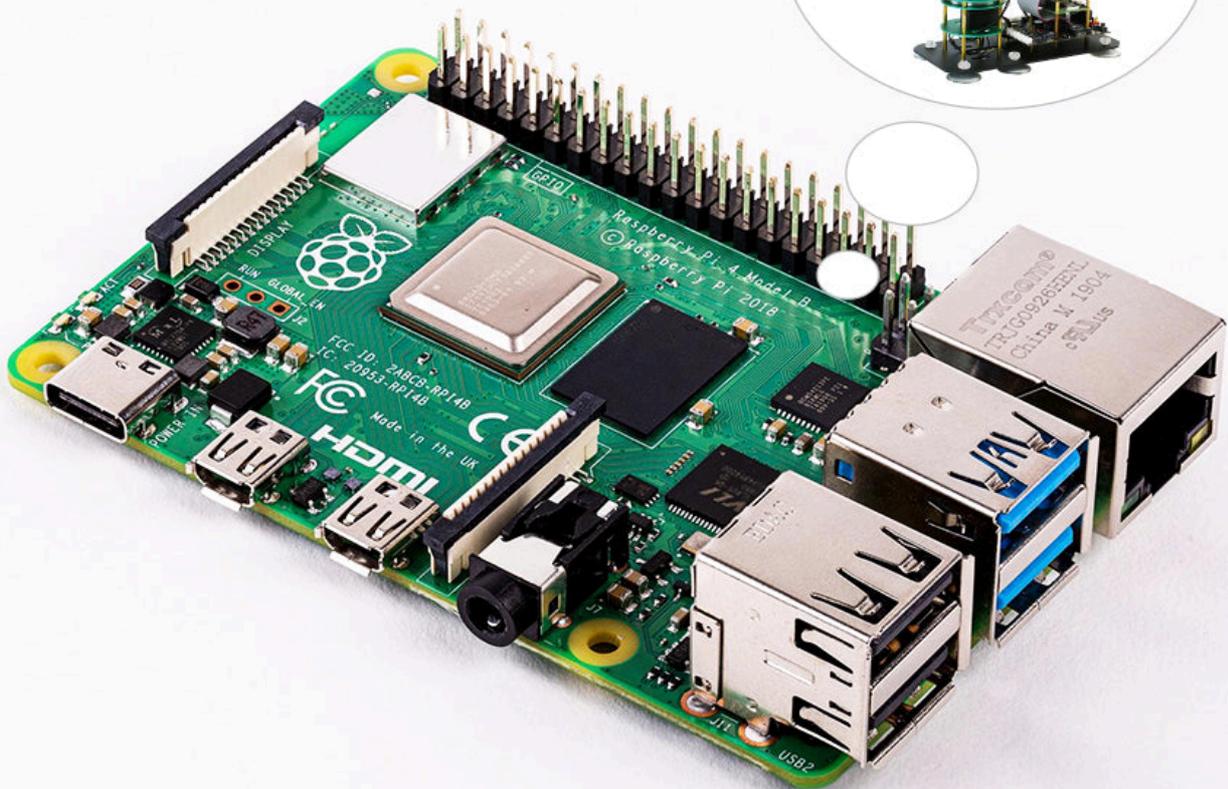
Raspberry Pi AI reaches a new height with strong performance support from Raspberry Pi 4B, significantly improving AI computing power

CPU

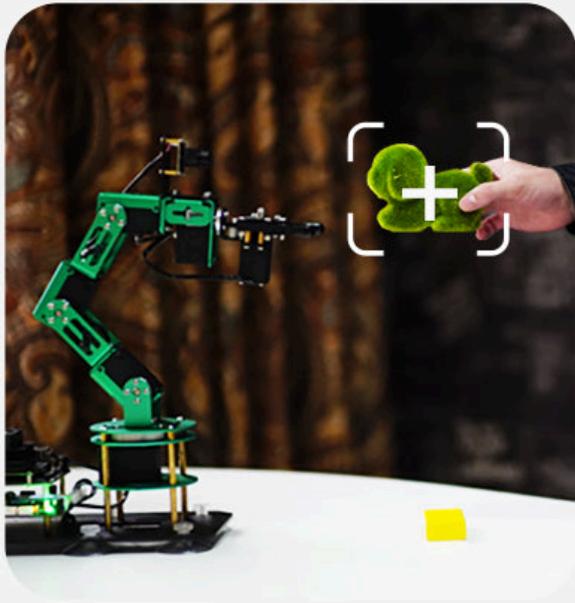
64-bit 1.5GHz quad core (28nm process)

GPU

Broadcom VideoCore VI @ 500MHz



AI Vision Interactive Function



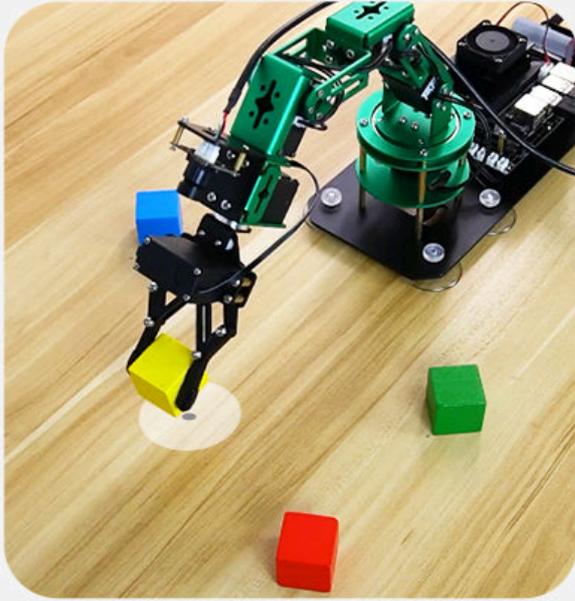
Color recognition tracking

In addition to standard colors, custom colors can also be extracted for color recognition and tracking.



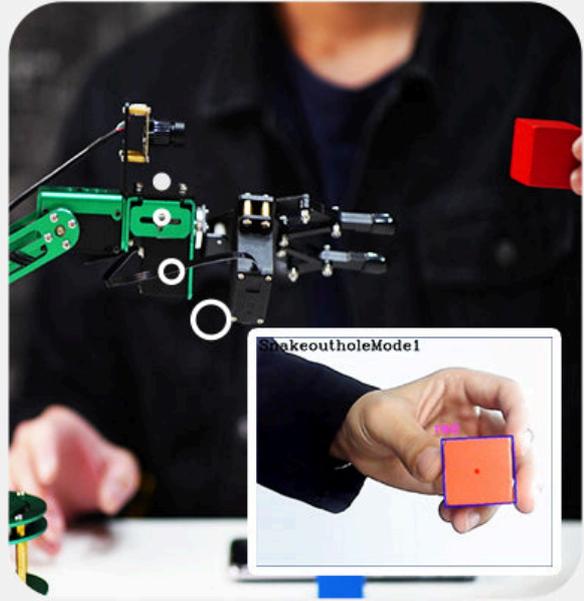
Catch game

Place the block into the map recognition area, robotic arm will automatically identify the current color and clamp the block to the corresponding color area on the map (shipping list include).



Grab color blocks

Put the block in front of the camera, and after the robotic arm recognizes the color of the block, it will clamp the block and put it into the middle area.



Color interaction

Select a color block, DOFBOT will move back and forth following the target color.



Garbage sorting

By loading the garbage classification model, the garbage classification on the blocks can be identified, and the robotic arm will sort the "garbage" to the corresponding classification area on the map.



Color block stacking

The camera recognizes the gesture numbers, and the robotic arm picks up the blocks on the map to the middle position for stacking.

AI Vision Human Feature Interaction

DOFBOT can recognize human body features such as gestures and faces through camera, and complete interactive actions such as grabbing, identity and tracking.



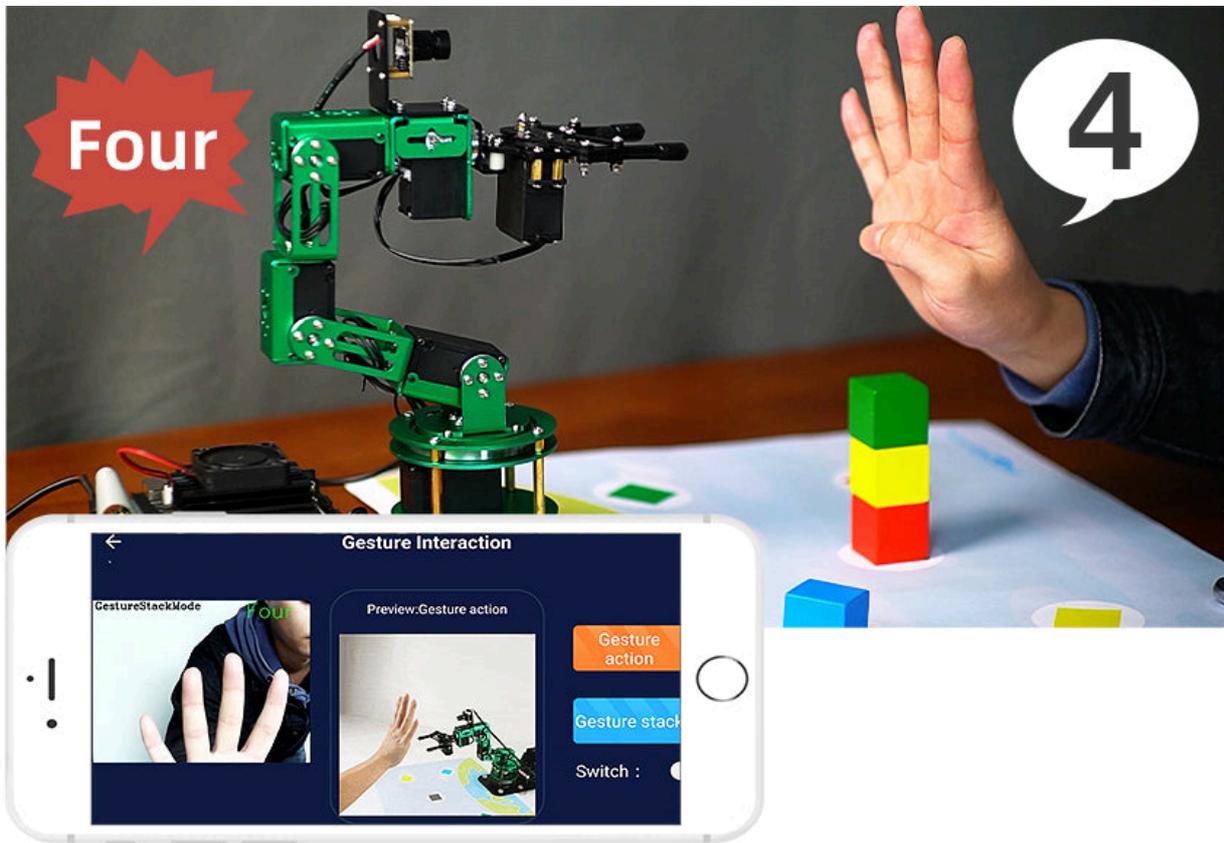
1/Gesture interaction

DOFBOT can recognize a variety of gestures and perform corresponding actions for interaction.



2/Gesture grabbing

Recognize the digital gestures, stack the corresponding number of layers, and finally push down the stacked blocks with the "fist" gesture.

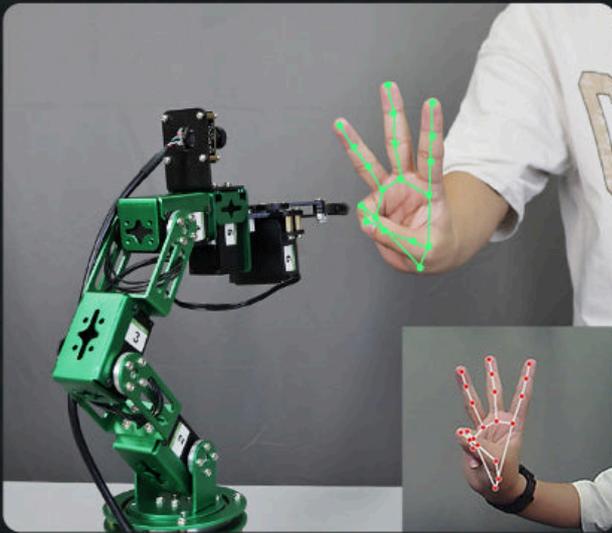


3/Face recognition and tracking

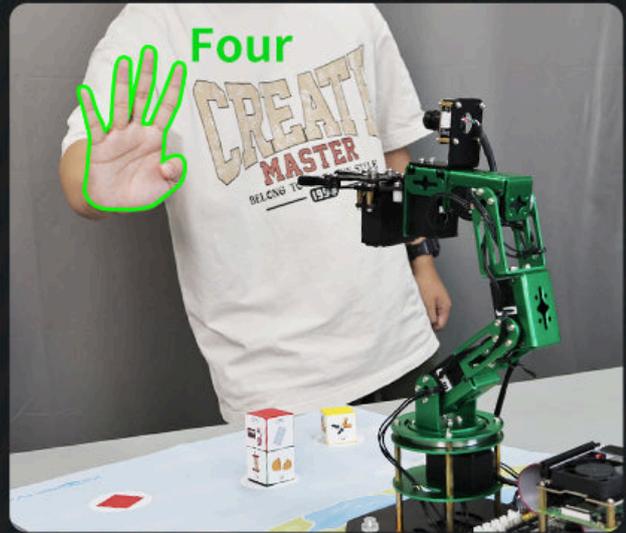
Detect whether there is a face in the current picture, mark the face after recognizing it and follow the movement.



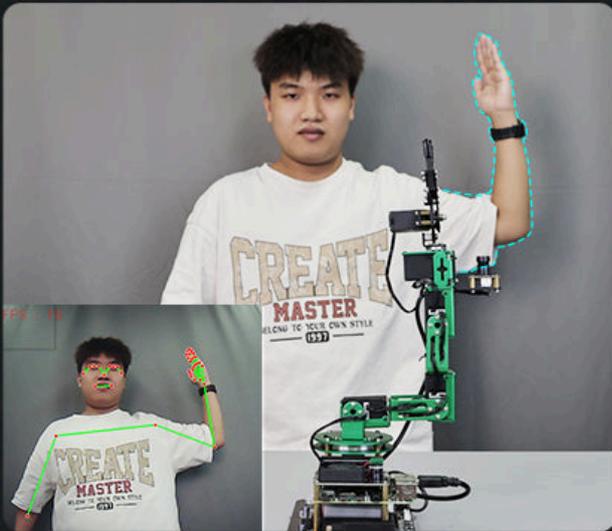
MediaPipe Machine Learning



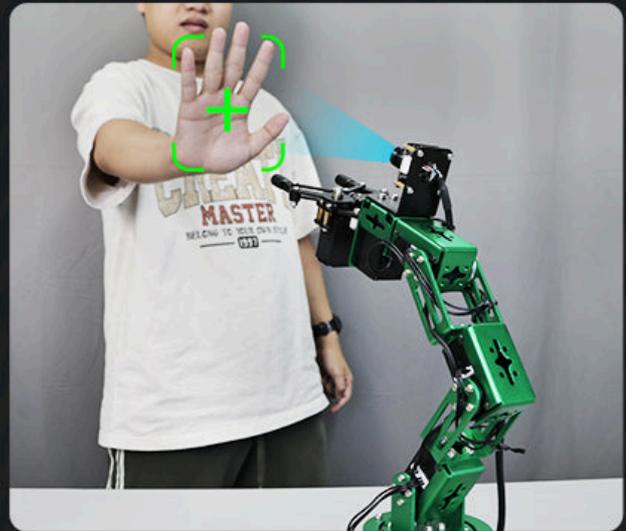
Gesture control robotic arm action group



Gesture recognition control robotic arm stacking



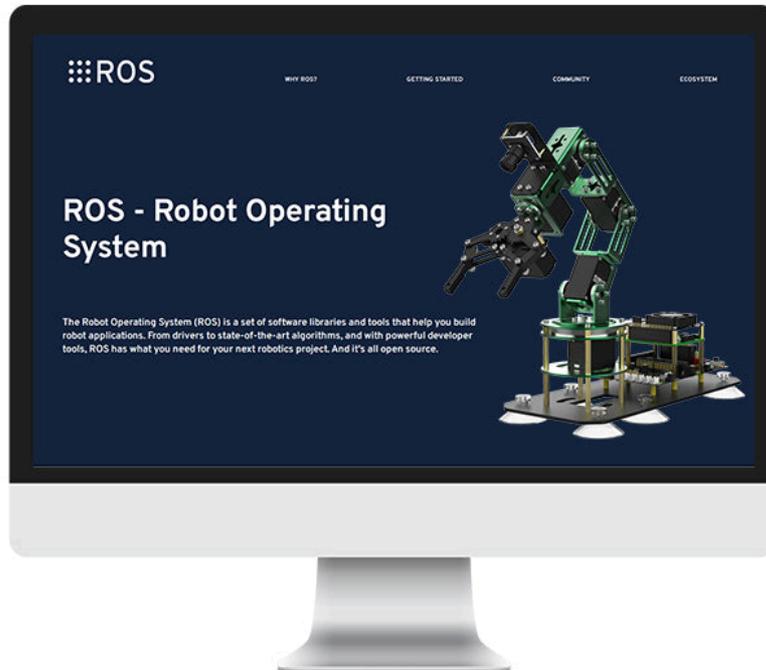
Arm posture control robotic arm



Robotic arm recognition and tracking palm

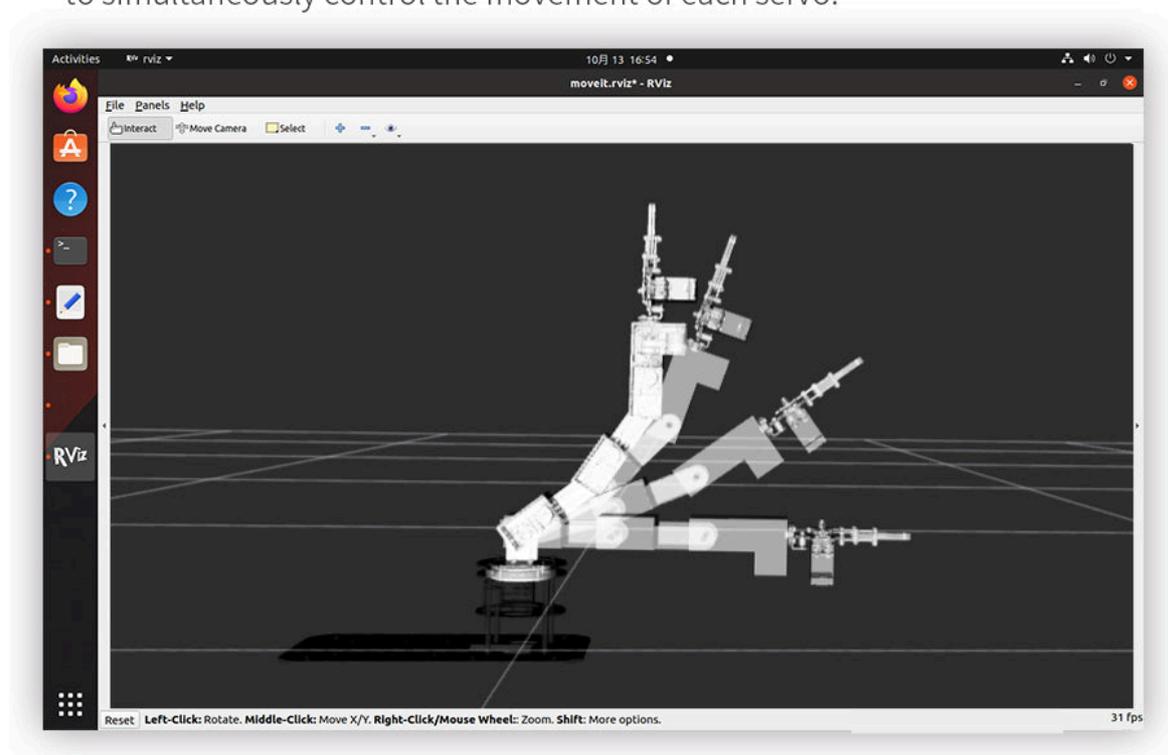
ROS operating system

It is a collection of tools, libraries, and protocols designed to simplify robotics platforms and build complex and powerful robots.



Inverse kinematics algorithm

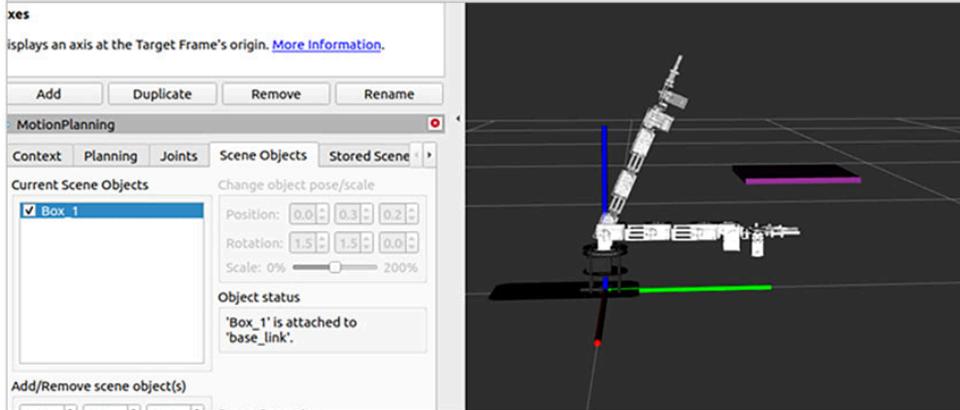
Decompose the movement of the six servos of the robotic arm, calculate the theoretical angle at which each servo needs to move by inputting the target coordinates, and then combine it with the servo control protocol to simultaneously control the movement of each servo.



Robotic arm MoveIt control

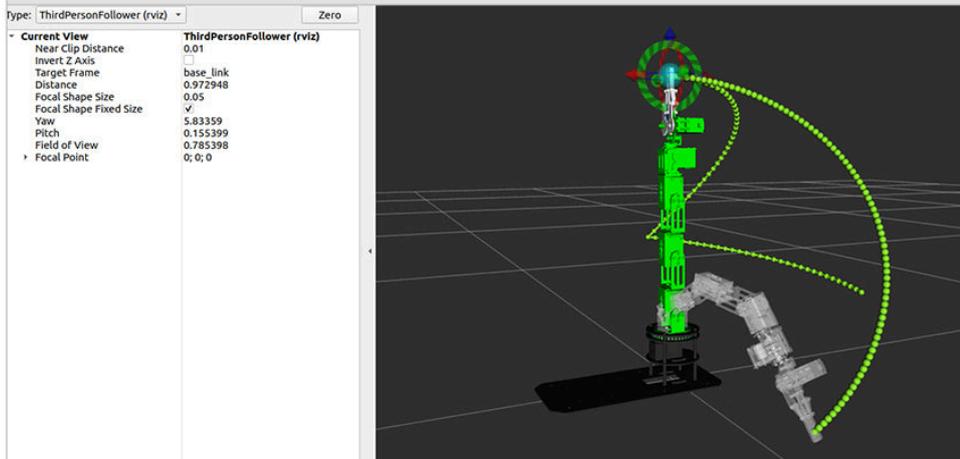
01.URDF kinematics simulation model

Real-scale fully restored model simulation, RVIZ tool visual control. Supports functions such as dragging, preset position control, and robotic arm movement to avoid obstacles.



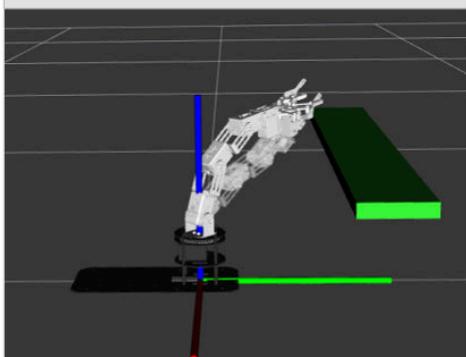
02.MoveIt simulation control/trajectory planning

While the simulation interface controls the movements of the robotic arm, the real robotic arm can make corresponding movements.



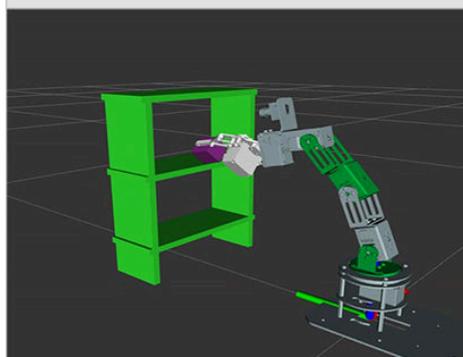
03.Collision detection

Create obstacles in the simulation interface and drive the robotic arm to complete random movements. The robotic arm will avoid obstacles while moving.



04.Space clamping

Import and set up the planned scene, drive the robotic arm to complete the action of picking up and placing objects in the scene, and perform collision detection on the scene.

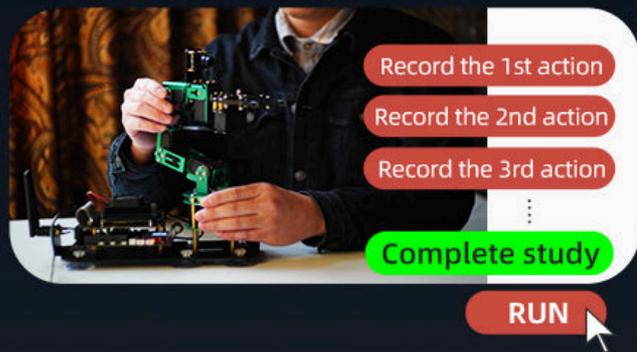


Robot Motion Control



01 Robotic arm custom learning action group

After DOFBOT enters the action learning mode, it can learn and repeatedly perform action groups by reading and recording the angle of each rotation.



02

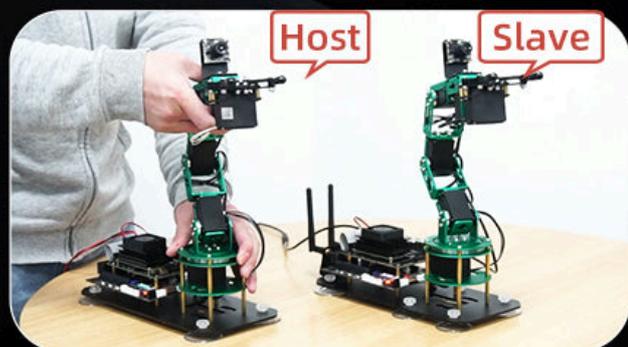
Fixed action group

Yahboom provides 8 fixed action groups on APP. User can click the serial number to preview, and click [Run] to start execution.



03 Robotic arm synchronous teaching

This function requires two robotic arms for teaching control. Read the servo joint angle of the DOFBOT host and transmit it to the slave in real time, allowing the slave to rotate synchronously according to the attitude of the host.

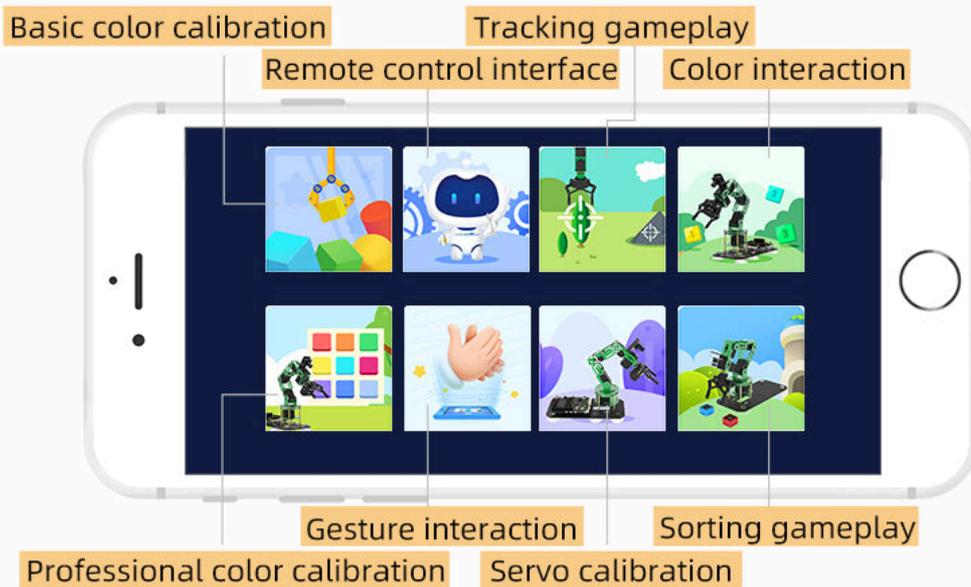


Host posture control
Slave synchronous action

Cross-platform interconnection control

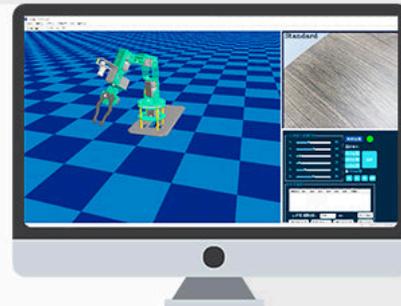
Multifunctional mobile APP (iOS/Android)

Built-in a variety of AI function, you can watch the robotic arm recognition screen in real time and support gesture interaction, mobile tracking, garbage sorting.



PC host computer control

In addition to the FPV camera video, the host computer also adds a 3D simulation model of the robotic arm. The 3D model rotates synchronously with the entity.



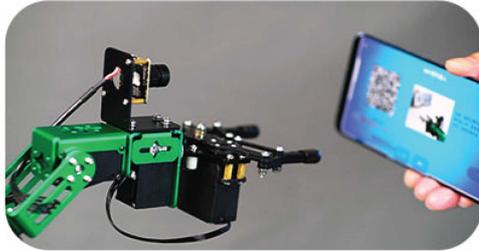
USB handle remote control

In addition to mobile APP and computer control, it also comes standard with a USB handle controller, allowing you to experience the funny of handle control.



Guided quick start

Guided teaching, follow the APP in three steps to get started



1) Scan the connect network

DOFBOT connects to WIFI and pairs by scanning the QR code.



2) Servo calibration

By reading the angle of the servo, perform angle calibration on the robotic arm entity.

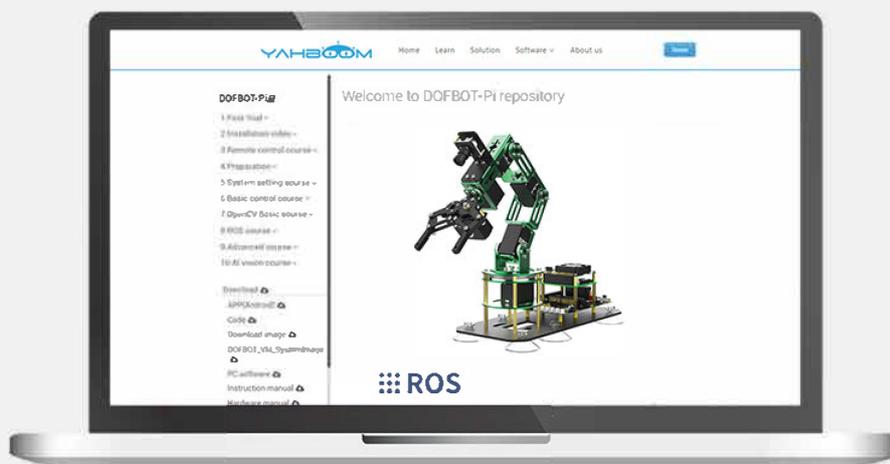


3) Color calibration

For beginners and developers, we provide basic color calibration and professional color calibration functions to avoid environmental factors affecting the robotic arm.

Gift information

Tutorial Link: <http://www.yahboom.net/study/Dofbot-Pi>



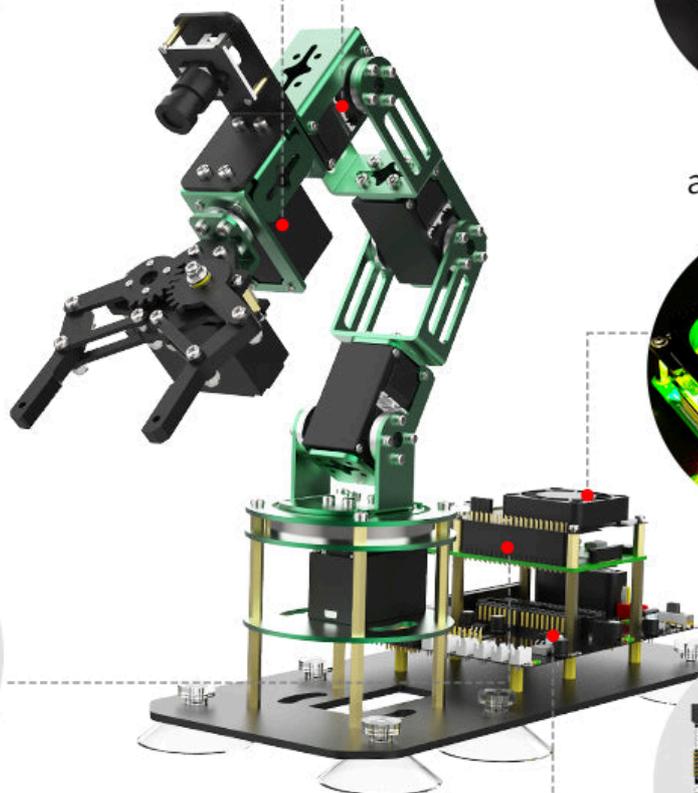
Powerful hardware configuration



Intelligent serial bus servo



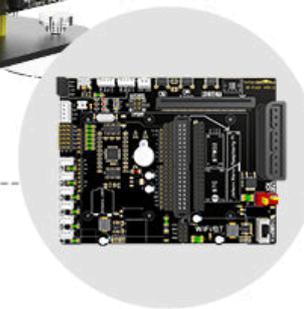
Anodized aluminum alloy



RGB cooling HAT



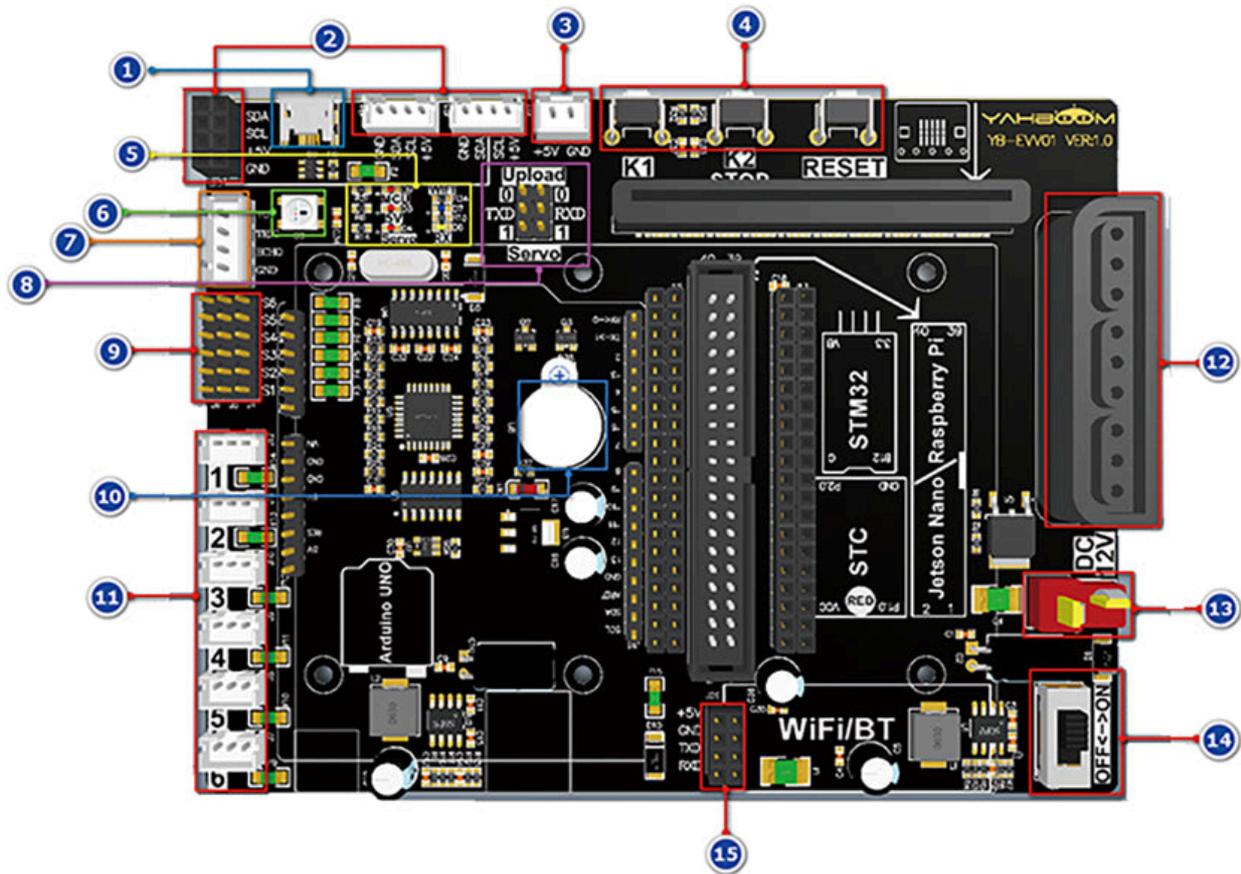
Raspberry Pi (optional)



Robotic arm control board

01 Robotic arm control board

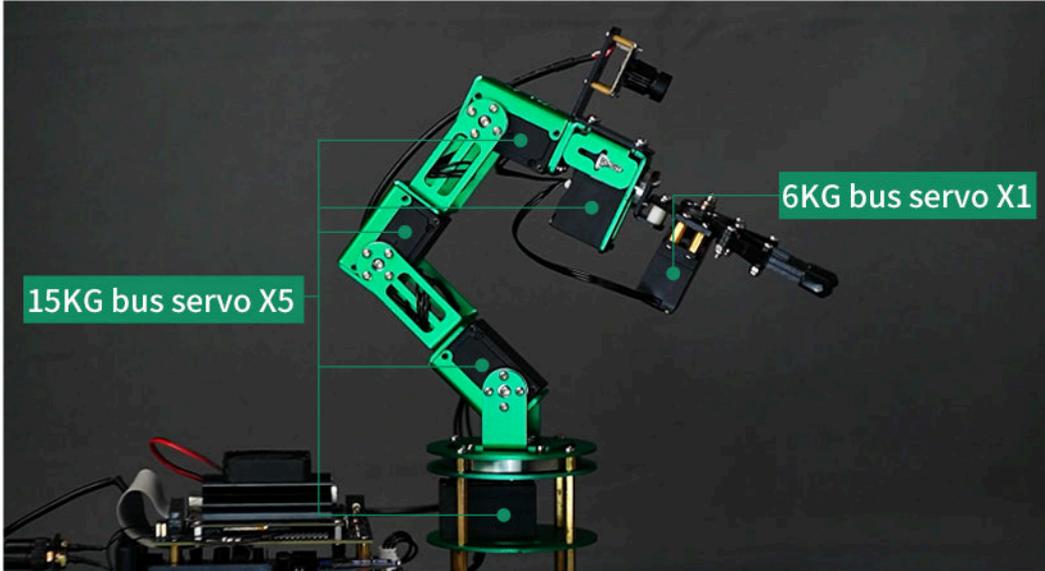
Supports STM32 core board direct plug-in, can drive serial bus servos and PWM servos, has serial ports, handles, and I2C control interfaces. Supports USB one-click download, and Yahboom will provide robotic arm control communication protocols.



- | | | |
|----------------------------------|----------------------------------|-----------------------------------|
| 1、Micro USB interface | 2、I2C interface | 3、Cooling fan interface |
| 4、Button: K1+K2 key+RESET key | 5、Status Indicator | 6、RGB light |
| 7、Ultrasonic interface | 8、Serial port function selection | 9、PWM servo interface |
| 10、Buzzer | 11、Bus servo interface | 12、PS2 controller receiver holder |
| 13、T-type power supply interface | 14、Power switch | 15、Serial interface |

02 Intelligent serial bus servo

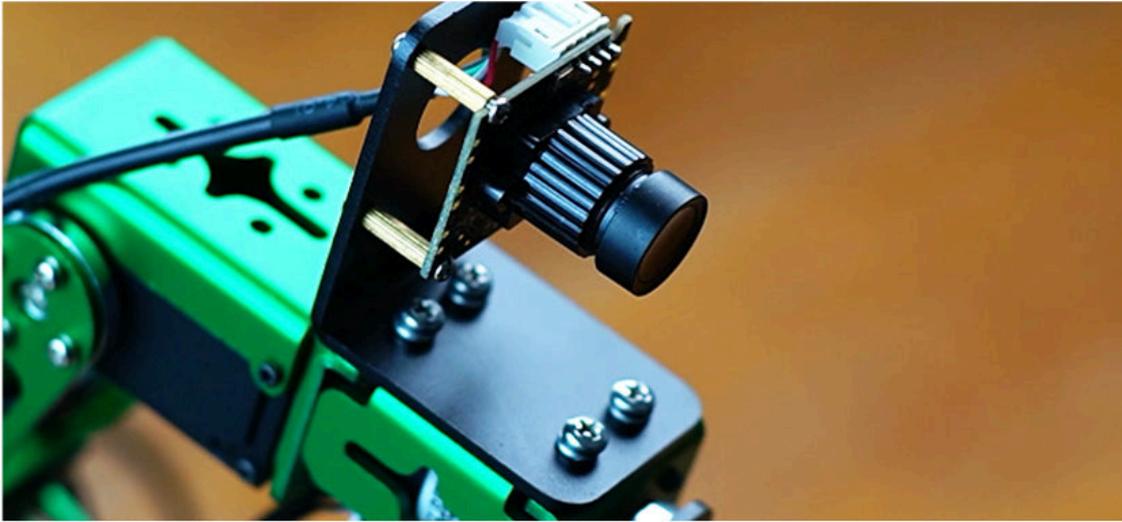
Equipped with five 15KG bus servos, and the gripper is equipped with one 6KG bus servo. It supports reading back the servo position, status and other information. Built-in metal gear is wear-resistant and durable. The high-precision imported potentiometer can effectively extend service life. On board anti-reverse connection interface, cascade control is possible.



Parameter	15KG bus servo (Body servo)	6KG bus servo (Gripper servo)
Size	44.37*23.06*35.12mm	40.6*20*41.8mm
Operating voltage	6.0~7.4V DC	4.8~6.0V DC
Rated torque	≥15kgf.cm at 7.4V	≥6kgf.cm at 6V
Rotation range	300°±15°	180°±10°
Noload current	≤310 mA at 7.4V	≤120 mA at 6V
Stalled rotor current	≤3.2 A at 7.4V	≤1.7A at 6V
Servo accuracy	≤ 1°	
Read back function	Supports reading back servo position, status and other information	
Control method	UART serial port instructions	
Communication baud rate	115200	
Storage	The servo settings are automatically saved when the power is turned off.	
Protection	Locked for 3 seconds and enter protection	
Servo ID	1, 2, 3, 4, 5	6
Gear type	Metal gear	Plastic gear
Interface model	PH2.0-3Pin	

03 USB camera

300,000 pixels, 110° field of view, 480p resolution, and 30fps frame rate, which can provide high-quality image capture and flexible focus control.



Pixel	300,000 pixels	Field of view	110°
Resolution	480P (640*480)	Frame rate	30fps
Connection method	USB 2.0	Focus mode	Focus can be adjusted manually

04 USB Handle

USB handle can control each joint of the robotic arm.



Open Python Source Code

```
#!/usr/bin/env python
# coding: utf-8
import Arm_Lib
import cv2 as cv
import threading
from time import sleep
import IPython.display as widgets
from IPython.display import display
from single_garbage_identify import single_garbage_identify

# 创建实例 初始化参数
single_garbage = single_garbage_identify()
# 初始化摄像头
model = "General"

# 初始化机械臂位置
import Arm_Lib
arm = Arm_Lib.Arm_Device()
joints_0 = [0, 90, 70, 11, 90, 30]
arm.arm_serial_servo_writes_array(joints_0, 1000)

gms = tf.config.experimental.list_physical_devices(device_type='GPU')
for gpu in gms: tf.config.experimental.set_memory_growth(gpu, True)

class garbage_identify:
    def __init__(self):
        self.score = 0.5
        self.loc = 0.3
        self.eager = False
        # 初始化摄像头
        self.arm = Arm_Lib.Arm_Device()
        self.anchors_path = '/home/robot/robot_ws/src/robot_garbage_identify/
        self.class_path = '/home/robot/robot_ws/src/robot_garbage_identify/
        self.model_path = '/home/robot/robot_ws/src/robot_garbage_identify/
        self.finet_path = '/home/robot/robot_ws/src/robot_garbage_identify/
        self.class_names = self.get_class()
        self.anchors = self.get_anchors()
        self.model_image_size = (416, 416)
        if not self.eager:
            tf.compat.v1.disable_eager_execution()
            self.sess = tf.Session()
        self.generate()
        # 生成随机初始值
        self.colors = [[random.randint(0, 255) for _ in range(3)] for _ in range
        self.ny = [0, 255]
        self.nx = [0, 255]
        # 初始化摄像头位置
        self.grap_move = garbage_grap_move()
        self.garbage_index = 0
        # 摄像头
        self.v = rospy.get_node('/robot_ros_arm', anonymous=True)
        # 摄像头订阅消息服务
        self.client = rospy.ServiceProxy('get_kinematics', Kinematics)
        # 摄像头订阅
        def get_class(self):
            # 摄像头订阅

static_image_mode=staticMode,
smooth_landmarks=landmarks,
min_detection_confidence=detectionCon,
min_tracking_confidence=trackingCon,
self.pub_point = rospy.Publisher('/mediapipe/points', PointArray, PointArray())
self.lndrawSpec = mp.solutions.drawing_utils.DrawingSpec(c
self.drawSpec = mp.solutions.drawing_utils.DrawingSpec(col

def findHolistic(self, frame, draw=True):
    pointArray = PointArray()
    img = np.zeros((frame.shape[0], frame.shape[1]), np.uint8)
    img_RGB = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
    self.results = self.mpholistic.process(img_RGB)
    if self.results.face_landmarks:
        if draw: self.mpDraw.draw_landmarks(frame, self.result
        self.mpDraw.draw_landmarks(img, self.results.face_land
        for id, lm in enumerate(self.results.face_landmarks.la
            point = Point()
            point.x, point.y, point.z = lm.x, lm.y, lm.z
            pointArray.points.append(point)
    if self.results.pose_landmarks:
        if draw: self.mpDraw.draw_landmarks(frame, self.result
        self.mpDraw.draw_landmarks(img, self.results.pose_land
        for id, lm in enumerate(self.results.pose_landmarks.la
            point = Point()
            point.x, point.y, point.z = lm.x, lm.y, lm.z
            pointArray.points.append(point)
    if self.results.left_hand_landmarks:
```

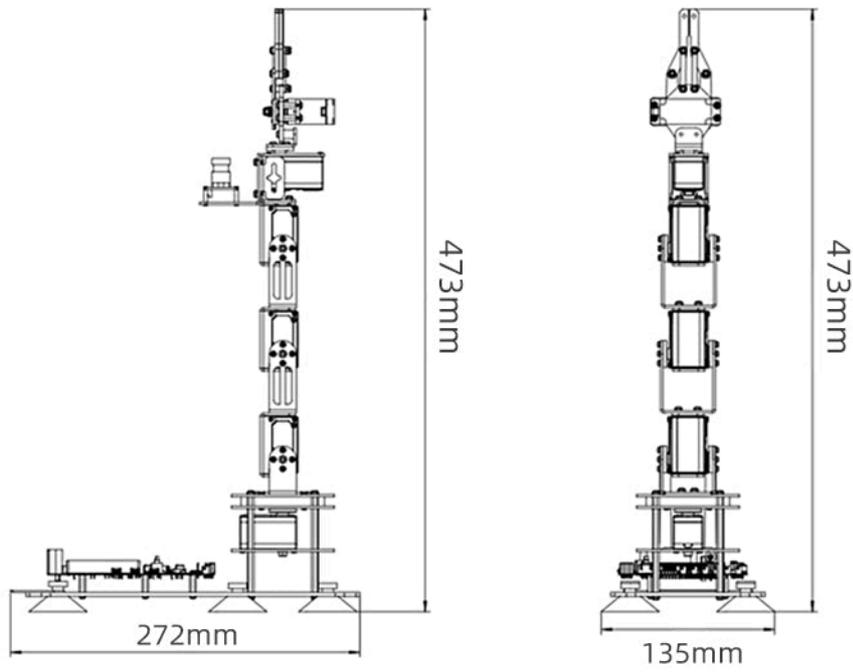


Provide product 3D models

Yahboom will provide 3D model of robotic arm for free.

Product Specifications

Product Size



Tutorial

[Yahboom Raspberry Pi DOFBOT AI Vision Robotic Arm](#)