

Palm keypoint classification

Palm keypoint classification

[Routine Experiment Effect](#)

[Code Explanation](#)

[Code structure](#)

[Code Explanation](#)

[flow chart](#)

Routine Experiment Effect

In this section, we classify based on key point detection

After connecting to the IDE, run the sample code in this section and make different gestures to K230, which can be recognized.

The supported gestures are as follows:

Fist: All fingers bent

Five fingers spread: all fingers are extended

Pistol: thumb and index finger extended, other fingers bent

Heart shape: thumb, index finger and pinky finger extended, other fingers bent

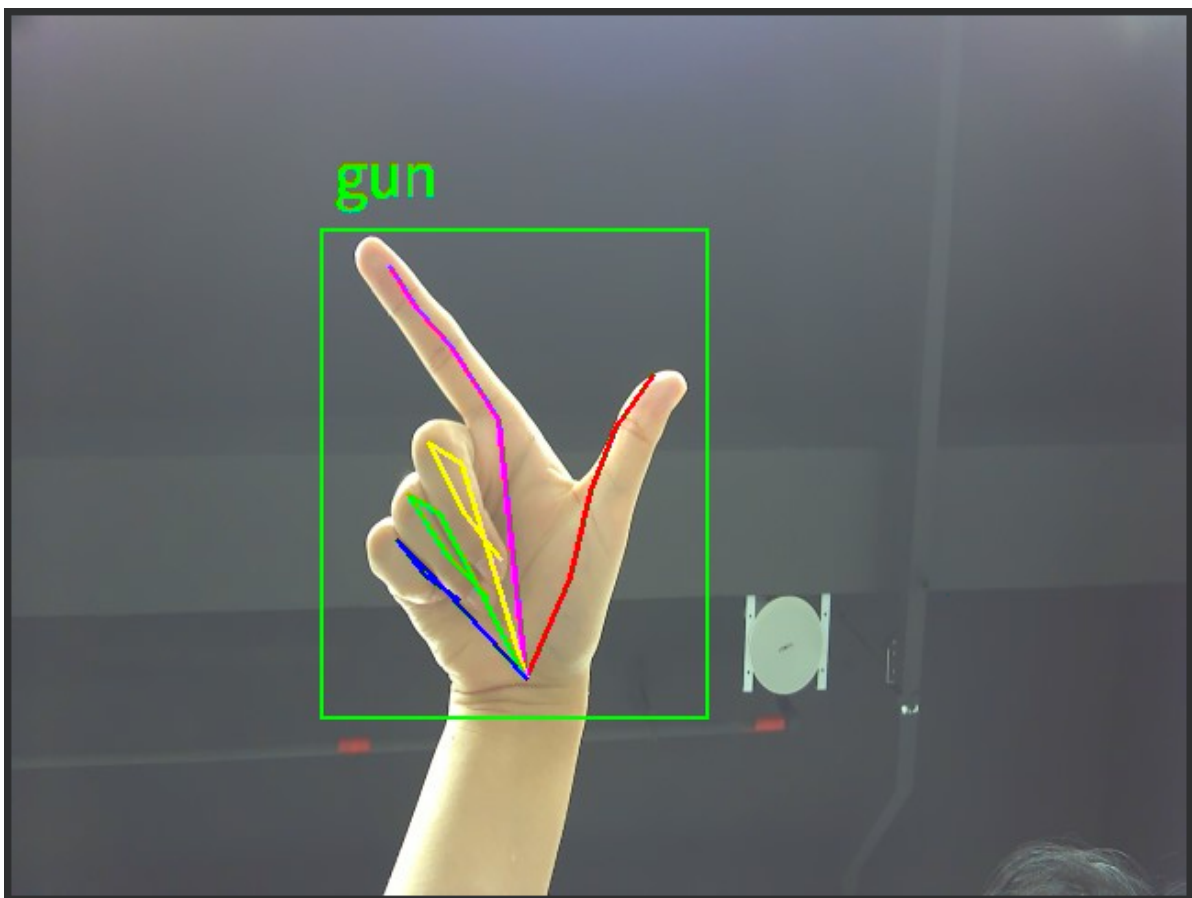
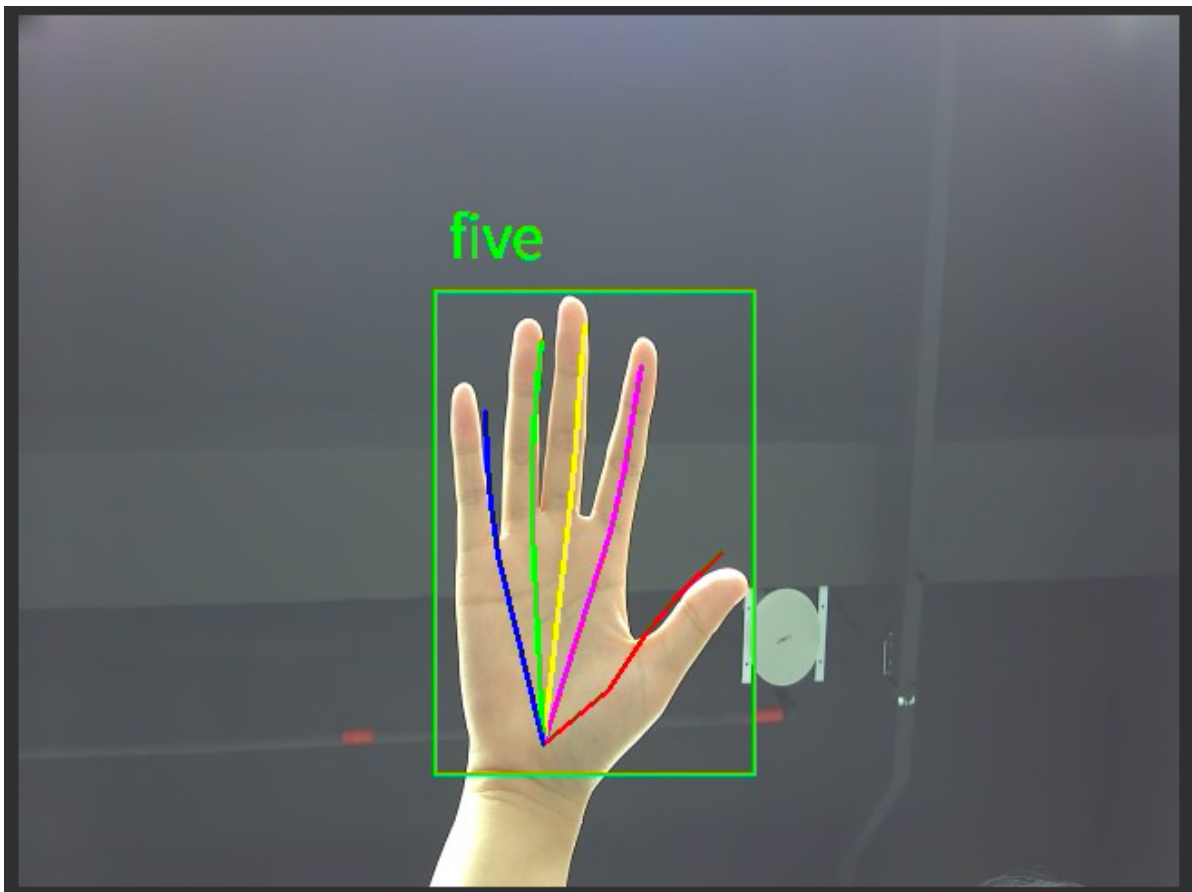
Digit One: Index finger extended, other fingers bent

Digit six: thumb and pinky extended, other fingers bent

Number three: Thumb bent, index, middle and ring fingers extended, pinky bent

Thumbs up: thumb extended, other fingers bent

Yay: Thumb bent, index and middle fingers extended, other fingers bent



Code Explanation

Code structure

1. Initialization Phase:
 - System parameter configuration
 - Create image pipeline
 - Initialize detector
2. Main Detection Loop:
 - Image acquisition
 - Hand detection
 - Detection box filtering
 - Keypoint detection
3. Keypoint Processing:
 - Image preprocessing
 - Keypoint extraction
 - Result visualization
4. Visualization:
 - Draw bounding boxes
 - Draw keypoints
 - Draw connection lines
5. Error Handling:
 - Exception catching
 - Resource cleanup

Code Explanation

For the complete code, please refer to the file [Source Code/08.Body/07. hand_keypoint_class.py]

The core code for judging gestures is as follows

```
# 根据手掌关键点检测结果判断手势类别
# Determine gesture category based on hand keypoint detection results
def hk_gesture(self, results):
    with ScopedTiming("hk_gesture", self.debug_mode > 0):
        # 初始化角度列表
        # Initialize angle list
        angle_list = []
        # 计算每个手指的角度
        # Calculate angle for each finger
        for i in range(5):
            angle = self.hk_vector_2d_angle([(results[0]-results[i*8+4]),
            (results[1]-results[i*8+5])], [(results[i*8+6]-results[i*8+8]), (results[i*8+7]-
            results[i*8+9])])
            angle_list.append(angle)
        # 设置角度阈值和初始手势字符串
        # Set angle thresholds and initial gesture string
        thr_angle, thr_angle_thumb, thr_angle_s, gesture_str = 65., 53., 49., None
        # 如果所有角度都有效, 则根据角度判断手势
        # If all angles are valid, determine gesture based on angles
        if 65535. not in angle_list:
            # 拳头: 所有手指都弯曲
            # Fist: all fingers are bent
```

```

        if (angle_list[0]>thr_angle_thumb) and
(angle_list[1]>thr_angle) and (angle_list[2]>thr_angle) and
(angle_list[3]>thr_angle) and (angle_list[4]>thr_angle):
            gesture_str = "fist"
            # 五指张开：所有手指都伸展
            # Five fingers open: all fingers are extended
        elif (angle_list[0]<thr_angle_s) and (angle_list[1]
<thr_angle_s) and (angle_list[2]<thr_angle_s) and (angle_list[3]<thr_angle_s)
and (angle_list[4]<thr_angle_s):
            gesture_str = "five"
            # 手枪：拇指和食指伸展，其他手指弯曲
            # Gun: thumb and index finger extended, other fingers bent
        elif (angle_list[0]<thr_angle_s) and (angle_list[1]
<thr_angle_s) and (angle_list[2]>thr_angle) and (angle_list[3]>thr_angle) and
(angle_list[4]>thr_angle):
            gesture_str = "gun"
            # 爱心：拇指、食指和小指伸展，其他手指弯曲
            # Love: thumb, index finger and pinky extended, other fingers
bent
        elif (angle_list[0]<thr_angle_s) and (angle_list[1]
<thr_angle_s) and (angle_list[2]>thr_angle) and (angle_list[3]>thr_angle) and
(angle_list[4]<thr_angle_s):
            gesture_str = "love"
            # 数字一：食指伸展，其他手指弯曲
            # Number one: index finger extended, other fingers bent
        elif (angle_list[0]>5) and (angle_list[1]<thr_angle_s) and
(angle_list[2]>thr_angle) and (angle_list[3]>thr_angle) and
(angle_list[4]>thr_angle):
            gesture_str = "one"
            # 数字六：拇指和小指伸展，其他手指弯曲
            # Number six: thumb and pinky extended, other fingers bent
        elif (angle_list[0]<thr_angle_s) and (angle_list[1]>thr_angle)
and (angle_list[2]>thr_angle) and (angle_list[3]>thr_angle) and (angle_list[4]
<thr_angle_s):
            gesture_str = "six"
            # 数字三：拇指弯曲，食指、中指和无名指伸展，小指弯曲
            # Number three: thumb bent, index, middle and ring fingers
extended, pinky bent
        elif (angle_list[0]>thr_angle_thumb) and (angle_list[1]
<thr_angle_s) and (angle_list[2]<thr_angle_s) and (angle_list[3]<thr_angle_s)
and (angle_list[4]>thr_angle):
            gesture_str = "three"
            # 竖大拇指：拇指伸展，其他手指弯曲
            # Thumbs up: thumb extended, other fingers bent
        elif (angle_list[0]<thr_angle_s) and (angle_list[1]>thr_angle)
and (angle_list[2]>thr_angle) and (angle_list[3]>thr_angle) and
(angle_list[4]>thr_angle):
            gesture_str = "thumbUp"
            # 耶：拇指弯曲，食指和中指伸展，其他手指弯曲
            # Yeah: thumb bent, index and middle fingers extended, other
fingers bent
        elif (angle_list[0]>thr_angle_thumb) and (angle_list[1]
<thr_angle_s) and (angle_list[2]<thr_angle_s) and (angle_list[3]>thr_angle) and
(angle_list[4]>thr_angle):
            gesture_str = "yeah"
    return gesture_str

```

flow chart

