代码：

import cv2

import numpy as np

import matplotlib.pyplot as plt

ori\_file\_path = "./pic.jpg"

def show\_image(desc, image):

    cv2.imshow(desc, image)

    cv2.waitKey(0)

    cv2.destroyAllWindows()

small\_size=(240, 320)

# 轮廓检测

def check\_contour():

    pic = cv2.imread(ori\_file\_path)

    pic\_gray = cv2.cvtColor(pic, cv2.COLOR\_BGR2GRAY)

    ret, binary\_pic = cv2.threshold(pic\_gray, 127, 255, cv2.THRESH\_BINARY)

    contours,\_ = cv2.findContours(binary\_pic, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

    cv2.drawContours(pic, contours, -1, (0, 255, 0), 3)

    pic = cv2.resize(pic,(240, 320))

    # show\_image("contour", pic)

    cv2.imwrite("contour.jpg",pic)

# 二值图像

def binary\_img():

    pic\_gray = cv2.imread(ori\_file\_path, 0)

    # pic\_gray = cv2.resize(pic\_gray,(127,170))

    ret,binary=cv2.threshold(pic\_gray,127,255,cv2.THRESH\_BINARY)

    # show\_image("binary", binary)

    cv2.imwrite("binary.jpg",binary)

# 形态学操作

def opt\_morphology():

    # 腐蚀

    pic = cv2.imread(ori\_file\_path, 0)

    kernel\_er = np.ones((9,9), np.uint8)

    eroded = cv2.morphologyEx(pic, cv2.MORPH\_ERODE, kernel\_er)

    cv2.imwrite("er.jpg", eroded)

    # plt.subplot(141)

    # plt.imshow(eroded)

    # 膨胀

    kernel\_ex = np.ones((9,9), np.uint8)

    extension = cv2.morphologyEx(pic, cv2.MORPH\_DILATE, kernel\_ex)

    # plt.subplot(142)

    # plt.imshow(extension)

    cv2.imwrite("ex.jpg", extension)

    # 开 腐蚀-膨胀

    open\_pic = cv2.morphologyEx(pic, cv2.MORPH\_ERODE, kernel\_er)

    open\_pic = cv2.morphologyEx(open\_pic, cv2.MORPH\_DILATE, kernel\_ex)

    # plt.subplot(143)

    # plt.imshow(open\_pic)

    cv2.imwrite("open.jpg", open\_pic)

    # 闭 膨胀-腐蚀

    close\_pic = cv2.morphologyEx(pic, cv2.MORPH\_DILATE, kernel\_ex)

    close\_pic = cv2.morphologyEx(close\_pic, cv2.MORPH\_ERODE, kernel\_er)

    # plt.subplot(144)

    # plt.imshow(close\_pic)

    cv2.imwrite("close.jpg", close\_pic)

    plt.show()

# canny检测边缘

def check\_outline():

    pic = cv2.imread(ori\_file\_path)

    edged = cv2.Canny(pic, 30, 200)

    result\_pic = cv2.resize(edged,(240,320))

    # show\_image('canny', result\_pic)

    cv2.imwrite("canny.jpg", result\_pic)

# 颜色识别 找到原图中最多的颜色

def color\_recognize():

    pic = cv2.imread(ori\_file\_path)

    ball\_color = 'green'

    color\_dist = {'red': {'Lower': np.array([0, 60, 60]), 'Upper': np.array([6, 255, 255])},

              'blue': {'Lower': np.array([100, 80, 46]), 'Upper': np.array([124, 255, 255])},

              'green': {'Lower': np.array([35, 43, 35]), 'Upper': np.array([90, 255, 255])},

              }

    gs\_frame = cv2.GaussianBlur(pic, (5, 5), 0)

    hsv = cv2.cvtColor(gs\_frame, cv2.COLOR\_BGR2HSV)

    erode\_hsv = cv2.erode(hsv, None, iterations=2)

    inRange\_hsv = cv2.inRange(erode\_hsv, color\_dist[ball\_color]['Lower'], color\_dist[ball\_color]['Upper'])

    cnts = cv2.findContours(inRange\_hsv.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)[-2]

    cv2.drawContours(pic, cnts, -1, (0, 255, 255), 2)

    # cv2.imshow('camera', pic)

    # cv2.waitKey(0)

    cv2.imwrite("col\_rec.jpg", pic)

# 图片滤波

def pic\_filter():

    pic = cv2.imread(ori\_file\_path)

    pic = cv2.resize(pic,(240,320))

    GaussianBlur=cv2.GaussianBlur(pic,(5,5),1)

    plt.subplot(211)

    plt.imshow(GaussianBlur)

    plt.subplot(212)

    plt.imshow(pic)

    plt.savefig("filter.jpg")

check\_contour()

binary\_img()

check\_outline()

pic\_filter()

opt\_morphology()

color\_recognize()

原图：



轮廓检测：



阈值处理



**形态学操作**

**腐蚀**

****

**膨胀**

****

**开**

****

**闭**



**边缘检测**



**颜色块识别（绿色）**



**图像滤波**

