第一题

代码：

import cv2  
import numpy as np  
  
img1 = cv2.imread(&apos;../label.jpg&apos;)  
img = cv2.GaussianBlur(img1, (5, 5), 0)#将图形进行二值化，用于后面的自适应阈值二值化  
                                       #高斯滤波，(5, 5)表示高斯矩阵的长与宽都是5，标准差取0，，没有此运用函数，得到的图形将有明显的噪声，白色中具有较多的黑点。  
gray= cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)#将图片灰度化  
dst = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, \  
              cv2.THRESH\_BINARY, 11, 2)#自适应阈值化，此函数的参数分别是：灰度的原图，最大阈值，阈值计算方法，二值化的计算方法，图片中分块的大小，阈值计算方法中的常数项  
kernel = np.ones((1, 1), np.uint8)# 使用一个1\*1的卷积核  
dst = cv2.dilate(dst, kernel, iterations=-1)# 腐蚀操作,参数分别是原图，卷积和，迭代为1.  
dst = cv2.erode(dst, kernel, iterations=1)# 膨胀操作  
  
cv2.namedWindow(&apos;window&apos;, cv2.WINDOW\_NORMAL)  
cv2.imshow(&apos;window&apos;,dst)  
cv2.waitKey(0)  
cv2.destroyWindow()

效果图：



第二题

代码：

import cv2  
import numpy as np  
  
img = cv2.imread(&apos;../gun.jpg&apos;)  
hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)# 将图片转化为HSV格式，此格式为三通道  
  
low = np.array([0, 153, 28])# 获取图像的阈值，创建掩膜  
up = np.array([80, 232, 197])  
mask = cv2.inRange(hsv, low, up)#黑白图  
img1 = cv2.bitwise\_and(img, img, mask=mask)#输出橙色图像（目标图像）  
  
gray = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY)# 将图形转为灰度，用于后面自适应二值化  
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)#进行自适应化二值化，把图片转化为黑白模式，用于寻找图片轮廓  
contours, hie = cv2.findContours(thresh, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)#cv2.findContours()函数用来查找检测物体的轮廓  
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:5]#在多个矩形轮廓中选取最适合图片的轮廓，即轮廓中最大的那个  
cnt = contours[0]  
M = cv2.moments(cnt)#cnt是矩形边框的点集  
  
x, y, w, h = cv2.boundingRect(cnt)#绘制最适应轮廓的矩形，并输出矩形的信息；cnt为轮廓点集合；x，y是矩阵左上点的坐标，w，h是矩阵的宽和高  
img = cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)  
  
x1= int(M["m10"] / M["m00"])#矩形中心  
y1 = int(M["m01"] / M["m00"])  
  
print("矩形中心和长宽分别为：")  
print((x1,y1),w,h)  
  
(x, y), radius = cv2.minEnclosingCircle(cnt)#绘制最小的外接圆，并输出圆的信息  
center = (int(x), int(y))  
radius = int(radius)  
img = cv2.circle(img, center, radius, (0, 255, 0), 2)  
print("圆心、半径为：")  
print((x, y),radius)  
  
# 获取四个极点  
left = tuple(cnt[cnt[:, :, 0].argmin()][0])  
right = tuple(cnt[cnt[:, :, 0].argmax()][0])  
top = tuple(cnt[cnt[:, :, 1].argmin()][0])  
bottom = tuple(cnt[cnt[:, :, 1].argmax()][0])  
print(&apos;极点参数为：&apos;)  
print(left)  
print(right)  
print(top)  
print(bottom)  
  
cv2.namedWindow(&apos;new&apos;, cv2.WINDOW\_NORMAL)  
cv2.imshow(&apos;new&apos;, img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

效果图：

