In [46]:

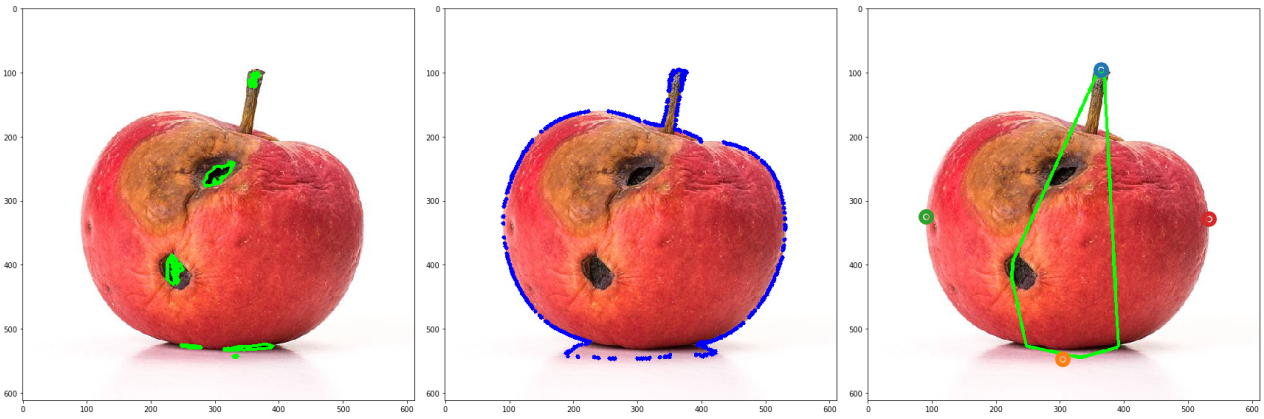
```
import cv2
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image


def find_defects(orig_image):
    lower_red1 = np.array([0,50,50])
    upper_red1 = np.array([255,255,255])

    hsv = cv2.cvtColor(orig_image, cv2.COLOR_RGB2HSV)
    imgray = cv2.cvtColor(orig_image,cv2.COLOR_RGB2GRAY)
    kernel_size = 3
    mask = cv2.inRange(hsv, lower_red1, upper_red1)


    res = cv2.bitwise_and(orig_image,orig_image, mask= mask)
    ret,thresh = cv2.threshold(res,1,255, cv2.THRESH_BINARY)
    imgray = cv2.cvtColor(thresh,cv2.COLOR_RGB2GRAY)
    image, contours, hierarchy = cv2.findContours(imgray,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)



    list_areas=[]
    list_contour=[]
    contour_mod=[]
    for contour in contours:
        area=cv2.contourArea(contour)
        if(area>10):
            list_areas.append(area)
            contour_mod.append(contour)

    list_areas = np.array(list_areas)
    list_contour = np.array(contour_mod)
    inds = list_areas.argsort()[::-1][:10]
    sorted_Contour= list_contour[inds]
    img = cv2.drawContours(np.copy(orig_image), sorted_Contour[1:], -1, (0,255,0), 3)
    return sorted_Contour,img
pixel_by_cm=1#6/1464

def get_four_object_pts(Contour):
    list_y=[]
    list_x=[]
    for point in Contour:
        x=point[0][0]
        y=point[0][1]
        list_y.append(y)
        list_x.append(x)
    list_y = np.array(list_y)
    list_x = np.array(list_x)
    inds = list_y.argsort()#[::-1]
    sorted_x_by_y= list_x[inds]
    sorted_y=sorted(list_y)

    inds = list_x.argsort()#[::-1]
    sorted_x=sorted(list_x)
    sorted_y_by_x=list_y[inds]

    top_point=[sorted_x_by_y[0], sorted_y[0]]
    bottom_point=[sorted_x_by_y[len(sorted_x_by_y)-1], sorted_y[len(sorted_y)-1]]
    left_point=[sorted_x[0], sorted_y_by_x[0]]
    right_point=[sorted_x[len(sorted_x)-1], sorted_y_by_x[len(sorted_y_by_x)-1]]

    height=(bottom_point[1]-top_point[1])*pixel_by_cm
    width=(right_point[0]-left_point[0])*pixel_by_cm
    print("Height",height)
    print("Width",width)
#     plt.imshow(orig_image)
    return top_point,bottom_point,left_point,right_point,height,width

def check_contour_in_object(sorted_Contour,top_point,bottom_point,left_point,right_point):
    for contour in sorted_Contour[1:]:
        M = cv2.moments(contour)
        cx = int(M['m10']/M['m00'])
        cy = int(M['m01']/M['m00'])
        if(top_point[1]<cy<bottom_point[1] and left_point[0]<cx<right_point[0]):

            return True
        else:

            return False

def draw_defect_convex(sorted_Contour,orig_image):
    defect_contour=[]
    hull=[]
    for contour in sorted_Contour[1:]:
        for points in contour:
            defect_contour.append(points)
```

```
    # creating convex hull object for each contour
    hull.append(cv2.convexHull(np.array(defect_contour), False))
    img = cv2.drawContours(np.copy(orig_image), hull, -1, (0,255,0), 3)
    return img

def plot_object_boundary_pts(top_point,bottom_point,left_point,right_point,height,width):
    plt.scatter([top_point[0]], [top_point[1]],linewidths=15)
    plt.scatter([bottom_point[0]],[bottom_point[1]],linewidths=15)
    plt.scatter([left_point[0]],[left_point[1]],linewidths=15)
    plt.scatter([right_point[0]],[right_point[1]],linewidths=15)

def plot_contour_boundary(orig_image,sorted_Contour):
    img = cv2.drawContours(np.copy(orig_image), sorted_Contour, -1, (0,0,255), 5)
    return img


def pipeline_fruits(orig_image):
    sorted_Contour,image_defects=find_defects(np.copy(orig_image))
    object_contour=sorted_Contour[0]
    top_point,bottom_point,left_point,right_point,height,width=get_four_object_pts(object_contour)
    ret=check_contour_in_object(sorted_Contour,top_point,bottom_point,left_point,right_point)
    boundary_img=plot_contour_boundary(np.copy(orig_image),object_contour)
    if(ret):
        print("Defects detected")
        convex_img=draw_defect_convex(sorted_Contour,np.copy(orig_image))
        f, [ax1, ax2,ax3] = plt.subplots(1, 3, figsize=(24, 9))
        f.tight_layout()
        ax1.imshow(image_defects)
        plot_object_boundary_pts(top_point,bottom_point,left_point,right_point,height,width)
        ax2.imshow(boundary_img)
        ax3.imshow(convex_img)
    else:
        print("No Defects detected")
        plt.imshow(boundary_img)
        plot_object_boundary_pts(top_point,bottom_point,left_point,right_point,height,width)
```

In [47]:

```
orig_image=plt.imread("rot_1.jpg")
pipeline_fruits(orig_image)
```

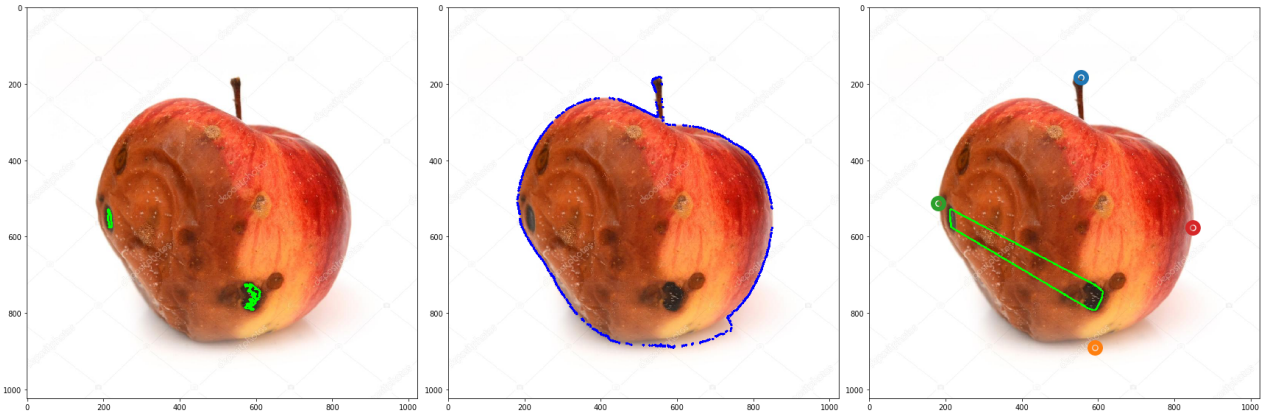
```
Height 452
Width 441
Defects detected
```



In [48]:

```
orig_image=plt.imread("rot_2.jpg")
pipeline_fruits(orig_image)
```

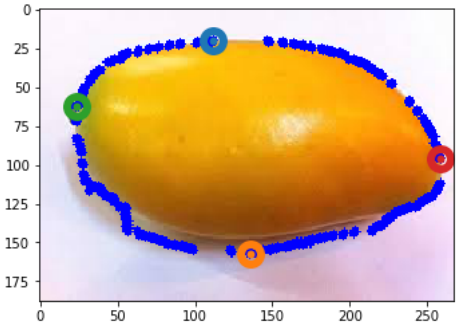
```
Height 708
Width 668
Defects detected
```



In [49]:

```
orig_image=plt.imread("mn_1.jpg")
pipeline_fruits(orig_image)
```

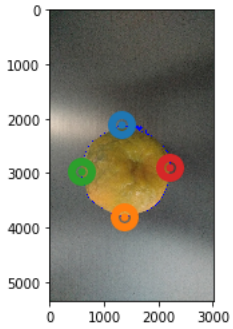
```
Height 137
Width 236
No Defects detected
```



In [50]:

```
orig_image=plt.imread("or_1.jpg")
pipeline_fruits(orig_image)
```

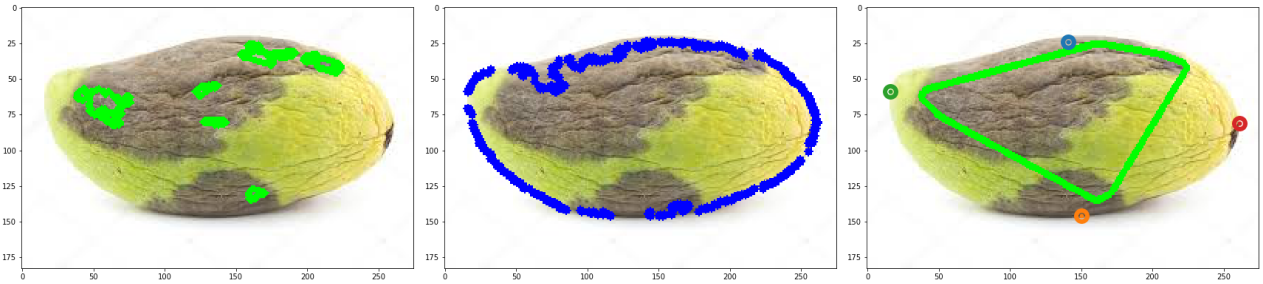
```
Height 1688
Width 1636
No Defects detected
```



In [51]:

```
orig_image=plt.imread("mnr_1.jpg")
pipeline_fruits(orig_image)
```

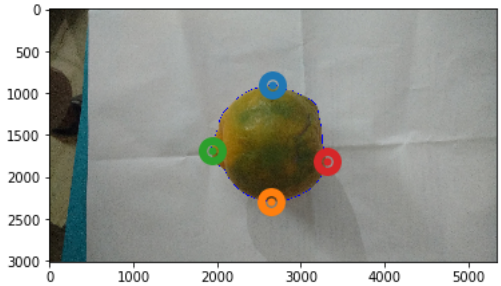
```
Height 122
Width 245
Defects detected
```



In [52]:

```
orig_image=plt.imread("or_4.jpg")
pipeline_fruits(orig_image)
```

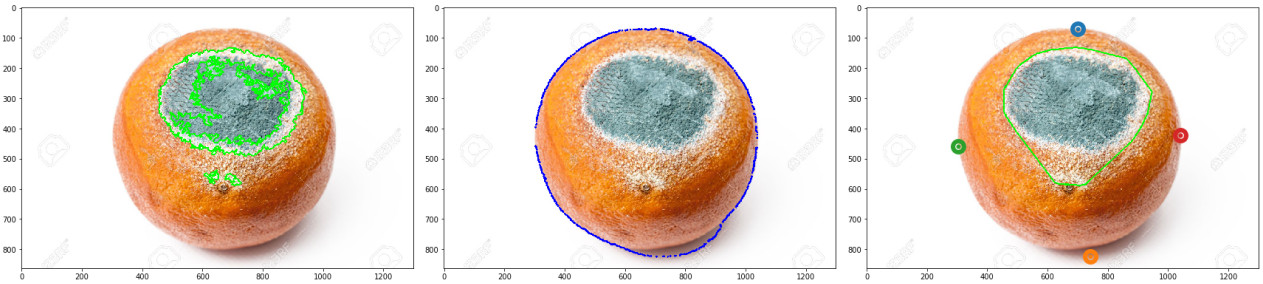
```
Height 1383
Width 1377
No Defects detected
```



In [56]:

```
orig_image=plt.imread("orr3.jpg")
pipeline_fruits(orig_image)
```

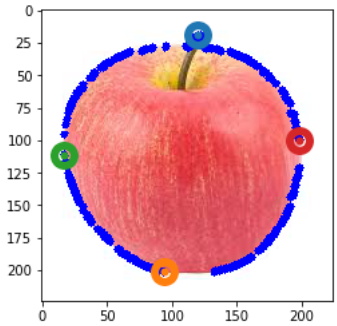
```
Height 755
Width 736
Defects detected
```



In [60]:

```
orig_image=plt.imread("a1.jpg")
pipeline_fruits(orig_image)
```

```
Height 182
Width 181
No Defects detected
```



In [ ]:

In [ ]: