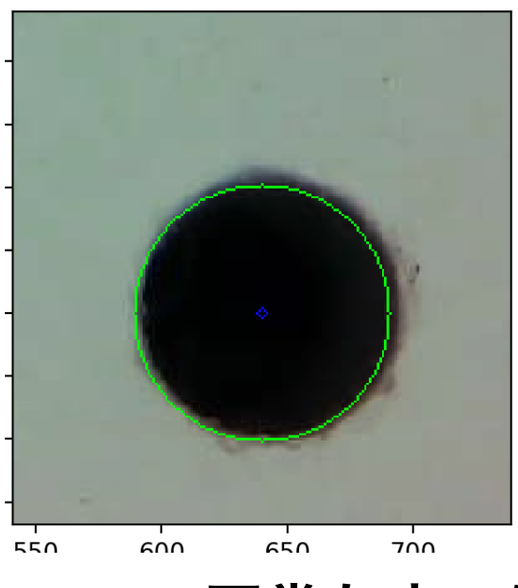
円検出、中心矯正の例

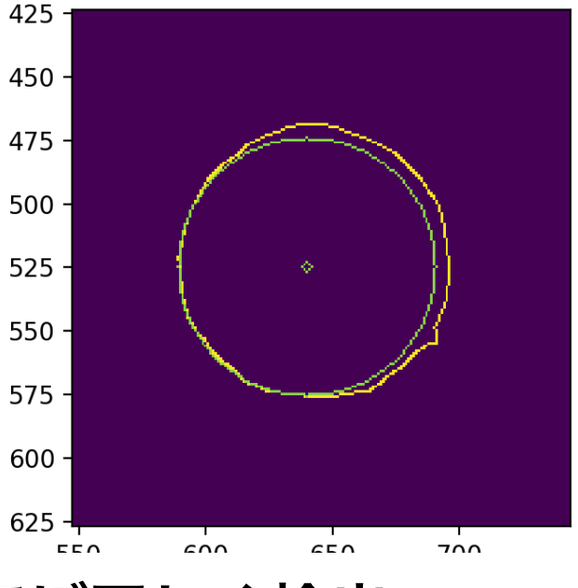
2019.11.24 谷

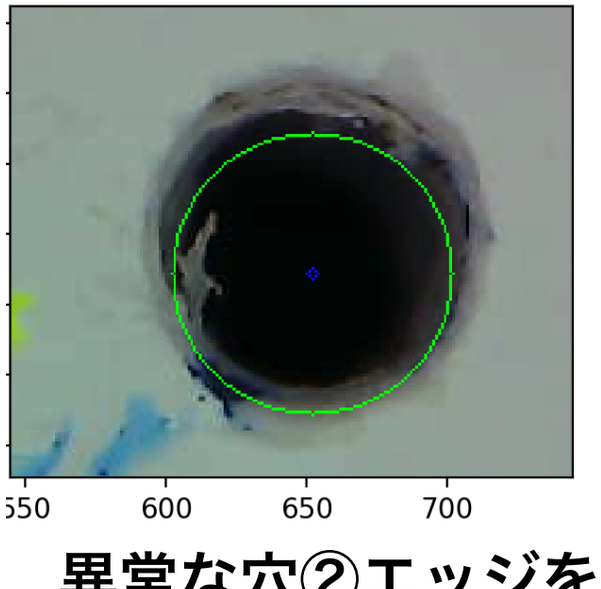
threshold の値ごとの穴の検出具合

- threshold 100（高すぎ）

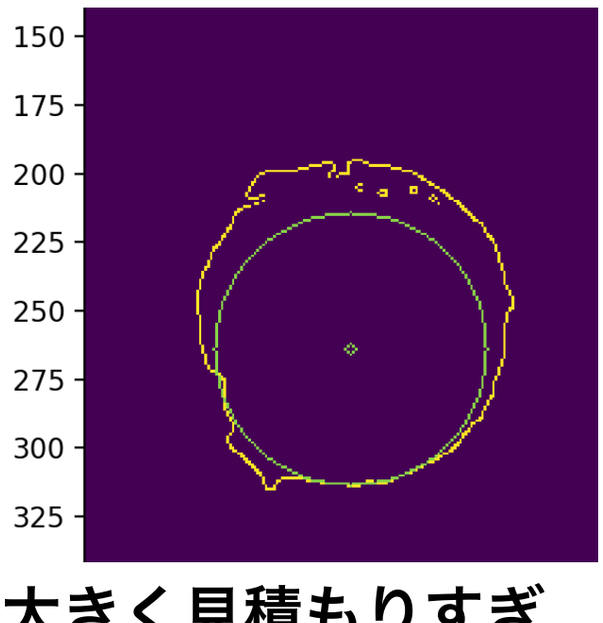


正常な穴、ほぼ正しく検出

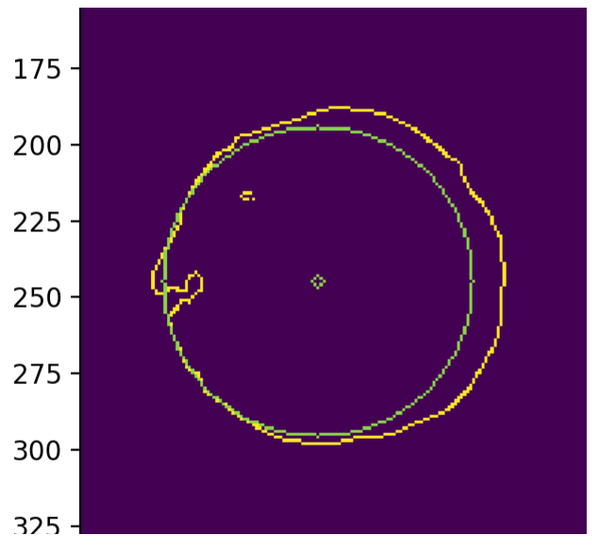




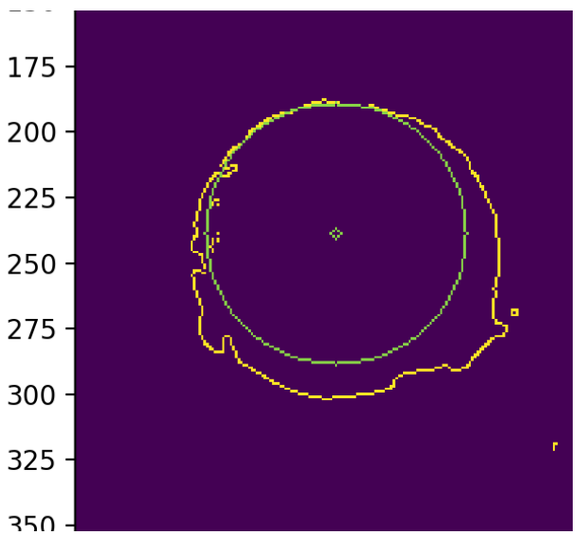
異常な穴②エッジを大きく見積もりすぎ



異常な穴①エッジを大きく見積もりすぎ

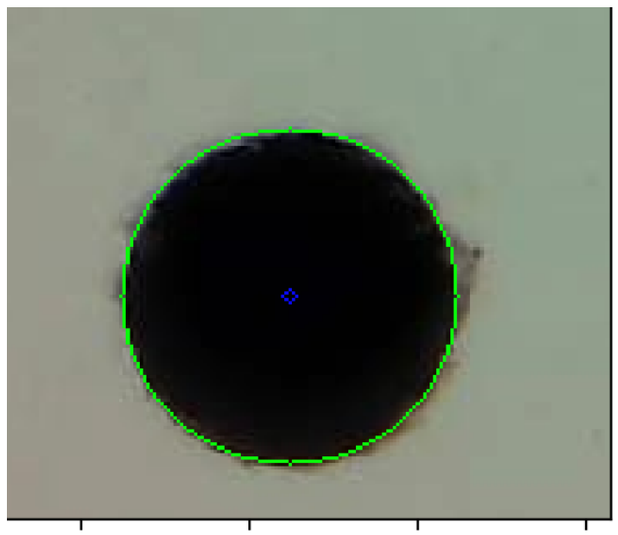


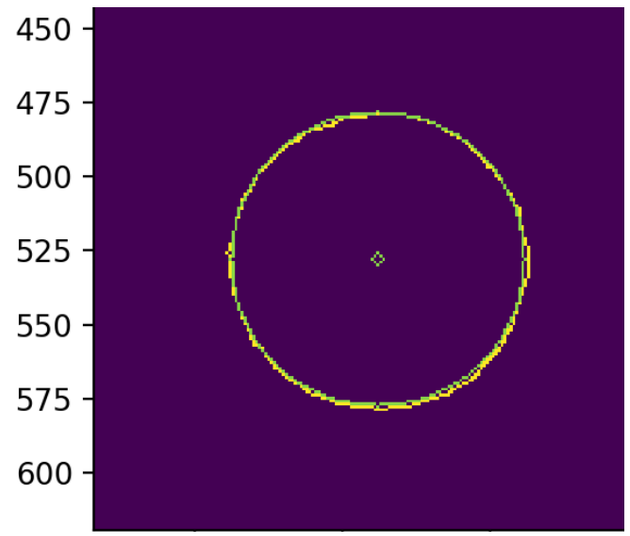
異常な穴②別角度



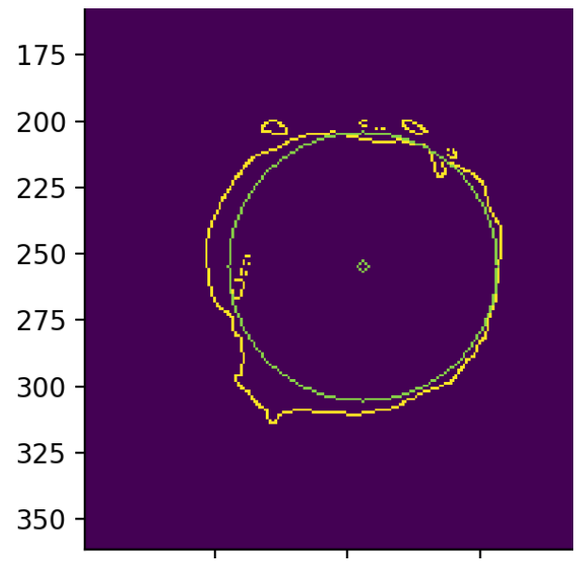
threshold の値ごとの穴の検出具合

- threshold 80（適正①）

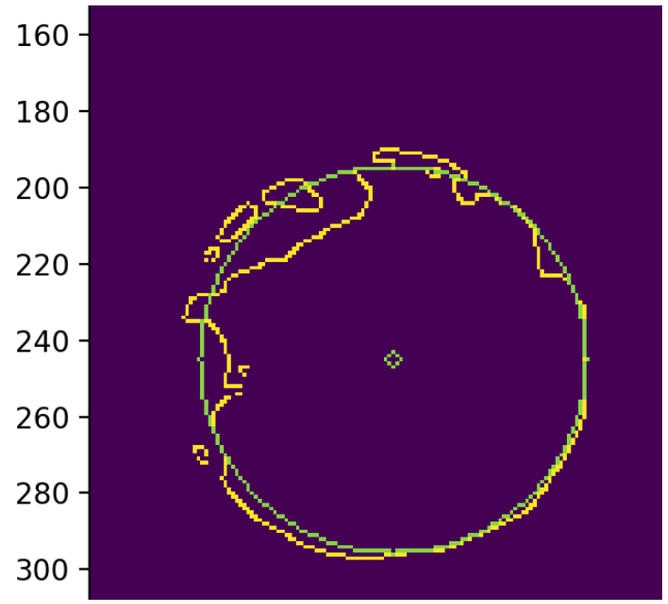




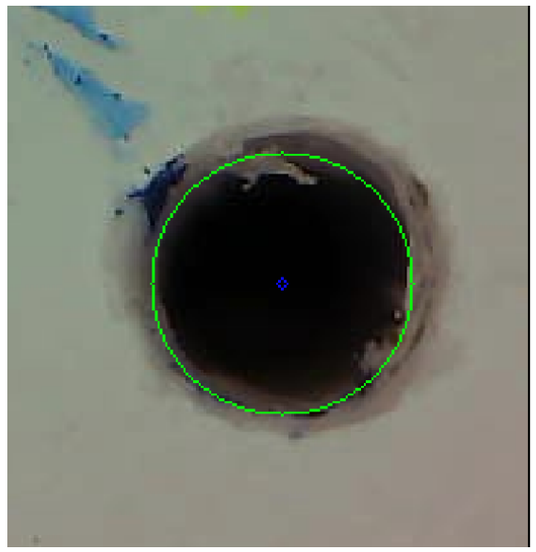
正常な穴、正しく検出

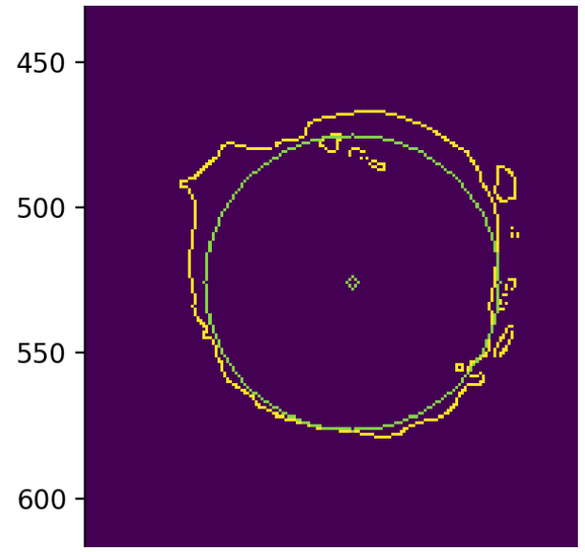


異常な穴②微妙



異常な穴①やや正しい円弧を検出

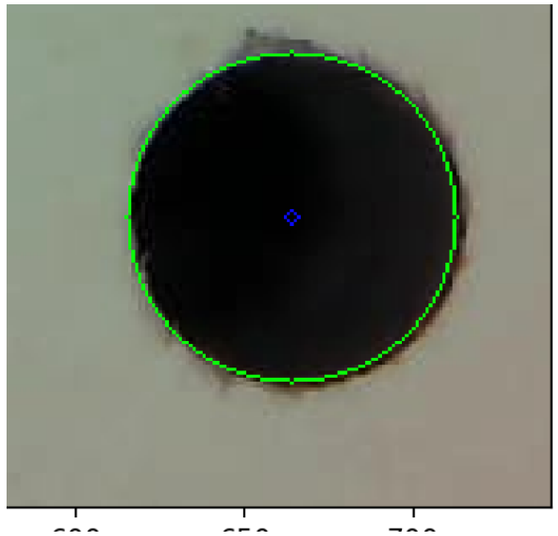


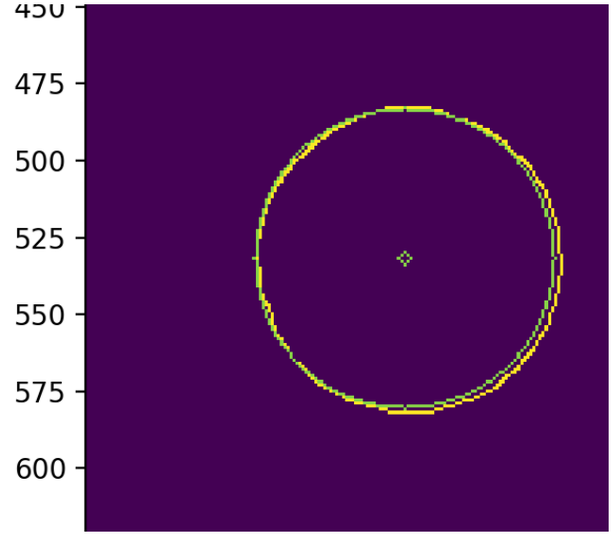


異常な穴②別角度

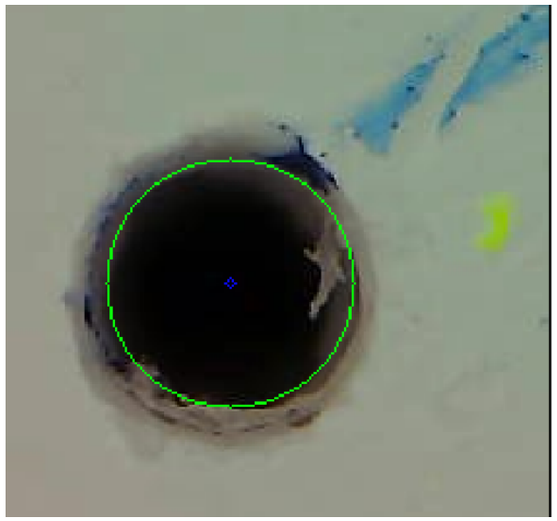
threshold の値ごとの穴の検出具合

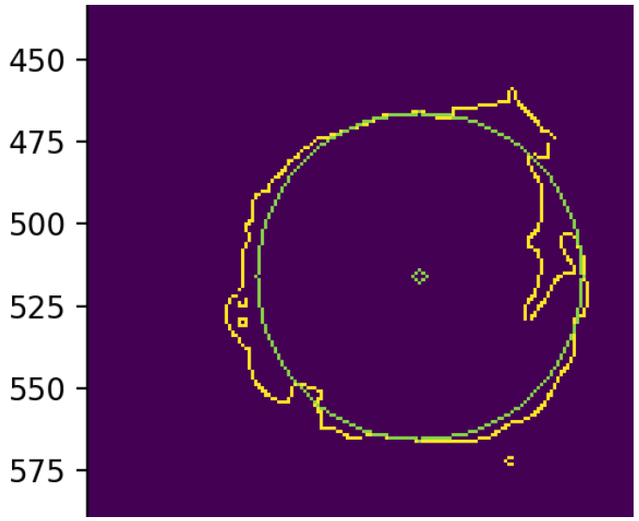
- threshold 60（適正②）



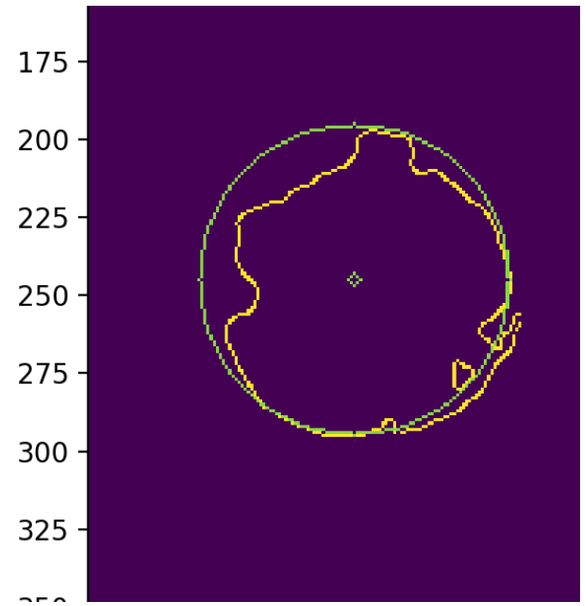


正常な穴、正しく検出

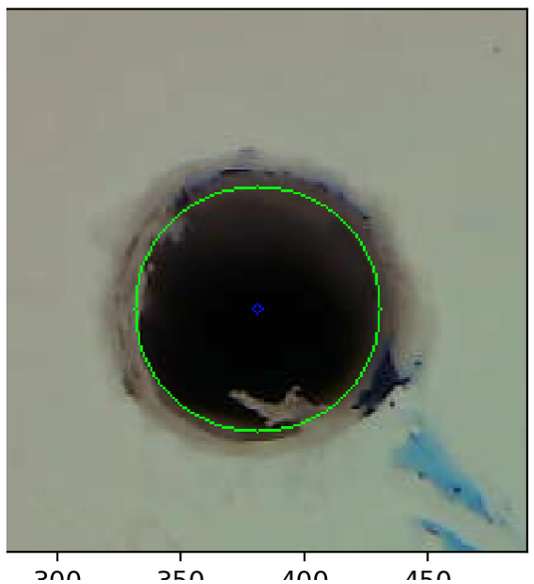


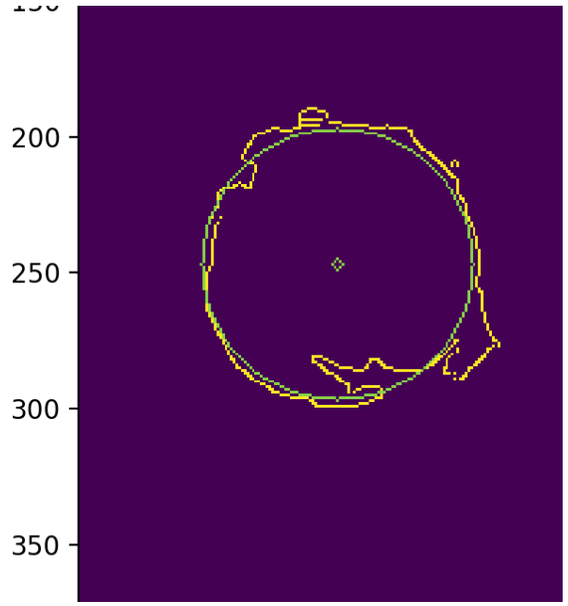


異常な穴②エッジは円ではないが正しい位置を検出



異常な穴①エッジがほぼ円でないが、
かろうじて検出

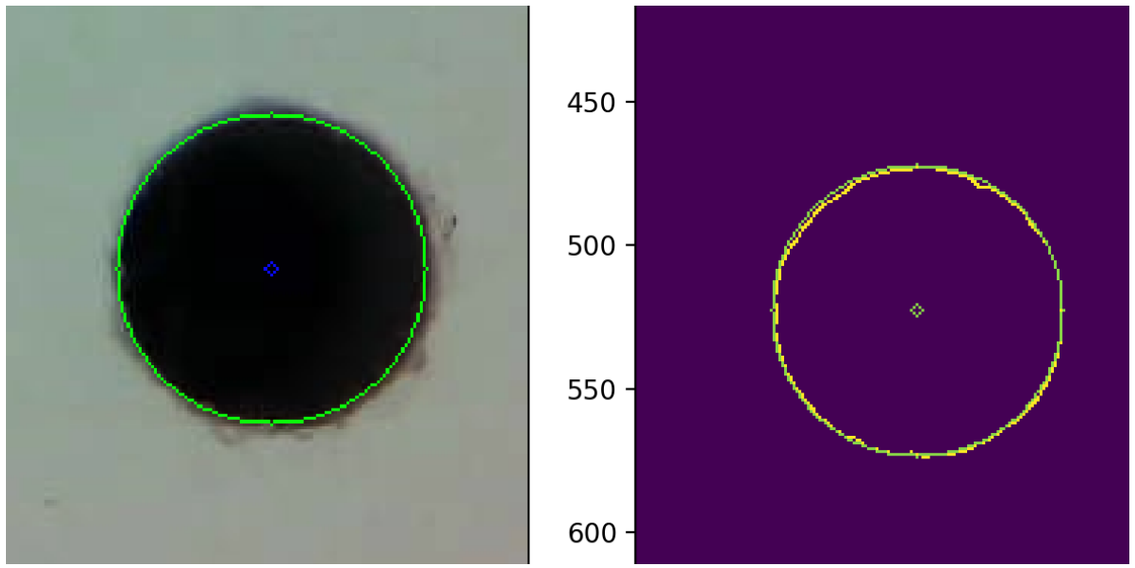




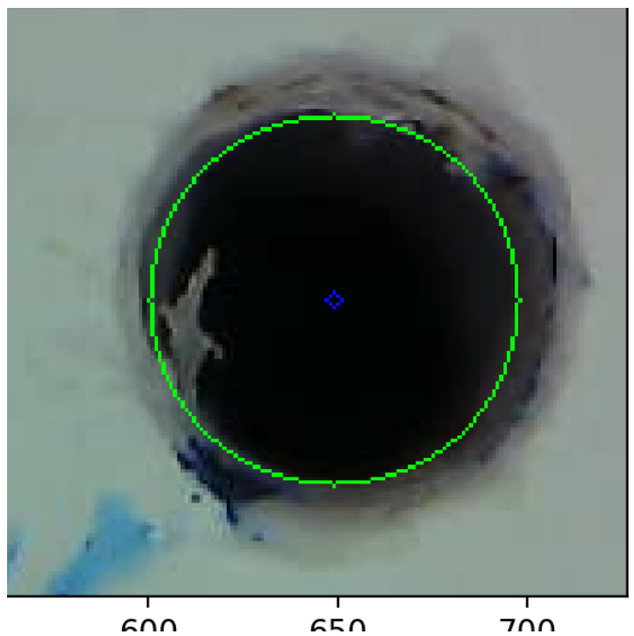
異常な穴②別角度

threshold の値ごとの穴の検出具合

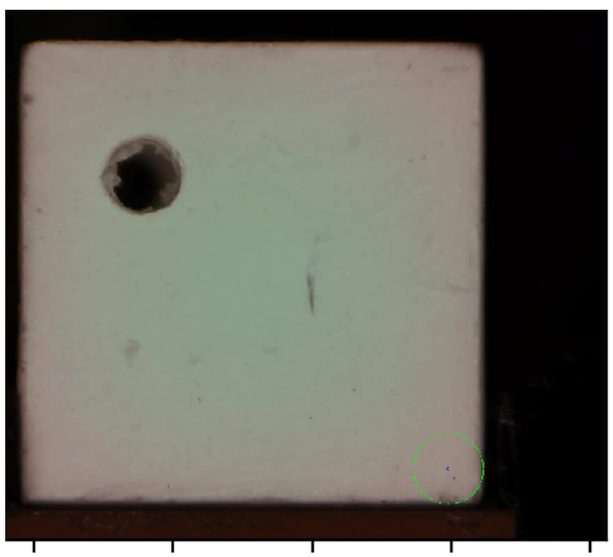
- threshold 40（低すぎ）

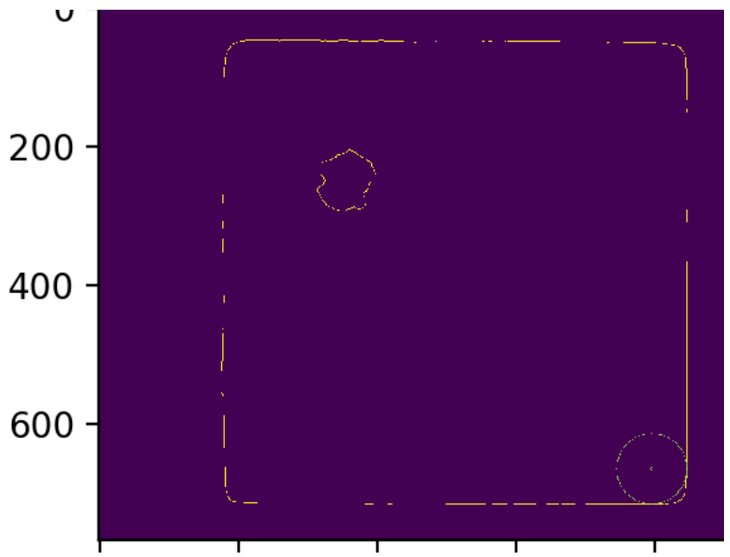


正常な穴、正しく検出

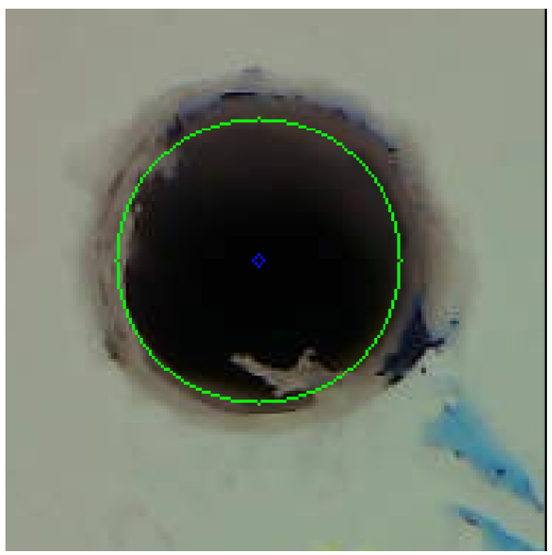


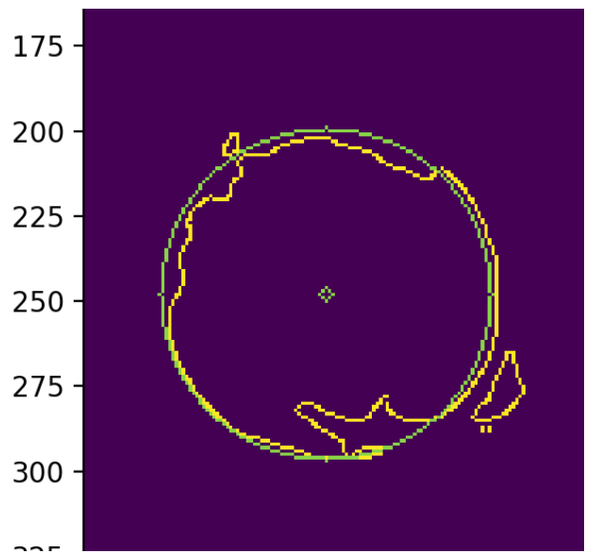
異常な穴②エッジは円に近い





異常な穴①検出失敗





異常な穴②別角度

穴の中心を探す関数

```
#穴の中心を検出する関数
#########################################################################
def detect_HoleCenter(img, img2):
    print("\n#call detect_Hole")

    th = 50
    img_forDraw = img2
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    height, width, ch = img.shape
    holeimg = np.zeros(gray.shape[:2], np.uint8)
    print ("height, width : {}, {}".format(height, width))
    xblack = np.zeros(width)
    x_mean = np.zeros(height)
    xscan = range(width)
    yscan = range(height)

    #検出した円の近傍をトリミングした画像に対して、
    #threshold(今回は50)以下の黒いピクセルのみ抜き出す。

    #まず、横方向のスキャンをして画像中の黒ピクセルの左端と右端を得る。
    #左端と右端の中心を黒ピクセルの中心位置とする（黄緑(100,200,10)でdrawMarker

    #縦方向も同様にスキャンをして上端と下端の中心を黒ピクセルの中心位置とする。

    #縦方向のスキャンと横方向のスキャンが重なったピクセルを、
    #その穴の中心とし（赤(250,10,10)で印）、出力する。

    #鈍い緑色(100,200,10)は円検出で得た円の中心（＝トリミングした画像の中心）を表す。
```

①各yiごとに、横方向のスキャン（明るい緑）

```
    for yi in range(height):
        for xj in range(width):
            if gray[yi,xj] > th :
                xblack[xj] = 0
           #     holeimg[yi,xj] = 255
            else :
                xblack[xj] = xj
                holeimg[yi,xj] = gray[yi,xj]

        xb_ex = [i for i in xblack if i > 0]
        if len(xb_ex) > 40 and len(xb_ex) < 120:
            xb_min = min(xb_ex)
            xb_max = max(xb_ex)

            x_mean[yi] = (xb_max + xb_min)*0.5
            cv2.drawMarker(img_forDraw, (int(x_mean[yi]), yi), (100,200,10),
                markerType=cv2.MARKER_STAR, markerSize=1, thickness=1, line_type=cv2.LINE_8)

            #print("x_mean[yi] is {}".format(x_mean[yi]))

    yblack = np.zeros(height)
    y_mean = np.zeros(width)
```

②各xiごとに、縦方向のスキャン（明るい緑）

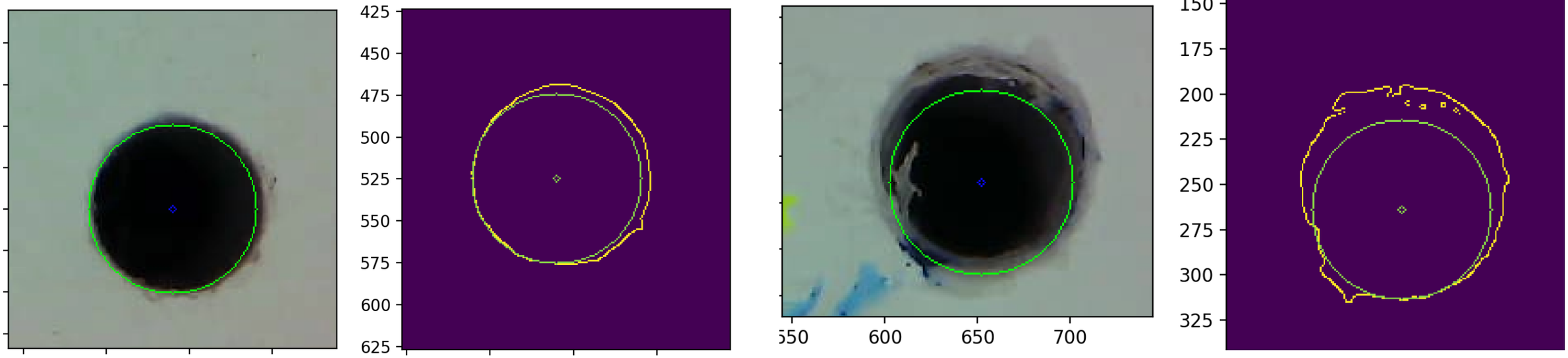
```
    for xi in range(width):
        for yj in range(height):
            if gray[yj,xi] > th :
                yblack[yj] = 0
          #      holeimg[yj,xi] = 255
            else :
                yblack[yj] = yj
                holeimg[yj,xi] = gray[yj,xi]

        yb_ex = [i for i in yblack if i > 0]
        if len(yb_ex) > 40 and len(yb_ex) < 120:
            yb_min = min(yb_ex)
            yb_max = max(yb_ex)

            y_mean[xi] = (yb_max + yb_min)*0.5
            cv2.drawMarker(img_forDraw, (xi, int(y_mean[xi])), (100,200,10),
                markerType=cv2.MARKER_STAR, markerSize=1, thickness=1, line_type=cv2.LINE_8)
```

③穴の中心（赤印）を出力、元の検出した円の中心を表示（暗い緑）

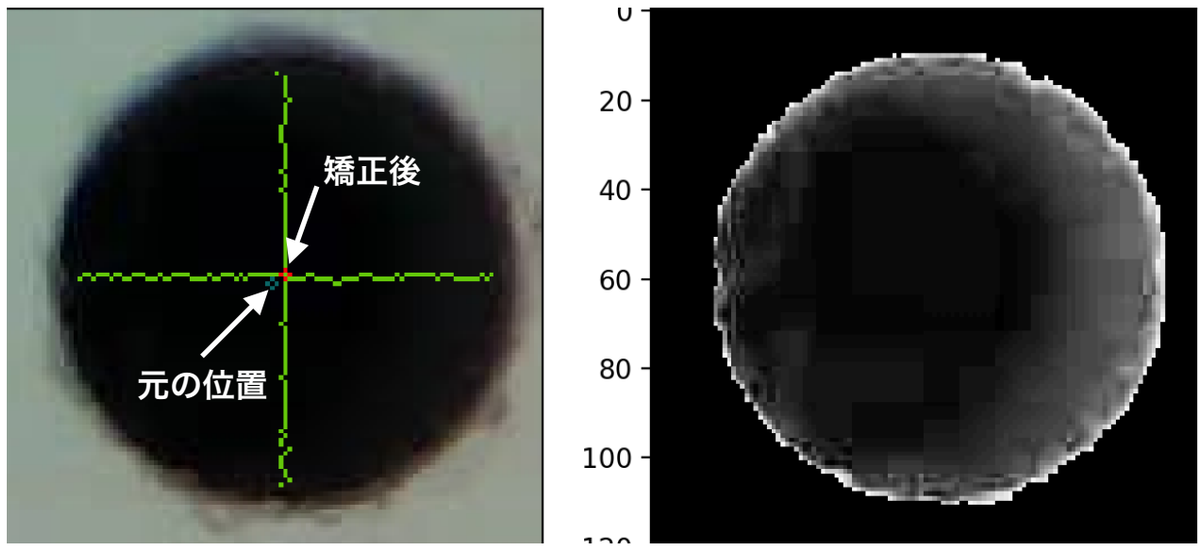
```
        for xiout in range(width):
            yiout = int(y_mean[xiout])
#            print (yiout)
            if int(x_mean[yiout])  == xiout :
                cv2.circle(img_forDraw, (xiout, yiout), 1, (250,10,10), 1)
                xholecenter = xiout
                yholecenter = yiout
    cv2.circle(img_forDraw, (int(height/2), int(width/2)), 1, (10,100, 100), 1)
    fig = plt.figure(figsize=(8,6) ,dpi=100)
    fig.suptitle('extract hole', fontsize=24)
    ax = fig.add_subplot(1,2,1), plt.imshow(img_forDraw, 'gray')
    plt.subplot(122)
    plt.imshow(holeimg, 'gray')
    print("xholecenter, yholecenter is {}, {}".format(xholecenter, yholecenter))

    return [xholecenter, yholecenter]
#########################################################################
```

検出円の中心（暗い緑）と穴（黒ピクセル）の中心（赤）

- threshold 100

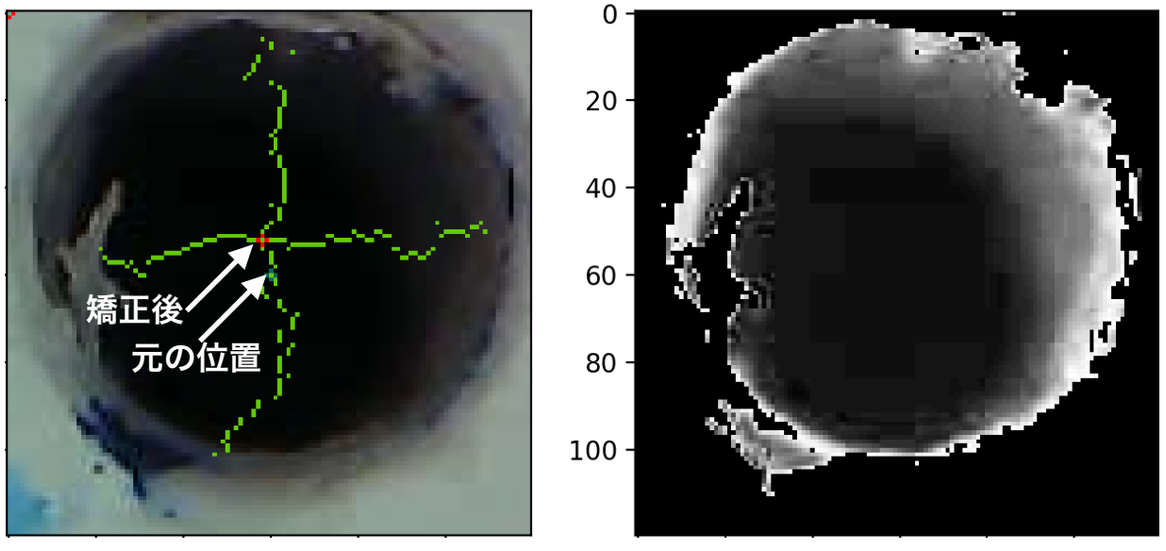


正常な穴・円検出



成功：中心を右上方に矯正

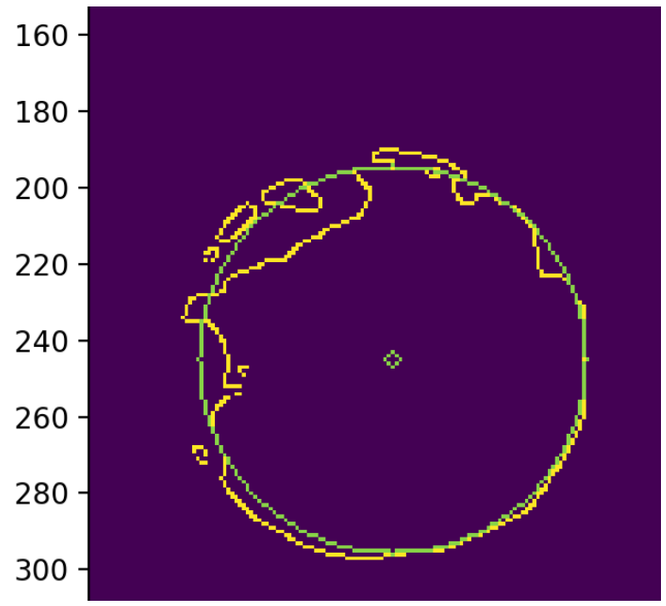
異常な穴②・円検出

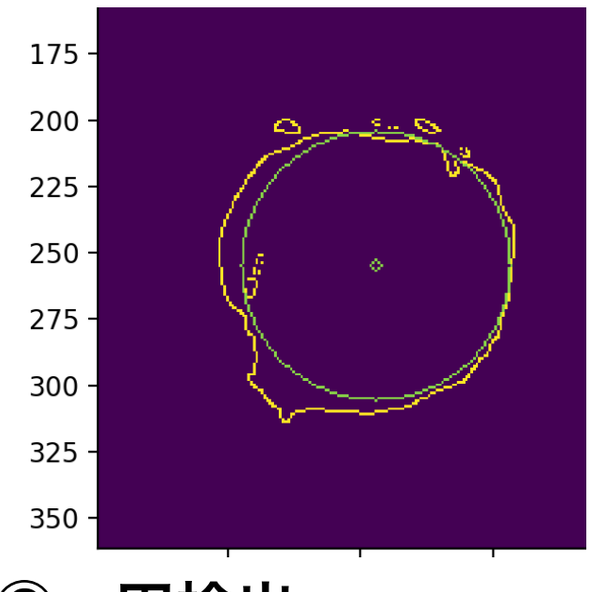


成功：中心を上方に矯正

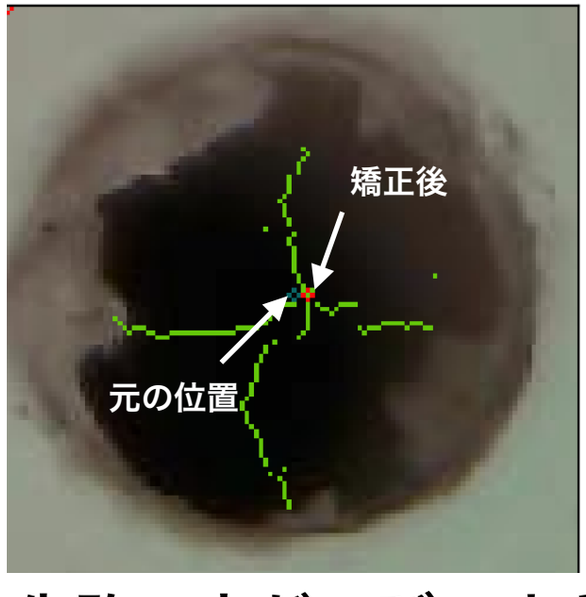
検出円の中心（暗い緑）と穴（黒ピクセル）の中心（赤）

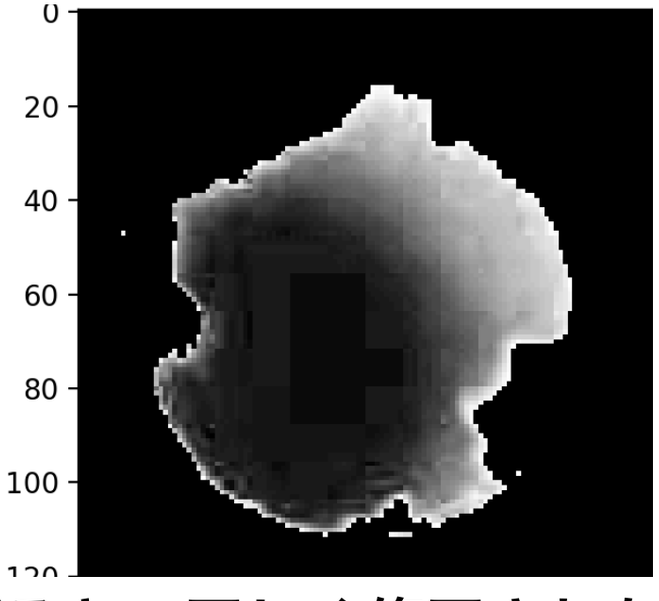
- threshold 80





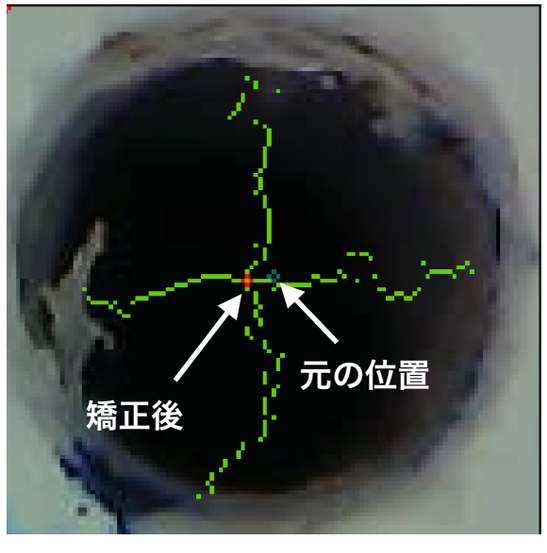
異常な穴①・円検出

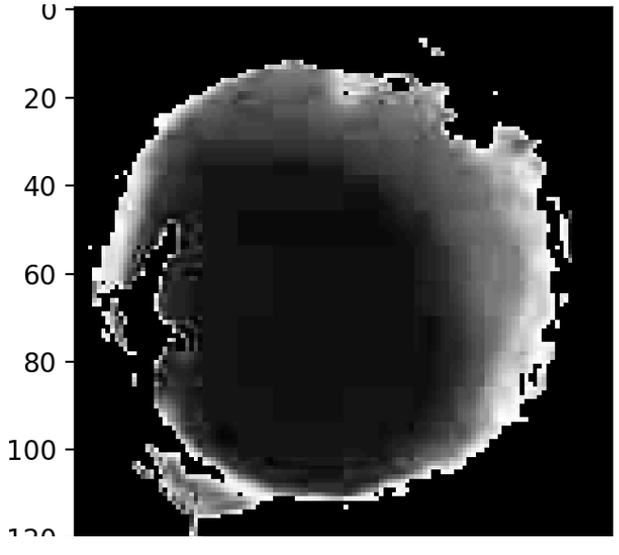




失敗：穴がいびつすぎると、正しく修正されない。

異常な穴②・円検出





成功：中心を左方に矯正

黒ピクセル抽出のthreshold(今回は50で固定)も最適化する必要あり。