

# Dynamic Watering Point Localization for Soil Channeling Prevention Using Computer Vision

## - Testing and Evaluation -

*Teng Tian*

*last edited: 01.02.2024*

```
In [ ]: %matplotlib inline
import library
from plantcv import plantcv as pcv
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
import time library to measure the execution time
import time
import math
import random
```

```
In [ ]: # turn off automaticly show image after every process
pcv.params.debug="none"
```

## Testing and Evaluation

### Testing with test images

I have prepared 10 images taken by the FarmBot camera for testing. Each image contains only one plant and one pot. Some images were captured on the edge of the farming platform. In the image, metal parts of the platform are visible. Some items are photographed under warm lighting, which affects their colors.

```
In [ ]: #img,path,filename=pcv.readimage(filename="../data/CAM_greenhouse/fotos_ideal_co
test_image_names = ["image 1", "image 2", "image 3", "image 4", "image 5", "imag
test_images = []
labeled_test_images = []
test_image_file_names = ["Farmbot_GH_ideal_2.jpg",
                        "Farmbot_GH_ideal_3.jpg",
                        "Farmbot_GH_ideal_4.jpg",
                        "Farmbot_GH_ideal_5.jpg",
                        "Farmbot_GH_ideal_6.jpg",
                        "Farmbot_GH_ideal_7.jpg",
                        "Farmbot_GH_ideal_8.jpg",
                        "Farmbot_GH_ideal_9.jpg",
                        "Farmbot_GH_ideal_10.jpg",
                        "Farmbot_GH_ideal_11.jpg"]

reset_debug = False
```

```

if pcv.params.debug=="plot":
    pcv.params.debug="None"
    reset_debug = True

full_path = "../data/CAM_greenhouse/fotos_ideal_condition/"
for i in range(0,10):
    img_read,__,__ = pcv.readimage(
        filename=full_path+test_image_file_names[i],
        mode="rgb"
    )
    test_images.append(img_read)
del img_read

# Plot Labels of each image
pcv.params.text_size = 5
pcv.params.text_thickness = 10

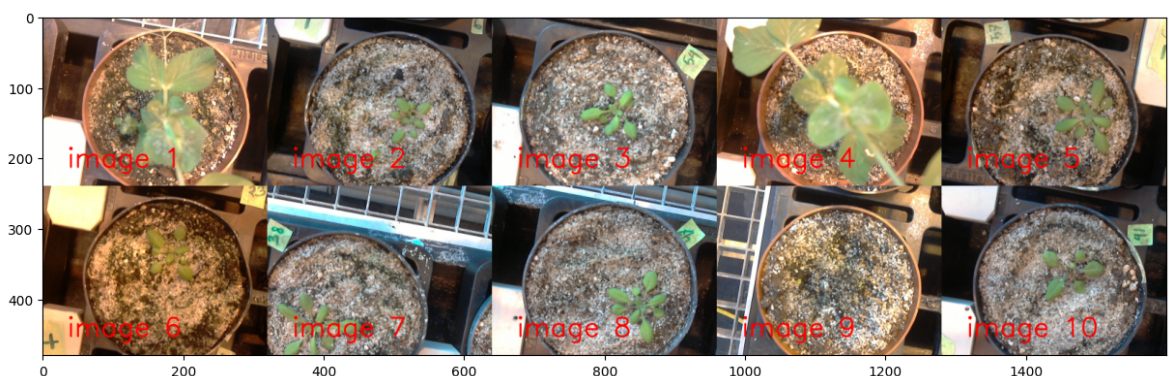
for i, test_img in enumerate(test_images):
    y = (int)(np.shape(test_img)[0]*8/9)
    x = (int)(np.shape(test_img)[1]*1/9)
    labeled=test_img.copy()
    labeled = cv.putText(img=labeled, text=test_image_names[i], org=(x,y),
                        fontFace=cv.FONT_HERSHEY_SIMPLEX,
                        fontScale=pcv.params.text_size, color=(0,0,255), thickn
    labeled_test_images.append(labeled)

plotting_img = np.vstack([np.hstack([labeled_test_images[0], labeled_test_images
                                np.hstack([labeled_test_images[5], labeled_test_images
plotting_img = pcv.transform.resize_factor(plotting_img, factors=(0.25,0.25))

#pcv.plot_image(plotting_img) this function will print the image on jupyter note
plt.figure(figsize=(15,10))
plt.imshow(cv.cvtColor(plotting_img, cv.COLOR_BGR2RGB))
plt.show()

if reset_debug==True:
    pcv.params.debug="plot"

```



The program will be executed for all test pictures.

```

In [ ]: #run all the test images at ones
pcv.params.debug = 'None'
test_images_out=[]

for i, test_img in enumerate(test_images):
    start_time = time.time()
    print('##### START #####')

```

```

print('processing image', test_image_names[i], ':')

# mask in H channel
img_H=pcv.rgb2gray_hsv(rgb_img=test_img, channel="H")
img_H_thresh, __ = pcv.threshold.custom_range(img=img_H, lower_thresh=[20],
# mask in A channel
img_A=pcv.rgb2gray_lab(rgb_img=test_img, channel="A")
img_A_hist_EQU=pcv.hist_equalization(img_A)
img_A_thresh, __ = pcv.threshold.custom_range(img=img_A_hist_EQU, lower_thre
# mask in V channel
img_V=pcv.rgb2gray_hsv(rgb_img=test_img, channel="V")
img_V_thresh_up = pcv.threshold.binary(gray_img=img_V, threshold = 250, obje
img_V_thresh_down = pcv.threshold.binary(gray_img=img_V, threshold = 50, obj
img_V_thresh = cv.bitwise_and(img_V_thresh_up,img_V_thresh_down)
# combine H A V masks
img_H_thresh_erode = pcv.erode(gray_img = img_H_thresh, ksize = 5, i = 1)
img_A_thresh_erode = pcv.erode(gray_img = img_A_thresh, ksize = 5, i = 1)
img_thresh = cv.bitwise_or(img_A_thresh_erode,img_H_thresh_erode)
img_thresh = cv.bitwise_and(img_thresh,img_V_thresh)
# closing method to the mask
mask_dilated = pcv.dilate(gray_img = img_thresh, ksize = 5, i = 2)
mask_erode = pcv.erode(gray_img = mask_dilated, ksize = 5, i =3)
mask_dilated = pcv.dilate(gray_img = mask_erode, ksize = 5, i = 3)
mask = mask_dilated

# Labeled the regions on the mask image
labeled_mask, num_mask = pcv.create_labels(mask=mask)
#pcv.plot_image(labeled_mask)
print('{:}'.format('\t'),'total', num_mask, 'region(s) found!')

# just keep the first 10 biggst region on the mask
count = 0
region_info={}
for region_id in range(1,num_mask+1,1):
    mask_region_cnt = cv.inRange(labeled_mask,region_id,region_id)
    count = cv.countNonZero(mask_region_cnt)
    region_info[region_id]= (region_id, count)
list_of_region = list(region_info.values())
sorted_data = sorted(list_of_region, key=lambda x: x[1], reverse=True)
sorted_data_cop = sorted_data[:10]

mask_cop = np.zeros(np.shape(mask),dtype=np.uint8)
for region_id in sorted_data_cop:
    id = (int)(region_id[0])
    mask_cop+=cv.inRange(labeled_mask,id,id)

pcv.plot_image(mask_cop)

# calculation the center of mass of the region
# this will locate the plant
contours, hierarchy = cv.findContours(mask_cop,cv.RETR_EXTERNAL,cv.CHAIN_APP
## cnt = contours[0]
## M = cv.moments(cnt)
## cx = int(M['m10']/M['m00'])
## cy = int(M['m01']/M['m00'])
## print('{:}'.format('\t'),'Location of the plant(x,y):',cx,cy)
mask_RGB=cv.cvtColor(mask_cop,cv.COLOR_GRAY2BGR)

```

```

##     cv.circle(mask_RGB, (cx, cy), 2, (0, 255, 0), 20)
##     pcv.plot_image(mask_RGB)
img_HoughCircles = test_img.copy()
#start_time = time.time()
circles_list = []
img_GRAY=cv.cvtColor(test_img, cv.COLOR_BGR2GRAY)
img_mB = cv.medianBlur(img_GRAY, 5)
circles= cv.HoughCircles(image=img_mB, method=cv.HOUGH_GRADIENT, dp=4, minDist=
circles = np.uint16(np.around(circles))
circles_list.append(circles)

# we take only the circle, whose center is closest to center of the plant
cnt = 1
index = 0
distance_prev = 0
cy = (int)(np.shape(test_img)[0]/2)
cx = (int)(np.shape(test_img)[1]/2)
for i in circles[0,:]:
    # draw the outer circle
##     cv.circle(mask_RGB, (i[0], i[1]), i[2], (255, 0, 255), 4)
    # draw the center of the circle
##     cv.circle(mask_RGB, (i[0], i[1]), 2, (0, 0, 255), 30)
    # draw a line from the center of the circle to center of mass
##     cv.line(mask_RGB, (i[0], i[1]), (cx, cy), (255, 0, 0), 4)
    distance = (int)(math.sqrt((cx-i[0])**2+(cy-i[1])**2))
    if distance<distance_prev:
        index = cnt - 1
##     print('Distance to ', cnt, '. circle =', distance)
        cnt=cnt+1
##     print('The', index+1, 'circle is the wanted circle')
    pot_x = circles[0][index][0]
    pot_y = circles[0][index][1]
    pot_radius = circles[0][index][2]
    print('{}'.format('\t'), 'Position of the flowerpot is', 'x=', pot_x, 'y=', pot_y)
##     pcv.plot_image(mask_RGB)

    # Let us define the watering point.
# It cannot be at the edge area, and either on top of the plant
edge_area = 150 # 100 Pixel to the edge will not be watered
num_watering_points = 4*7 # 4 time a day and 7 days a week
# create watering point
#angel = 30 # must be 15, 30, 45, 60, 90
# the previous mask will be enlarged, so that there will be a safty zone, th
mask_with_saftyzone = pcv.dilate(gray_img = mask_cop, ksize = 15, i = 3)
watering_points_list = []
count = 0
while (count<=num_watering_points):
    angel = random.randint(0, 360)
    rel_radius = random.random()
    x_watering_point = (int)(math.cos(angel/180*math.pi)*(pot_radius - edge_
    y_watering_point = (int)(math.sin(angel/180*math.pi)*(pot_radius - edge_
    if mask_with_saftyzone[y_watering_point, x_watering_point] != 255:
        watering_points_list.append((x_watering_point, y_watering_point))
        count+=1
    #watering_point_List.append((x_watering_point, y_watering_point))

# Lets draw everything on image

```

```

img_out = test_img.copy()
cv.circle(img_out,(pot_x,pot_y),pot_radius,(255,0,255),10)
for i in watering_points_list:
    cv.circle(img_out,i,2,(0,255,0),20)
    print('{}\t'.format('t'),'possible watering point:',i)
cv.drawContours(img_out, contours, contourIdx=-1, color=(255,0,0), thickness
#pcv.plot_image(img_out)
test_images_out.append(img_out)

end_time = time.time()
print('{}\t'.format('t'),'Execution time:', round(end_time - start_time, 2),
print('##### END #####')
print('{}\n'.format('n'))
#endfor

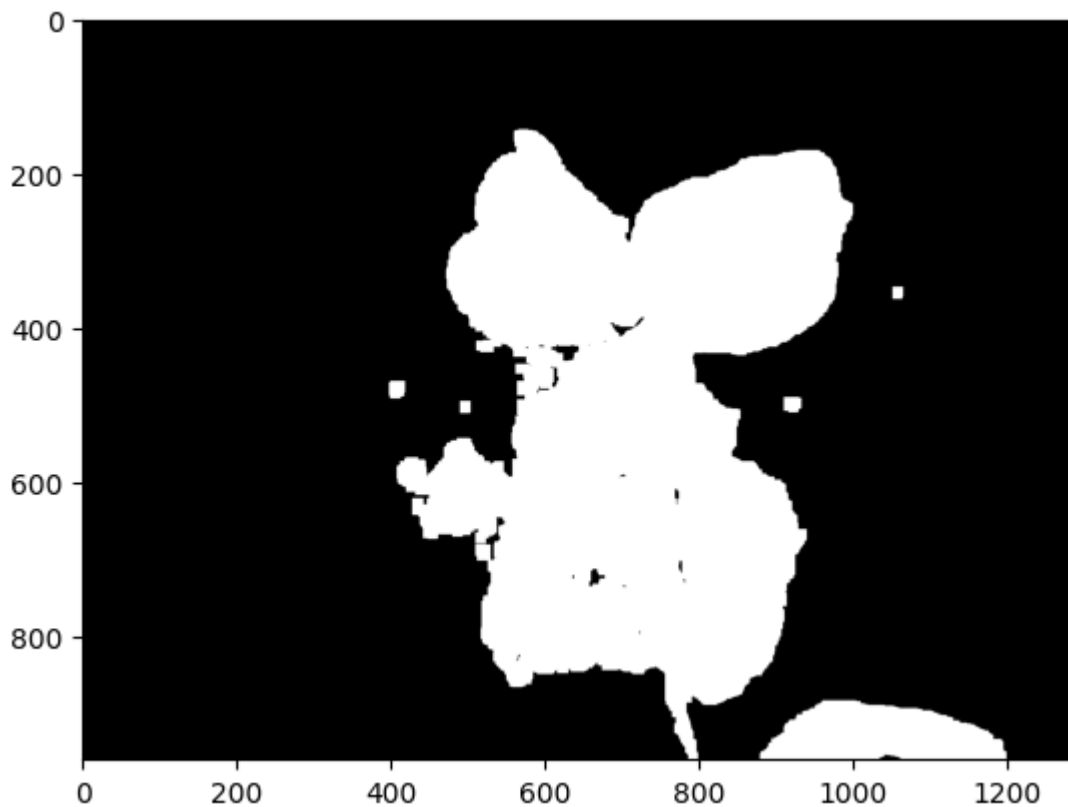
print('##### ALL DONE #####')

```

```

##### START #####
processing image image 1 :
    total 6 region(s) found!

```

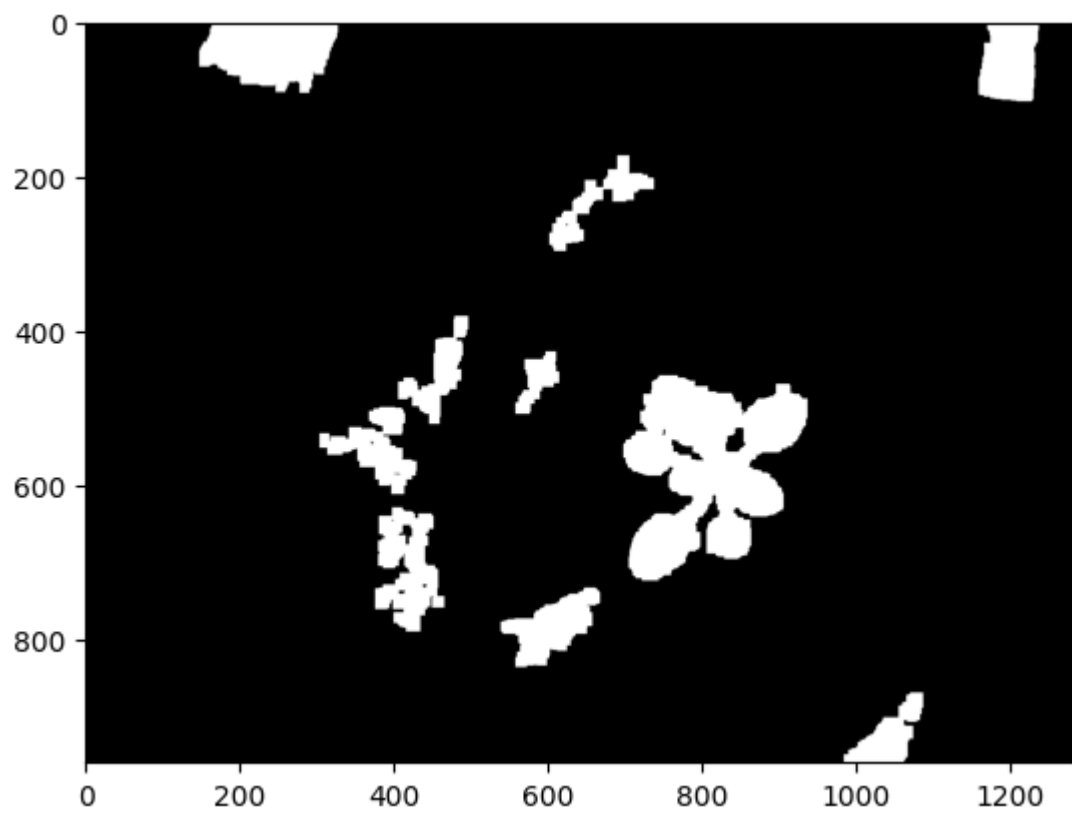


Position of the flowerpot is x= 690 y= 550 radius= 467  
possible watering point: (976, 559)  
possible watering point: (464, 739)  
possible watering point: (949, 540)  
possible watering point: (961, 521)  
possible watering point: (413, 443)  
possible watering point: (533, 533)  
possible watering point: (427, 720)  
possible watering point: (935, 588)  
possible watering point: (450, 411)  
possible watering point: (491, 473)  
possible watering point: (945, 532)  
possible watering point: (973, 505)  
possible watering point: (488, 781)  
possible watering point: (896, 462)  
possible watering point: (995, 555)  
possible watering point: (958, 462)  
possible watering point: (964, 550)  
possible watering point: (449, 738)  
possible watering point: (950, 600)  
possible watering point: (474, 757)  
possible watering point: (880, 550)  
possible watering point: (483, 729)  
possible watering point: (891, 546)  
possible watering point: (424, 431)  
possible watering point: (400, 444)  
possible watering point: (386, 608)  
possible watering point: (403, 659)  
possible watering point: (452, 503)  
possible watering point: (500, 469)  
Execution time: 0.69 seconds

##### END #####

##### START #####

processing image image 2 :  
total 59 region(s) found!



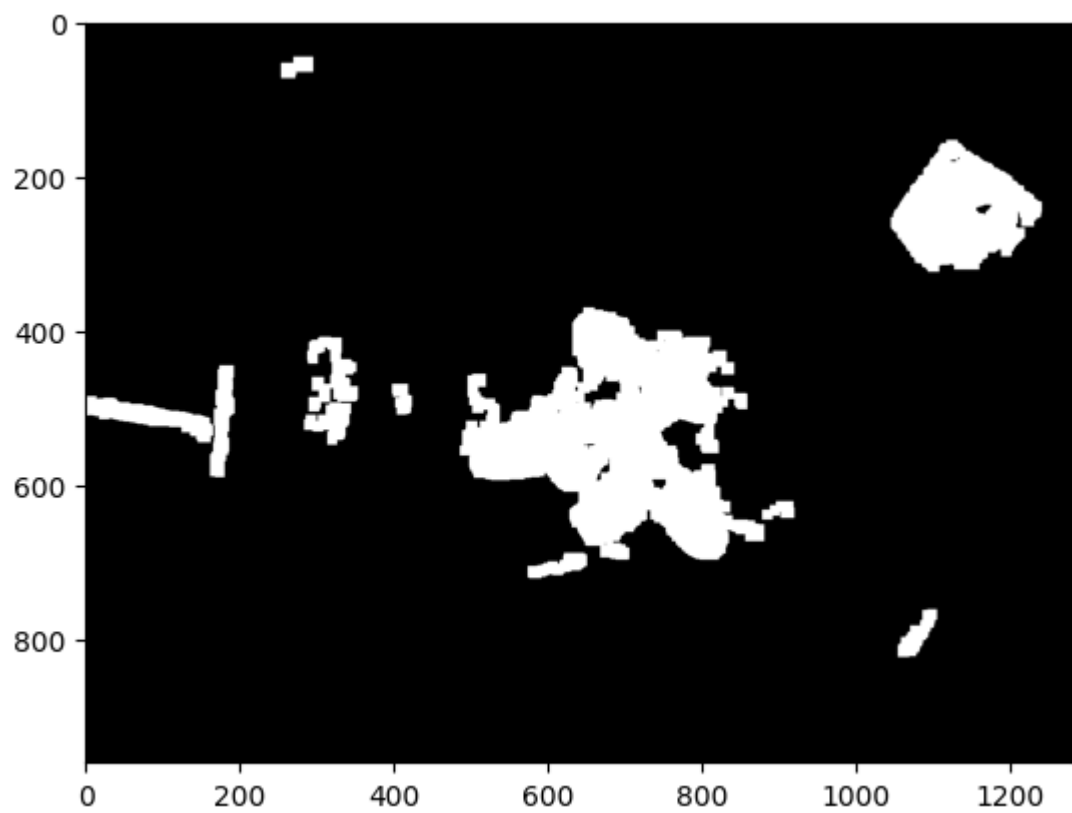
Position of the flowerpot is x= 710 y= 566 radius= 462  
possible watering point: (518, 676)  
possible watering point: (683, 814)  
possible watering point: (676, 539)  
possible watering point: (478, 586)  
possible watering point: (684, 491)  
possible watering point: (831, 355)  
possible watering point: (629, 547)  
possible watering point: (676, 330)  
possible watering point: (633, 571)  
possible watering point: (585, 694)  
possible watering point: (516, 625)  
possible watering point: (692, 465)  
possible watering point: (819, 772)  
possible watering point: (844, 322)  
possible watering point: (659, 605)  
possible watering point: (653, 410)  
possible watering point: (806, 726)  
possible watering point: (611, 552)  
possible watering point: (908, 715)  
possible watering point: (860, 788)  
possible watering point: (527, 456)  
possible watering point: (641, 326)  
possible watering point: (682, 421)  
possible watering point: (746, 829)  
possible watering point: (553, 616)  
possible watering point: (656, 611)  
possible watering point: (624, 355)  
possible watering point: (942, 645)  
possible watering point: (887, 777)  
Execution time: 0.93 seconds

##### END #####

##### START #####

processing image image 3 :  
total 71 region(s) found!



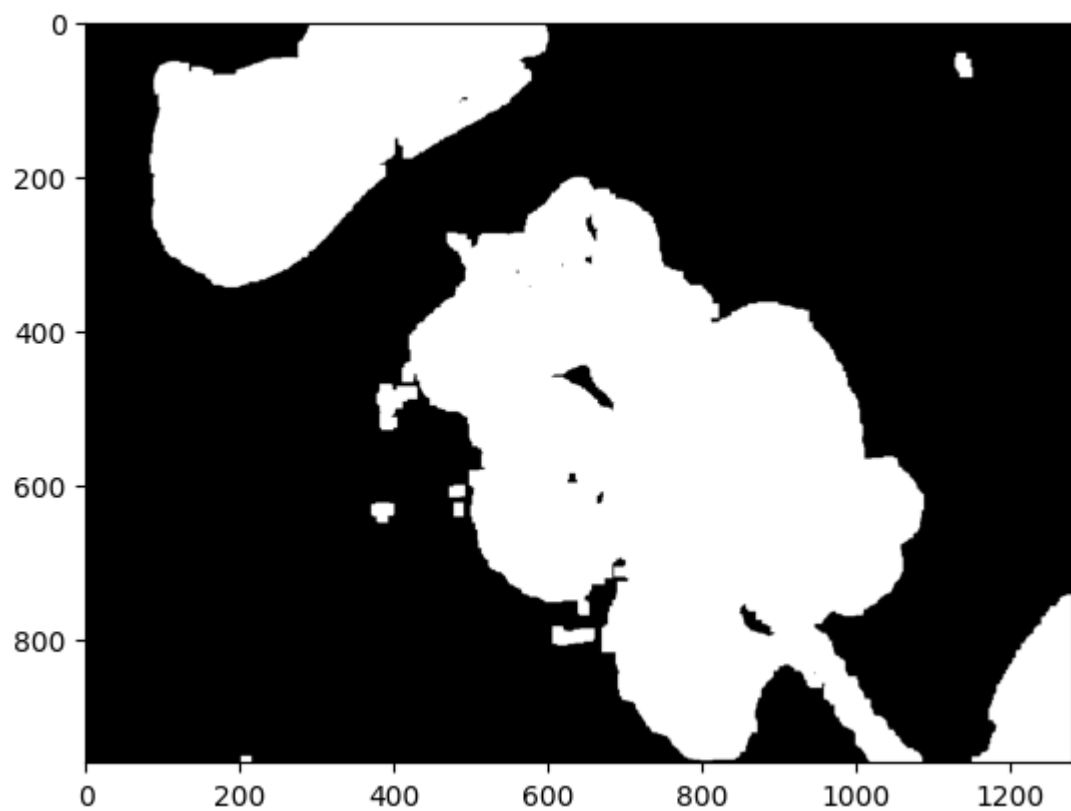


Position of the flowerpot is x= 650 y= 566 radius= 454  
possible watering point: (577, 651)  
possible watering point: (386, 570)  
possible watering point: (875, 736)  
possible watering point: (848, 534)  
possible watering point: (479, 614)  
possible watering point: (637, 320)  
possible watering point: (424, 676)  
possible watering point: (398, 601)  
possible watering point: (786, 309)  
possible watering point: (888, 469)  
possible watering point: (730, 286)  
possible watering point: (436, 631)  
possible watering point: (395, 400)  
possible watering point: (888, 483)  
possible watering point: (523, 430)  
possible watering point: (650, 303)  
possible watering point: (348, 587)  
possible watering point: (432, 395)  
possible watering point: (477, 399)  
possible watering point: (425, 666)  
possible watering point: (551, 671)  
possible watering point: (523, 803)  
possible watering point: (484, 810)  
possible watering point: (586, 660)  
possible watering point: (512, 794)  
possible watering point: (926, 677)  
possible watering point: (888, 405)  
possible watering point: (571, 623)  
possible watering point: (711, 351)  
Execution time: 0.97 seconds

##### END #####

##### START #####

processing image image 4 :  
total 10 region(s) found!

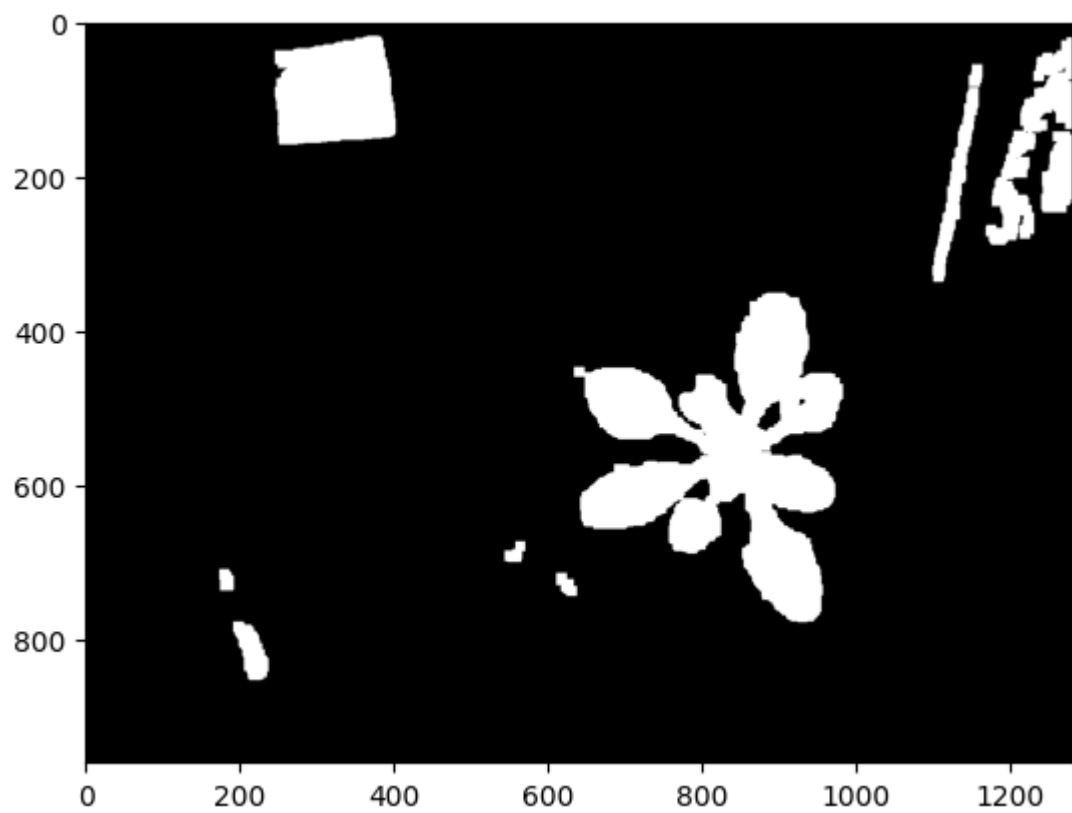


Position of the flowerpot is x= 702 y= 554 radius= 476  
possible watering point: (582, 788)  
possible watering point: (946, 341)  
possible watering point: (461, 748)  
possible watering point: (616, 832)  
possible watering point: (828, 282)  
possible watering point: (448, 540)  
possible watering point: (577, 788)  
possible watering point: (547, 782)  
possible watering point: (544, 770)  
possible watering point: (431, 558)  
possible watering point: (656, 840)  
possible watering point: (427, 648)  
possible watering point: (504, 788)  
possible watering point: (609, 837)  
possible watering point: (440, 675)  
possible watering point: (420, 691)  
possible watering point: (805, 270)  
possible watering point: (625, 840)  
possible watering point: (421, 661)  
possible watering point: (839, 341)  
possible watering point: (415, 574)  
possible watering point: (457, 724)  
possible watering point: (541, 792)  
possible watering point: (473, 782)  
possible watering point: (474, 542)  
possible watering point: (888, 297)  
possible watering point: (487, 698)  
possible watering point: (546, 767)  
possible watering point: (834, 333)  
Execution time: 0.63 seconds

##### END #####

##### START #####

processing image image 5 :  
total 27 region(s) found!

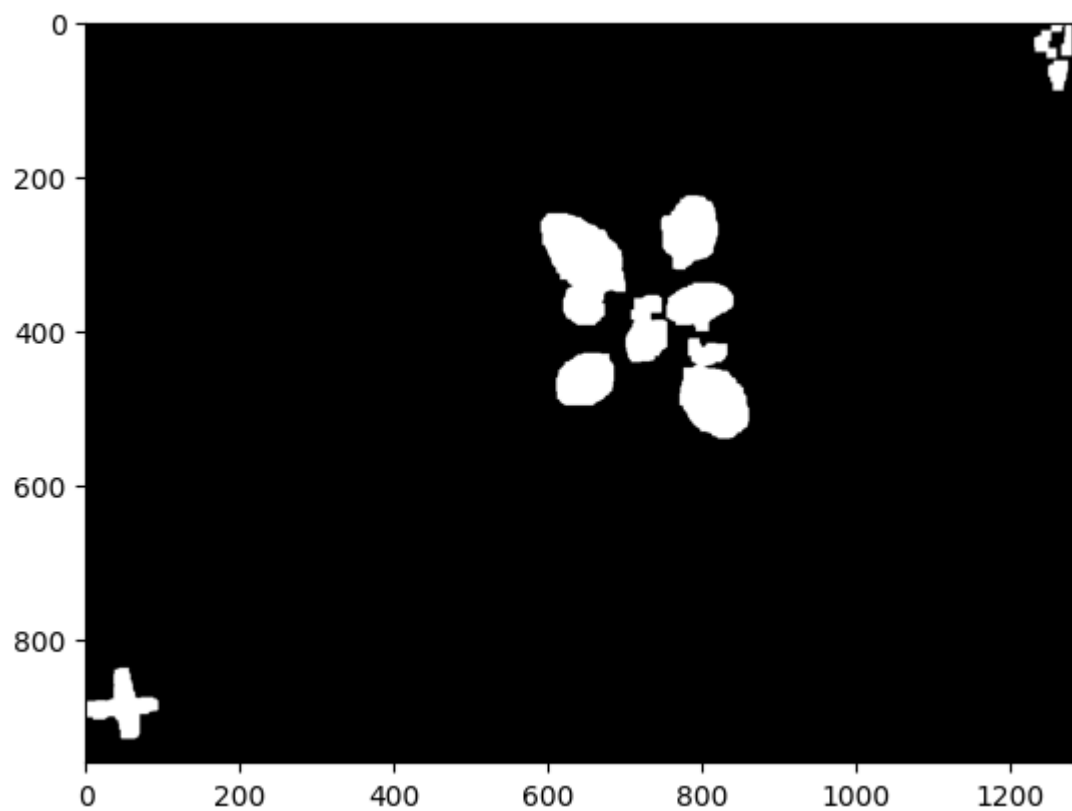


Position of the flowerpot is x= 630 y= 590 radius= 476  
possible watering point: (698, 768)  
possible watering point: (442, 374)  
possible watering point: (428, 408)  
possible watering point: (644, 795)  
possible watering point: (499, 445)  
possible watering point: (340, 620)  
possible watering point: (594, 620)  
possible watering point: (644, 389)  
possible watering point: (353, 743)  
possible watering point: (676, 427)  
possible watering point: (419, 683)  
possible watering point: (491, 648)  
possible watering point: (583, 491)  
possible watering point: (615, 423)  
possible watering point: (503, 696)  
possible watering point: (529, 899)  
possible watering point: (728, 287)  
possible watering point: (770, 711)  
possible watering point: (479, 579)  
possible watering point: (386, 538)  
possible watering point: (474, 768)  
possible watering point: (552, 553)  
possible watering point: (511, 617)  
possible watering point: (603, 595)  
possible watering point: (607, 568)  
possible watering point: (865, 809)  
possible watering point: (529, 460)  
possible watering point: (476, 378)  
possible watering point: (505, 482)  
Execution time: 0.82 seconds

##### END #####

##### START #####

processing image image 6 :  
total 17 region(s) found!



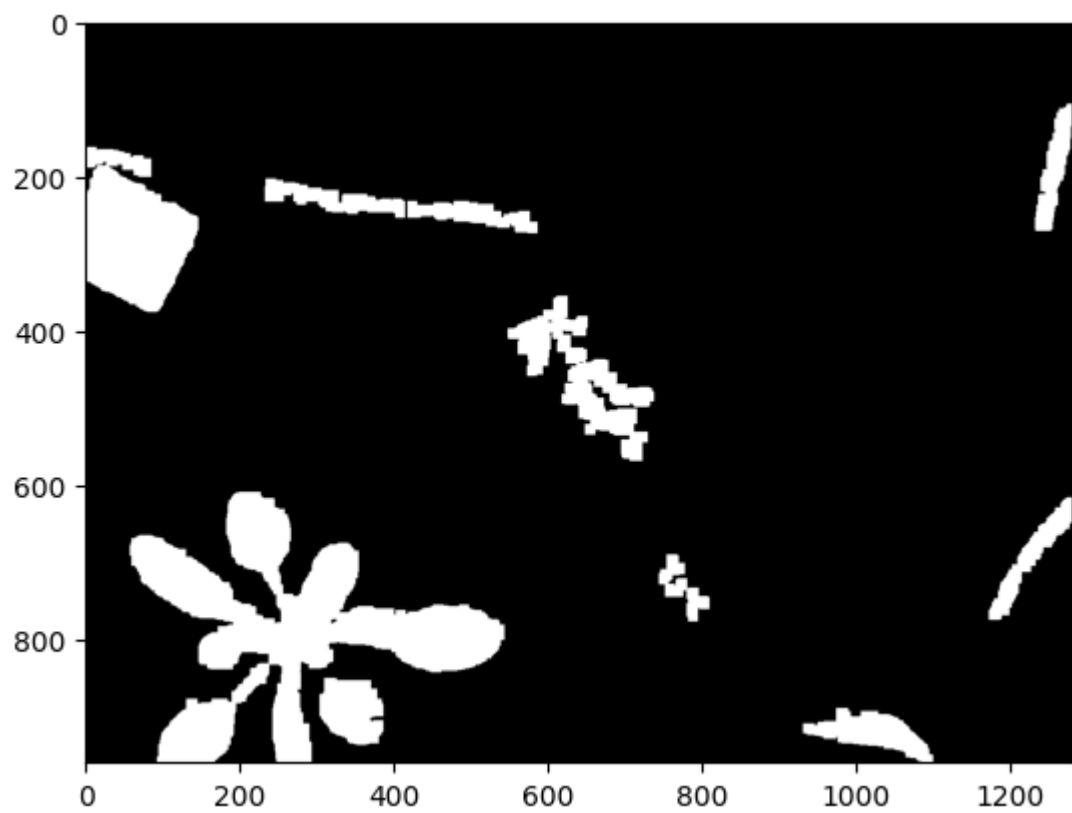
Position of the flowerpot is x= 710 y= 514 radius= 454  
possible watering point: (817, 573)  
possible watering point: (642, 549)  
possible watering point: (654, 676)  
possible watering point: (898, 459)  
possible watering point: (508, 654)  
possible watering point: (777, 558)  
possible watering point: (685, 578)  
possible watering point: (738, 534)  
possible watering point: (727, 492)  
possible watering point: (675, 563)  
possible watering point: (720, 807)  
possible watering point: (947, 560)  
possible watering point: (477, 390)  
possible watering point: (818, 782)  
possible watering point: (926, 597)  
possible watering point: (722, 491)  
possible watering point: (677, 546)  
possible watering point: (865, 727)  
possible watering point: (676, 565)  
possible watering point: (557, 748)  
possible watering point: (714, 495)  
possible watering point: (766, 758)  
possible watering point: (856, 747)  
possible watering point: (505, 481)  
possible watering point: (928, 455)  
possible watering point: (923, 622)  
possible watering point: (721, 522)  
possible watering point: (545, 552)  
possible watering point: (573, 484)  
Execution time: 0.78 seconds

##### END #####

##### START #####

processing image image 7 :  
total 85 region(s) found!



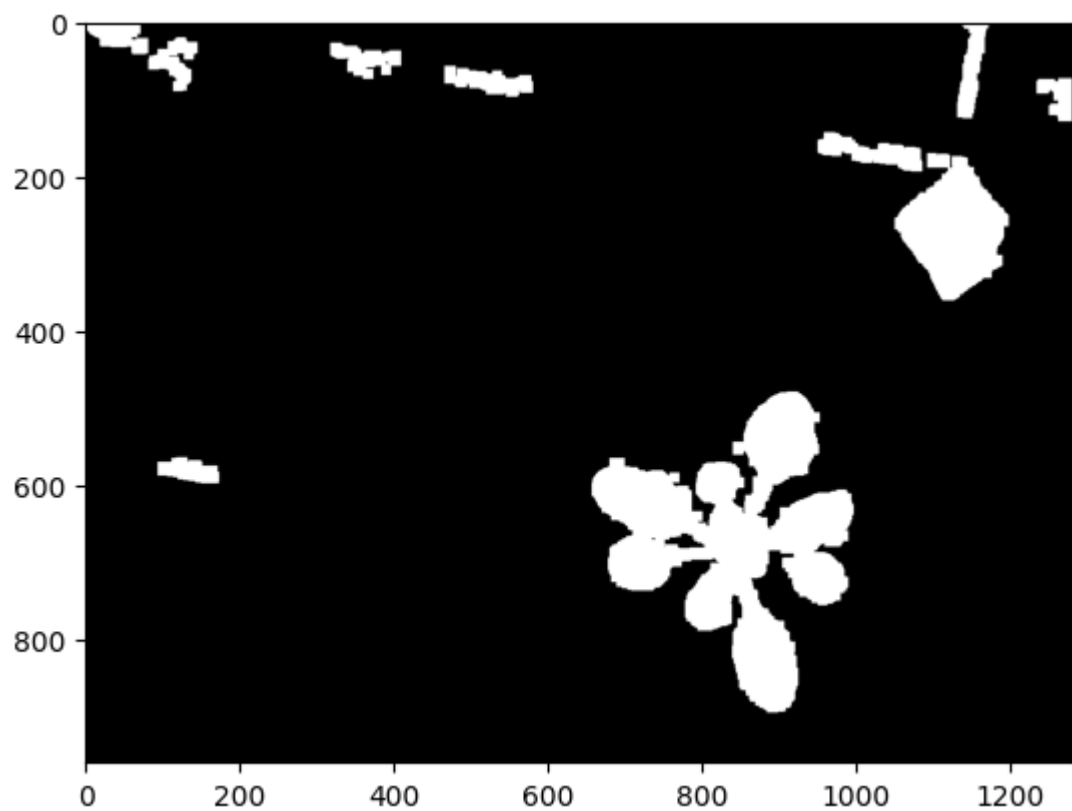


Position of the flowerpot is x= 474 y= 482 radius= 467  
possible watering point: (506, 562)  
possible watering point: (479, 474)  
possible watering point: (434, 294)  
possible watering point: (539, 644)  
possible watering point: (322, 482)  
possible watering point: (585, 563)  
possible watering point: (477, 537)  
possible watering point: (287, 295)  
possible watering point: (517, 363)  
possible watering point: (566, 519)  
possible watering point: (327, 521)  
possible watering point: (571, 631)  
possible watering point: (759, 457)  
possible watering point: (458, 568)  
possible watering point: (446, 557)  
possible watering point: (522, 449)  
possible watering point: (387, 389)  
possible watering point: (570, 780)  
possible watering point: (473, 481)  
possible watering point: (753, 401)  
possible watering point: (501, 567)  
possible watering point: (619, 723)  
possible watering point: (760, 360)  
possible watering point: (326, 563)  
possible watering point: (671, 327)  
possible watering point: (472, 500)  
possible watering point: (312, 522)  
possible watering point: (374, 541)  
possible watering point: (432, 608)  
Execution time: 1.03 seconds

##### END #####

##### START #####

processing image image 8 :  
total 60 region(s) found!

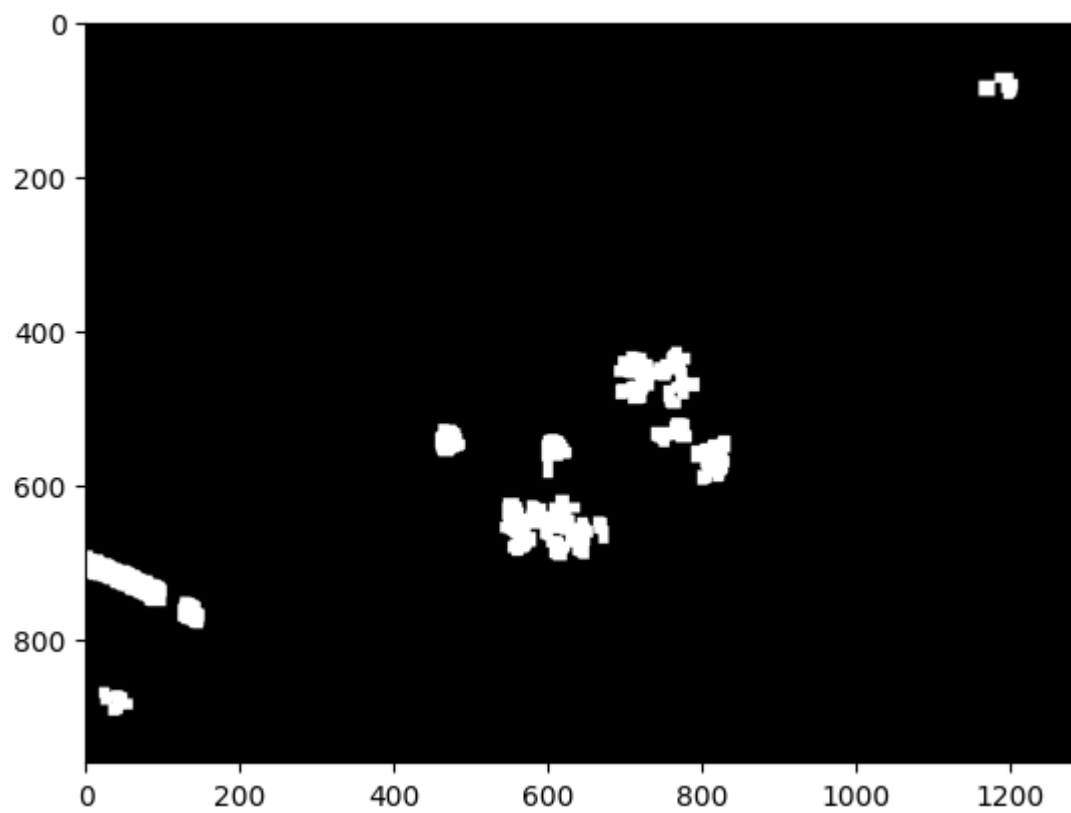


Position of the flowerpot is x= 658 y= 586 radius= 471  
possible watering point: (620, 537)  
possible watering point: (564, 298)  
possible watering point: (482, 567)  
possible watering point: (519, 469)  
possible watering point: (617, 467)  
possible watering point: (479, 521)  
possible watering point: (423, 395)  
possible watering point: (429, 669)  
possible watering point: (577, 646)  
possible watering point: (471, 482)  
possible watering point: (440, 439)  
possible watering point: (572, 607)  
possible watering point: (548, 327)  
possible watering point: (643, 754)  
possible watering point: (685, 322)  
possible watering point: (688, 414)  
possible watering point: (512, 521)  
possible watering point: (739, 760)  
possible watering point: (609, 537)  
possible watering point: (651, 409)  
possible watering point: (606, 696)  
possible watering point: (428, 520)  
possible watering point: (622, 662)  
possible watering point: (608, 633)  
possible watering point: (869, 395)  
possible watering point: (770, 382)  
possible watering point: (894, 394)  
possible watering point: (575, 813)  
possible watering point: (610, 593)  
Execution time: 0.97 seconds

##### END #####

##### START #####

processing image image 9 :  
total 51 region(s) found!

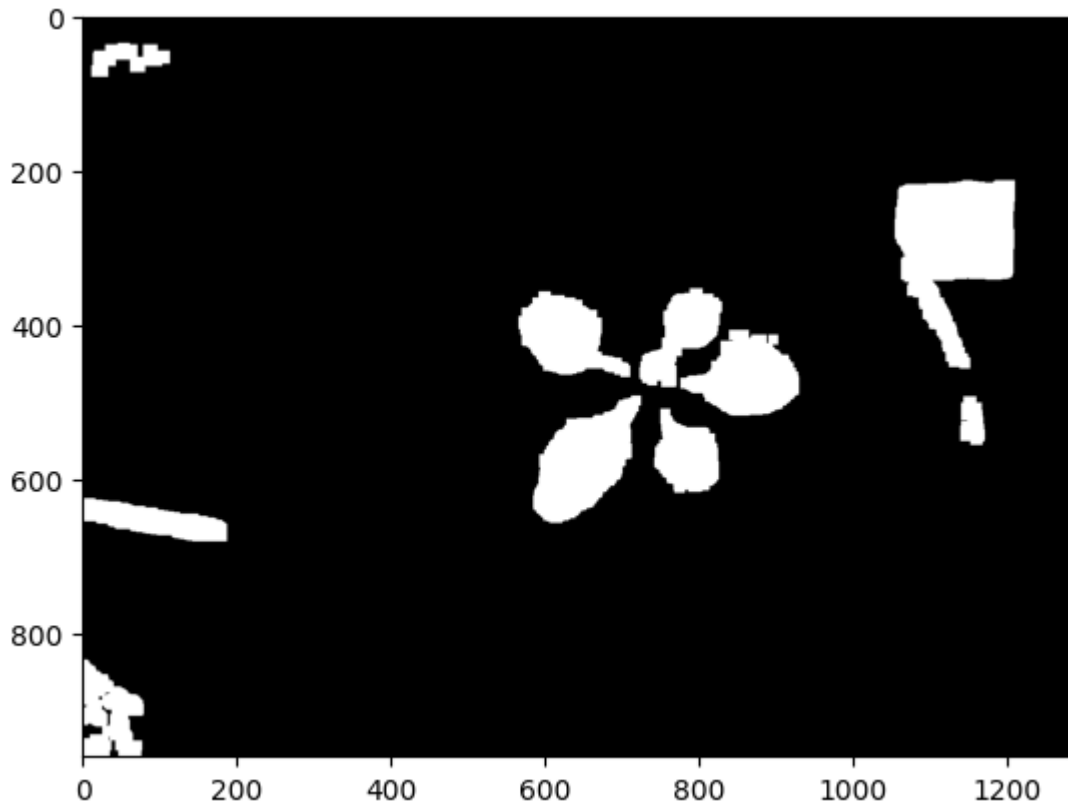


Position of the flowerpot is x= 694 y= 598 radius= 458  
possible watering point: (663, 617)  
possible watering point: (547, 506)  
possible watering point: (806, 808)  
possible watering point: (727, 867)  
possible watering point: (660, 567)  
possible watering point: (718, 367)  
possible watering point: (859, 732)  
possible watering point: (441, 660)  
possible watering point: (724, 590)  
possible watering point: (968, 656)  
possible watering point: (462, 668)  
possible watering point: (748, 622)  
possible watering point: (412, 573)  
possible watering point: (826, 648)  
possible watering point: (976, 522)  
possible watering point: (790, 878)  
possible watering point: (766, 388)  
possible watering point: (891, 501)  
possible watering point: (708, 599)  
possible watering point: (638, 501)  
possible watering point: (692, 566)  
possible watering point: (665, 392)  
possible watering point: (798, 631)  
possible watering point: (694, 586)  
possible watering point: (522, 495)  
possible watering point: (705, 530)  
possible watering point: (884, 825)  
possible watering point: (943, 497)  
possible watering point: (502, 587)  
Execution time: 0.95 seconds

##### END #####

##### START #####

processing image image 10 :  
total 47 region(s) found!



Position of the flowerpot is x= 714 y= 614 radius= 454

possible watering point: (694, 750)

possible watering point: (862, 713)

possible watering point: (714, 829)

possible watering point: (998, 558)

possible watering point: (473, 622)

possible watering point: (677, 771)

possible watering point: (475, 725)

possible watering point: (799, 788)

possible watering point: (599, 706)

possible watering point: (717, 649)

possible watering point: (501, 786)

possible watering point: (574, 691)

possible watering point: (556, 614)

possible watering point: (727, 637)

possible watering point: (490, 532)

possible watering point: (711, 759)

possible watering point: (554, 555)

possible watering point: (729, 318)

possible watering point: (838, 722)

possible watering point: (960, 635)

possible watering point: (740, 664)

possible watering point: (532, 645)

possible watering point: (587, 693)

possible watering point: (708, 780)

possible watering point: (675, 855)

possible watering point: (799, 762)

possible watering point: (894, 853)

possible watering point: (695, 346)

possible watering point: (525, 623)

Execution time: 0.99 seconds

##### END #####

##### ALL DONE #####

```
In [ ]: # Plot Labels of each image
pcv.params.debug = 'None'

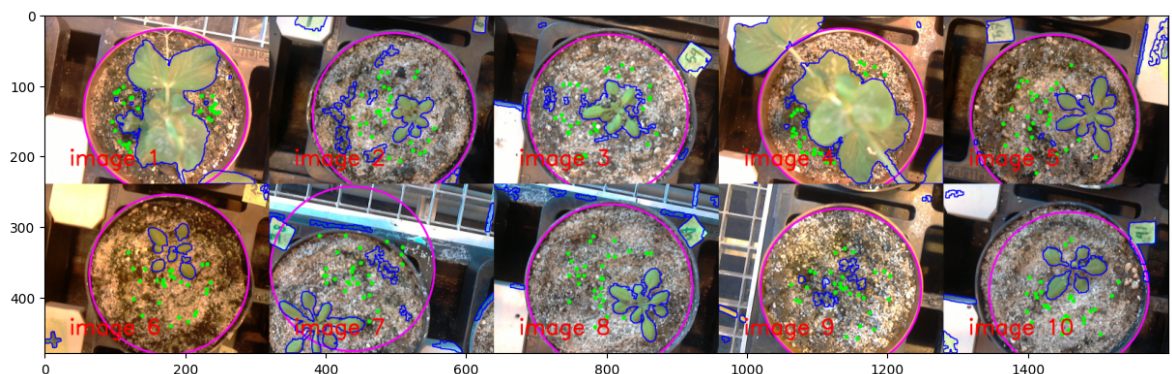
pcv.params.text_size = 4
pcv.params.text_thickness = 10
labeled_test_out_images = []

for i, test_img_out in enumerate(test_images_out):
    labeled=test_img_out.copy()
    labeled = cv.putText(img=labeled, text=test_image_names[i], org=(x,y),
                        fontFace=cv.FONT_HERSHEY_SIMPLEX,
                        fontScale=pcv.params.text_size, color=(0,0,255), thickn
    labeled_test_out_images.append(labeled)

plotting_img = np.vstack([np.hstack([labeled_test_out_images[0], labeled_test_out
                                np.hstack([labeled_test_out_images[5], labeled_test_out
plotting_img = pcv.transform.resize_factor(plotting_img, factors=(0.25,0.25))

#pcv.plot_image(plotting_img) this function will print the image on jupyter note
plt.figure(figsize=(15,10))
plt.imshow(cv.cvtColor(plotting_img, cv.COLOR_BGR2RGB))
plt.show()

pcv.params.debug = 'plot'
```



## evaluation

First, it is important to explain the legend of the results.

- The magenta circle represents the position of the flower pot, which is recognized by the program.
- The blue contour indicates the area of the plant's leaves that the program identifies.
- The green dots show the optimal positions for watering the plant, avoiding pouring water on the leaves or too close to the edge of the flower pot.

As demonstrated, the program generates usable results in nearly all 10 cases. In image 2, the program mistakenly recognized some soil particles as leaves due to the color of the dark soil or possibly moss. This phenomenon also occurred in some other test images, but it did not affect the final results. In Image 7, the program failed to detect the flower top due to the interference of the metal grills on the top.



The program relies on contextual knowledge to perform calculations. These are:

- The object's color.
- The flower pot should be positioned near the center of the image.
- Most of the circle should be visible in the image.
- Yellow and green post-it labels may affect the results.
- The presence of moss on the soil may also impact the results.