



Εργαστήριο 3

Τεχνητής Όρασης

Κώστας Μαριάς

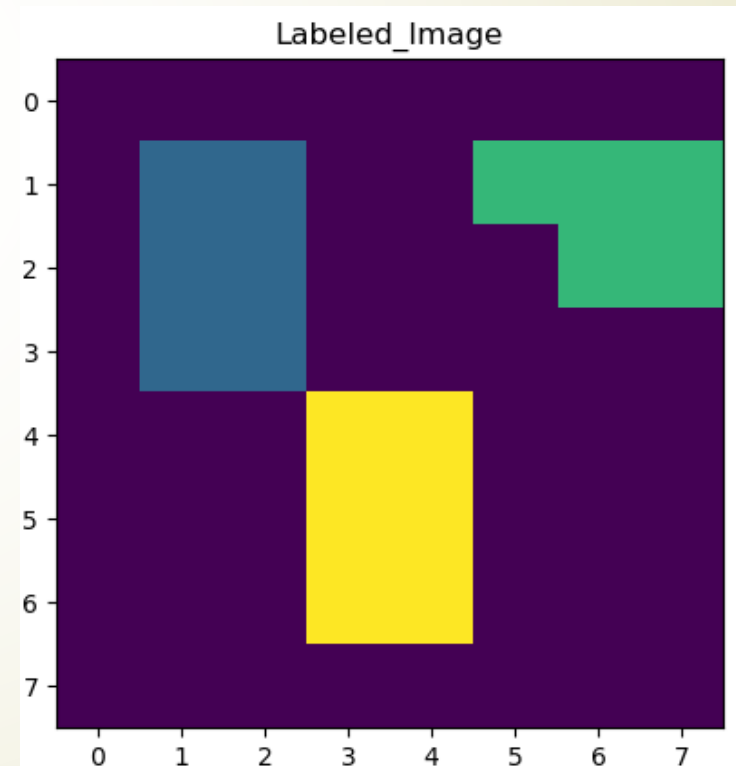
Labeling

3

```
[Labeled_image,num_of_neighbors] = measure.label(Image_Binary,  
neighbors=4,background=False,return_num=True)
```

Για να οπτικοποιήσουμε το αποτέλεσμα, πρέπει να εμφανίσουμε τη μήτρα ετικετών ως εικόνα ψευδοχρώματος RGB.

```
plt.figure(1)  
plt.imshow(Labeled_image)  
plt.title('Labeled_Image')
```



Πως βρίσκουμε το εμβαδό κάθε περιοχής?

Group connected list=image tool.measure.regionprops(Labeled image)

Group_connected_list - List (3 elements)

Index	Type	Size	Value
0	measure._regionprops._RegionProperties	1	_RegionProperties object of skima...
1	measure._regionprops._RegionProperties	1	_RegionProperties object of skima...
2	measure._regionprops._RegionProperties	1	_RegionProperties object of skima...

Save and Close Close

0 - skimage.measure._regionprops._RegionProperties

Attribute	Type	Size	Value
_slice	tuple	2	(slice, slice)
_transpose_moments	bool	1	True
_use_xy_warning	bool	1	True
area	int32	1	6
bbox	tuple	4	(1, 1, 4, 3)
bbox_area	int	1	6
centroid	tuple	2	(2.0, 1.5)
convex_area	int32	1	6
convex_image	bool	(3, 2)	[[True True] [True True]
coords	int64	(6, 2)	[[1 1] [1 2]
eccentricity	float	1	0.7905694150420949

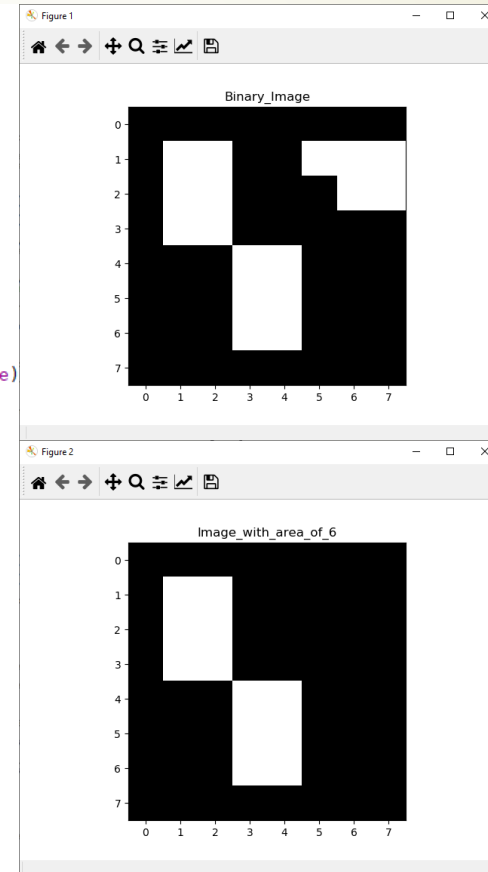
Save and Close Close

Βρίσκοντας και εμφανίζοντας μεγαλύτερες περιοχές

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import skimage as image_tool
4 from skimage import measure
5
6
7 Image_Binary=[[0, 0, 0, 0, 0, 0, 0, 0],[0, 1, 1, 0, 0, 1, 1, 1],
8              [0, 1, 1, 0, 0, 0, 1, 1],[0, 1, 1, 0, 0, 0, 0, 0],
9              [0, 0, 0, 1, 1, 0, 0, 0],[0, 0, 0, 1, 1, 0, 0, 0],
10             [0, 0, 0, 1, 1, 0, 0, 0],[0, 0, 0, 0, 0, 0, 0, 0]]
11
12 Image_Binary=np.asarray(Image_Binary, dtype=np.bool)
13 Image_Binary=Image_Binary.astype(int)
14
15 plt.figure(1)
16 plt.imshow(Image_Binary,cmap=plt.cm.gray)
17 plt.title('Binary_Image')
18
19 [Labeled_image,num_of_neighbors] = measure.label(Image_Binary, neighbors=4,background=False,return_num=True)
20
21 Group_connected_list=image_tool.measure.regionprops(Labeled_image)
22
23
24 List_of_Areas=[]
25 List_of_Areas.append([d['area'] for d in Group_connected_list ] )
26
27 List_of_Areas = np.transpose(List_of_Areas)
28
29 list_of_Areas_6=[i for i, e in enumerate(List_of_Areas) if e > 5]
30 list_of_Areas_6=np.transpose(list_of_Areas_6)+1
31
32
33 Image_with_area_6=[]
34
35 for i in range(len(list_of_Areas_6)):
36     Image_with_area_6.append(Labeled_image==list_of_Areas_6[i])
37
38
39
40 Image_with_area_6=sum(Image_with_area_6)
41
42 plt.figure(2)
43 plt.imshow(Image_with_area_6,cmap="gray")
44 plt.title('Image_with_area_of_6')
45

```



Challenge # 1

Φτιάξτε πάλι τον πίνακα BW 8x8 όπως στην αρχή της διάλεξης.

1) Φτιάξτε το RGB_label και βάλτε ετικέτες=αριθμό κάθε περιοχής πάνω στα αντικείμενα (στο κέντρο βάρους/centroid).

2) Φτιάξτε το RGB_label και βάλτε ετικέτες=εμβαδό κάθε περιοχής πάνω στα αντικείμενα (στο κέντρο βάρους/centroid).

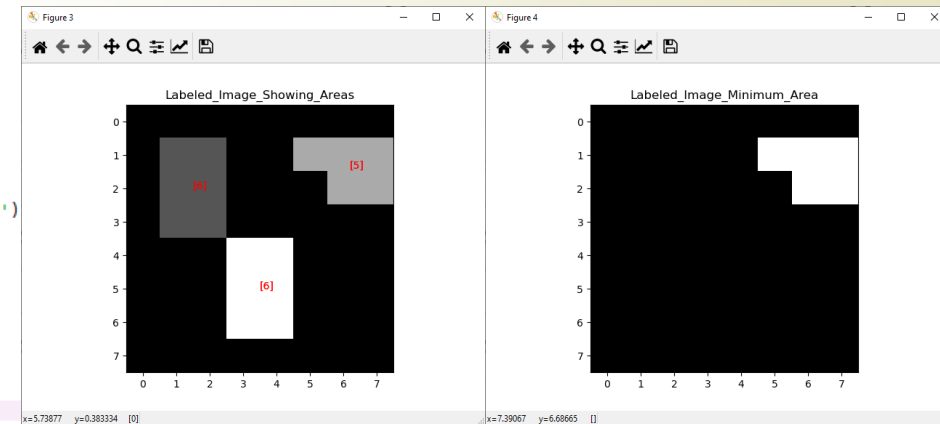
3) Βρείτε την μικρότερη περιοχή και εμφανίστε την ξεχωριστά σαν εικόνα.

Challenge #1

```

3 import skimage as image_tool
4 from skimage import measure
5
6
7 Image_Binary=[[0, 0, 0, 0, 0, 0, 0, 0],[0, 1, 1, 0, 0, 1, 1, 1],
8              [0, 1, 1, 0, 0, 0, 1, 1],[0, 1, 1, 0, 0, 0, 0, 0],
9              [0, 0, 0, 1, 1, 0, 0, 0],[0, 0, 0, 1, 1, 0, 0, 0],
10             [0, 0, 0, 1, 1, 0, 0, 0],[0, 0, 0, 0, 0, 0, 0, 0]]
11
12 Image_Binary=np.asarray(Image_Binary, dtype=np.bool)
13 Image_Binary=Image_Binary.astype(int)
14
15 plt.figure(1)
16 plt.imshow(Image_Binary,cmap=plt.cm.gray)
17 plt.title('Binary_Image')
18
19 [Labeled_image,num_of_neighbors] = measure.label(Image_Binary, neighbors=4,background=False,return_num=True)
20
21 Group_connected_list=image_tool.measure.regionprops(Labeled_image)
22
23 plt.figure(2)
24 plt.imshow(Labeled_image,cmap=plt.cm.gray)
25 plt.title('Labeled_Image_Numbering_Groups')
26
27 for i in range(num_of_neighbors):
28     plt.annotate(i, xy=(Group_connected_list[i].centroid[1],Group_connected_list[i].centroid[0]),color='red')
29
30 #-----
31 List_of_Areas=[]
32 List_of_Areas.append([d['area'] for d in Group_connected_list ] )
33 List_of_Areas = np.transpose(List_of_Areas)
34
35 plt.figure(3)
36 plt.imshow(Labeled_image,cmap=plt.cm.gray)
37 plt.title('Labeled_Image_Showing_Areas')
38
39 for i in range(num_of_neighbors):
40     plt.annotate(List_of_Areas[i], xy=(Group_connected_list[i].centroid[1],Group_connected_list[i].centroid[0]),color='red')
41
42
43
44 Position_of_Minimum_Area=[i for i, e in enumerate(List_of_Areas) if e==min(List_of_Areas)]
45 Position_of_Minimum_Area = np.transpose(Position_of_Minimum_Area)+1
46
47 plt.figure(4)
48 plt.imshow(Labeled_image==Position_of_Minimum_Area,cmap=plt.cm.gray)
49 plt.title('Labeled_Image_Minimum_Area')
50

```



Challenge #2

- ▶ Να διαβάσετε την εικόνα `rice.png` και να υπολογίσετε την δυαδική εικόνα με τους κόκκους ρυζιού.
- ▶ Τι παρατηρείτε? Εξετάστε το ιστόγραμμα των εμβαδών των περιοχών της εικόνας και δείτε ξεχωριστά 3 κατηγορίες ανάλογα με το εμβαδό των περιοχών που έχουν προκύψει από το labelling.
- ▶ Τι συμπεράσματα μπορείτε να πάρετε κοιτώντας την κάθε περιοχή ξεχωριστά (δηλαδή τις περιοχές που προκύπτουν αν κρατήσουμε μόνο αυτές που έχουν εμβαδό από <150 , $150:300$, >300)

Challenge #2

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 import numpy as np
4 from skimage.filters import threshold_otsu
5 from skimage import measure
6 import skimage as image_tool
7
8
9 Grayscale_image = mpimg.imread('rice.png')
10
11
12 thresh = threshold_otsu(Grayscale_image)
13 Image_Binary = Grayscale_image > thresh
14
15 plt.figure(1)
16 plt.imshow(Image_Binary,cmap=plt.cm.gray)
17 plt.title('Image_Binary')
18
19 [Labeled_image,num_of_neighbors] = measure.label(Image_Binary, neighbors=4,background=False,return_num=True)
20
21 Group_connected_list=image_tool.measure.regionprops(Labeled_image)
22
23
24 List_of_Areas=[]
25 List_of_Areas.append([d['area'] for d in Group_connected_list ] )
26
27
28 List_of_Areas = np.transpose(List_of_Areas)
29
30 Position_of_Areas=[i for i, e in enumerate(Group_connected_list)]
31 Position_of_Areas=np.asarray(Position_of_Areas, dtype=np.int32)
32
33 plt.figure(2)
34 hist1= plt.hist(List_of_Areas, bins='auto')
35 plt.title("Histogram with 'auto' bins")
36 plt.show()
37
38 Position_from_0_to_150=[i for i, e in enumerate(List_of_Areas) if e <=150]
39 Position_from_0_to_150=np.transpose(Position_from_0_to_150)+1
40
41 Position_from_151_to_300=[i for i, e in enumerate(List_of_Areas) if e >150 and e<301]
42 Position_from_151_to_300=np.transpose(Position_from_151_to_300)+1
43
44 Position_from_301_MAX=[i for i, e in enumerate(List_of_Areas) if e >300]
45 Position_from_301_MAX=np.transpose(Position_from_301_MAX)+1
46 #-----
47
```

Challenge #2

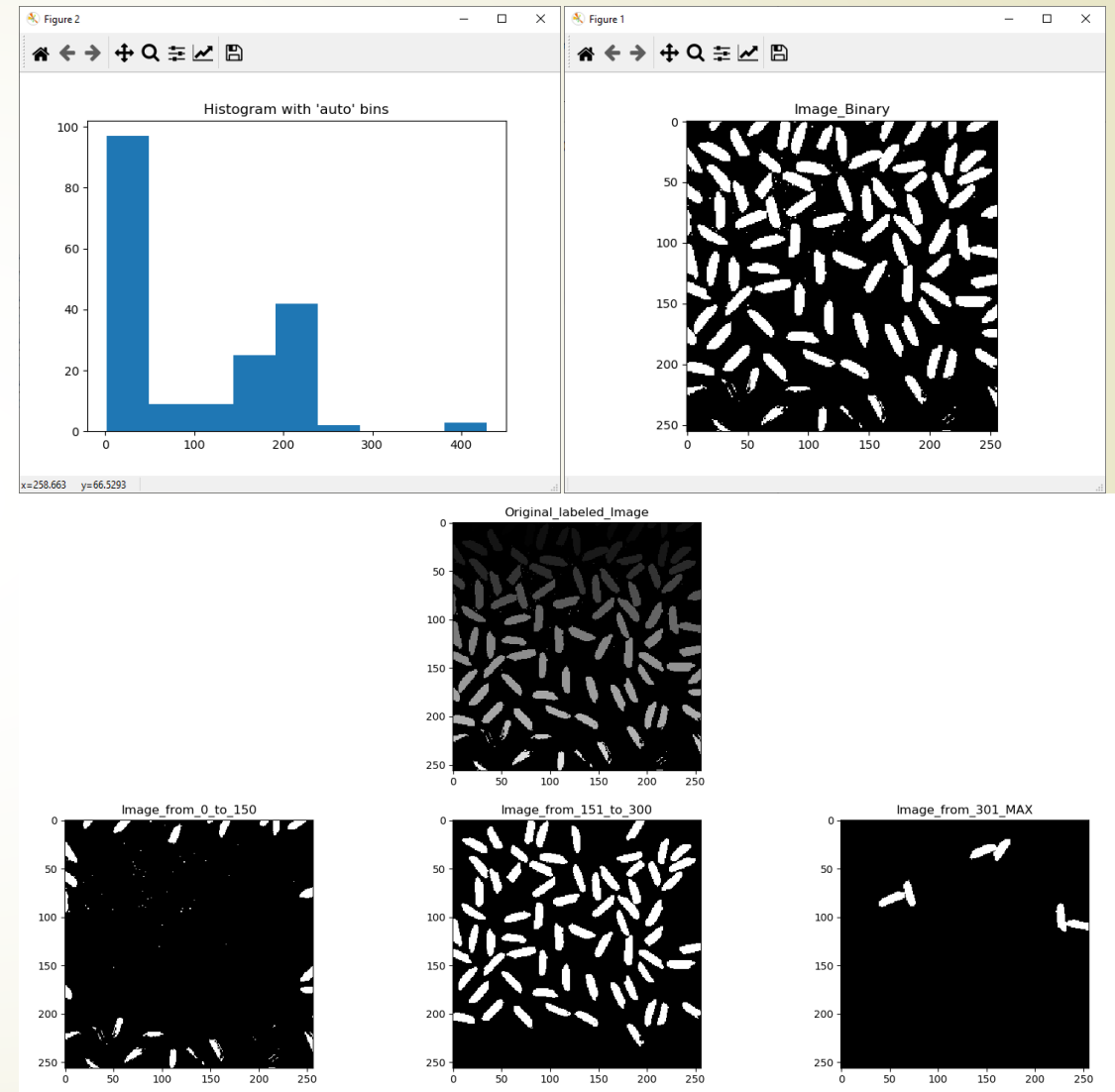
```
47
48 Image_from_0_to_150=[]
49 Image_from_151_to_300=[]
50 Image_from_301_MAX=[]
51
52 for i in range(len(Position_from_0_to_150)):
53
54     Image_from_0_to_150.append(Labeled_image==Position_from_0_to_150[i])
55
56 for i in range(len(Position_from_151_to_300)):
57
58     Image_from_151_to_300.append(Labeled_image==Position_from_151_to_300[i])
59
60 for i in range(len(Position_from_301_MAX)):
61     Image_from_301_MAX.append(Labeled_image==Position_from_301_MAX[i])
62
63 #-----
64 for i in range(len(Position_from_0_to_150)):
65
66     Image_from_0_to_150[i]=np.asarray(Image_from_0_to_150[i], dtype=np.int32)
67
68 for i in range(len(Position_from_151_to_300)):
69
70     Image_from_151_to_300[i]=np.asarray(Image_from_151_to_300[i], dtype=np.int32)
71
72 for i in range(len(Position_from_301_MAX)):
73
74     Image_from_301_MAX[i]=np.asarray(Image_from_301_MAX[i], dtype=np.int32)
75
76 #-----
77 Image_from_0_to_150=sum(Image_from_0_to_150)
78 Image_from_151_to_300=sum(Image_from_151_to_300)
79 Image_from_301_MAX=sum(Image_from_301_MAX)
80
81 plt.figure(3)
__
```

Challenge #2

```

81 plt.figure(3)
82
83
84
85 fig3=plt.figure(3)
86
87 subplot3=fig3.add_subplot(2, 3, 2)
88 plt.imshow(Labeled_image,cmap = "gray")
89 subplot3.set_title('Original_labeled_Image')
90 |
91 subplot3=fig3.add_subplot(2, 3, 4)
92 plt.imshow(Image_from_0_to_150,cmap = "gray")
93 subplot3.set_title('Image_from_0_to_150')
94
95 subplot3=fig3.add_subplot(2, 3, 5)
96 plt.imshow(Image_from_151_to_300,cmap = "gray")
97 subplot3.set_title('Image_from_151_to_300')
98
99 subplot3=fig3.add_subplot(2, 3, 6)
100 plt.imshow(Image_from_301_MAX,cmap = "gray")
101 subplot3.set_title('Image_from_301_MAX')

```

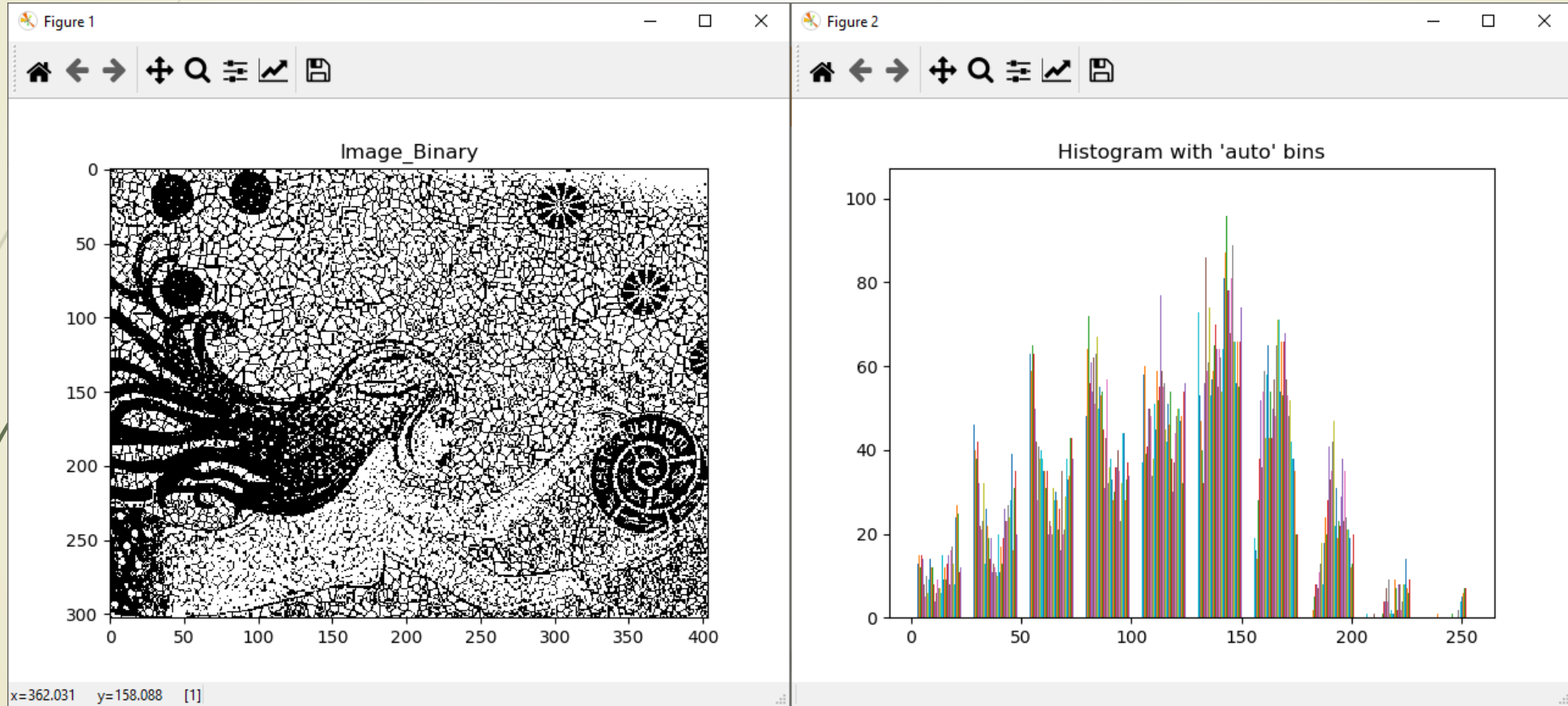


Challenge #3?

► You have **one week** to complete Project #1.

1. Read the RGB (3 channel) image "Mermaid.jpg"
2. Find the appropriate channel OR do grayscale conversion of RGB image
3. Then convert it to binary.
4. Measure the areas of the image (Tip!!! List_of_Areas=2658)
5. Measure the histogram of the 1 channel image
6. The results must be identical to the figures below
7. I will NOT answer any **basic** question on my e-mail because you are trained enough to accomplish this project.
8. The entire project, must be compressed as .rar file with name:
Tp_YourName.rar

Project #1



Project #1

