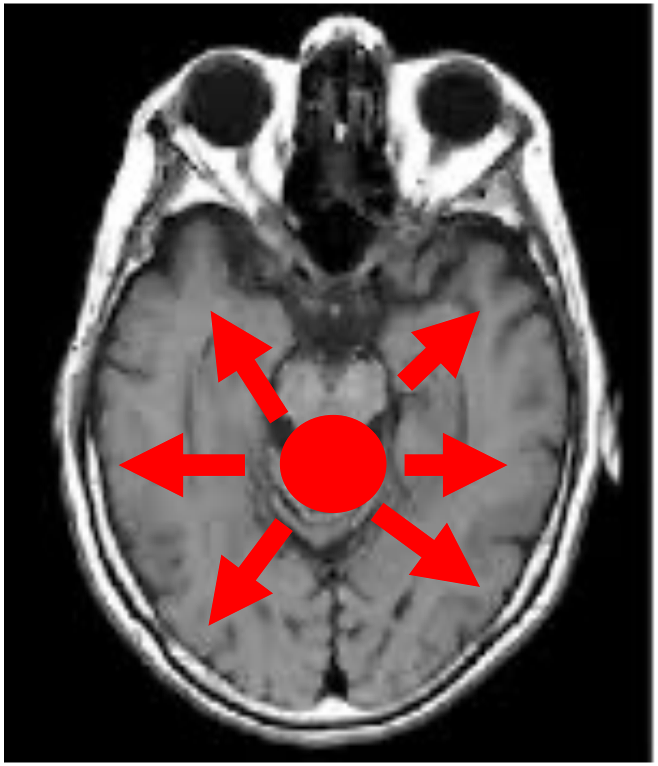
Lecture 8 Morphological Segmentation

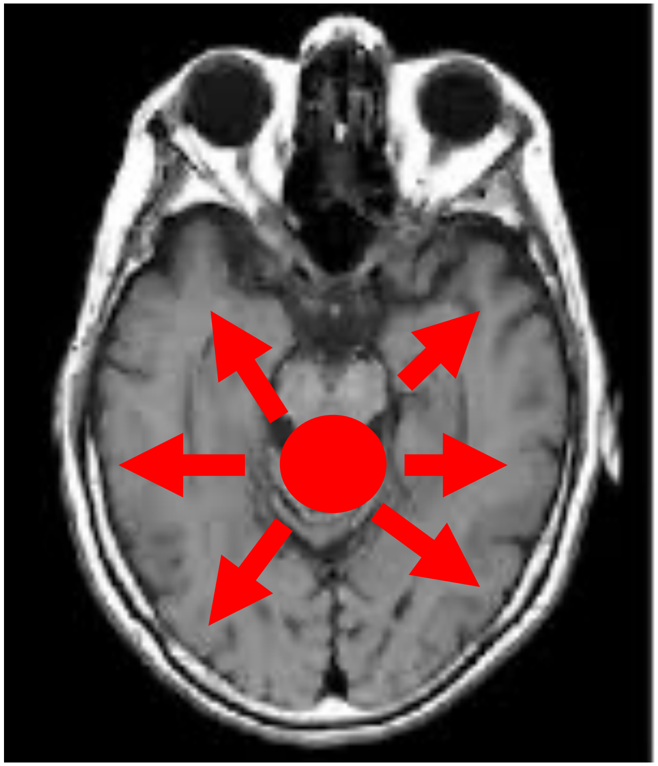
ECE 1390/2390

Watershed algorithm



A Watershed
rain
precipitation
snow
watershed divide
tributaries
watershed divide
percolation
percolation
groundwater
(aquifer)

Watershed algorithm



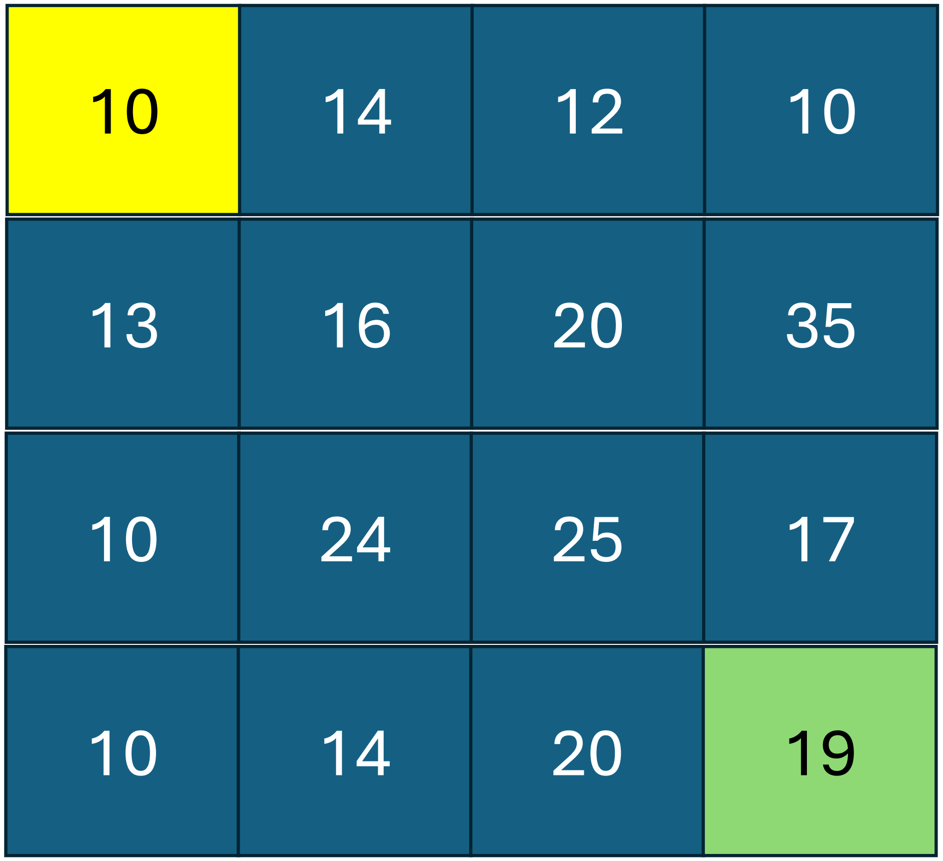
Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

Watershed algorithm

Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

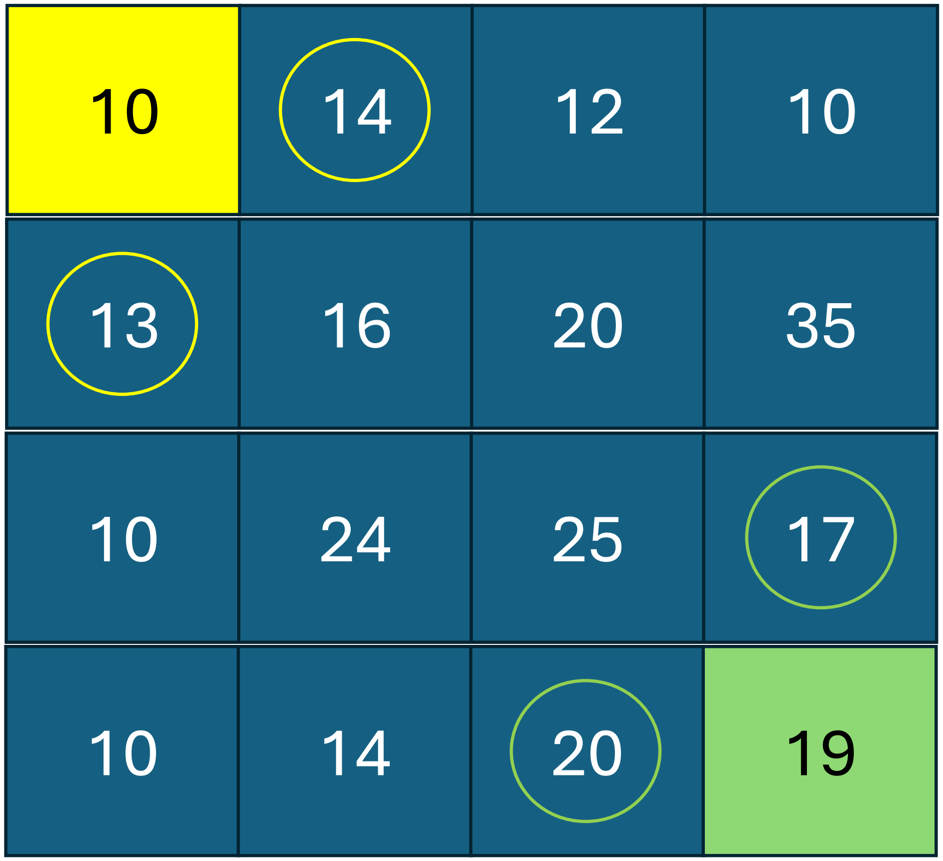


10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

Watershed algorithm

Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

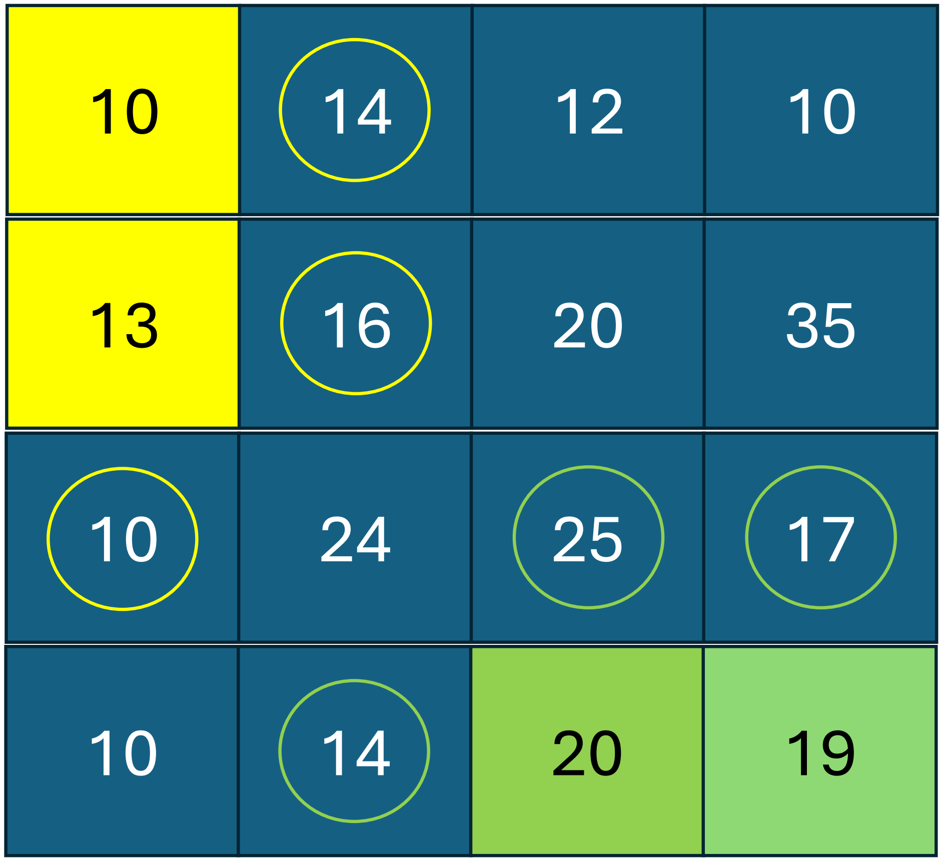


10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

Watershed algorithm

Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

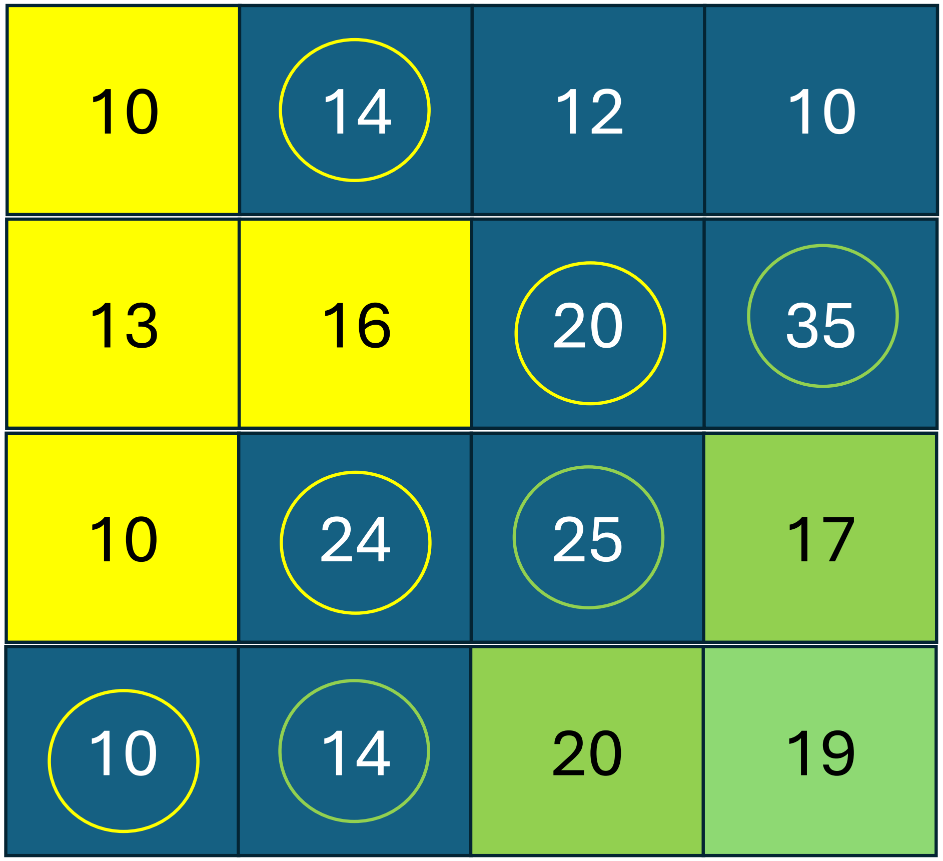


10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

Watershed algorithm

Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

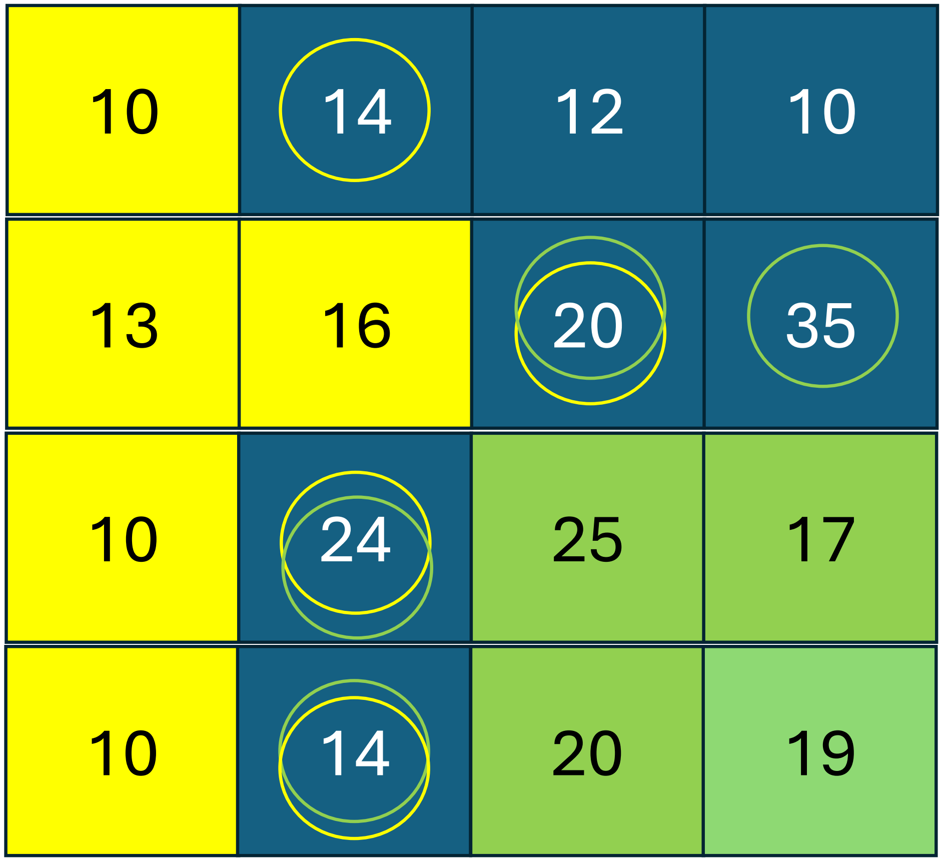


10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

Watershed algorithm

Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

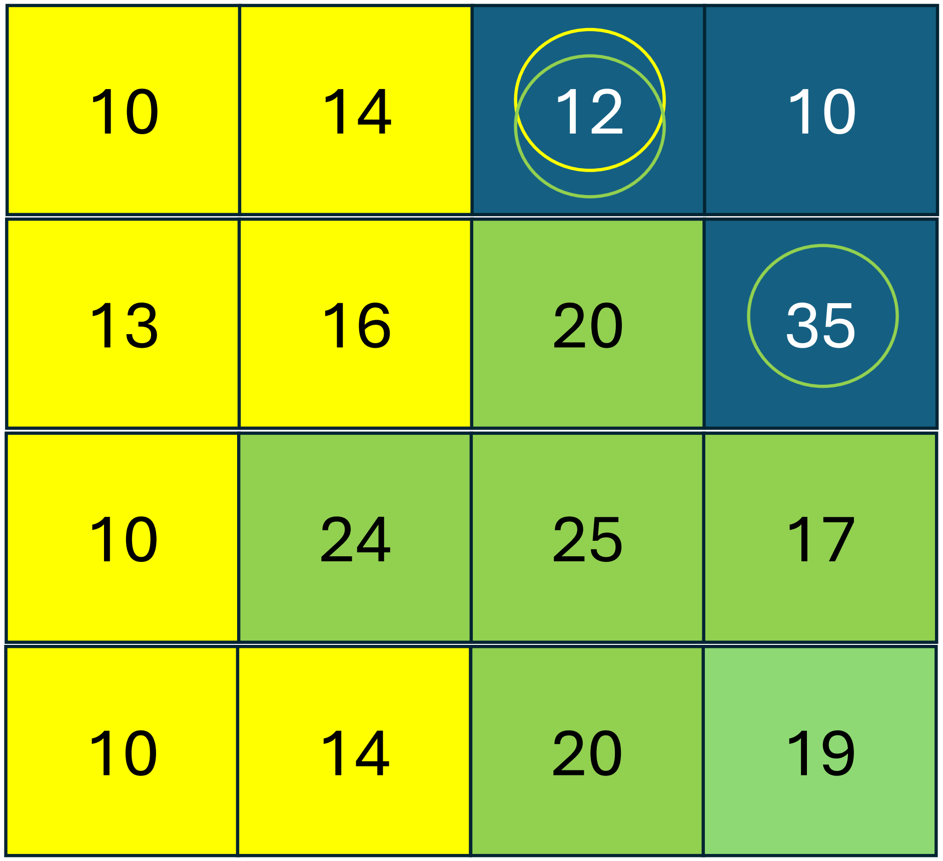


10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

Watershed algorithm

Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**

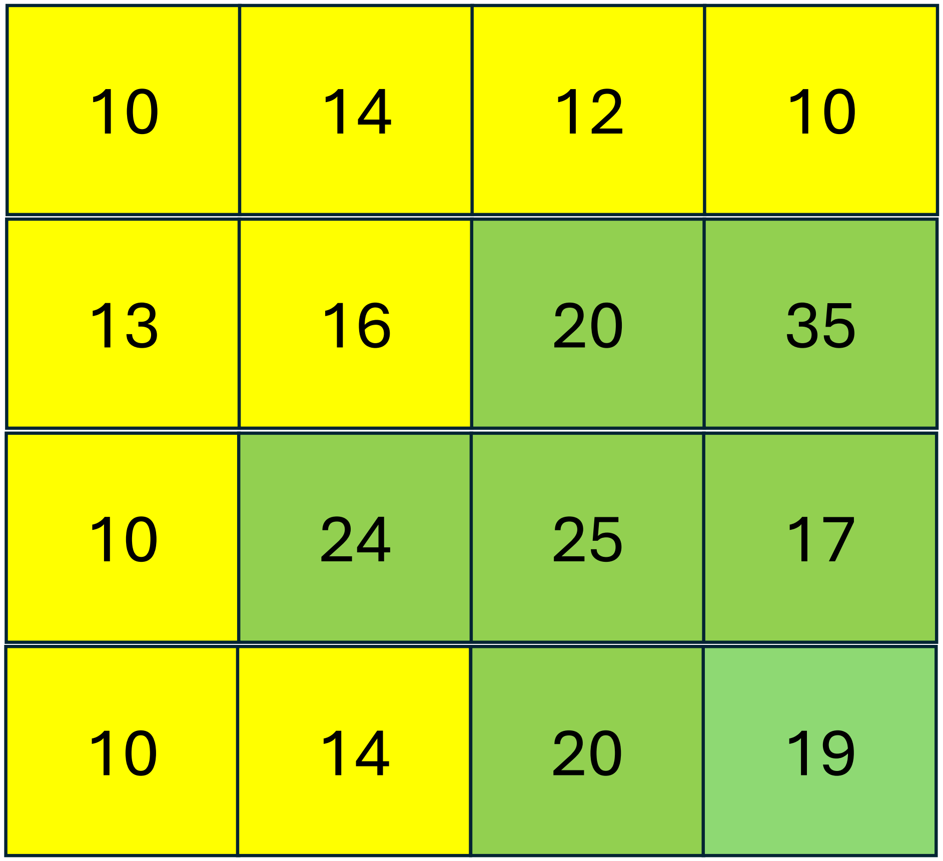


10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

Watershed algorithm

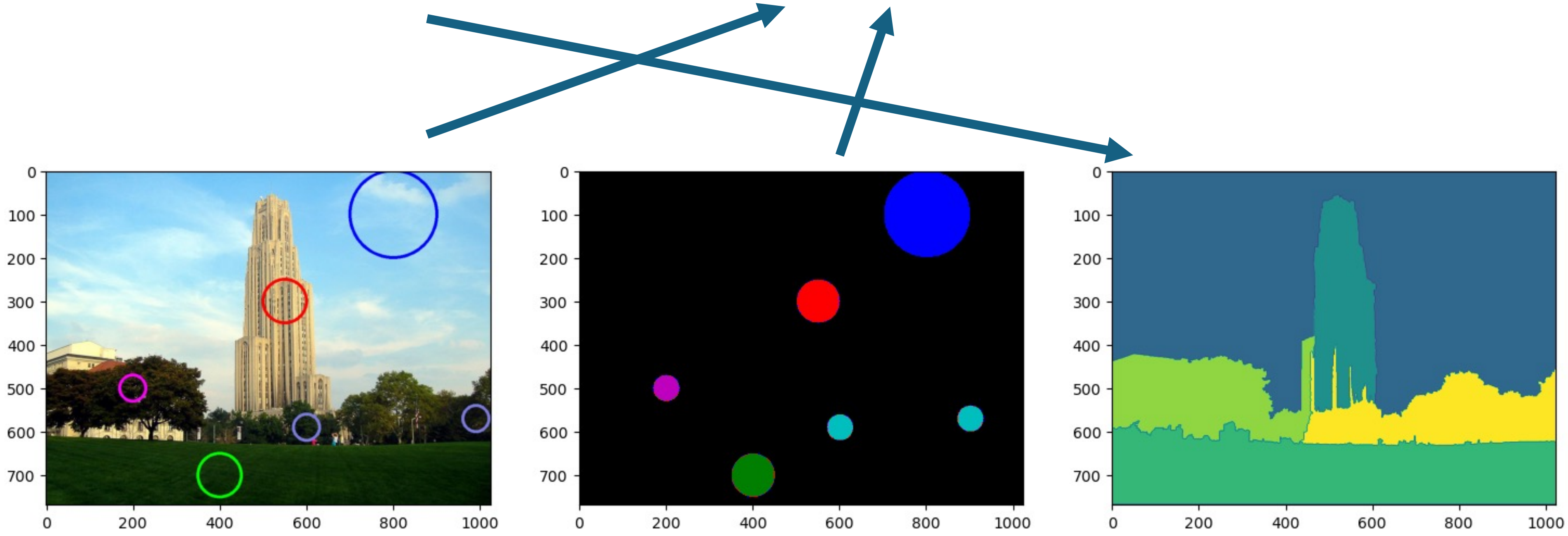
Watershed by flooding

- **Define a set of markers in the image**
- **Define the flood priority from region into neighbors based on the gradient of the image**
- **Flood the lowest priority neighbor (lowest gradient)**



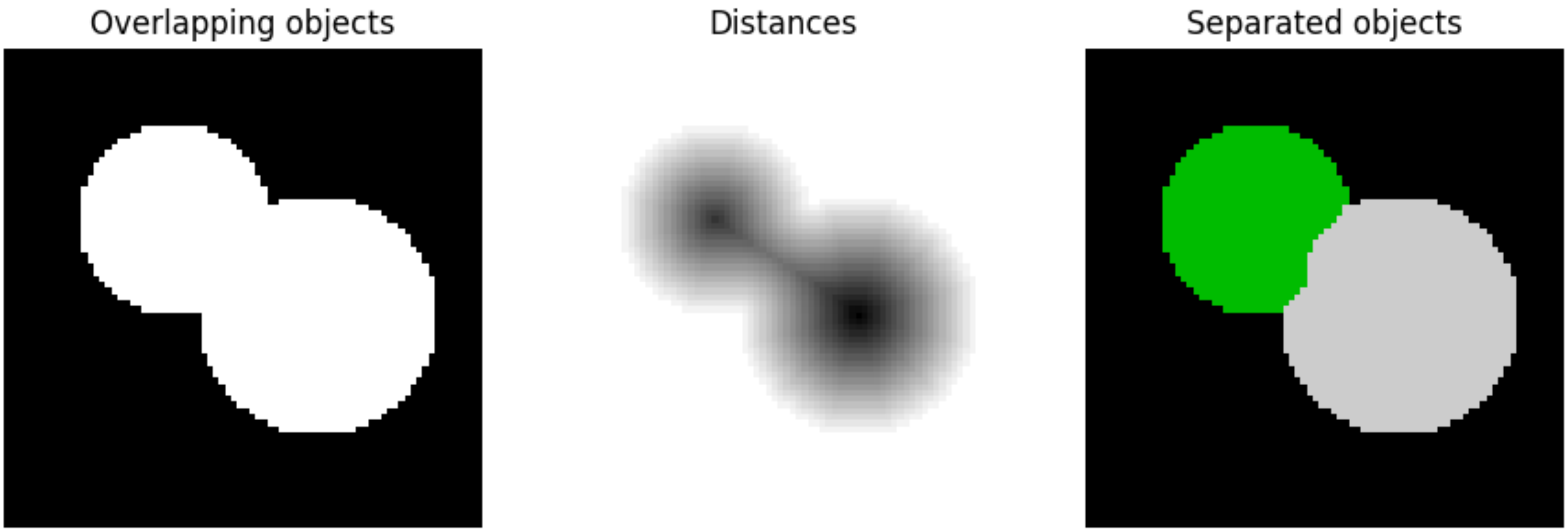
10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

```
markersWS = cv2.watershed(im,markers)
```



Watershed algorithm

Watershed by topographic distance

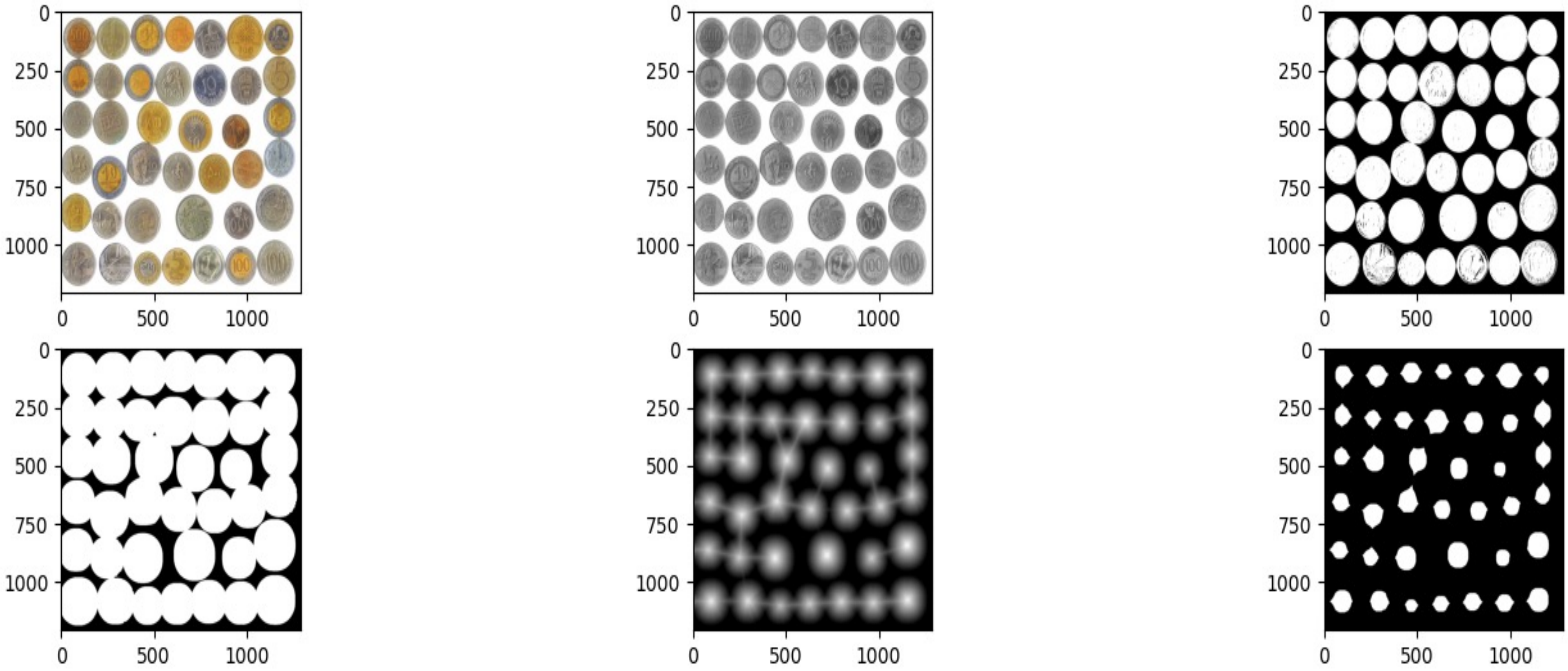


Overlapping objects

Distances

Separated objects

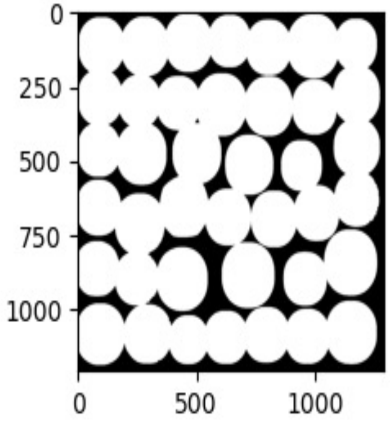
Watershed algorithm

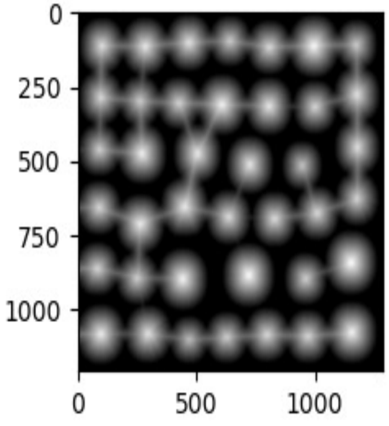


Watershed algorithm

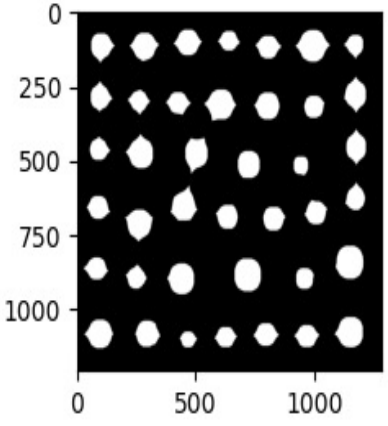
```
dist = cv2.distanceTransform(imbin_dil, cv2.DIST_L2, 5)
```

Mask size



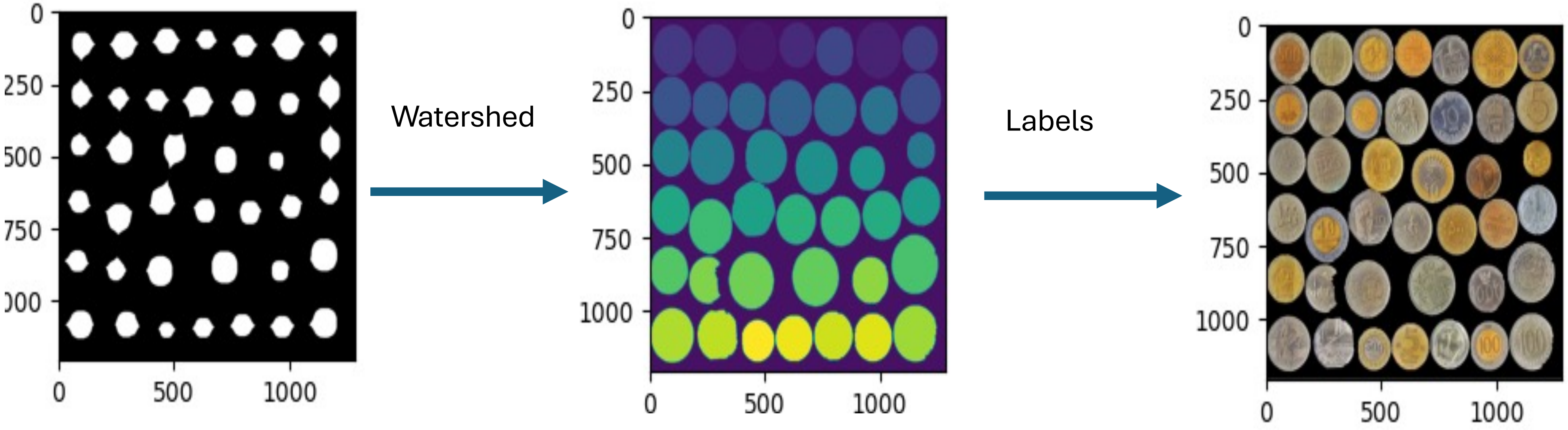


Threshold



Watershed algorithm

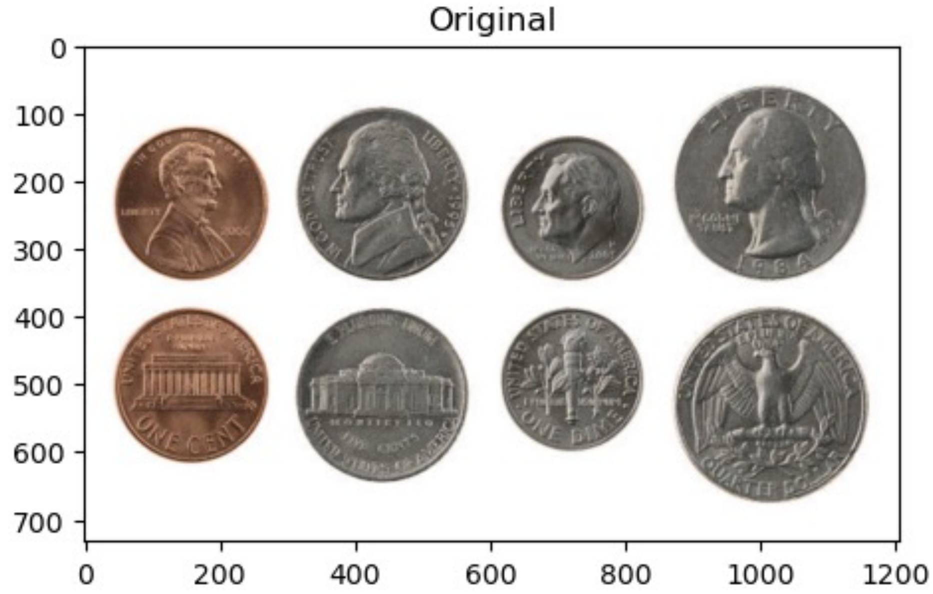
```
_,markers = cv2.connectedComponents(coin_centers)
```

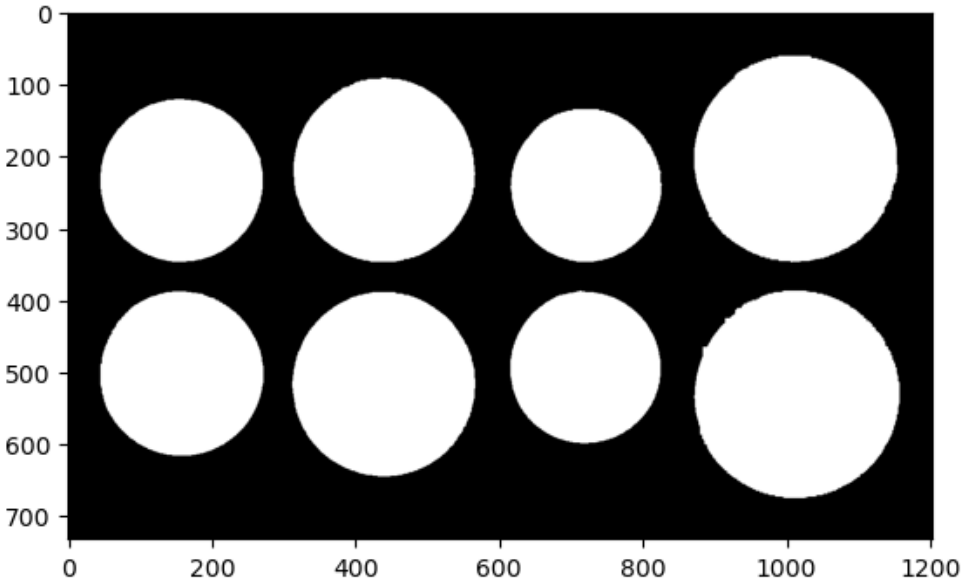


Watershed

Labels

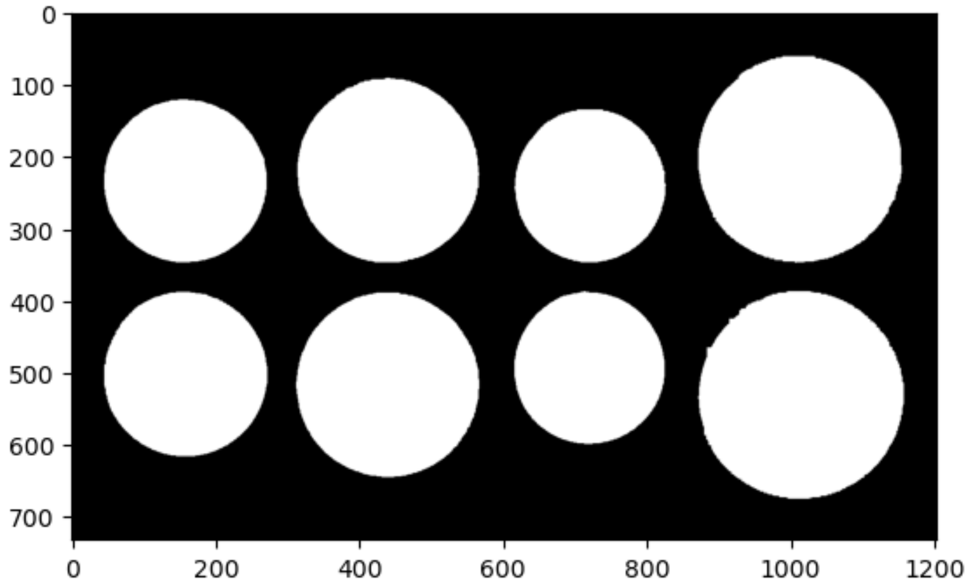
Contour segmentation

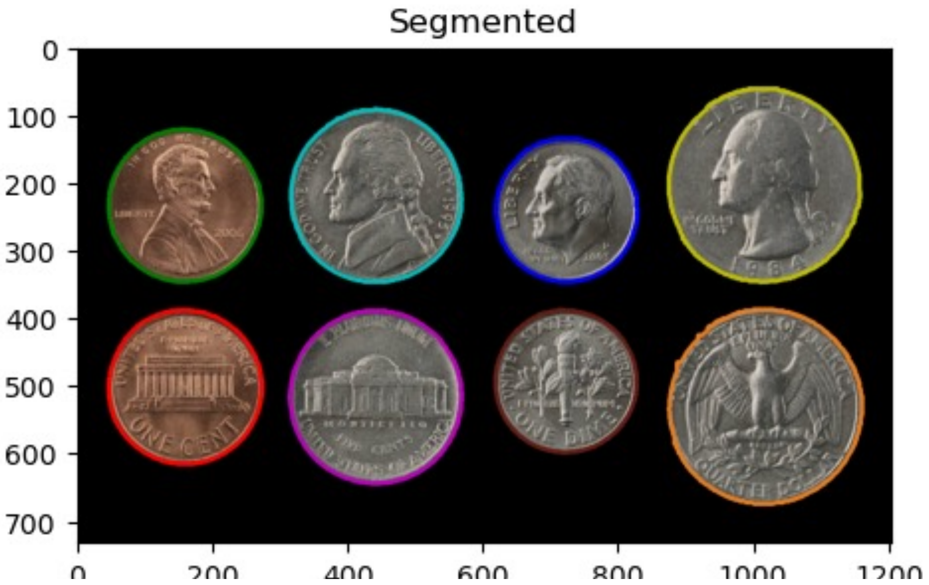




Contour segmentation

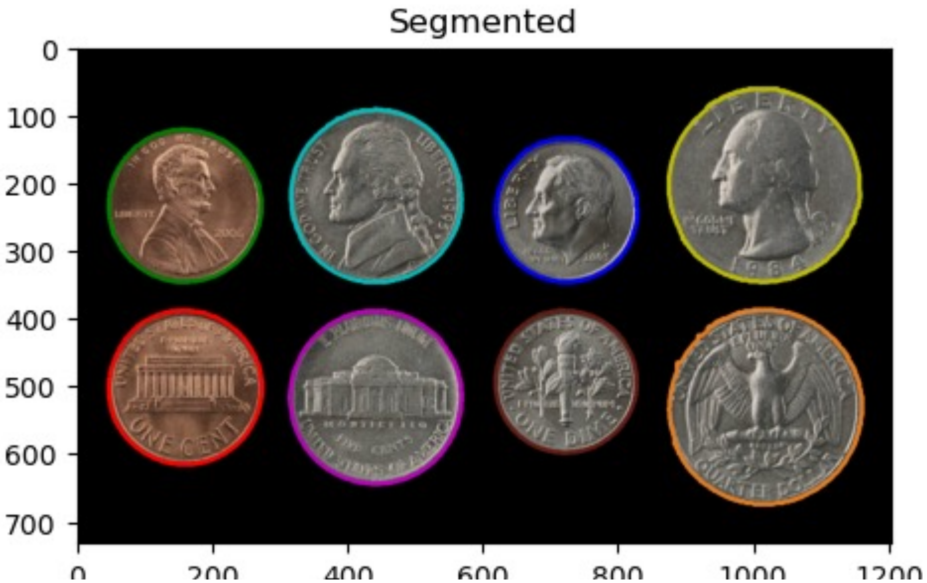
```
contours,hierarchy = cv2.findContours(thres,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
```





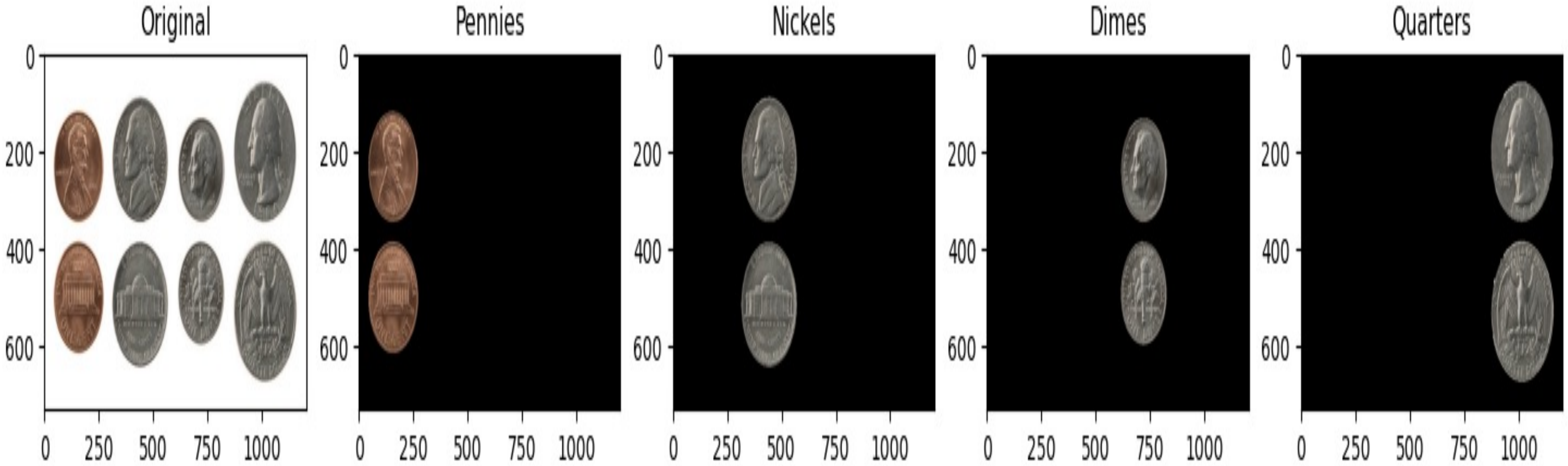
Contour segmentation

```
area=np.zeros((len(sorted_contour)))
for i in range(0,len(sorted_contour)):
       area[i]=cv2.contourArea(sorted_contour[i])
```



Contour segmentation

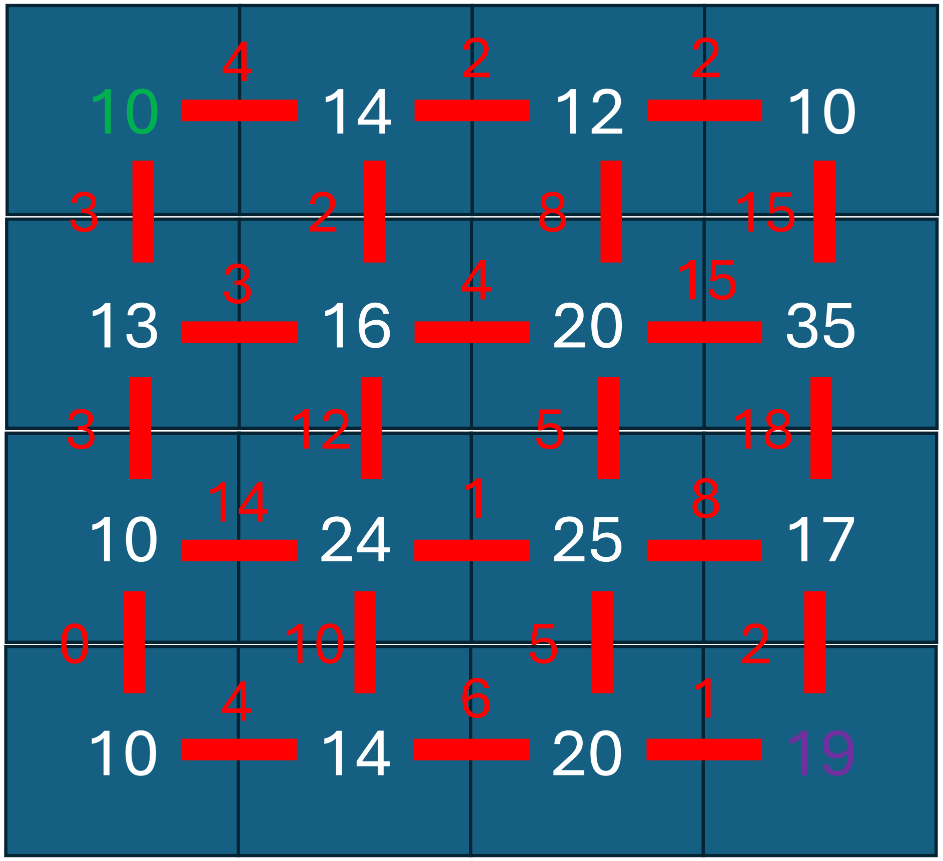
```
_,label,center=cv2.kmeans(area,4,None,criteria,10,cv2.KMEANS_RANDOM_CENTERS)
```



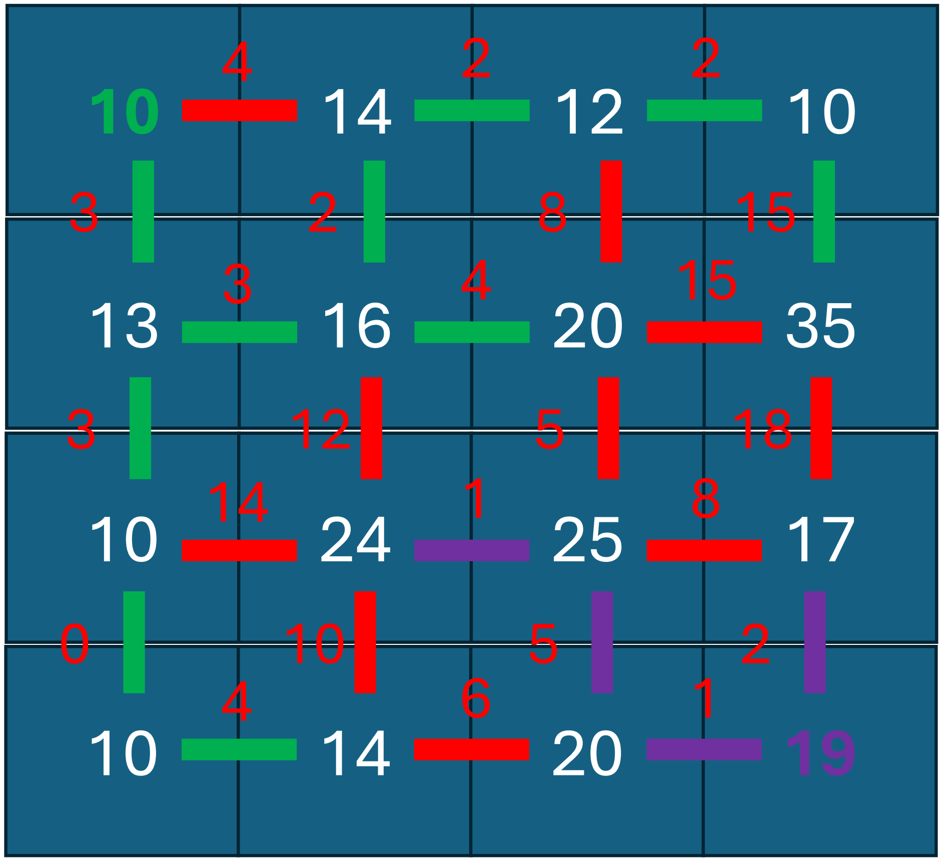
Minimum spanning forest

10	14	12	10
13	16	20	35
10	24	25	17
10	14	20	19

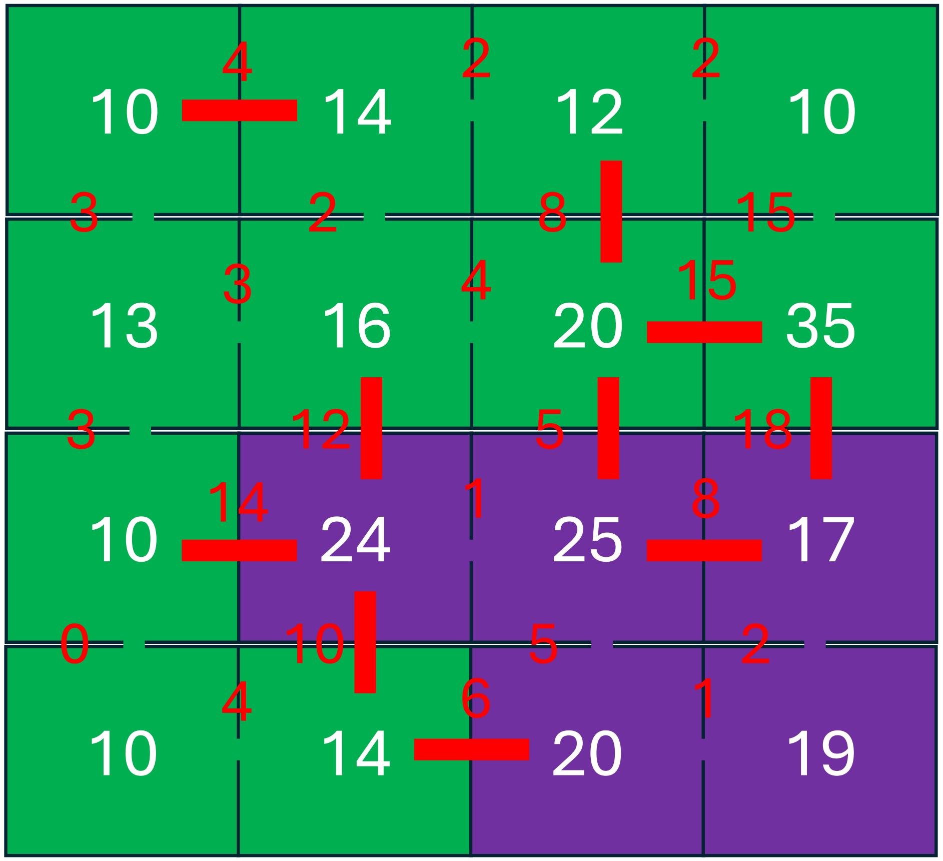
Minimum spanning forest



Minimum spanning forest



Minimum spanning forest


4
2
2
10
14
12
10
3
2
8
15
3
4
15
13
16
20
35
3
12
5
18
14
1
8
10
24
25
17
0
10
5
2
4
6
1
10
14
20
19