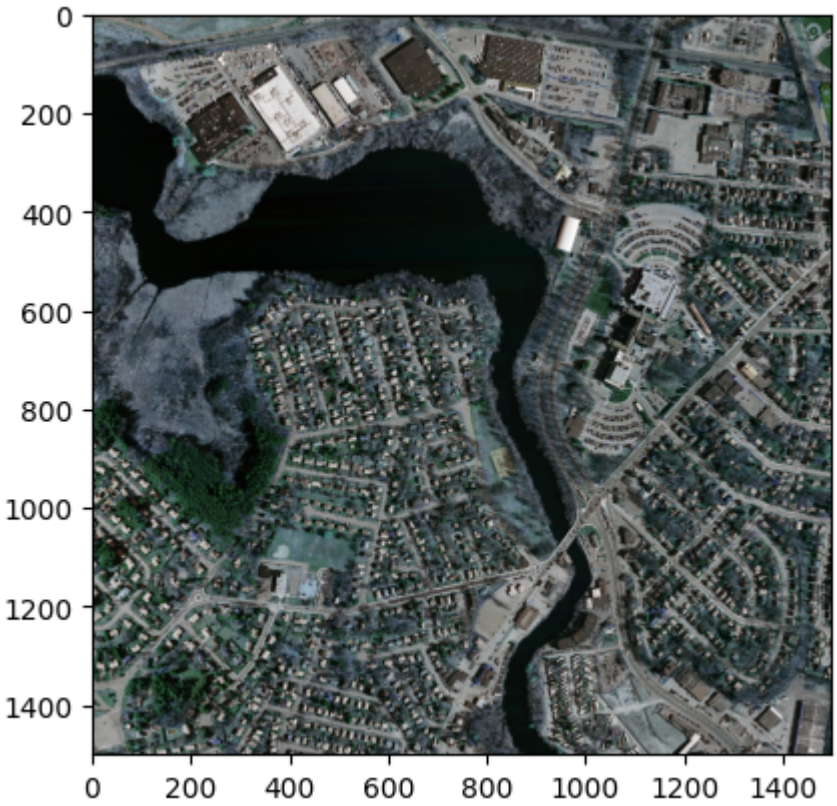
In [2]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.vq import *
```

In [45]:
```
img = cv2.imread('/content/drive/MyDrive/png (1)/train (1)/22678915_15 (1).png')
plt.imshow(img)
```

Out[45]:
```
<matplotlib.image.AxesImage at 0x7e4c7d578b50>
```



In [46]:
```
center = 3
gray = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
datalab = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

In [47]:
```
column = len(datalab[1,])
rows = len(datalab)
```

In [48]:
```
ab = datalab[:,:,2:3]
```

In [49]:
```
ab = np.reshape(ab,rows*column,order="C")
print(ab.shape)
```

```
(2250000,)
```

In [50]:
```
ans ,arr = kmeans2(ab.astype(float),center,iter=15,missing='warn')
```

In [51]:
```
cluster = []
a = []
for i in range(center):
    cluster.append(a)
    a = []
```

In [52]:
```
arr = np.reshape(arr,(rows,column),order='C')
```

In [53]:
```
img_backup = img.copy()
print (ans)
```

```
[ 23.38354738  81.30227219 148.79786973]
```

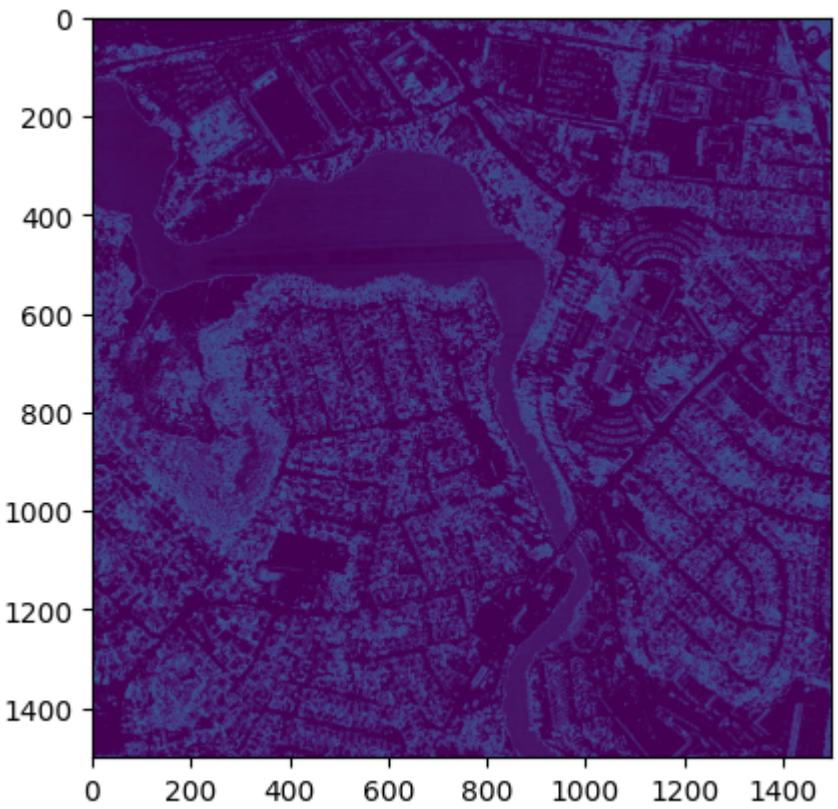
In [54]:
```
for i in range(rows):
    for j in range(column):
        img_backup[i,j] = [0,0,0]
```

In [55]:
```
for z in range(center):
    for x in range(rows):
        for y in range(column):
            if arr[x,y] == z:
                #print z
                img_backup[x,y] = img[x,y]
                #cluster[z].append([x,y])
    cv2.imwrite('%s.jpg'%z,img_backup)
    print ('cluster%s'%z)
```

```
cluster0
cluster1
cluster2
```

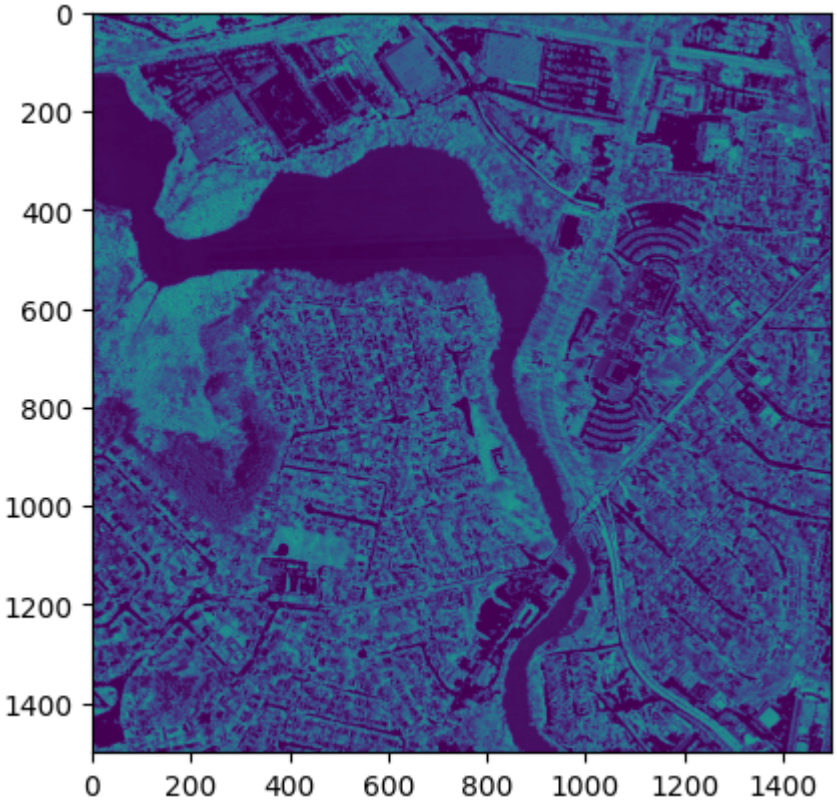
In [56]:
```
img_0 = cv2.imread('./0.jpg')
img_0 = img_0[:,:,0]
plt.imshow(img_0)
```

Out[56]:
```
<matplotlib.image.AxesImage at 0x7e4c71891c00>
```



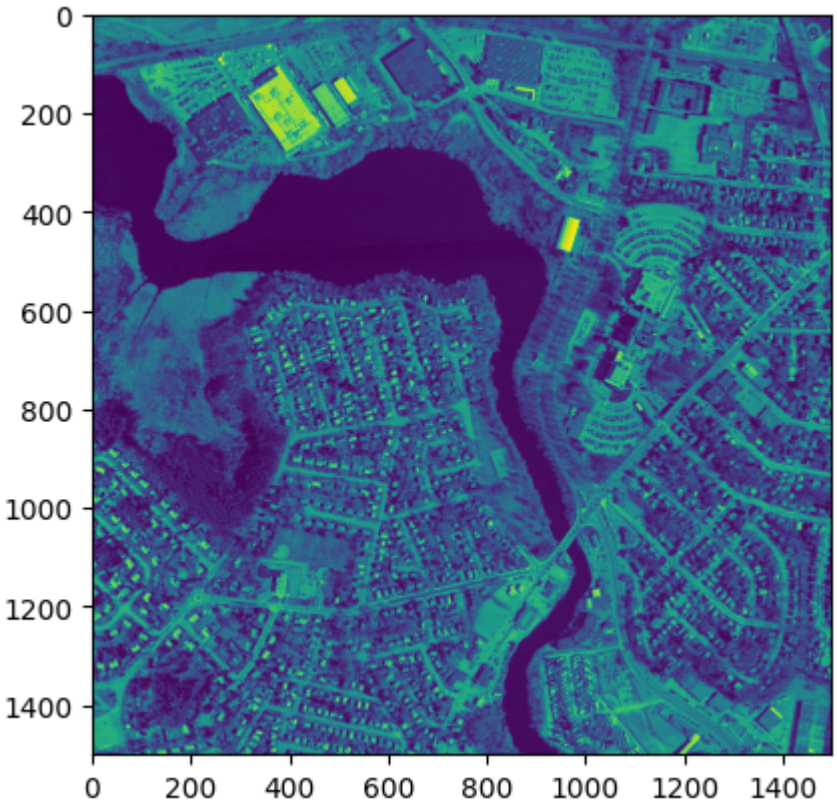
In [57]:
```
img_1 = cv2.imread('./1.jpg')
img_1 = img_1[:,:,0]
plt.imshow(img_1)
```

Out[57]:
```
<matplotlib.image.AxesImage at 0x7e4c71b12d40>
```



In [58]:
```
img_2 = cv2.imread('./2.jpg')
img_2 = img_2[:,:,0]
plt.imshow(img_2)
```

Out[58]:
```
<matplotlib.image.AxesImage at 0x7e4c71a61570>
```



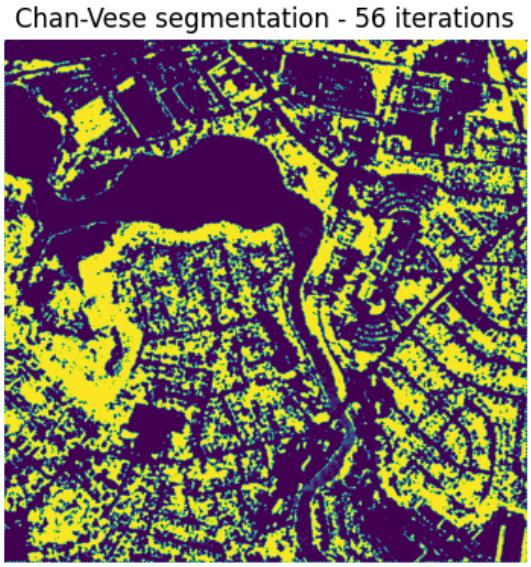
In [59]:
```
img_2.shape
```

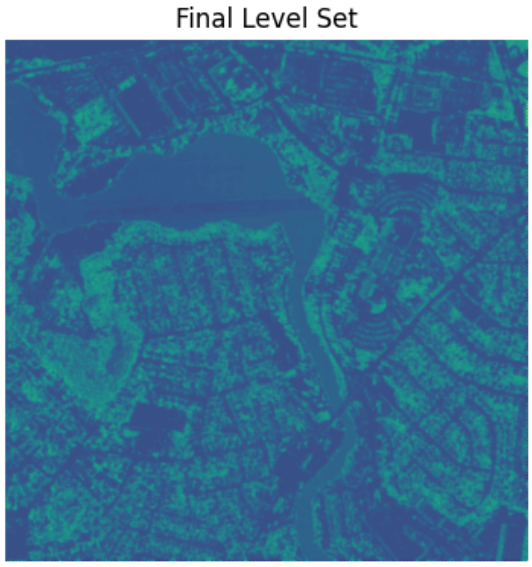
Out[59]:
```
(1500, 1500)
```

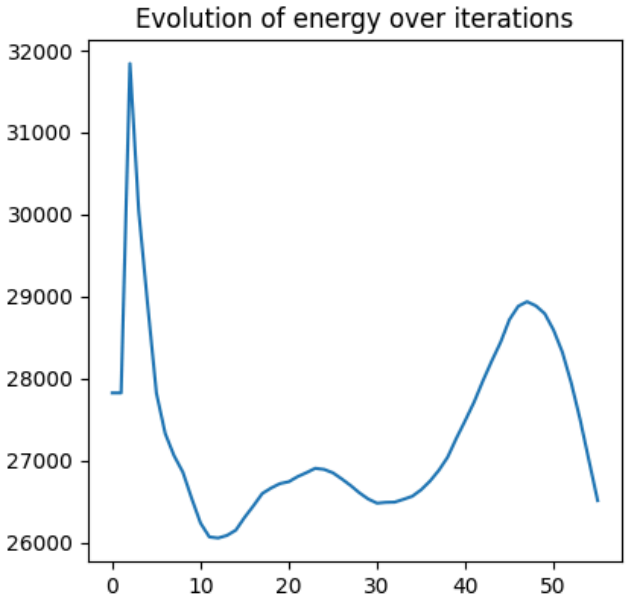
In [60]:
```
import cv2
import numpy
from scipy.cluster.vq import *
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage as ndi

from skimage import data
from skimage.metrics import (adapted_rand_error,
                             variation_of_information)
from skimage.filters import sobel
from skimage.measure import label
from skimage.util import img_as_float
from skimage.feature import canny
from skimage.morphology import remove_small_objects
from skimage.segmentation import (morphological_geodesic_active_contour,
                                  inverse_gaussian_gradient,
                                  watershed,
                                  mark_boundaries)

import matplotlib.pyplot as plt
from skimage import data, img_as_float
from skimage.segmentation import chan_vese
```

In [67]:
```
cv = chan_vese(img_0, mu=0.15, lambda1=1, lambda2=1, tol=1e-3, max_num_iter=107,
               dt=0.5, init_level_set="checkerboard", extended_output=True)

fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.flatten()

ax[0].imshow(img)
ax[0].set_axis_off()
ax[0].set_title("Original Image", fontsize=12)

ax[1].imshow(cv[0])
ax[1].set_axis_off()
title = "Chan-Vese segmentation - {} iterations".format(len(cv[2]))
ax[1].set_title(title, fontsize=12)

ax[2].imshow(cv[1])
ax[2].set_axis_off()
ax[2].set_title("Final Level Set", fontsize=12)

ax[3].plot(cv[2])
ax[3].set_title("Evolution of energy over iterations", fontsize=12)

fig.tight_layout()
plt.show()
```


Original Image


Chan-Vese segmentation - 56 iterations


Final Level Set


Evolution of energy over iterations
32000
31000
30000
29000
28000
27000
26000
0
10
20
30
40
50

In [ ]: