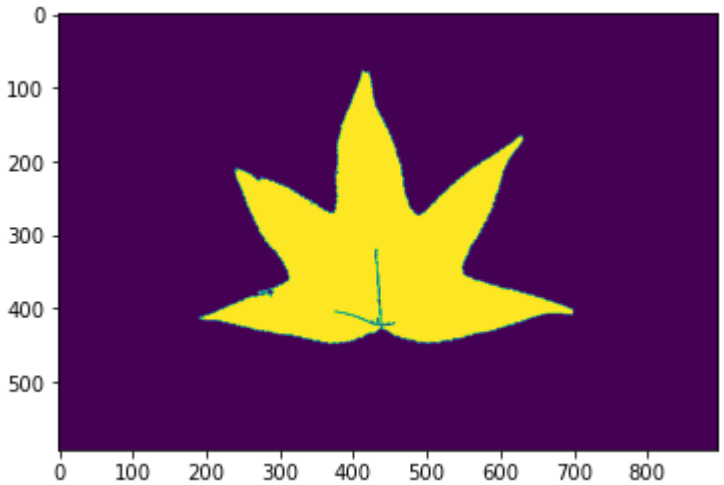
Segmentation Algorithm

In [62]:

```
import skimage.util as util
import skimage.io as io
import numpy as np
import os as os
import skimage.morphology  as  morph
import skimage.segmentation as seg

#TESTS imports
import matplotlib.pyplot as plt
import skimage.measure as measure
import skimage.metrics as metric

def segmentation(I):
    '''
    Segments a leaf image.
    '''
    image = util.img_as_float(I)
    red = I[:,:,0]
    green = I[:,:,1]
    blue = I[:,:,2]

    #Threshold
    newImage = ((green > red) & (green > blue) & (red < 100) & (blue < 100) & (green >
    newImage = morph.remove_small_holes(newImage, area_threshold=128)
    newImage = morph.remove_small_objects(newImage, min_size=256)
    labels = measure.label(newImage, return_num=False)
    result_img = labels == np.argmax(np.bincount(labels.flat, weights=newImage.flat))

    return result_img

#TESTS
img = io.imread('./images/image_0001.png')
result_img = segmentation(img)
plt.figure()
plt.imshow(result_img)
```

Out[62]: <matplotlib.image.AxesImage at 0x2b42a5a9790>



MSD/HD/DSC Measures

In [63]:
```
def MSD(B,G):
    B = np.tile(B, (B.shape[0], 1, 1))
    G = np.tile(G, (G.shape[0], 1, 1))
    G = np.transpose(G, (1,0,2))

    d= (np.bitwise_xor(B, G))
    d = d**2
    d  = d.sum(axis=2)
    d = np.sqrt(d)
    d = d.min(axis=0)
    d = d.sum()
    msd = d/B.shape[0]
    return msd

def HD(B,G):
    B = np.tile(B,(B.shape[0],1,1))
    G = np.tile(G,(G.shape[0],1,1))
    B=np.transpose(B,(1,0,2))
    G=np.transpose(G,(1,0,2))
    distB = (np.bitwise_xor(B,G))
    distG = (np.bitwise_xor(G,B))

    sumB =distB.sum(axis=2)
    sumG = distG.sum(axis=2)

    b=np.sqrt(sumB)
    b=sumB.min(axis=0)
    b =sumB.max()

    g=np.sqrt(sumG)
    g=sumG.min(axis=0)
    g=sumG.max()


    result = np.maximum(b,g)
    return result

def DSC(B,G):
    sets = np.absolute(((B==1) & (G==1))).sum()
    sets = sets *2 / np.absolute(B.sum() + G.sum())
    return sets
```

Validation driver

In [64]:
```
#### Validate ####
images_path = os.path.join('.', 'images')
gt_path = os.path.join('.', 'groundtruth')

msdList = np.empty(0)
hdList= np.empty(0)
dscList=np.empty(0)
dscRecognized = 0
# Iterate over all files in the original images folder
for root, dirs, files in os.walk(images_path):
    for filename in files:
        # ignore files that are not PNG files.
        if filename[-4:] != '.png':
            continue
        images = io.imread('./images/'+ filename)
```

```
        Groundimages = io.imread('./groundtruth/thresh'+ filename)

        binaryImage = segmentation(images)
        label = morph.label(binaryImage,connectivity = 2)
        label_img = seg.find_boundaries(label,connectivity=2,mode='inner')
        Groundimages = util.img_as_bool(Groundimages)
        Groundlabel = morph.label(Groundimages,connectivity = 2)
        Groundlabel_img = seg.find_boundaries(Groundlabel,connectivity=2,mode='inner')

        msdList =  np.append(msdList, MSD(binaryImage,Groundimages))
        hdList = np.append(hdList,HD(binaryImage,Groundimages))
        dscList = np.append(dscList,DSC(binaryImage,Groundimages))

        print("DSC for "+ filename +": " + str(dscList[-1]))
        print("MSD for "+ filename +": " + str(msdList[-1]))
        print("HD for "+ filename +": " + str(hdList[-1]))



for r in dscList:
    if (r > 0.6):
        dscRecognized +=1

print("The mean Dice coefficient was: "+ str(dscList.mean()))
print("The std. deviation of Dice coefficient was: "+ str(dscList.std()))
print("The mean MSD was: "+ str(msdList.mean()))
print("The std. deviation of MSD was: "+ str(msdList.std()))
print("The mean HD  was: "+ str(hdList.mean()))
print("The std. deviation of HD was: "+ str(hdList.std()))
print(((dscRecognized)/dscList.shape[0])*100, "%" + " of leaves were recognized.")
```

```
DSC for image_0001.png: 0.9931080141679632
MSD for image_0001.png: 0.7551934505523868
HD for image_0001.png: 21.0
DSC for image_0002.png: 0.991004941895695
MSD for image_0002.png: 0.6938349258820639
HD for image_0002.png: 15.0
DSC for image_0005.png: 0.9945545579689842
MSD for image_0005.png: 0.6633201794627971
HD for image_0005.png: 17.0
DSC for image_0007.png: 0.6735185457699844
MSD for image_0007.png: 4.384948771741827
HD for image_0007.png: 301.0
DSC for image_0009.png: 0.9629119527791441
MSD for image_0009.png: 1.768532325937548
HD for image_0009.png: 92.0
DSC for image_0010.png: 0.6289430133230806
MSD for image_0010.png: 4.3375030645496375
HD for image_0010.png: 350.0
DSC for image_0011.png: 0.9936929470336866
MSD for image_0011.png: 0.5266627477468985
HD for image_0011.png: 12.0
DSC for image_0015.png: 0.9921791344185857
MSD for image_0015.png: 0.7688728059418946
HD for image_0015.png: 24.0
DSC for image_0018.png: 0.9915570440740128
MSD for image_0018.png: 0.8904735161651973
HD for image_0018.png: 32.0
DSC for image_0019.png: 0.9932562046491678
MSD for image_0019.png: 0.7489579888750161
HD for image_0019.png: 18.0
DSC for image_0078.png: 0.6504684390772989
```

```
MSD for image_0078.png: 4.295768584067295
HD for image_0078.png: 224.0
DSC for image_0080.png: 0.986083518953304
MSD for image_0080.png: 1.363578999551431
HD for image_0080.png: 26.0
DSC for image_0089.png: 0.986967924694529
MSD for image_0089.png: 0.9895024029955148
HD for image_0089.png: 28.0
DSC for image_0090.png: 0.9825737459916714
MSD for image_0090.png: 1.0167619864459545
HD for image_0090.png: 51.0
DSC for image_0099.png: 0.9824784963364128
MSD for image_0099.png: 1.0013963589300219
HD for image_0099.png: 25.0
DSC for image_0100.png: 0.9880749410470807
MSD for image_0100.png: 1.1719883246907676
HD for image_0100.png: 30.0
DSC for image_0104.png: 0.9918343557220103
MSD for image_0104.png: 0.7150097808498833
HD for image_0104.png: 23.0
DSC for image_0105.png: 0.9931193688645839
MSD for image_0105.png: 0.6439702004186355
HD for image_0105.png: 25.0
DSC for image_0110.png: 0.992633471817449
MSD for image_0110.png: 0.7377578578671727
HD for image_0110.png: 16.0
DSC for image_0113.png: 0.9807744923358225
MSD for image_0113.png: 1.20223389881107
HD for image_0113.png: 43.0
DSC for image_0132.png: 0.966453659641967
MSD for image_0132.png: 1.326646681485802
HD for image_0132.png: 61.0
DSC for image_0160.png: 0.9687352535692145
MSD for image_0160.png: 1.1583975266933648
HD for image_0160.png: 70.0
DSC for image_0161.png: 0.9581668585359246
MSD for image_0161.png: 1.6243944915928827
HD for image_0161.png: 49.0
DSC for image_0162.png: 0.5409519236540423
MSD for image_0162.png: 3.1231871479653446
HD for image_0162.png: 243.0
DSC for image_0163.png: 0.9279492418539628
MSD for image_0163.png: 2.0514604059615813
HD for image_0163.png: 76.0
DSC for image_0165.png: 0.8976808655125259
MSD for image_0165.png: 2.6469200303004965
HD for image_0165.png: 83.0
DSC for image_0166.png: 0.9931421400361717
MSD for image_0166.png: 0.5166196119721587
HD for image_0166.png: 25.0
DSC for image_0171.png: 0.9892184412184079
MSD for image_0171.png: 0.6462920418817801
HD for image_0171.png: 26.0
DSC for image_0174.png: 0.9860035820216627
MSD for image_0174.png: 0.6597137717372277
HD for image_0174.png: 37.0
DSC for image_0175.png: 0.9903783916169551
MSD for image_0175.png: 0.626393282185138
HD for image_0175.png: 21.0
The mean Dice coefficient was: 0.9322805156193767
The std. deviation of Dice coefficient was: 0.1242261009347081
The mean MSD was: 1.4352097721086268
The std. deviation of MSD was: 1.1391263286550957
The mean HD  was: 68.8
```

```
The std. deviation of HD was: 87.07330245258875
96.66666666666667 % of leaves were recognized.
```

Display Examples

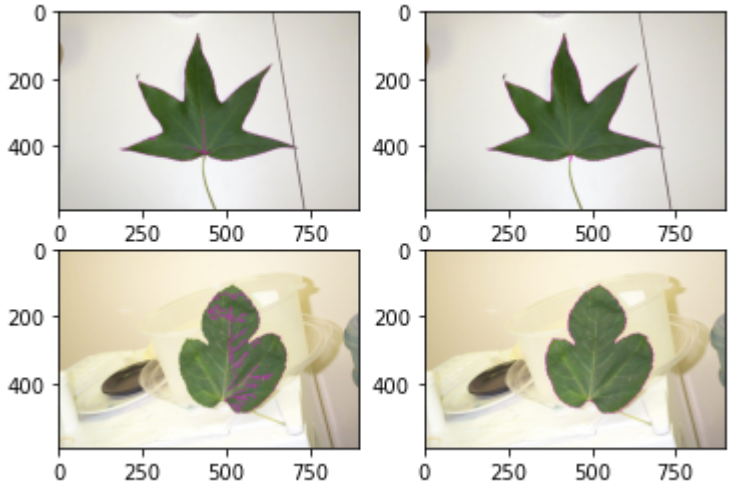
In [65]:

```
import matplotlib.pyplot as plt

#good result
image = io.imread('./images/image_0001.png')
binary = segmentation(image)
L = morph.label(binary , connectivity =2)
boundaries = seg.find_boundaries(L, connectivity =2,mode='inner')
image_with_boundaries = seg.mark_boundaries(image , L,color =(1 ,0 ,1))

plt.figure()
plt.subplots(2,2)
plt.subplot(2,2,1)
plt.imshow(image_with_boundaries)

Groundimage = io.imread('./groundtruth/threshimage_0001.png')
L = morph.label(Groundimage , connectivity =2)
boundaries = seg.find_boundaries(L, connectivity =2,mode='inner')
image_with_boundaries = seg.mark_boundaries(image , L,color =(1 ,0 ,1))

plt.subplot(2,2,2)
plt.imshow(image_with_boundaries)



#Bad result
image = io.imread('./images/image_0078.png')
binary = segmentation(image)
L = morph.label(binary , connectivity =2)
boundaries = seg.find_boundaries(L, connectivity =2,mode='inner')
image_with_boundaries = seg.mark_boundaries(image , L,color =(1 ,0 ,1))

plt.subplot(2,2,3)
plt.imshow(image_with_boundaries)

Groundimage = io.imread('./groundtruth/threshimage_0078.png')
L = morph.label(Groundimage , connectivity =2)
boundaries = seg.find_boundaries(L, connectivity =2,mode='inner')
image_with_boundaries = seg.mark_boundaries(image , L,color =(1 ,0 ,1))

plt.subplot(2,2,4)
plt.imshow(image_with_boundaries)
```

Out[65]:

```
<matplotlib.image.AxesImage at 0x2b42bc07790>
```

```
<Figure size 432x288 with 0 Axes>
```



In [9]:

In [ ]: