HOMEWORK 9

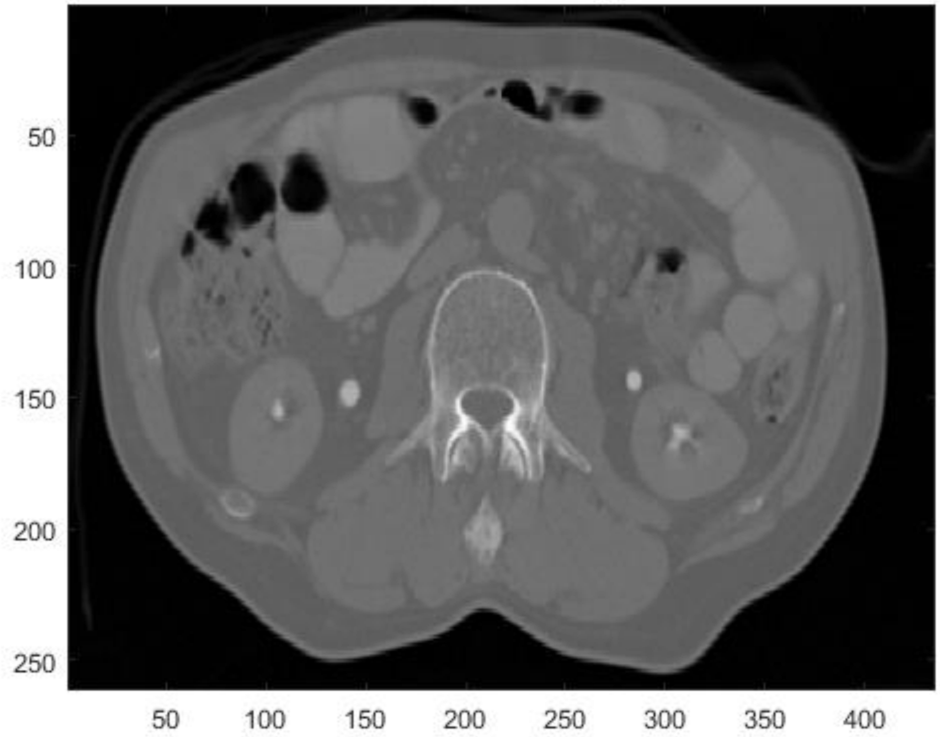
SOLUTIONS

1. The result of `Threshold_6.m` is a binary image with pixel intensities of 0 and 64; by using image algebra techniques such as the ones introduced in Chap. 3 (i.e., using an intensity transform function), it is possible to generate a greyscale image of the segmented ROI. Modify `Threshold_6.m` accordingly.

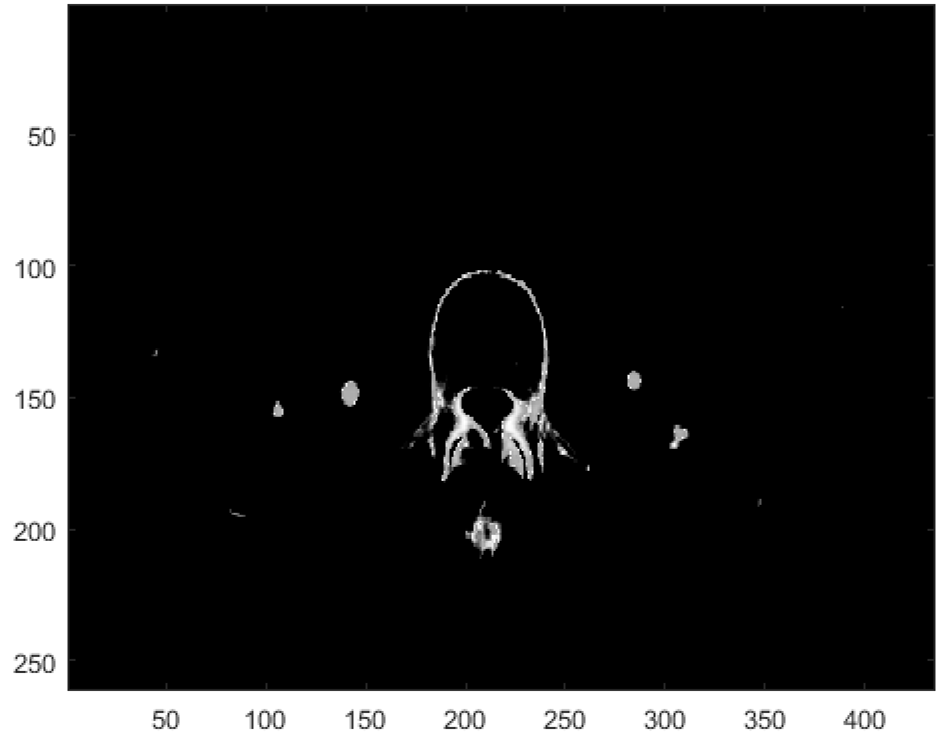
Solution:

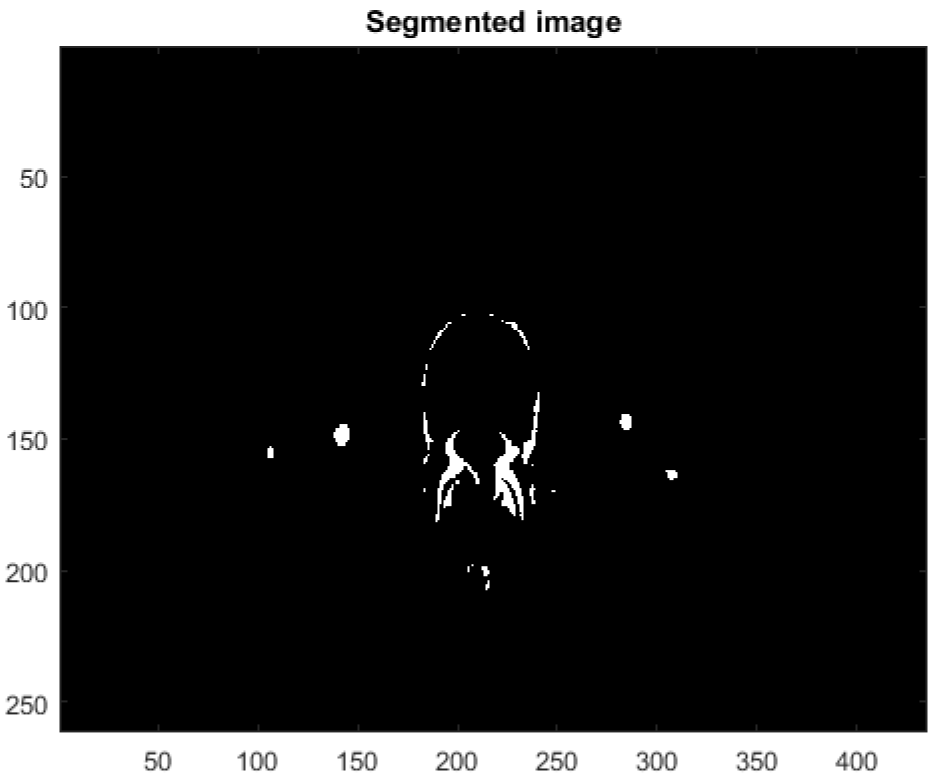
```
clc

clear all
close all
img = imread('ABD_CT.jpg');
img1=single(img);
imgt=(256.*(img1-min(img1(:))))/(max(img1(:)-min(img1(:))));
colormap(gray(256))
figure(1)
image(imgt),title('transformed image')
segimg = zeros(261,435);
thre_low=140;
thre_high=160;
for i=1:261
  for j=1:435
     if imgt(i,j)>= thre_high
       segimg(i,j)=64;
     else if imgt(i,j) <= thre_low
        imgt(i,j) = 0;
         else
        imgt(i,j)=(imgt(i,j)-thre_low)*256/(thre_high-thre_low);
         end
     end
  end
end
figure(2)
colormap(gray(256)); image(imgt)
title('ROI in transformed image')
figure(3)
colormap(gray)
image(segimg)
title('Segmented image')
```

transformed image



ROI in transformed image



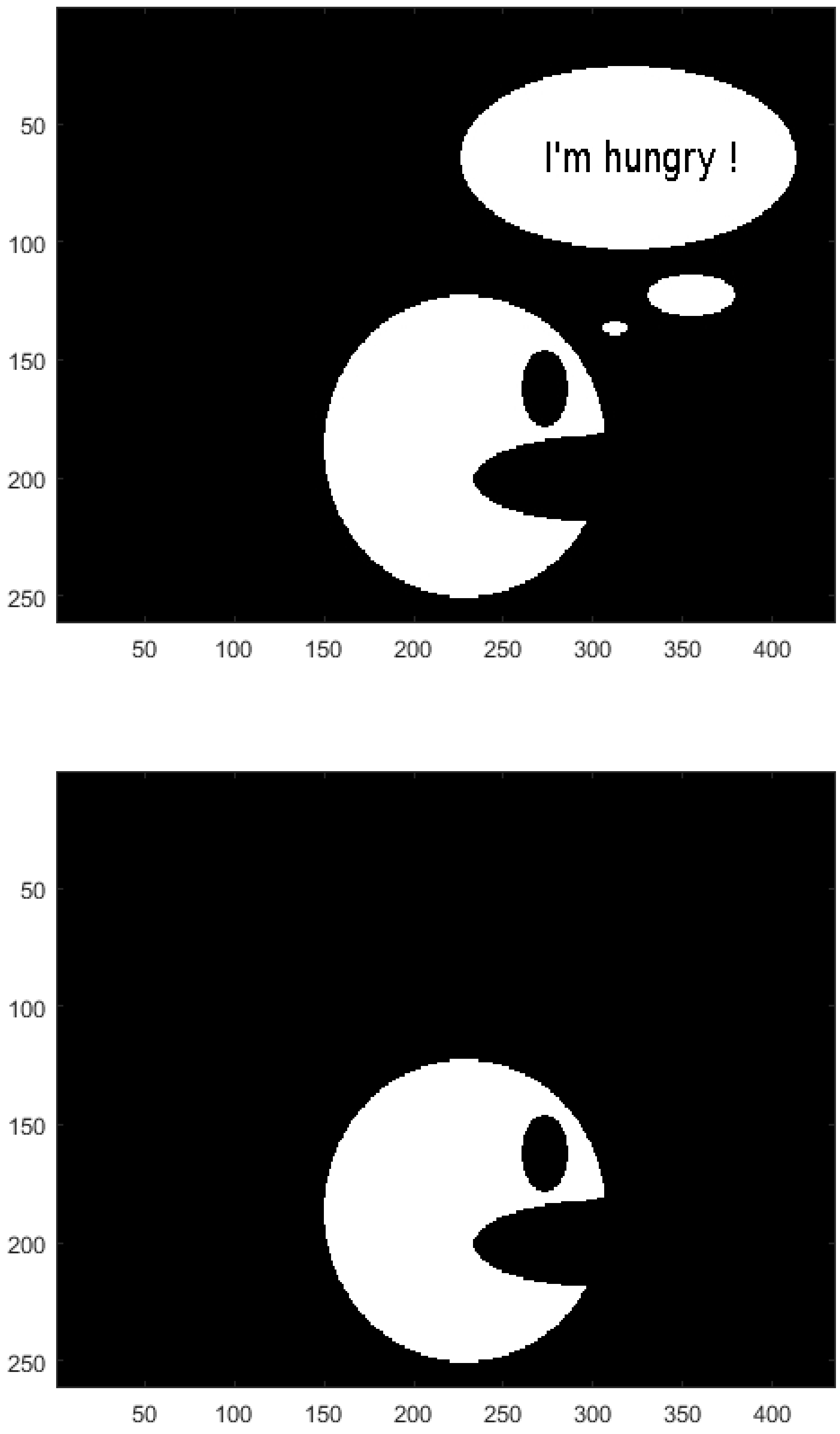


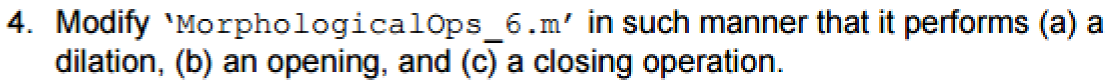
2. In class, we talked about region growing using ‘`RegionGrowing_FourConnected_6.m`’. Please shorten this code or m-file by using functions.

Solution:

```
function img_out=regrow(img)
k=uint8(imread(img));
figure(1)
colormap(gray(256));
image(k)
seedmask=zeros(size(k));
seedmask(130,210)=64;
seedint=k(130,210);
seedrangemin=seedint-20;
if seedrangemin < 0
    seedrangemin=0;
end
seedrangemax=seedint+20;
if seedrangemax > 255
    seedrangemax=255;
end
oldseeds=1;
newseeds=0;
while newseeds ~= oldseeds
    oldseeds=newseeds;
    newseeds=0;
    for i=2:260
        for j=2:434
            if seedmask(i,j)>0
                intensi = k((i-1),j);
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask((i-1),j)=64;
                end
                intensi = k((i+1),j);
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask((i+1),j)=64;
                end
                intensi = k(i,(j-1));
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask(i,(j-1))=64;
                end
                intensi = k(i,(j+1));
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask(i,(j+1))=64;
                end
            end
        end
    end
end
figure(2)
colormap(gray)
img_out=seedmask;
image(img_out)
```

```
clc

clear all
close all
img=regrow('PACMAN_THINKS.jpg');
```


I'm hungry !
50
100
150
200
250
50
100
150
200
250
300
350
400
50
100
150
200
250
50
100
150
200
250
300
350
400



4. Modify 'MorphologicalOps_6.m' in such manner that it performs (a) a dilation, (b) an opening, and (c) a closing operation.

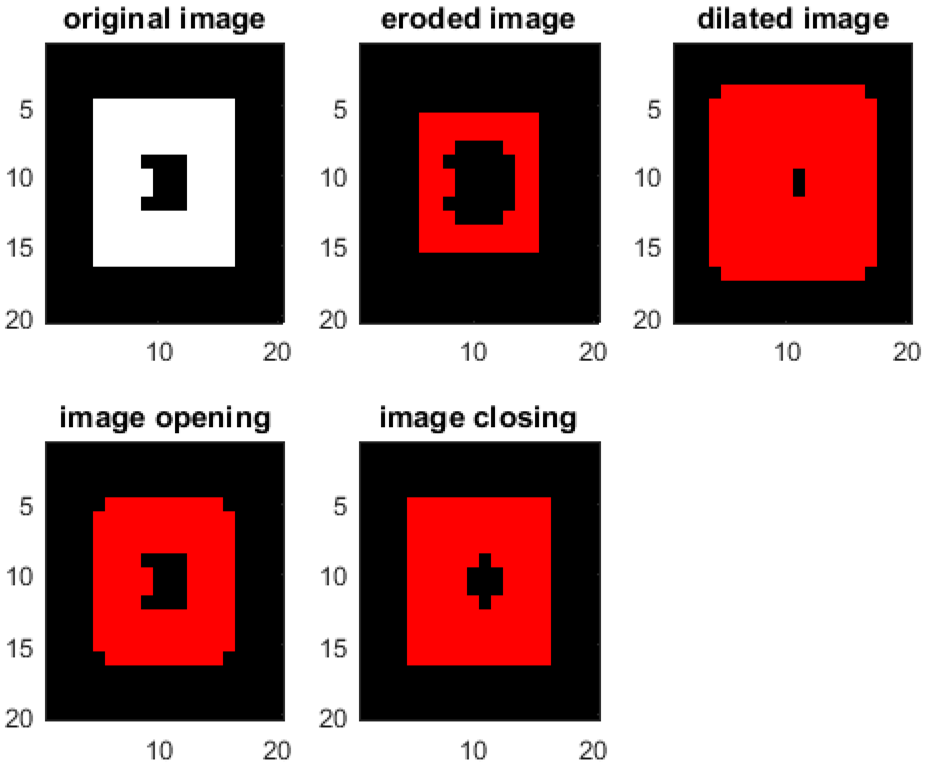
Solution:

```
function [erodedimage]=erosion(img,modifyimg)
for i=2:19
    for j=2:19
        if img(i,j)>0
            modifyimg(i,j)=256;
            if (img((i-1),j) == 0) || (img((i+1),j) == 0) || (img(i,(j-1)) == 0) ||
(img(i,(j+1)) == 0)
                modifyimg(i,j)=0;
            end
        end
    end
end
erodedimage=modifyimg;
```

```
function [dilatedimage]=dilation(img,modifyimg)
for i=2:19
    for j=2:19
        if (img((i-1),j) ~= 0) || (img((i+1),j) ~= 0) || (img(i,(j-1)) ~= 0) ||
(img(i,(j+1)) ~= 0)
                modifyimg(i,j)=256;
        end
    end
end
dilatedimage=modifyimg;
```

```
clc

clear all
close all
colormap(gray(256))
% Erosion
img=single(imread('MorphologicalSquare.jpg'));
subplot(2,3,1)
% colormap(gray)
image(img)
title('original image')
modifyimg=zeros(size(img));
erodedimage=erosion(img,modifyimg);
subplot(2,3,2)
image(erodedimage)
title('eroded image')
```

```
dilatedimage=dilation(img,modifyimg);
subplot(2,3,3)
image(dilatedimage)
title('dilated image')
dilatedimage1=dilation(erodedimage,modifyimg);
subplot(2,3,4)
image(dilatedimage1)
title('image opening')
erodedimage=erosion(dilatedimage,modifyimg);
subplot(2,3,5)
image(erodedimage)
title('image closing')
```

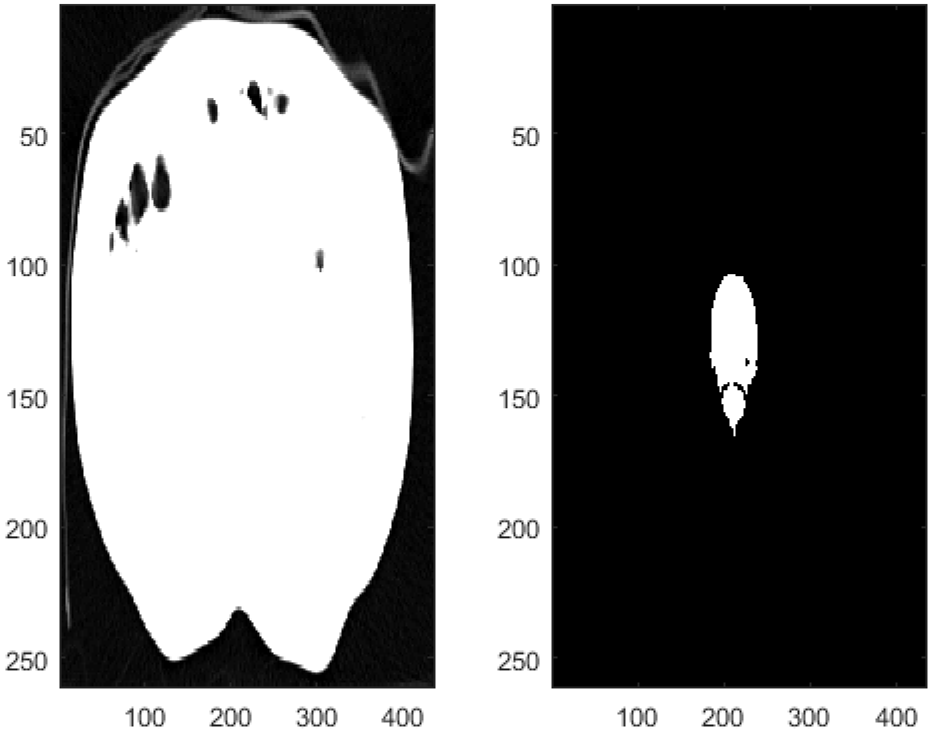


5. Threshold 'ABD_CT.jpg' and perform some morphological operations to show/segment vertebral region. (You can compare your results with Fig. 6.3.)

Solution:

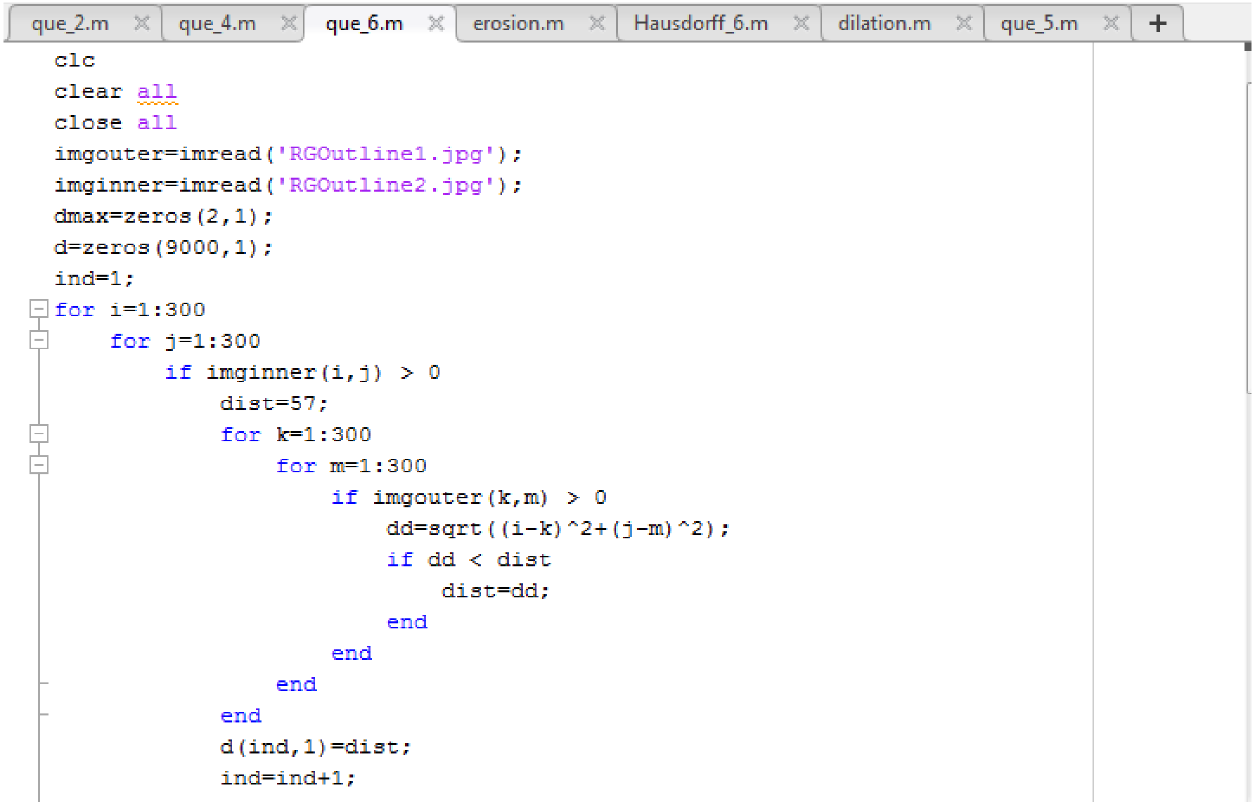
```
clc

clear all
close all
img = uint8(imread('ABD_CT.jpg'));
```

```
subplot(1,2,1)
colormap(gray)
image(img)
modifyimg=zeros(size(img));
seedmask=zeros(261,435);
seedmask(129,209)=64;
seedintensity=img(129,209);
seedrangemin=seedintensity-20;
if seedrangemin < 0
    seedrangemin=0;
end
seedrangemax=seedintensity+20;
if seedrangemax > 255
    seedrangemax=255;
end
oldseeds=1;
newseeds=0;
while newseeds ~= oldseeds
    oldseeds=newseeds;
    newseeds=0;
    for i=2:260
        for j=2:434
            if seedmask(i,j)>0
                intensi = img((i-1),j);
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask((i-1),j)=64;
                end
                intensi = img((i+1),j);
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask((i+1),j)=64;
                end
                intensi = img(i,(j-1));
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask(i,(j-1))=64;
                end
                intensi = img(i,(j+1));
                if (intensi >= seedrangemin) & (intensi <= seedrangemax)
                    newseeds = newseeds + 1;
                    seedmask(i,(j+1))=64;
                end
            end
        end
    end
end
subplot(1,2,2)
colormap(gray)
image(seedmask)
```

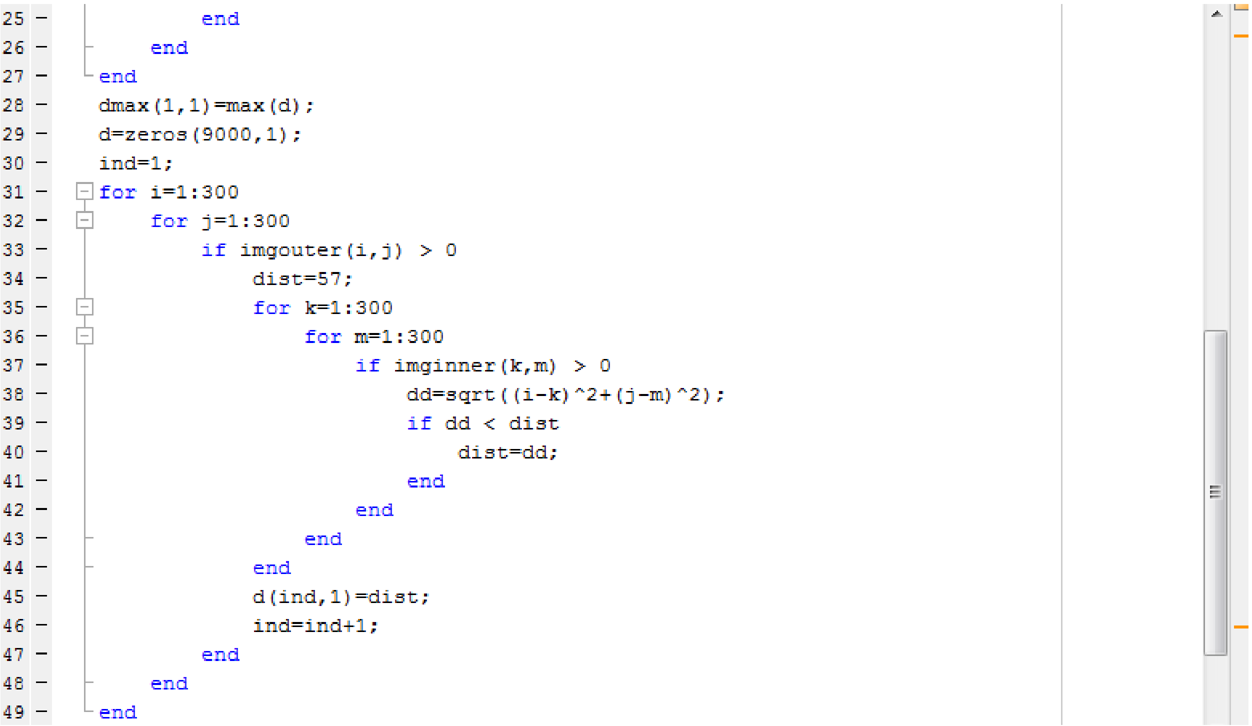


6. In the `LessonData` or Blackboard class folder, you will find the outline of the two segmentation results from Figure 6.15. These are named `'RGOutline1.jpg'` and `'RGOutline2.jpg'`. Compute the Hausdorff-distance, and compare the two values in `dmax`.

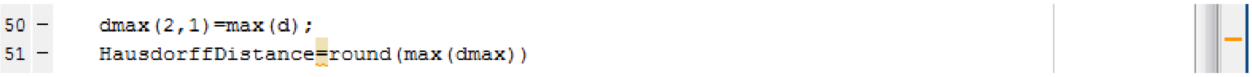
Solution:



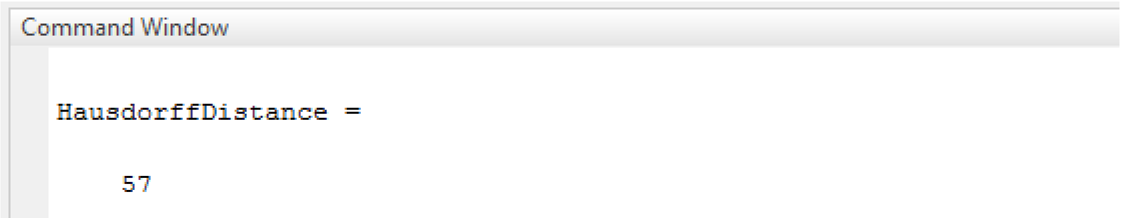
```
clc
clear all
close all
imgouter=imread('RGOutline1.jpg');
imginner=imread('RGOutline2.jpg');
dmax=zeros(2,1);
d=zeros(9000,1);
ind=1;
for i=1:300
    for j=1:300
        if imginner(i,j) > 0
            dist=57;
            for k=1:300
                for m=1:300
                    if imgouter(k,m) > 0
                        dd=sqrt((i-k)^2+(j-m)^2);
                        if dd < dist
                            dist=dd;
                        end
                    end
                end
            end
            d(ind,1)=dist;
            ind=ind+1;
```



```
        end
    end
end
dmax(1,1)=max(d);
d=zeros(9000,1);
ind=1;
for i=1:300
    for j=1:300
        if imgouter(i,j) > 0
            dist=57;
            for k=1:300
                for m=1:300
                    if imginner(k,m) > 0
                        dd=sqrt((i-k)^2+(j-m)^2);
                        if dd < dist
                            dist=dd;
                        end
                    end
                end
            end
            d(ind,1)=dist;
            ind=ind+1;
        end
    end
end
```



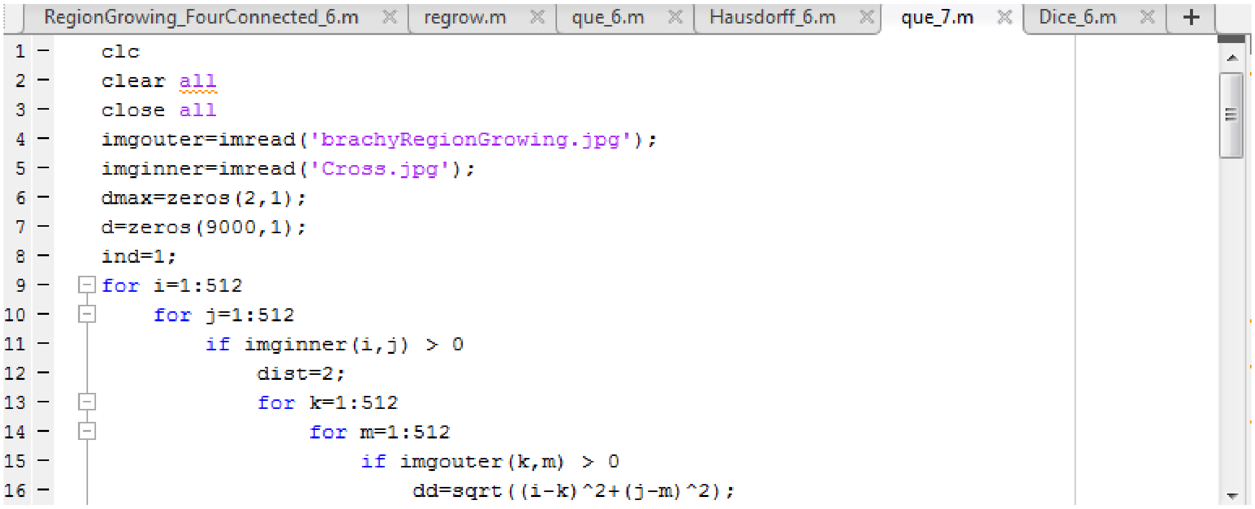
```
dmax(2,1)=max(d);
HausdorffDistance=round(max(dmax))
```



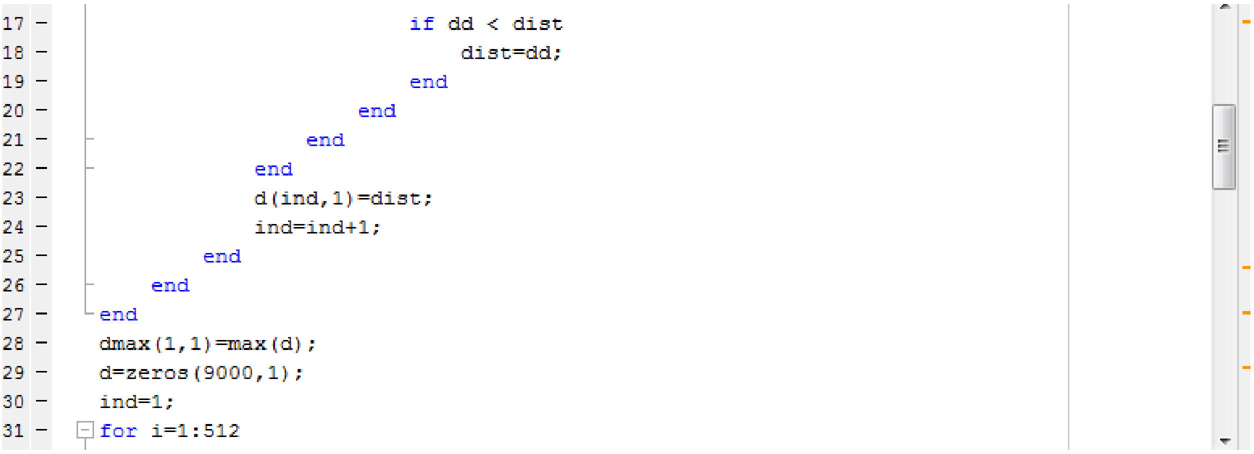
```
HausdorffDistance =

    57
```

7. Compute the Dice-coefficient and the Hausdorff-distance for the images ‘brachyRegionGrowing.jpg’ (Figure 6.26) and ‘Cross.jpg’, which is the output of the ‘BrachyCleanup_6.m’ script. Both images can be found in the LessonData or Blackboard class folder.

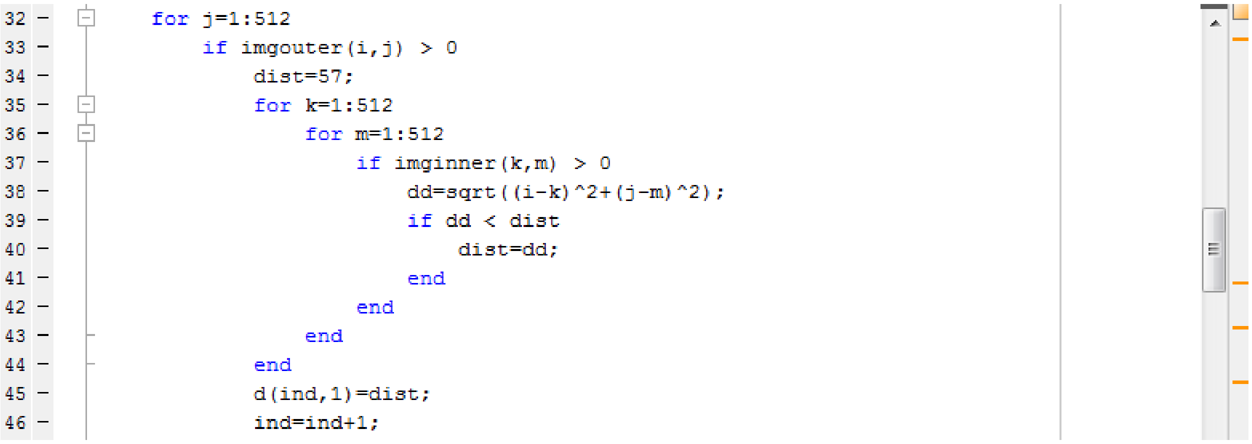
Solution:



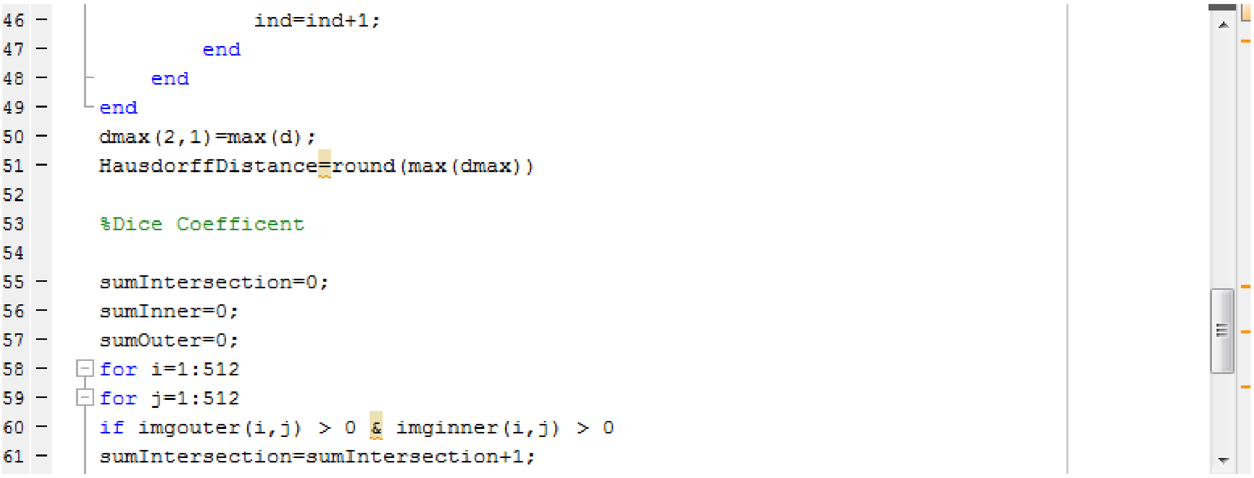
```
clc
clear all
close all
imgouter=imread('brachyRegionGrowing.jpg');
imginner=imread('Cross.jpg');
dmax=zeros(2,1);
d=zeros(9000,1);
ind=1;
for i=1:512
    for j=1:512
        if imginner(i,j) > 0
            dist=2;
            for k=1:512
                for m=1:512
                    if imgouter(k,m) > 0
                        dd=sqrt((i-k)^2+(j-m)^2);
```



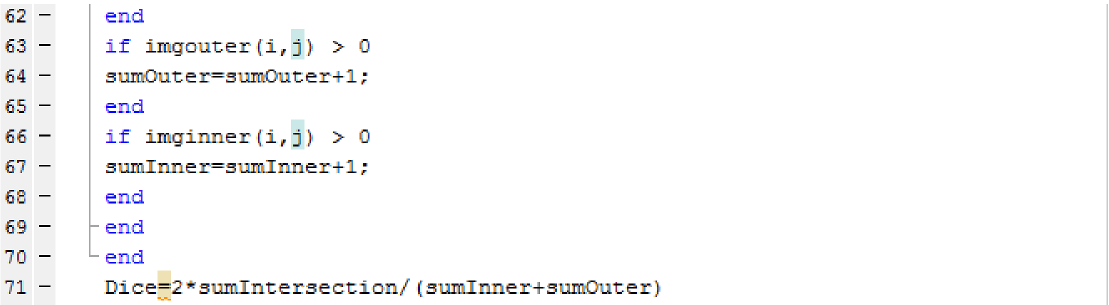
```
                        if dd < dist
                            dist=dd;
                        end
                    end
                end
            end
            d(ind,1)=dist;
            ind=ind+1;
        end
    end
end
dmax(1,1)=max(d);
d=zeros(9000,1);
ind=1;
for i=1:512
```



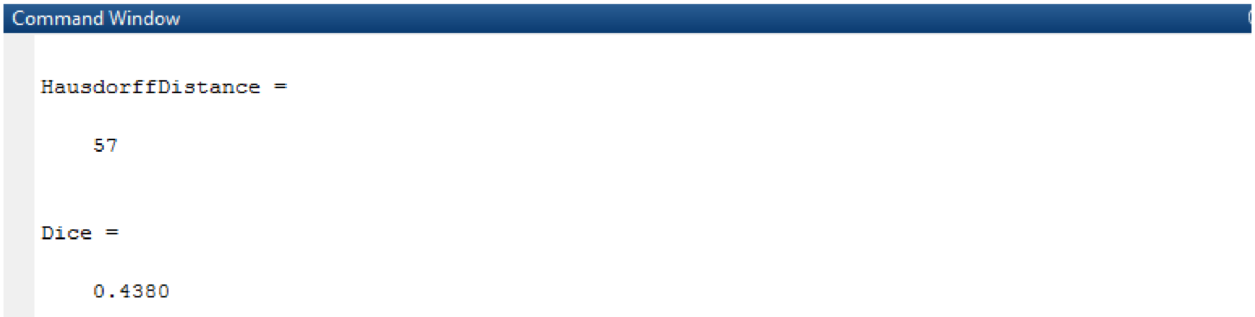
```
    for j=1:512
        if imgouter(i,j) > 0
            dist=57;
            for k=1:512
                for m=1:512
                    if imginner(k,m) > 0
                        dd=sqrt((i-k)^2+(j-m)^2);
                        if dd < dist
                            dist=dd;
                        end
                    end
                end
            end
            d(ind,1)=dist;
            ind=ind+1;
```



```
            ind=ind+1;
        end
    end
end
dmax(2,1)=max(d);
HausdorffDistance=round(max(dmax))

%Dice Coefficent

sumIntersection=0;
sumInner=0;
sumOuter=0;
for i=1:512
for j=1:512
if imgouter(i,j) > 0 & imginner(i,j) > 0
sumIntersection=sumIntersection+1;
```



```
end
if imgouter(i,j) > 0
sumOuter=sumOuter+1;
end
if imginner(i,j) > 0
sumInner=sumInner+1;
end
end
end
Dice=2*sumIntersection/(sumInner+sumOuter)
```



```
HausdorffDistance =

    57


Dice =

    0.4380
```