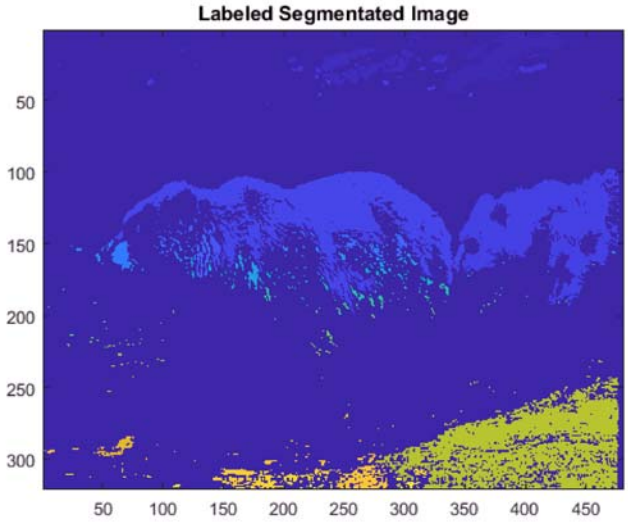
Part(b)

In this part, performance of three different functions, which are designed to calculate GCE and LCE, are compared. Please note that "compute_GCE_LCE functions" are presented in apendix section.

```
%read the image
a=double(imread([cd '\BSDS300\images\train\100075.jpg']));
%turn the image into grayscale
a_gray=(0.2125*a(:,:,1)+0.7154*a(:,:,2)+0.0721*a(:,:,3))/255;
%next three lines are for thresholding
b=a_gray;
b(b>127/255)=1;
b(b<=127/255)=0;
figure,imshow(b),title('Image Thresholded at 127')
```


```
%labeling segments in threshold image
Seg=bwlabel(b')';
figure,imagesc(Seg),title('Labeled Segmentated Image')
```




```
%read the ground truth image
GT=double(imread([cd '\mask\gray\train\100075.png']));
%computing GCE and LCE using loopy function
tic
[GCE,LCE]=compute_GCE_LCE_loopy(Seg,GT);
toc
```

```
Elapsed time is 438.782674 seconds.
```

```
disp(['GCE=',num2str(GCE)])
```

```
GCE=0.28648
```

```
disp(['LCE=',num2str(LCE)])
```

```
LCE=0.15609
```

```
%computing GCE and LCE using sample optimized function posted on canvas
tic
[GCE1,LCE1]=compute_GCE_LCE_sample(Seg,GT);
toc
```

```
Elapsed time is 20.359284 seconds.
```

```
disp(['GCE=',num2str(GCE1)])
```

```
GCE=0.28648
```

```
disp(['LCE=',num2str(LCE1)])
```

```
LCE=0.15609
```

```
%computing GCE and LCE using optimized function
tic
[GCE2,LCE2]=compute_GCE_LCE(Seg,GT);
toc
```

```
Elapsed time is 21.223063 seconds.
```

```
disp(['GCE=',num2str(GCE2)])
```

```
GCE=0.28648
```

```
disp(['LCE=',num2str(LCE2)])
```

```
LCE=0.15609
```

Appendix 1

In this part, two functions that are designed to calculate GCE and LCE are presented.

```
function [GCE,LCE]=compute_GCE_LCE_loopy(Seg,GT)
%in this part of the function, two segmentations are separated and stored in 3D matrices, in which each layer contains only one segment
L1=Seg+1;
n1=numel(unique(L1));
L2=reshape(1:n1,1,1,[]);
L2=repmat(L2,size(L1));
L3=repmat(L1,1,1,n1);
L4=L3-L2;
L4(L4~=0)=1;
L4=abs(L4-1);

mL1=GT+1;
n2=numel(unique(mL1));
mL2=reshape(unique(mL1),1,1,[]);
mL2=repmat(mL2,size(mL1));
mL3=repmat(mL1,1,1,n2);
mL4=mL3-mL2;
mL4(mL4~=0)=1;
mL4=abs(mL4-1);
mL2=reshape(1:n2,1,1,[]);
mL2=repmat(mL2,size(mL1));
mL1=sum(mL4.*mL2,3);
%in this section, empty matrices are defined to store GCE and LCE values
[m,n]=size(L1);
LCE=0;
GCE=0;
den1=sum(sum(L4));
den1=den1(:);
den2=sum(sum(mL4));
den2=den2(:);
%in this section, function loops over every pixel in the segmentation and calculates ES1S2 and ES2S1
for i=1:m
    for j=1:n
        ES1S2=sum(sum((L4(:,:,L1(i,j)).*abs(mL4(:,:,mL1(i,j))-1))))/den1(L1(i,j));
        ES2S1=sum(sum((mL4(:,:,mL1(i,j)).*abs(L4(:,:,L1(i,j))-1))))/den2(mL1(i,j));
        LCE=LCE+min([ES1S2,ES2S1]);
        GCE=GCE+[ES1S2,ES2S1];
    end
end
LCE=LCE/numel(L1);
GCE=min(GCE)/numel(L1);

end
```

Appendix 2

```
function [GCE,LCE]=compute_GCE_LCE(Seg,GT)
%in this part of the function, two segmentations are separated and stored in 3D matrices, in which each layer contains only one segment
L1=Seg+1;
n1=numel(unique(L1));
L2=reshape(1:n1,1,1,[]);
L2=repmat(L2,size(L1));
L3=repmat(L1,1,1,n1);
L4=L3-L2;
L4(L4~=0)=1;
L4=abs(L4-1);

mL1=GT+1;
n2=numel(unique(mL1));
mL2=reshape(unique(mL1),1,1,[]);
mL2=repmat(mL2,size(mL1));
mL3=repmat(mL1,1,1,n2);
mL4=mL3-mL2;
mL4(mL4~=0)=1;
mL4=abs(mL4-1);

%in this section, empty matrices are defined to store GCE and LCE values
[~,~,m]=size(L4);
[~,~,n]=size(mL4);
LCE=zeros(2,m*n);
GCE1=0;
GCE2=0;
den1=sum(sum(L4));
den1=den1(:);
den2=sum(sum(mL4));
den2=den2(:);
%in this part, function loops over every possible combination of segments from two segmentations
for i=1:m
    for j=1:n
        %np is number of pixels in intersection of two segments. obviously we are interested in combination of segments that have non-empty intersections
        np=L4(:,:,i)+mL4(:,:,j);
        np(np~=2)=0;
        np=sum(sum(np/2));
        if (np~=0)
            %Num1 and Num2 are numerators in calculating ES1S2 and ES2S1
            Num1=sum(sum(L4(:,:,i)))-np;
            if Num1~=0
                GCE1=GCE1+np*(Num1/den1(i));
                LCE(1,(i-1)*n+j)=np*(Num1/den1(i));
            end
            Num2=sum(sum(mL4(:,:,j)))-np;
            if ( Num2~=0 )
                GCE2=GCE2+np*(Num2/den2(j));
                LCE(2,(i-1)*n+j)=np*(Num2/den2(j));
            end
        end
    end
end
LCE=sum(min(LCE))/numel(L1);
GCE=min([GCE1,GCE2])/numel(L1);

end
```