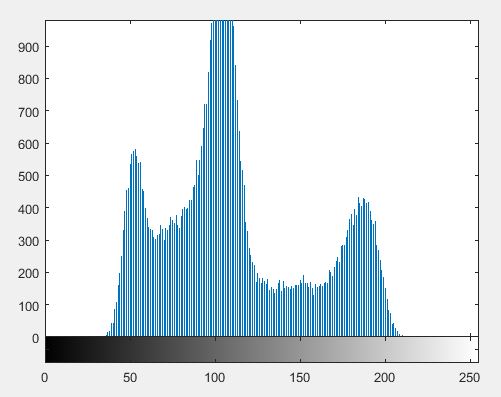
1) Thresholding by histogram Analysis:

Histogram:



Output:



Code:

I=imread('rice.JPG');

J=rgb2gray(I);

figure,imshow(J);

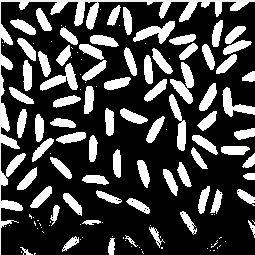
figure,imhist(J);

T=130;

It=im2bw(J,T/255);

figure,imshow(It);

2) Using graythresh function:



I=imread('rice.JPG');

J=rgb2gray(I);

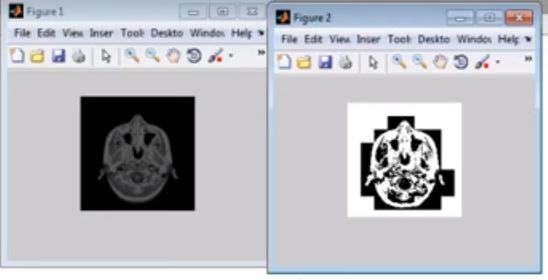
figure,imshow(J);

figure,imhist(J);

T=graythresh(J);

It=im2bw(J,T);

figure,imshow(It);

3) Adaptive Threshold:

I=imread('rice.JPG');

figure,imshow(I);

imshow(I);

it=blkproc(I,[15 15],@adaptt);

figure,imshow(it);

function [ y ]=adaptt(x)

if std2(x) <1

y=ones(size(x,1),size(x,2));

else

y=im2bw(x,graythresh(x));

end

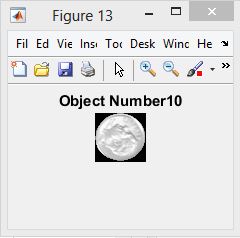
//Coins

The Otsu’s method has been implemented in an application where an image consisting of several coins is segmented to show the coins as separate objects.

Input Image:



Output Image:



I=imread('coins.png');

figure,imshow(I);

J=I;

T=graythresh(J);

%to apply calculate threshold by otsu method

K=im2bw(J,T); %converts the image to binary

figure,imshow(K);

C=imfill(K,'holes'); %to completely fill the holes

figure,imshow(C);

label=bwlabel(C); %labels the independent objects in an image

for j=1:max(max(label))

[row,col]=find(label==j);

len=max(row)-min(row)+2;

br=max(col)-min(col)+2;

target=uint8(zeros([len br]));

sy=min(col)-1;

sx=min(row)-1;

for i=1:size(row,1);

x=row(i,1)-sx;

y=col(i,1)-sy;

target(x,y)=I(row(i,1),col(i,1));

end

t=strcat('Object Number',num2str(j));

figure,imshow(target);title(t);

end