

Ee421 Image processıng

**Mini Project 1 : Coin Counting**



17 Kasım 2014

**NURAY GÜL 130505009**

**M.CAN ALPTEKİN 120505006**

**INTRODUCTION:**

This project is aim to count coins on a captured image by using image processing methods. We will try to find number of the coins and their total values. The finding numbers of the coins is required the labeling and segmentation methods. So we should prepare our images for segmentation by doing preprocessing operations. After these processes we can get labels and objects. Then there would be so many methods to classify the coins but we prefer to use their areas in this project. The areas of the most coins are not same even if they are same kind of coin but their values are close to each other. So we use this feature to define their types. As a result, after finding the numbers and types of the coins we can calculate the total values of them.

**PROCEDURE:**

1. **Reading Image**

Firstly, we use imread to read the captured image and used imshow to show it as in the Figure1.

>>x=imread('para.jpg');

>>imshow(x);



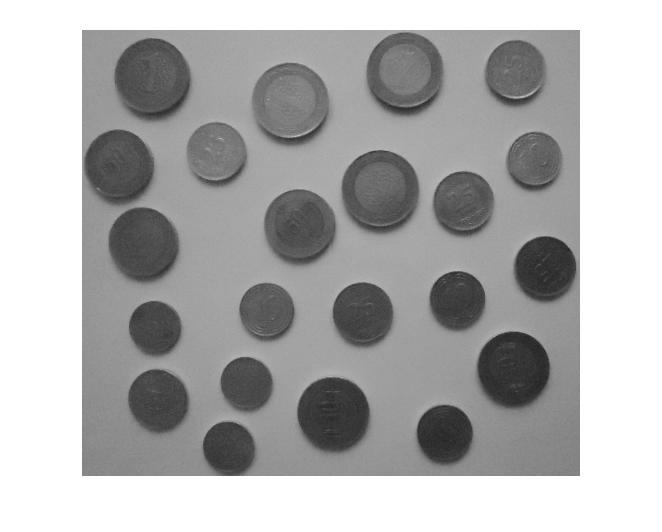
**Figure1**

1. **Convert RGB image to grayscale.**

We convert the RGB image to grayscale image by using rgb2gray and show it as shown in the Figure2.

>>x1=rgb2gray(x);

>>figure,imshow(x1)

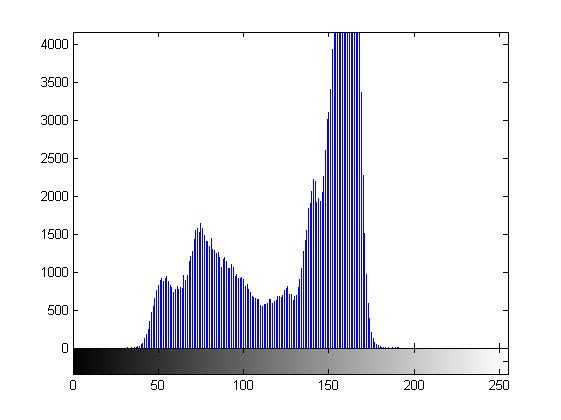


**Figure2**

1. **Histogram**

We plot the histogram of the image by using imhist (Figure3)

>>figure,imhist(x1)



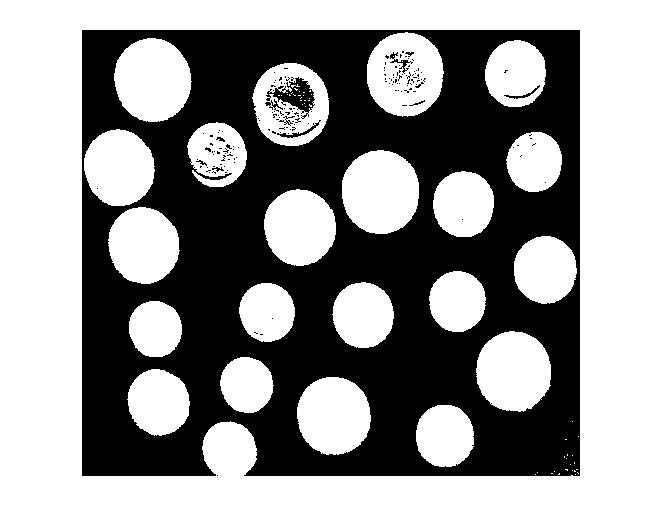
**Figure 3**

1. **Convert image to black and white image by thresholding**

We define a threshold value by using histogram of image. This threshold will help us to subtract background from objects easily. And also we can use it to make our image black and white binary image. We choose threshold value as 128. Background is light and objects are dark. 128 is an appropriate value to separate each other.

>>y=x1<128; %Threshold=128. Decide it by using the histogram.

>>figure, imshow(y)



**Figure4**

1. **Remove noise and Fill the Holes**

When we look at the Figure4 we saw some salt and pepper noise on it and also objects are not fully white. So we will try to fix these parts by using imfill and medfilt2.

%% Filling image regions and holes.

t=imfill(y,'holes'); % fill the holes with white

figure, imshow(t) %Figure5

%% FILTERING (2-D median filtering.)

imFiltered = medfilt2(t); % medfilt2 is using to remove salt and pepper noise in image t.

figure,imshow(imFiltered) %Figure6

|  |  |
| --- | --- |
|  |  |

**Figure5 Figure6**

1. **Labeling**

Now our image is ready for Labeling and we can find the total number of the coins.

[L,num]=bwlabel(imFiltered,8); % returns in NUM the number of connected objects found in imFiltered.

%8 is the number of connected objects,

%L is the number of labels

%num is the number of objects

disp(['There are ', num2str(num), ' coins.'])

When we write this code we can see on the command window the number of coins as 22.

>>There are 22 coins.

1. **Finding Objects**

In this part we use find(L==i) to find the objects one by one. And also define the rows and colums each of them. Then we use their max and minimum row and colum values to crop them from the original image.Then calculate their areas by using bwarea. We can see all of them separetly in the Figure7.

for i=1:1:num

[r,c]=find(L==i); %this line finds the objects and their rows and columns

rmin=min(r);rmax= max(r); %minimum and maximum row values of objects

cmin=min(c);cmax=max(c) ; %minimum and maximum columns values of objects

m=x(rmin:rmax,cmin:cmax); % crops the objects from original image

figure, imshow(m)

title(['Coin number:',num2str(i)]);

area(i)= bwarea(m);% Calculates the areas of each objects

end

total=0;

sort1=sort(area);%sorts the value of areas from smaller to bigger

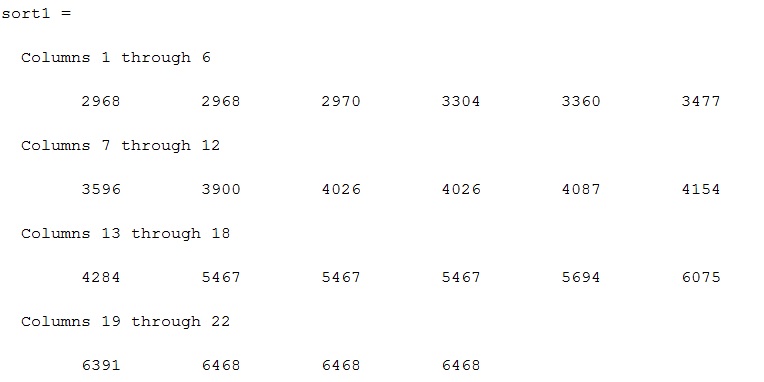
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Figure7**

1. **Finding Values of each Coins**

We try to find values of coins by using their area. We define a loop and we can say if the area values are closest to each other, they can be in same type. For example, the smallest areas are area of 5 Kuruş which is the smallest coin. After that 10 Kuruş, 25 Kuruş, 50 Kuruş and 1 TL is sorted according their areas from smaller to bigger.

Sorted area values are :



for j=1:1:num

if sort1(j)<=3000

c1=c1+1; % 5 Kuruş

elseif sort1(j)<=3600 & sort1(j)>3300

c2=c2+1;%10 Kuruş

elseif sort1(j)<=4300 & sort1(j)>3850

c3=c3+1; %25 Kuruş

elseif sort1(j)<=6100 & sort1(j)>5400

c4=c4+1;%50 Kuruş

else

c5=c5+1; %1 TL

end

end

1. **Final**

We display the results on the command window by using following codes. (Figure8)

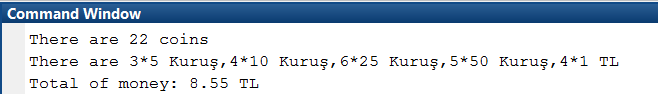
disp(['There are ',num2str(c1),'\*5 Kuruş,',num2str(c2),'\*10 Kuruş,',num2str(c3),'\*25 Kuruş,',num2str(c4),'\*50 Kuruş,',num2str(c5), '\*1 TL' ])

total=c1\*50+c2\*100+c3\*250+c4\*500+c5\*1000;%total amount of the coins

total=total/1000;

disp(['Total of money: ',num2str(total),' TL'])

**RESULT:**



**Figure8**