

DIGITAL IMAGE PROCESSING
ASSIGNMENT 3

BY
M V S SRAVAN
201531197

QUESTION 1: STRUCTURES IN IMAGES

A) Circles of different sizes in images

CODE

```
close all;
clear all;
im=imread('circle3.jpg');
im=rgb2gray(im);
im=edge(im, 'canny');
im=255*double(im);
imshow(uint8(im)); figure,
Threshold=180;

[M N]=size(im);

count=zeros(M,N,100);

for i=1:size(im,1)
for j=1:size(im,2)
if(im(i,j)==255)
for r=10:100
for angle=0:360
a=floor(i-r*cos(angle*pi/180));
b=floor(j-r*sin(angle*pi/180));
if(a>=1 && b>=1 && a<=M && b<=N)
count(a,b,r)=count(a,b,r)+1;
end
end
end
end
end

output=zeros(M,N);
k=1;
for i=1:M
for j=1:N
for r=1:100
if(count(i,j,r)>=Threshold)
store(k,1)=i;store(k,2)=j;store(k,3)=r;k=k+1;
for angle=0:360
```

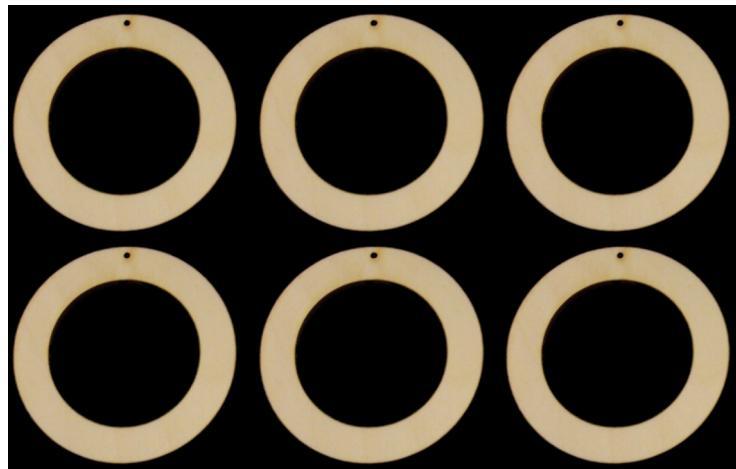
```

a=floor(i-r*cos(angle*pi/180));
b=floor(j-r*sin(angle*pi/180));
if(a>=1 && b>=1 && a<=M && b<=N)
output(a,b)=255;
end
end
end
end
end
end
imshow(uint8(output));

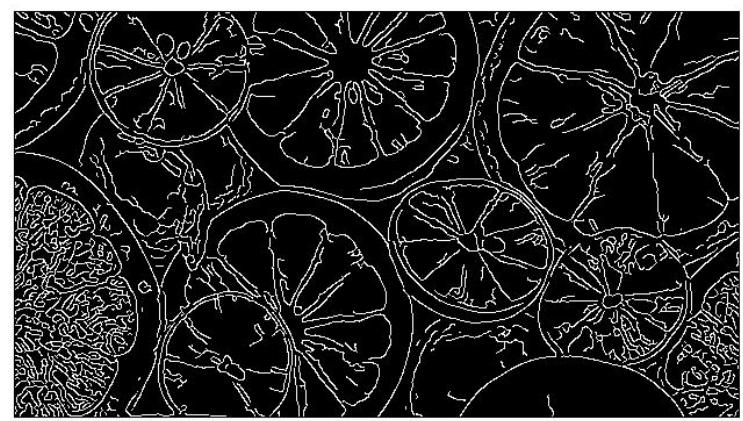
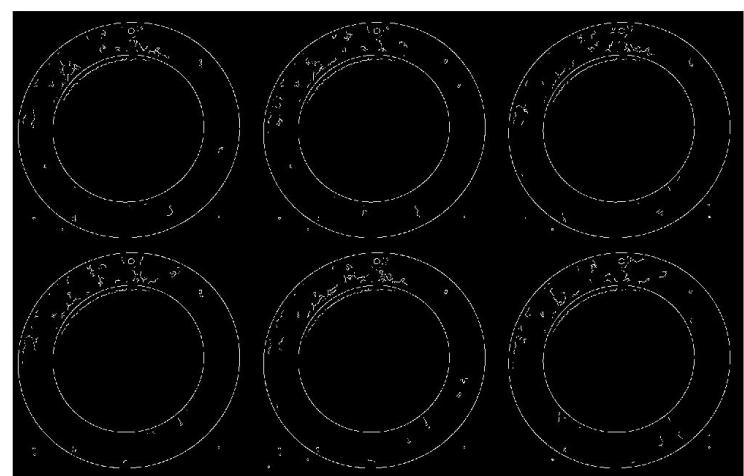
```

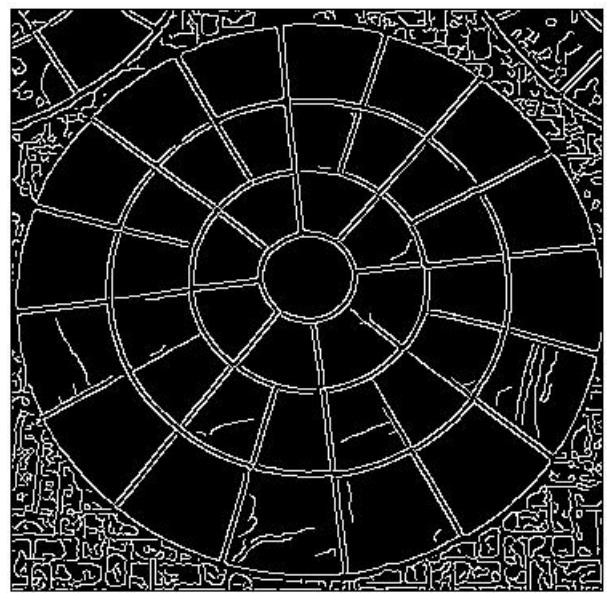
RESULTS

INPUT IMAGE



OUTPUT IMAGE





B) Parabolas of different sizes

CODE

```
img=imread('parabolal.jpg');
I=rgb2gray(img);
[N,M]=size(I);
[E,thresh]=edge(I,'canny',0.35);

rtrain = [0.003,0.004,0.005,0.006,0.007,0.008];

R=6;
counter=zeros(N,M,R);
[yindex xindex]=find(E);

for index=1:length(xindex)
    for r=1:R
        for x0=1:M
            y0=yindex(index)-rtrain(r)*(xindex(index)-x0)^2;
            y0=round(y0);
            if y0< N & y0>=1
                counter(y0,x0,r) = counter(y0,x0,r)+1;
            end
        end
    end
end

Ar=max(counter,[],3);

SE=strel('disk',40);
```

```

countmax=imdilate (Ar,SE) ;

thresh=90;

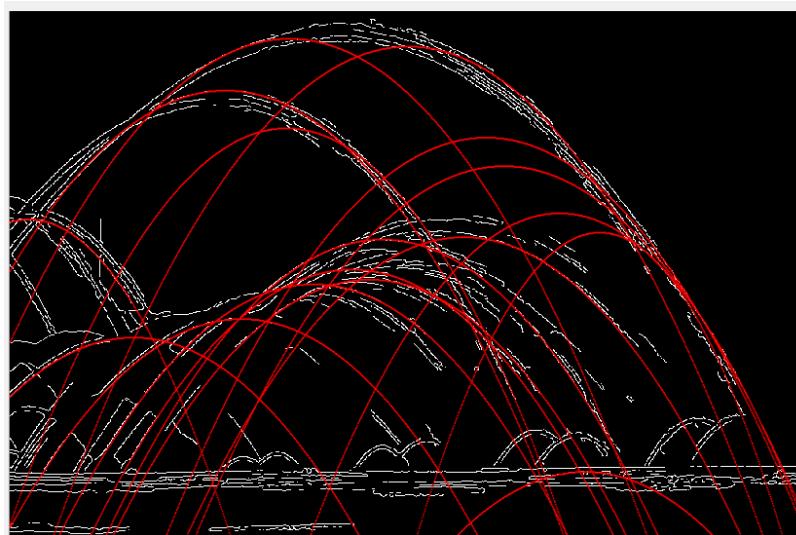
ydetect = [];
xdetect= [];
rdetect= [];

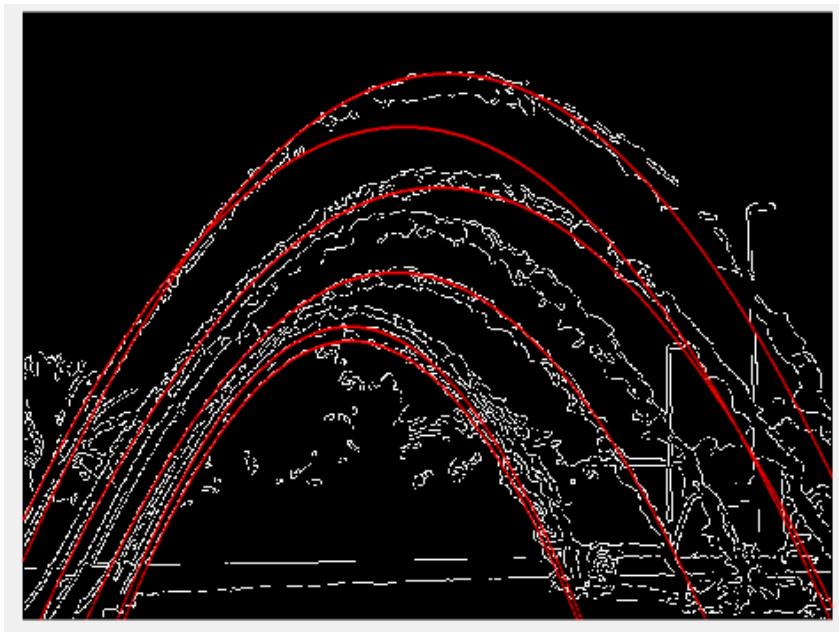
for r=1:R
    [y0 x0]=find((countmax(:,:,:) == counter(:,:,r)) &
counter(:,:,r) > thresh);
    ydetect=[ydetect; y0];
    xdetect=[xdetect; x0];
    rdetect=[rdetect; rtrain(r)*ones(length(x0),1)];
end
i2=img;
for i=1:size(img,1)
    for j=1:size(img,2)
        i2(size(img,1)-i+1,size(img,2)-j+1,:)=img(i,j,:);
    end
end
imshow(i2);

hold on;
for i=1:length(xdetect)
    x0=xdetect(i);
    y0=ydetect(i);
    r0=rdetect(i);
    for x=1:M
        y=round(y0+r0*(x-x0)^2);
        if y<=N & y>=1
            rectangle('Position',[x y 1 1], 'Edgecolor', 'r');
        end
    end
end

```

RESULTS





Q2:

A)Separate players in image Cricket1.jpg

Using Kmeans clustering

CODE

```
clear;
close all;
clc;
i=imread('Cricket4.jpeg');
%imshow(i);
%{
cform = makecform('srgb2lab');
lab_he = applycform(i,cform);
%imshow(lab_he)

ab = double(lab_he(:,:,2:3));
nrows = size(i,1);
ncols = size(i,2);
ab = reshape(ab,nrows*ncols,2);
%imshow(ab);
%}

nrows = size(i,1);
ncols = size(i,2);
ab=double(reshape(i,nrows*ncols,3));
nColors=3;
iters=150;
%initial_cluster_centers = random_initialization( ab,
nColors );
randidx = randperm(size(ab, 1));
initial_cluster_centers = ab(randidx(1:nColors), :);

[unnecessary, cluster_center,cluster_idx] =
run_KMeans(ab,initial_cluster_centers,iters);

for i=1:size(cluster_idx,1)
    if(cluster_idx(i)==3)
        cluster_idx(i)=255;
    else
        cluster_idx(i)=0;
    end
end

pixel_labels = uint8(reshape(cluster_idx,nrows,ncols));
%pixel_labels=pixel_labels/max(pixel_labels(:))
```

```

%pixel_labels=255-pixel_labels;
imshow(pixel_labels);

function [ J, cluster_centers, idx ] = run_KMeans( X,
initial_clusters_centers, max_iters )

[m, ~] = size(X);
K = size(initial_clusters_centers, 1);
cluster_centers = initial_clusters_centers;
previous_cluster_centers = cluster_centers;
idx = zeros(m, 1);

for i=1:max_iters
    % For each example in X, assign it to the closest centroid
    idx = findClosestClusters( X, cluster_centers );

    % Optionally, plot progress here
    %
    if plot_progress
        plotProgresskMeans(X, cluster_centers,
previous_cluster_centers, idx, K, i);
        previous_cluster_centers = cluster_centers;
        %fprintf('Press enter to continue.\n');
        %pause;
    end
    %
    % Given the memberships, compute new centroids
    cluster_centers = compute_cluster_centers( X, idx, K );
end
J = 0;
for i = 1 : size(X,1)
    J = J + sqrt(sum((X(i,:) - cluster_centers(idx(i),:)).^2));
end
%if plot_progress
%hold off;

end

function [ cluster_centers ] = compute_cluster_centers( X, idx,
K )
[m, n] = size(X);
cluster_centers = zeros(K, n);
number_of_samples=zeros(K,1);
for i=1:m

```

```

number_of_samples(idx(i))=number_of_samples(idx(i))+1;
cluster_centers(idx(i),:)=(cluster_centers(idx(i),:)) +
X(i,:));
end
cluster_centers=bsxfun(@rdivide,cluster_centers,number_of_samples);
end

function [ idx ] = findClosestClusters( X, cluster_centers )
K = size(cluster_centers,1);
idx = zeros(size(X,1), 1);
distances=zeros(K,1);
for i=1:size(X,1)
    for k=1:K
        distances(k)=sum((X(i,:)-cluster_centers(k,:)).^2);
    end
    [~,idx(i)]=min(distances);
end
end

```

RESULTS

ORIGINAL IMAGE



OUTPUT





USING ITERATIVE THRESHOLDING

CODE

```
clc;
clear;
close all;
I=imread('Cricket4.jpeg');
I = rgb2gray(I);
I=double(I);
globalmean = mean(mean(mean(I(:,:, :))));
a=0.0085;b=1.265;
xwd=1;ywd=1;
k=3;
for i=1:size(I,1)-k
for j=1:size(I,2)-k
Itemp=I(i:i+2,j:j+2);
vxy=var(var(Iterp(:,:, :)));
Txy(j) = a*vxy+b*globalmean;
if(Txy(j)>255 && j~=1)
Txy(j)=Txy(j-1);
end
for s=1:3
```

```
for e=1:3  
  
if Itemp(s,e)>TxY(j)  
out(i+xwd,j+ywd) = 1;  
else  
out(i+xwd,j+ywd) = 0;  
end  
  
end  
end  
end  
end  
  
figure,imshow(out);
```

ORIGINAL IMAGE



RESULT



ORIGINAL IMAGE

RESULT



ORIGINAL



FINAL IMAGE



ORIGINAL



RESULT

