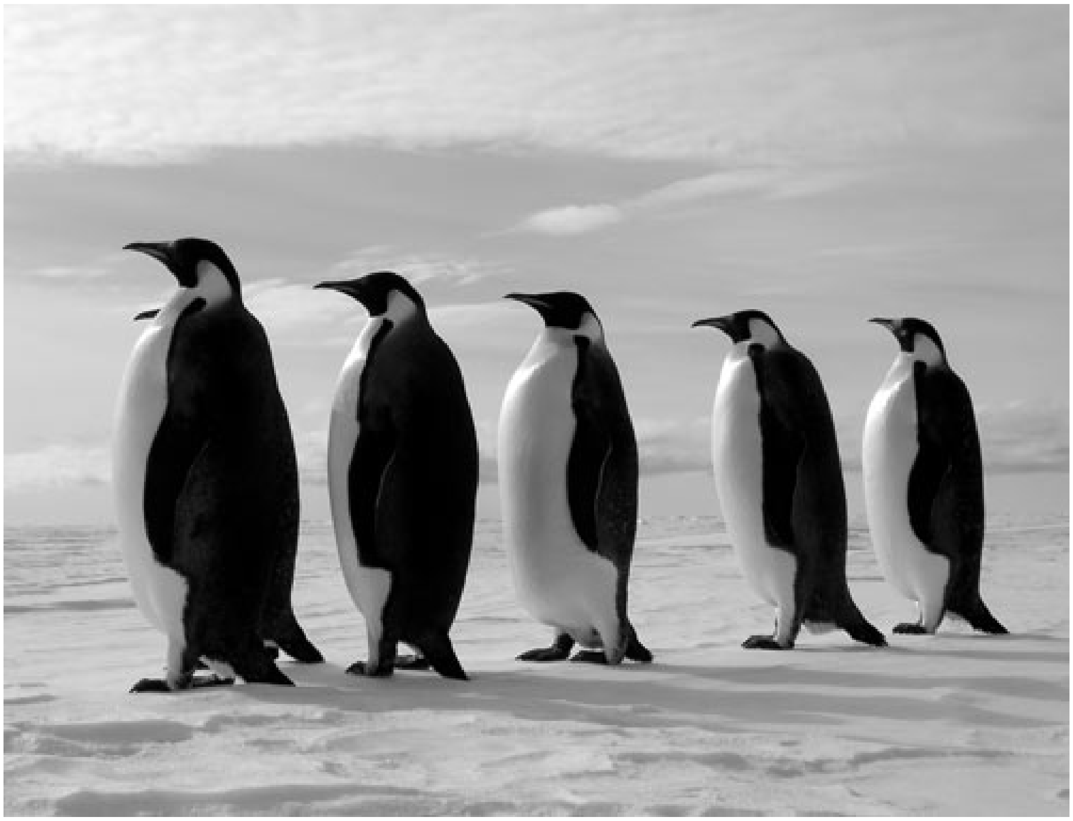
Contents

```
%Calvin and Keene Robotics Takehome
```

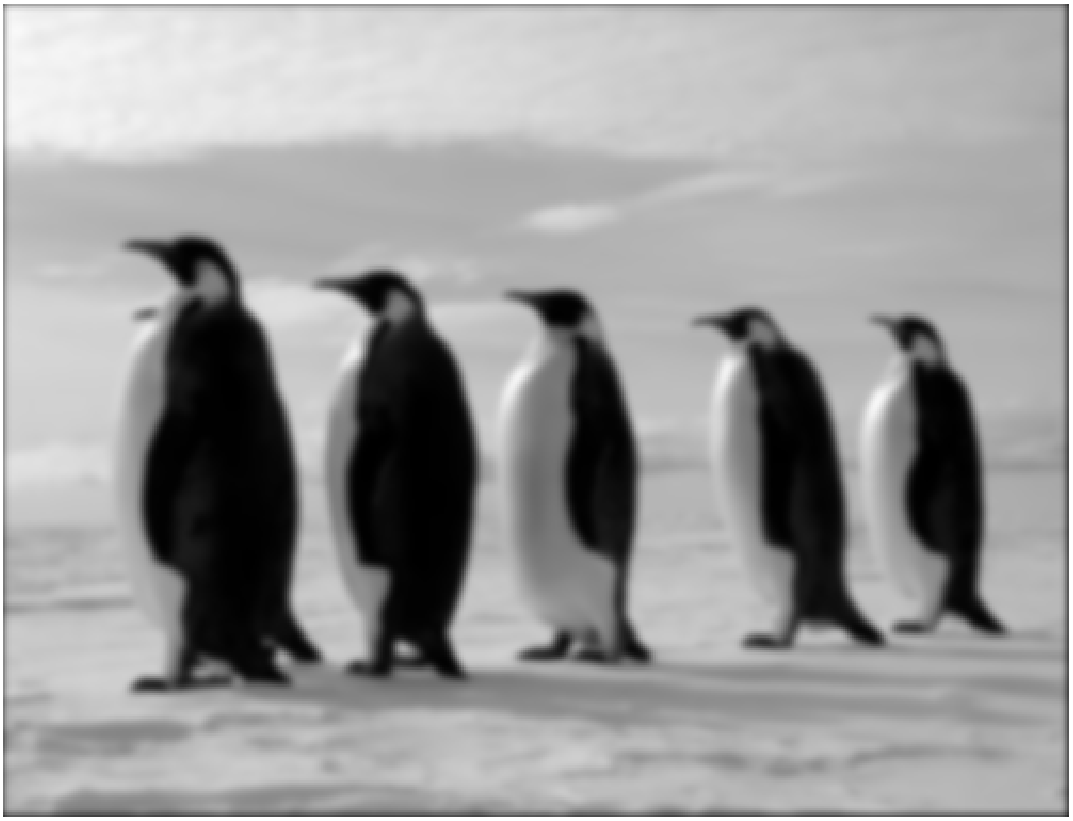
Open the color images and convert to grayscale

```
clear all
cpens=imread('penguins.jpg'); %read in a file, convert to matlab matrices
gpens=rgb2gray(cpens); %converts from color to grayscale
imshow(cpens) %displays the image in a window
figure
imshow(gpens)
```



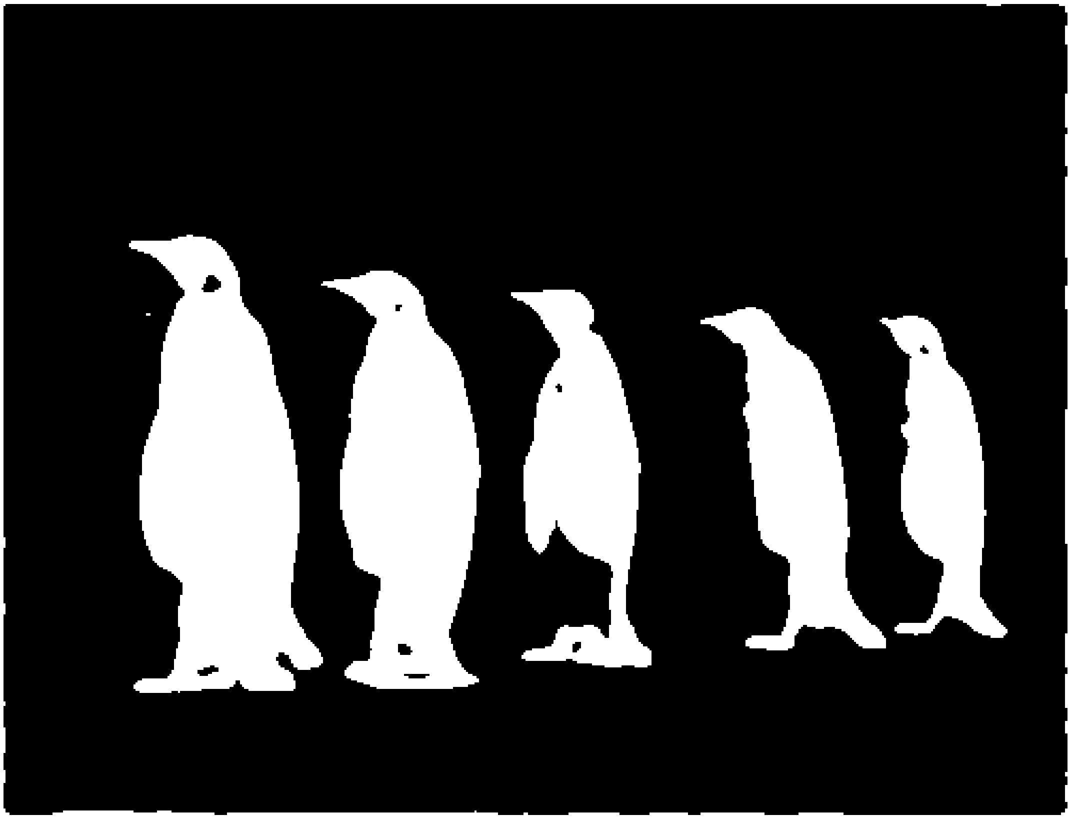
Do some filtering

```
G = fspecial('gaussian',[10 10],2); %create 10x10 gaussian blur kernel
blurpens= imfilter(gpens,G,'same'); %filters image with gaussian blur(best to use before thre
hsolding
imshow(blurpens)
```



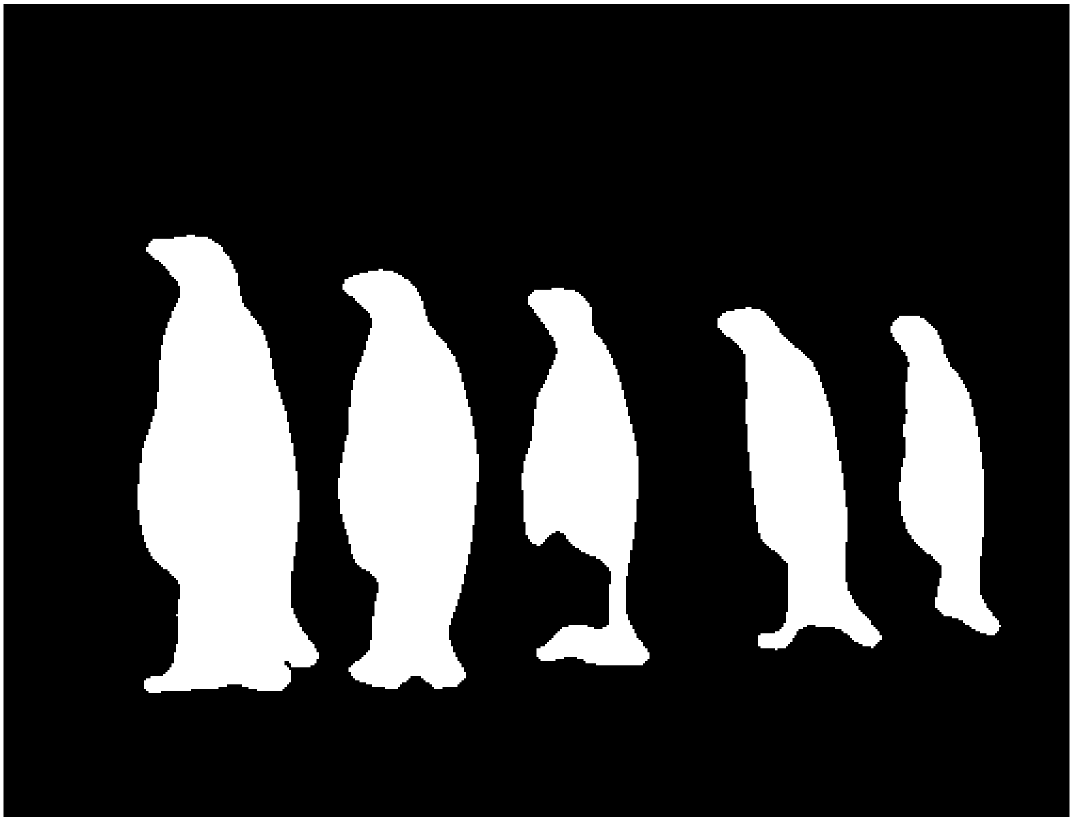
Threshold the image

```
threshpens = calvthresh(blurpens,120);
%display the thresholded image
figure
imshow(threshpens)
```



Morphological filtering

```
se = strel('square',6); %creates a 8x8 kernel for morph. filtering
erodedpens = imerode(threshpens,se); %erodes image (best to use afterthresholding)
dilatedpens = imdilate(erodedpens,se); %dilates image (best to use afterthresholding)
%displays the image. imscale(dilatedpens) will beuseful if you have images with pixel values
outside the standard range, suchas very large numbers after your semgmentation labeling.
medpens= medfilt2(dilatedpens, [10 10]); %median filters image (best to use afterthresholding
imshow(medpens)
%imsave %saves the current image
```

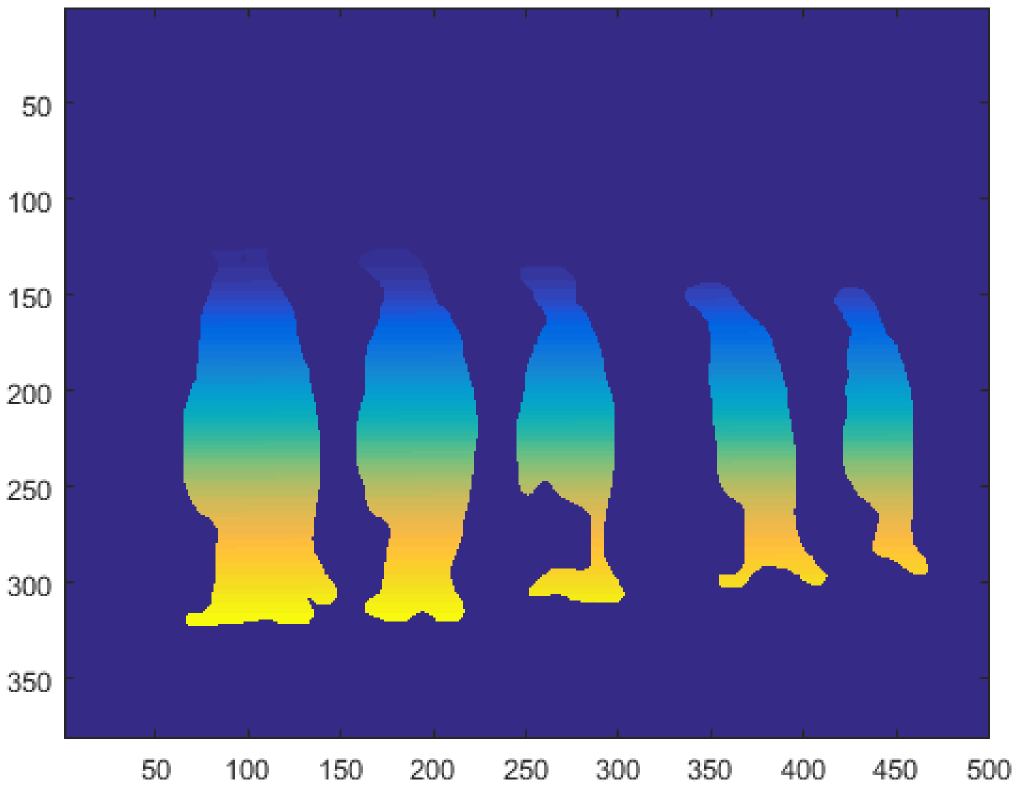


Segmentation

```
%assign a unique value to each white pixel in the image
seg1 = calvseg(medpens);
```

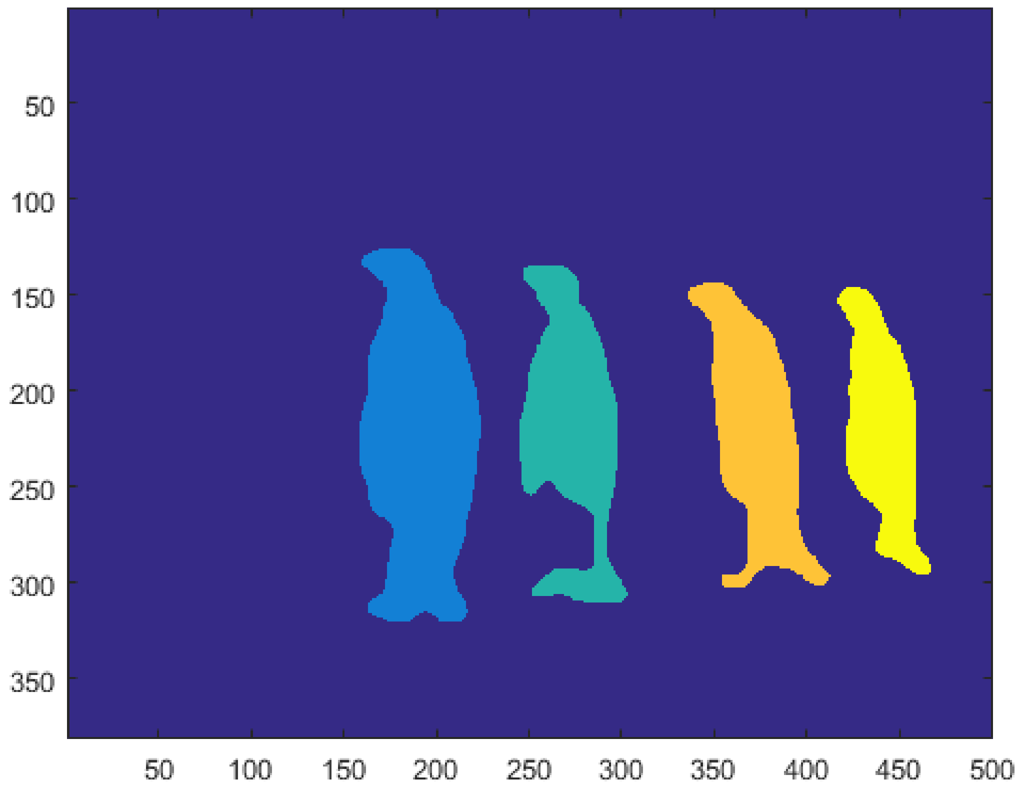
should appear as gradiated thresholded image as pixel label value increases

```
figure
imagesc(seg1)
%segment the image, each segment has a unique value
seg2 = segim2(seg1);
```



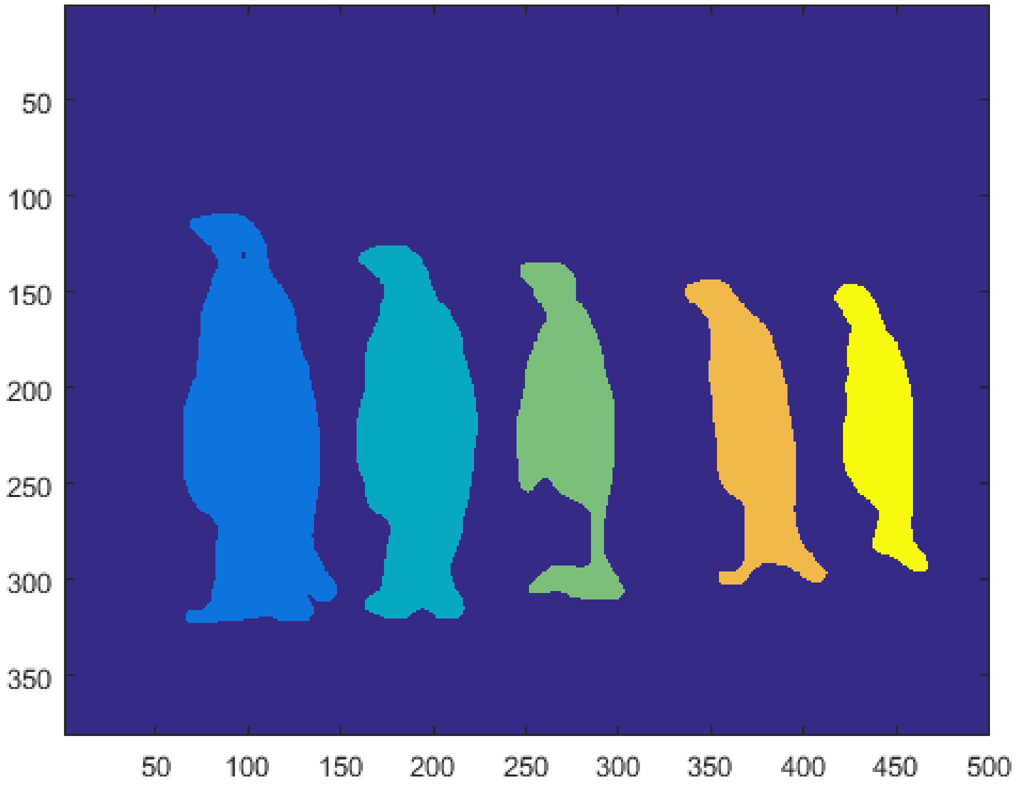
show the segments in grayscale

```
figure
imagesc(seg2)
%relabel the segments so they have a logical order that is sequential
seg3=relabel(seg2);
```



display the final segmented image

```
figure
imagesc(seg3)
```



```
%calculate the segment parameters
segpars=getallseginfo(seg3);
%put them in column form for putting in a table
penguin=segpars(:,1);
area = segpars(:,2);
cx = segpars(:,3);
cy = segpars(:,4);
orientation = segpars(:,5);
table(penguin,area,cx,cy,orientation)
```

```
ans = 

    penguin    area      cx        cy       orientation
    _______    _____    ______    ______    ___________

    1          11936    102.87    225.88    -78.323    
    2           9228    190.48    225.99     76.527    
    3           5910     272.9    219.91     73.261    
    4           5495    372.49    223.32    -85.283    
    5           4098    441.17    220.47      72.49    
```

Published with MATLAB® R2016a

```
function imbw = calvthresh(imgr,thresh)
imsz = size(imgr);
imht = imsz(1);
imwd = imsz(2);
for i = 1:imht
    for j = 1:imwd
        if imgr(i,j) < thresh
            imbw(i,j) = 255;
        else
            imbw(i,j) = 0;
        end
    end
end
end

function imseg = calvseg(imbw)
imsz = size(imbw);
imht = imsz(1);
imwd = imsz(2);
x = 0;
imseg = zeros(imht,imwd);
for i = 1:imht
    for j = 1:imwd
        if imbw(i,j) == 255
            x = x + 1;
            imseg(i,j) = x;
        end
    end
end

end


function [ imseg2 ] = segim2( imseg )
imsz = size(imseg);
imht = imsz(1);
imwd = imsz(2);
imseg2 = imseg;
%set neighboring pixel values to lowest from top left to bottom right
for i = 2:(imht-1)
    for j = 2:(imwd-1)

       if imseg2(i,j)>0
        neighbors = [imseg2(i,j-1),imseg2(i+1,j),imseg2(i,j+1),imseg2(i-
1,j)];
        for k=1:4
           if neighbors(k)==0;
            neighbors(k)=max(neighbors(:));
           end
        end
        newlabel=min(neighbors);
        if imseg2(i,j)>newlabel
            imseg2(i,j)=newlabel;

        end
       end
    end
end
%set neighboring pixel values to lowest from bottom right to top left
for i = (imht-1):-1:2
    for j = (imwd-1):-1:2
       if imseg2(i,j)>0
        neighbors = [imseg2(i,j-1),imseg2(i+1,j),imseg2(i,j+1),imseg2(i-
1,j)];
        for k=1:4
           if neighbors(k)==0;
            neighbors(k)=max(neighbors(:));
           end
        end
        newlabel=min(neighbors);
        if imseg2(i,j)>newlabel
            imseg2(i,j)=newlabel;
        end
       end

    end
end

%set neighboring pixel values to lowest from top right to bottom left to
%clean up lower left corners of penguins
for i = 2:(imht-1)
    for j = (imwd-1):-1:2
       if imseg2(i,j)>0
        neighbors = [imseg2(i,j-1),imseg2(i+1,j),imseg2(i,j+1),imseg2(i-
1,j)];
        for k=1:4
           if neighbors(k)==0;
            neighbors(k)=max(neighbors(:));
           end
        end
        newlabel=min(neighbors);
        if imseg2(i,j)>newlabel
            imseg2(i,j)=newlabel;
        end
       end

    end
end
end

function[imseg3]=relabel(imseg2)
labels=unique(imseg2);
imsz = size(imseg2);
imht = imsz(1);
imwd = imsz(2);
imseg3 = imseg2;
for i = 1:imht
    for j = 1:imwd
        for x=2:size(labels)
            if (imseg3(i,j)==labels(x))
                imseg3(i,j)=x-1;
            end
        end
    end
end
end



function[segpars]=getallseginfo(img)
labels=unique(img);
for i=2:size(labels);
     [label,area,cx,cy,orientation] = segmentparams(img,labels(i));
     segpars(i-1,:)= [label,area,cx,cy,orientation];
end
end



function [ label,area,cx,cy,orientation] = segmentparams( img,label )
area=getmoment(img,label,0,0);
cx = getmoment(img,label,1,0)/area;
cy = getmoment(img,label,0,1)/area;
c20=centmoment(img,label,cx,cy,2,0);
c11=centmoment(img,label,cx,cy,1,1);
c02=centmoment(img,label,cx,cy,0,2);
orientation = rad2deg(.5*atan2(2.*c11,(c20-c02)));
end

function [ moment ] = getmoment( img,label,p,q )
imsz = size(img);
imht = imsz(1);
imwd = imsz(2);
moment = 0;
for i = 1:imht
    for j = 1:imwd
       if img(i,j) == label
           moment = moment + j.^p*i.^q;
       end
    end
end
end






function [ centralmoment ] = centmoment(img,label,cx,cy,p,q)
imsz = size(img);
imht = imsz(1);
imwd = imsz(2);
moment = 0;
for i = 1:imht
    for j = 1:imwd
       if img(i,j) == label
           centralmoment = (j - cx).^p.*(i-cy).^q;
       end
    end
```

```
end
end
```