LAB 8: SPATIAL DOMAIN

Name: Soundarya G

Register Number: 2048057

Feature Extraction

1. **Means of Red:**

Average pixel values of R component seperately.

2. **Means of Green:**

Average pixel values of G component seperately.

3. **Means of Blue:**

Average pixel values of B component seperately.

4. **Standard Deviation**

It is a measure of the amount of variation or dispersion of a set of values.

5. **Mean**

Average pixel value of gray scale image.

6. **Median**

It is the value separating the higher half from the lower half of a data sample.

7. **Maximum**

Maximum pixel value of gray scale image.

8. **Minimum**

Minimum pixel value of gray scale image.

9. **Entropy**

It is a statistical measure of randomness that can be used to characterize the texture of the input image.

10. **Above & Below R**

The no. of pixels which are above and below the average values of R components separately.

11. **Mode**

It is the most frequently occurring value in sample.

12. **Above & Below G**

The no. of pixels which are above and below the average values of G components separately.

13. **Above & Below B**

The no. of pixels which are above and below the average values of B components separately.

14. M**idpoint**

It measures the average of minimum and maximum pixels.

15. **Variance**

It measures variability from the average or mean.

16. Skewness

It is a measure of symmetry, or more precisely, the lack of symmetry

17. **Kurtosis**

It is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution

These 17 features are extracted for original images, after Noise Reduction and Edge Detection

```
mydir='/MATLAB Drive/IVA_LABS/Lab8&9/BirdsSet/class-9';
fileformat='*.jpg';
dd=dir(fullfile(mydir,fileformat));
assert(numel(dd) > 0, 'No file was found. Check that the path is correct');
my_img = struct('img', cell(size(dd)));
k=numel(dd)+1;
for zz=1:numel(dd)
 my_img(zz).img = imread(fullfile(mydir,dd(zz).name));
end

check = 0;
while check == 0
    fprintf('This Program does the following');
    fprintf('\n1. Original Image');
    fprintf('\n2. NOISE REDUCTION: Median Filtering');
    fprintf('\n3. EDGE DETECTION: Canny Filter');
    x = input('\nPlease select your choice: ');

    switch(x)
        case 1
            fprintf('\nOriginal Image');
            fprintf('\n***************');
            c_r_1=0;
            c_r_2=0;
            c_g_1=0;
            c_g_2=0;
            c_b_1=0;
            c_b_2=0;
            for i=1:numel(dd)
                current=imresize(my_img(i).img,[400,400]);

                %rgb means
                r=mean(mean(current(:,:,1)));
                g=mean(mean(current(:,:,2)));
```

```
            b=mean(mean(current(:,:,3)));

            %grayscale
            g_img=double(rgb2gray(current));

            %statistical measures
            av=mean(mean(g_img));
            med=median(median(g_img));
            st_dev=std(std(double(g_img)));
            max_=max(max(g_img));
            mode_=mode(mode(g_img));
            min_=min(min(g_img));
            midpoint_=0.5*(max_+min_);
            var_=var(var(g_img));
            kur_=kurtosis(kurtosis(g_img));
            skew_=skewness(skewness(g_img));
            [M,N]=size(g_img);
            area=M*N;

            %entropy values
            e=entropy(g_img);

            %Above & Below
            for i = 1:M
                for j = 1:N
                    if(current(i,j,1)>r)
                        c_r_1=c_r_1+1;
                    end
                    if(current(i,j,2)>g)
                        c_g_1=c_g_1+1;
                    end
                    if(current(i,j,3)>b)
                        c_b_1=c_b_1+1;
                    end
                    if(current(i,j,1)<r)
                        c_r_2=c_r_2+1;
                    end
                    if(current(i,j,2)<g)
                        c_g_2=c_g_2+1;
                    end
                    if(current(i,j,1)<b)
                        c_b_2=c_b_2+1;
                    end
                end
            end
            %Column Values
            rgb=[r,g,b,av,med,st_dev,max_,min_,area,e,c_r_1,c_r_2,c_g_1,c_g_2,c_b_1
            %Writing into Excel Sheets
            writematrix(rgb,'/MATLAB Drive/IVA_LABS/Lab8&9/Spatial Domain/c9_1.csv'

        end
        fprintf('\nExcel sheet generated');
        fprintf('\n************************************');
        check = input('\nPress 0 to return or any other key to exit.');
```

```
        case 2
            fprintf('\nNOISE REDUCTION: Median Filtering');
            fprintf('\n************************************');
            %Above & Below
            c_r_1=0;
            c_r_2=0;
            c_g_1=0;
            c_g_2=0;
            c_b_1=0;
            c_b_2=0;
            for i=1:numel(dd)
                current=imresize(my_img(i).img,[400,400]);

                %rgb means
                r=mean(mean(current(:,:,1)));
                g=mean(mean(current(:,:,2)));
                b=mean(mean(current(:,:,3)));

                %grayscale
                g_img1=rgb2gray(current);
                g_img=double(medfilt2(g_img1));

                %statistical measures
                av=mean(mean(g_img));
                med=median(median(g_img));
                st_dev=std(std(double(g_img)));
                max_=max(max(g_img));
                min_=min(min(g_img));
                mode_=mode(mode(g_img));
                midpoint_=0.5*(max_+min_);
                var_=var(var(g_img));
                kur_=kurtosis(kurtosis(g_img));
                skew_=skewness(skewness(g_img));
                [M,N]=size(g_img);
                area=M*N;

                %entropy values
                e=entropy(g_img);

                %Above & Below
                for i = 1:M
                    for j = 1:N
                        if(current(i,j,1)>r)
                            c_r_1=c_r_1+1;
                        end
                        if(current(i,j,2)>g)
                            c_g_1=c_g_1+1;
                        end
                        if(current(i,j,3)>b)
                            c_b_1=c_b_1+1;
                        end
                        if(current(i,j,1)<r)
                            c_r_2=c_r_2+1;
                        end
```

```
                    if(current(i,j,2)<g)
                        c_g_2=c_g_2+1;
                    end
                    if(current(i,j,1)<b)
                        c_b_2=c_b_2+1;
                    end
                end
            end
            %Column Values
            rgb1=[r,g,b,av,med,st_dev,max_,min_,area,e,c_r_1,c_r_2,c_g_1,c_g_2,c_b_
            %Writing into Excel Sheets
            writematrix(rgb1,'/MATLAB Drive/IVA_LABS/Lab8&9/Spatial Domain/c9_2.csv
        end
        fprintf('\nExcel sheet generated');
        fprintf('\n************************************');
        check = input('\nPress 0 to return or any other key to exit.');

    case 3
        fprintf('\nEDGE DETECTION: Canny Filter');
        fprintf('\n************************************');
        for i=1:numel(dd)
            c_r_1=0;
            c_r_2=0;
            c_g_1=0;
            c_g_2=0;
            c_b_1=0;
            c_b_2=0;
            current=imresize(my_img(i).img,[400,400]);

            %rgb means
            r=mean(mean(current(:,:,1)));
            g=mean(mean(current(:,:,2)));
            b=mean(mean(current(:,:,3)));

            %grayscale
            g_img1=rgb2gray(current);
            g_img=double(edge(g_img1,'Canny',0.2));

            %statistical measures
            av=mean(mean(g_img));
            med=median(median(g_img));
            st_dev=std(std(double(g_img)));
            max_=max(max(g_img));
            min_=min(min(g_img));
            mode_=mode(mode(g_img));
            midpoint_=0.5*(max_+min_);
            var_=var(var(g_img));
            kur_=kurtosis(kurtosis(g_img));
            skew_=skewness(skewness(g_img));
            [M,N]=size(g_img);
            area=M*N;

            %entropy values
            e=entropy(g_img);
```

```
                %Above & Below
                for i = 1:M
                    for j = 1:N
                        R=current(i,j,1);
                        G=current(i,j,2);
                        B=current(i,j,3);
                        if(R>r)
                            c_r_1=c_r_1+1;
                        end
                        if(G>g)
                            c_g_1=c_g_1+1;
                        end
                        if(B>b)
                            c_b_1=c_b_1+1;
                        end
                        if(R<r)
                            c_r_2=c_r_2+1;
                        end
                        if(G<g)
                            c_g_2=c_g_2+1;
                        end
                        if(B<b)
                            c_b_2=c_b_2+1;
                        end
                    end
                end
                %Column Values
                rgb1=[r,g,b,av,med,st_dev,max_,min_,area,e,c_r_1,c_r_2,c_g_1,c_g_2,c_b_
                %Writing into Excel Sheets
                writematrix(rgb1,'/MATLAB Drive/IVA_LABS/Lab8&9/Spatial Domain/c9_3.csv
            end
            fprintf('\nExcel sheet generated');
            fprintf('\n************************************');
            check = input('\nPress 0 to return or any other key to exit.');
    end

end
fprintf('Thank you, have a Nice Day!')
```