LAB 9: FREQUENCY DOMAIN

Name: Soundarya G

Register Number: 2048057

```
mydir='/MATLAB Drive/IVA_LABS/Lab8&9/BirdsSet/class-1';
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)));

for zz=1:numel(dd)
    my_img(zz).img = imread(fullfile(mydir,dd(zz).name));
end
```

RED COMPONENT OF IMAGE

```
r_img = struct('img', cell(size(dd)));
for zz=1:numel(dd)
    r_img(zz).img = my_img(zz).img(:,:,1);
end
```

GREEN COMPONENT OF IMAGE

```
g_img = struct('img', cell(size(dd)));
for zz=1:numel(dd)
    g_img(zz).img = my_img(zz).img(:,:,2);
end
```

BLUE COMPONENT OF IMAGE

```
b_img = struct('img', cell(size(dd)));
for zz=1:numel(dd)
    b_img(zz).img = my_img(zz).img(:,:,3);
end
```

GRAY SCALE IMAGE

```
gray_img = struct('img', cell(size(dd)));
for zz=1:numel(dd)
    gray_img(zz).img = rgb2gray(my_img(zz).img);
end
```

EDGE DETECTION IMAGE

```
edge_img = struct('img', cell(size(dd)));
for zz=1:numel(dd)
    edge_img(zz).img = fourrier(gray_img(zz).img,0.09,4);
end
```

FAST FOURIER TRANSFORM

1.Gray Scale Image

```
%Fast Fourier transform :gray scale image

for i=1:numel(dd)

    current=gray_img(i).img;

    % fourier transform

    fft_img=fft2(current);

    %statistical measures

    av=real(mean(mean(fft_img)));
    med=real(median(median(fft_img)));
    st_dev=real(std(std(double(fft_img))));
    max_=real(max(max(fft_img)));
    min_=real(min(min(fft_img)));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_FFT_gray.csv','WriteMode', 'append');
end
```

2. Red component image

```
%Fast Fourier transform :red image

for i=1:numel(dd)

    current=r_img(i).img;

    % fourier transform

    fft_img=fft2(current);

    %statistical measures

    av=real(mean(mean(fft_img)));
    med=real(median(median(fft_img)));
```

```
    st_dev=real(std(std(double(fft_img))));
    max_=real(max(max(fft_img)));
    min_=real(min(min(fft_img)));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_FFT_r.csv','WriteMode', 'append');
end
```

3. Green component Image

```
%Fast Fourier transform :green image

for i=1:numel(dd)

    current=g_img(i).img;

    % fourier transform

    fft_img=fft2(current);

    %statistical measures

    av=real(mean(mean(fft_img)));
    med=real(median(median(fft_img)));
    st_dev=real(std(std(double(fft_img))));
    max_=real(max(max(fft_img)));
    min_=real(min(min(fft_img)));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_FFT_g.csv','WriteMode', 'append');
end
```

4. Blue component Image

```
%Fast Fourier transform :blue image

for i=1:numel(dd)

    current=b_img(i).img;

    % fourier transform

    fft_img=fft2(current);

    %statistical measures

    av=real(mean(mean(fft_img)));
    med=real(median(median(fft_img)));
    st_dev=real(std(std(double(fft_img))));
    max_=real(max(max(fft_img)));
```

```
    min_=real(min(min(fft_img)));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_FFT_b.csv','WriteMode', 'append');
end
```

5. Edge Detected Image

```
%Fast Fourier transform :edge image

for i=1:numel(dd)

    current=edge_img(i).img;

    % fourier transform

    fft_img=fft2(current);

    %statistical measures

    av=real(mean(mean(fft_img)));
    med=real(median(median(fft_img)));
    st_dev=real(std(std(double(fft_img))));
    max_=real(max(max(fft_img)));
    min_=real(min(min(fft_img)));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_FFT_e.csv','WriteMode', 'append');
end
```

DISCRETE COSINE TRANSFORM

1. Gray Scale Image

```
% Discrete Cosine Transform : gray scale image

for i=1:numel(dd)
    current=gray_img(i).img;

    %dct
    dct_img=dct2(current);
    dc=dct_img(1,1);

    writematrix(dc,'Class1_DCT_gray.csv','WriteMode', 'append')
end
```

2. Red Component Image

```
% Discrete Cosine Transform :red image

for i=1:numel(dd)
    current=r_img(i).img;

    %dct
    dct_img=dct2(current);
    dc=dct_img(1,1);

    writematrix(dc,'Class1_DCT_r.csv','WriteMode', 'append')
end
```

3. Green Component Image

```
% Discrete Cosine Transform : green image

for i=1:numel(dd)
    current=g_img(i).img;

    %dct
    dct_img=dct2(current);
    dc=dct_img(1,1);

    writematrix(dc,'Class1_DCT_g.csv','WriteMode', 'append')
end
```

4. Blue Component Image

```
% Discrete Cosine Transform : blue image

for i=1:numel(dd)
    current=b_img(i).img;

    %dct
    dct_img=dct2(current);
    dc=dct_img(1,1);

    writematrix(dc,'Class1_DCT_b.csv','WriteMode', 'append')
end
```

5. Edge Detected Image

```
% Discrete Cosine Transform : edge detected image

for i=1:numel(dd)
    current=edge_img(i).img;

    %dct
    dct_img=dct2(current);
    dc=dct_img(1,1);

    writematrix(dc,'Class1_DCT_e.csv','WriteMode', 'append')
end
```

WAVELET

1. Gray scale Image

```
%wavelet : gray scale image
for i=1:numel(dd)
    current=gray_img(i).img; %r_img(i).img;

    %wavelet transform

    wave_img=wave(current,'haar',3);

    %statistical measures

    av=mean(mean(wave_img));
    med=median((median(wave_img)));
    st_dev=std(std(double(wave_img)));
    max_=max(max(wave_img));
    min_=min(min(wave_img));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_wavelet_gray.csv','WriteMode', 'append');
end
```

2. Red Component Image

```
%wavelet : red component image
for i=1:numel(dd)
    current=r_img(i).img;

    %wavelet transform

    wave_img=wave(current,'haar',3);

    %statistical measures

    av=mean(mean(wave_img));
    med=median((median(wave_img)));
    st_dev=std(std(double(wave_img)));
    max_=max(max(wave_img));
    min_=min(min(wave_img));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_wavelet_r.csv','WriteMode', 'append');
end
```

3. Green Component Image

```
%wavelet : green component image
for i=1:numel(dd)
    current=g_img(i).img;

    %wavelet transform

    wave_img=wave(current,'haar',3);

    %statistical measures

    av=mean(mean(wave_img));
    med=median((median(wave_img)));
    st_dev=std(std(double(wave_img)));
    max_=max(max(wave_img));
    min_=min(min(wave_img));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_wavelet_g.csv','WriteMode', 'append');
end
```

4. Blue Component Image

```
%wavelet : blue component image
for i=1:numel(dd)
    current=b_img(i).img;

    %wavelet transform

    wave_img=wave(current,'haar',3);

    %statistical measures

    av=mean(mean(wave_img));
    med=median((median(wave_img)));
    st_dev=std(std(double(wave_img)));
    max_=max(max(wave_img));
    min_=min(min(wave_img));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_wavelet_b.csv','WriteMode', 'append');
end
```

5. Edge Detected Image

```
%wavelet : edge detected image
for i=1:numel(dd)
    current=edge_img(i).img;
```

```
    %wavelet transform

    wave_img=wave(current,'haar',3);

    %statistical measures

    av=mean(mean(wave_img));
    med=median((median(wave_img)));
    st_dev=std(std(double(wave_img)));
    max_=max(max(wave_img));
    min_=min(min(wave_img));

    %Column Values

    rgb=[av,med,st_dev,max_,min_];
    writematrix(rgb,'Class1_wavelet_e.csv','WriteMode', 'append');
end
```