FUNCTION_LBG.M

```
%LBG Algorithm on color image for vector dimension 12

function [centroid]=Function_EV_Rotation(s,CB);

block_len=12;
mx=size(s);

if mod(mx(1),2) == 0
else mx(1)=mx(1)-1;
end

if mod(mx(2),2) == 0
else mx(2)=mx(2)-1;
end

E=[1 1 1 1 1 1 1 1 1 1 1 1];

vect_count=0;
k=1;
i=1;
for t=1:mx(1)/2;
    j=1;
    for p=1:mx(2)/2;
       v(k,1:3)=s(i,j,1:3);
       v(k,4:6)=s(i,j+1,1:3);
       v(k,7:9)=s(i+1,j,1:3);
       v(k,10:12)=s(i+1,j+1,1:3);

       j=j+2;
       k=k+1;

       vect_count = vect_count + 1;
    end
    i=i+2;
end

v1(1,1:vect_count)=[1:vect_count];
for j=1:block_len;
    sum1=0;
    for k=1:vect_count;
        sum1 = sum1 + double(v(k,j));
    end
    centroid(1,j)=sum1/vect_count;
end

centroid(1,1:block_len);

c_pt=[k];

d1=0;
for k = 1:c_pt(1); %Calculation of mean square error
    for j=1:block_len;
```

```
        d1=d1+(double(v(k,j)) - centroid(1,j))*(double(v(k,j)) -
centroid(1,j));
    end
end

mse(1)=d1/(c_pt(1)*block_len);

netmse=mse(1);

cb_size=1;
cb=centroid;

%clear k;
%clear s;
%clear v;
pack;
ff=1;

while cb_size<=CB/2
    pack;
    cu=1;
    c=1;
    vc=1;

    for out=1:cb_size

        c0(c,:)=cb(out,:) + E;
        c0(c+1,:)=cb(out,:) - E;
        c0;
        m=0;
        q=0;
        vect_count1=c_pt(vc);
        for k = 1:vect_count1; %Creation of clusters
            d1=0;
            d2=0;
            for j=1:block_len;
                d1=d1+(double(v(v1(out,k),j)) -
c0(c,j))*(double(v(v1(out,k),j)) - c0(c,j));
                d2=d2+(double(v(v1(out,k),j)) -
c0(c+1,j))*(double(v(v1(out,k),j)) - c0(c+1,j));
            end

            if d1<d2
                m=m+1;
                %for j=1:block_len;
                    clu(cu,m)=v1(out,k);
                %end
            else
                q=q+1;
                %for j=1:block_len;
                    clu(cu+1,q)=v1(out,k);
                %end
            end
        end
        clu_pt(c)=[m];
        clu_pt(c+1)=[q];
        pack;
        for j=1:block_len; %Calculates centroid for the cluster 1
            sum1=0;
            for k=1:clu_pt(c);
                sum1 = sum1 + double(v(clu(cu,k),j));
            end
            if clu_pt(c)== 0
               centroid(cu,j)=-1;
            else
                centroid(cu,j)=sum1/clu_pt(c);
            end
        end

        for j=1:block_len; %Calculates centroid for the cluster 2
            sum1=0;
            for k=1:clu_pt(c+1);
                sum1 = sum1 + double(v(clu(cu+1,k),j));
            end
            if clu_pt(c+1)== 0
                centroid(cu+1,j)=0;
            else
                centroid(cu+1,j)=sum1/clu_pt(c+1);
            end
        end

    cu=cu+2;
    c=c+2;
    vc=vc+1;

    end %End Of Outer For Loop
    cb=centroid;
    c_pt=clu_pt;
    v1=clu;
    netmse=0;
    count=0;

    ff=ff+1;

    cb_size=cb_size*2;
    cb_size;

end %End Of While Loop
```

```
CBIR-LBG-RGB.M

%Encoding Algorithm For Gray Scale Images for block size 2x2

clear all;
clc;
block_len=12;
CB=8;

for g=1:1:150
    A=imread(strcat('C:\Users\vinayak-pc\Documents\matlab\Data\input\',int2str(g),'.jpg'));
    centroid=Function_LBG(A,CB);
    fprintf('%d ',g);
    Codebook(1:CB,1:12,g)=centroid(1:CB,1:12);
end %End Of While Loop
str=strcat('C:\Users\vinayak-pc\Documents\matlab\Data\',int2str(g),'.mat');
save(str,'g');
```

BAG OF WORDS:-

```
%%
imgSets=imageSet('Data\Training1','recursive');
[imgSets.Count];

%%
minSetCount=min([imgSets.Count]);
trainingSets=partition(imgSets,minSetCount,'randomize');
[trainingSets.Count];

%%
bag=bagOfFeatures(trainingSets,'VocabularySize',100,'PointSelection','Detector');

%%
categoryClassifier=trainImageCategoryClassifier(trainingSets,bag);

%%
confMatrix=evaluate(categoryClassifier,trainingSets);
```