APPENDIX

SUPERVISED CLASSIFICATION FOR ENTROPY BASED LOCAL DESCRIPTOR OPERATOR

KB = imread('fig02.gif');
KC = imread('fig03.gif');
KD = imread('fig04.gif');
KE = imread('fig05.gif');
KF = imread('fig06.gif');
KG = imread('fig07.gif');
KH = imread('fig08.gif');
KI = imread('fig09.gif');

K = [KA KB KC; KD KE KF; KG KH KI];
KX = K;
figure(1)
hx = subplot(1,1,1);
imshow(K)
axis('square')
set(hx, 'Xtick', [], 'Xticklabel', [], 'Ytick', [], 'Yticklabel', []);

Z1 = size(KA);
di = 1;

K1 = KA(1:30,1:30); % ==
K2 = KB(1:30,1:30); % |
sample images
K3 = KC(1:30,1:30); % |
K4 = KD(1:30,1:30); % |
K5 = KE(1:30,1:30); % |
K6 = KF(1:30,1:30); % ==
K7 = KG(1:30,1:30);
K8 = KH(1:30,1:30);
K9 = KI(1:30,1:30);

Ntux1 = tso(K1);
KS1 = hist(Ntux1, 6560);
Ntux2 = tso(K2);
KS2 = hist(Ntux2, 6560);
Ntux3 = tso(K3);
KS3 = hist(Ntux3, 6560);
Ntux4 = tso(K4);
KS4 = hist(Ntux4, 6560);
Ntux5 = tso(K5);
KS5 = hist(Ntux5, 6560);
Ntux6 = tso(K6);
KS6 = hist(Ntux6, 6560);
Ntux7=tso(K7);
KS7=hist(Ntux7,6560);
Ntux8=tso(K8);
KS8=hist(Ntux8,6560);
Ntux9=tso(K9);
KS9=hist(Ntux9,6560);

figure(31)
h1=subplot(1,1,1);
S1=plot(KS1);
set(h1,'Ytick',[]);
ylabel('Occurences')

figure(32)
h2=subplot(1,1,1);
S2=plot(KS2);
set(h2,'Ytick',[]);
ylabel('Occurences')

figure(33)
h3=subplot(1,1,1);
S3=plot(KS3);
set(h3,'Ytick',[]);
ylabel('Occurences')

figure(34)
h4=subplot(1,1,1);
S4=plot(KS4);
set(h4,'Ytick',[]);
ylabel('Occurences')

figure(35)
h5=subplot(1,1,1);
S5=plot(KS5);
set(h5,'Ytick',[]);
ylabel('Occurences')

figure(36)
h6=subplot(1,1,1);
S6=plot(KS6);
set(h6,'Ytick',[]);
ylabel('Occurences')

figure(37)
h7=subplot(1,1,1);
S7=plot(KS7);
set(h7,'Ytick',[]);
ylabel('Occurences')

figure(38)
h8=subplot(1,1,1);
S8=plot(KS8);
set(h8,'Ytick',[]);
ylabel('Occurences')

figure(39)
h9 = subplot(1,1,1);
S9 = plot(KS9);
set(h9, 'Ytick', []);
ylabel('Occurrences')

Zc = size(K);
si = 1;
ji = 1;

while ji < (Zc(1) - 29)
    ki = 1;
    while ki < (Zc(2) - 29)
        KN = K(ji:ji + 29, ki:ki + 29);
        Ntux = tso(KN);
        HS = hist(Ntux, 6560);
        Ntua(si,:) = HS;
        Sx = [ji ki];
        Sxi(si,:) = Sx;
        si = si + 1;
        ki = ki + 4;
    end
    ji = ji + 4;
end

% Query Image Classification

if true
    gFv = KS1;
    d = distfcm(gFvs, gFv) % Eucleidian Distance Calculation
    [du J] = sort(d) % Sorting the evaluated distance for min.
    distance calculation.
end

Ind1 = Sxi(J(1),:);
Ind2 = Sxi(J(2),:);
Ind3 = Sxi(J(3),:);
Ind4 = Sxi(J(4),:);
Ind5 = Sxi(J(5),:);
Ind6 = Sxi(J(6),:);
Ind7 = Sxi(J(7),:);
Ind8 = Sxi(J(8),:);
Ind9 = Sxi(J(9),:);
Ind10 = Sxi(J(10),:);

figure(11)
subplot(5,2,1), imshow(KX(Ind1(1):Ind1(1)+29, Ind1(2):Ind1(2)+29))
subplot(5,2,2), imshow(KX(Ind2(1):Ind2(1)+29, Ind2(2):Ind2(2)+29))
subplot(5,2,3), imshow(KX(Ind3(1):Ind3(1)+29, Ind3(2):Ind3(2)+29))
subplot(5,2,4), imshow(KX(Ind4(1):Ind4(1)+29, Ind4(2):Ind4(2)+29))
subplot(5,2,5), imshow(KX(Ind5(1):Ind5(1)+29, Ind5(2):Ind5(2)+29))
subplot(5,2,6), imshow(KX(Ind6(1):Ind6(1)+29, Ind6(2):Ind6(2)+29))
function ss=entropymain(img,clusters)
% To count the number of pixels Segmented in first texture image
% To calculate the percentage of first image
count1(1,clusters)=0;
percentage1(1,clusters)=0;

% To count the number of pixels Segmented in second texture image
% To calculate the percentage of second image
count2(1,clusters)=0;
percentage2(1,clusters)=0;

% To count the number of pixels Segmented in third texture image
% To calculate the percentage of third image
count3(1,clusters)=0;
percentage3(1,clusters)=0;

% To count the number of pixels in fourth texture image
% To calculate the percentage of fourth image
count4(1,clusters)=0;
percentage4(1,clusters)=0;

% To count the number of pixels Segmented in fifth texture image
% To calculate the percentage of fifth image
count5(1,clusters)=0;
percentage5(1,clusters)=0;

% To count the number of pixels Segmented in sixth texture image
% To calculate the percentage of sixth image
count6(1,clusters)=0;
percentage6(1,clusters)=0;

% define the color combinations for coloring the labels
c=['0 0 0; 0 0 1; 0 1 0; 0 1 1; 1 0 0; 1 0 1; 1 1 0; 1 1 1;'];

% find the size of the image
[x y z]=size(img);
[m n f]= size(img);

if (f==3)
img = (double(img(:, :, 1)) + double(img(:, :, 2)) + double(img(:, :, 3)))/3;
else
    img = double(img);
end

% copy the image from img to aa
aa = img;
[m1 n1] = size(img);

% initialize the size of the window to move vertically
k = 1;
l = 30;
q = 1;

while l <= m1
    q1 = 1;
    % initialize the size of the window to move horizontally
    m = 1;
n = 30;

    while n <= n1
        s = aa(k:l, m:n, 1);
        % call the function Entropy
        filterimage(q, q1) = entropy(s);
        q1 = q1 + 1;

        % increase the size of the window to move horizontally
        m = m + 3;
n = n + 3;
        end

    q = q + 1;

    % increase the size of the window to move vertically
    k = k + 3;
l = l + 3;
    end

q2 = q - 1;
q3 = q1 - 1;

% FCM clustering

% reshape the image into one column to perform FCM clustering
patt = reshape(filterimage,q2*q3,1);

% call the FCM function
[c u o] = fcm(patt,clusters);
dummy J]max(u);

% reshape the labeled image to the original size of the image
ss=reshape(J,q2,q3);

% resize the labeled image to the original size of the image
ss=resizem(ss, [x y]);

% display the image
figure,imshow(ss,ct);

% count the number of pixels Segmented in first texture image
for i=1:256
    for j=1:256
        for k=1:clusters
            if ss(i,j)==k
                count1(i,k)=count1(i,k)+1;
            end
        end
    end
end

% calculate the percentage of first image
for l=1:clusters
    percentage1(l,1)=(count1(1,l)/(256*256))*100;
end

% count the number of pixels Segmented in second texture image
for i=1:256
    for j=257:512
        for k=1:clusters
            if ss(i,j)==k
                count2(i,k)=count2(i,k)+1;
            end
        end
    end
end

% calculate the percentage of second image
for l=1:clusters
    percentage2(1,l)=(count2(1,l)/(256*256))*100;
end

% count the number of pixels Segmented in third texture image
for i=1:256
    for j=513:768
        for k=1:clusters
            if ss(i,j)==k
                count3(i,k)=count3(i,k)+1;
            end
        end
    end
end

% calculate the percentage of third image
for l=1:clusters
    percentage3(1,l)=(count3(1,l)/(256*256))*100;
end
count3(1,k)=count3(1,k)+1;
end
end
end

% calculate the percentage of third image
for l=1:clusters
    percentage3(1,l)=(count3(1,l)/(256*256))*100;
end

% count the number of pixels Segmented in fourth texture image
for i=257:512
    for j=1:256
        for k=1:clusters
            if ss(i,j)==k
                count4(1,k)=count4(1,k)+1;
            end
        end
    end
end

% calculate the percentage of fourth image
for l=1:clusters
    percentage4(1,l)=(count4(1,l)/(256*256))*100;
end

% count the number of pixels Segmented in fifth texture image
for i=257:512
    for j=257:512
        for k=1:clusters
            if ss(i,j)==k
                count5(1,k)=count5(1,k)+1;
            end
        end
    end
end

% calculate the percentage of fifth image
for l=1:clusters
    percentage5(1,l)=(count5(1,l)/(256*256))*100;
end

% count the number of pixels Segmented in sixth texture image
for i=257:512
    for j=513:768
        for k=1:clusters
            if ss(i,j)==k
                count6(1,k)=count6(1,k)+1;
            end
        end
    end
end
% calculate the percentage of sixth image
for l=1:clusters
    percentage6(1,l)=(count6(1,l)/(256*256))*100;
end

% display percentages
percentage1
    percentage2
percentage3
    percentage4
percentage5
    percentage6