function [feat\_val,im\_processed] = featureextraction(i)

i2=imresize(i,[512 ,512]);

i3=rgb2gray(i2);

i3=im2double(i3);

i3=im2bw(i3); %converting image to black and white

i3 = bwmorph(~i3, 'thin', inf); %thining the image

i3=~i3;

im1 = i3;

%extracting the black pixels

k=1;

for i=1:512

for j=1:512

if(i3(i,j)==0)

u(k)=i;

v(k)=j;

k=k+1;

i3(i,j)=1;

end

end

end

c=[u;v];

n=k-1;%the number of pixels in the signature

oub=sum(c(1,:))/n; %the original x co-ordinate center of mass of the image

ovb=sum(c(2,:))/n; %the original y co-ordinate center of mass of the image

%rotate

%moving the signature to the origin

for i=1:n

u(i)=u(i)-oub+1;

v(i)=v(i)-ovb+1;

end

% the new curve of the signature

c=[u;v];

ub=sum(c(1,:))/n;

vb=sum(c(2,:))/n;

ubsq=sum((c(1,:)-ub).^2)/n;

vbsq=sum((c(2,:)-vb).^2)/n;

for i=1:n

uv(i)=u(i)\*v(i);

end

uvb=sum(uv)/n;

m=[ubsq uvb;

uvb vbsq];

%calculating minimum igen value of the matrix

minigen=min(abs(eig(m)));

%the eigen vector

mi=[ubsq-minigen uvb;uvb vbsq-minigen];

theta=(atan((-mi(1))/mi(2))\*180)/pi;

thetarad=(theta\*pi)/180;

rotmat=[cos(thetarad) -sin(thetarad);

sin(thetarad) cos(thetarad)];

%% rotating the signature and passing the new co-ordinates

for i=1:n

v(i)=(c(2,i)\*cos(thetarad))-(c(1,i)\*sin(thetarad));

u(i)=(c(2,i)\*sin(thetarad))+(c(1,i)\*cos(thetarad));

end

c=[u;v];

%moving the signature to its original position

for i=1:n

u(i)=round(u(i)+oub-1);

v(i)=round(v(i)+ovb-1);

end

%after rotating the image the signature might go out of the boundry (128x128) therefore

%we have to move the signature curve

mx=0; %the moving x co-ordinate

my=0; %the moving y co-ordinate

if (min(u)<0)

mx=-min(u);

for i=1:n

u(i)=u(i)+mx+1;

end

end

if (min(v)<0)

my=-min(v);

for i=1:n

v(i)=v(i)+my+1;

end

end

c=[u;v];

for i=1:n

i3((u(i)),(v(i)))=0;

end

% removing extra white space in sides

xstart=512;

xend=1;

ystart=512;

yend=1;

for r=1:512

for c=1:512

if((i3(r,c)==0))

if (r<ystart)

ystart=r;

end

if((r>yend))

yend=r;

end

if (c<xstart)

xstart=c;

end

if (c>xend)

xend=c;

end

end

end

end

%cutting the image and copying it to another matrix

for i=ystart:yend

for j=xstart:xend

im((i-ystart+1),(j-xstart+1))=i3(i,j);

end

end

im\_processed = im;

% feature extraction - nsa

pixelb = 0;

pixela = 0;

for i=ystart:yend

for j=xstart:xend

if (im(i-ystart+1,j-xstart+1)== 0)

pixelb = pixelb + 1;

end

pixela = pixela + 1;

end

end

nsa = pixelb/pixela;

% feature extration - aspect ratio

height\_sign = yend-ystart;

length\_sign = xend-xstart;

aspect\_ratio = length\_sign/height\_sign;

% feature extraction - maximum horizontal and vertical projection

max=0;

for i=ystart:yend

summ=0;

for j=xstart:xend

if(im((i-ystart+1),(j-xstart+1))==0)

summ=summ+1;

end

end

if (summ>max)

max=summ;

end

end

max;

max1=0;

for i=xstart:xend

summ=0;

for j=ystart:yend

if(im((j-ystart+1),(i-xstart+1))==0)

summ=summ+1;

end

end

if (summ>max1)

max1=summ;

end

end

max1;

xdiff=xend-xstart;

ydiff=yend-ystart;

hor\_proj = max/xdiff;

ver\_proj = max1/ydiff;

% feature extraction end points

i1 = im1;

disp('i1 matrix');

disp(i1);

disp(size(i1));

[row, col, depth] = size(i1);

%add row%

addrow = ones(1, col);

i1 = [addrow; addrow; i1; addrow];

[row, col, depth] = size(i1);

%add column%

addcol = ones(row, 1);

i1 = horzcat(addcol, i1, addcol, addcol);

[row, col, depth] = size(i1);

i1=~i1;

crosspoints=0;

for r = 3:row-1

for c = 2:col-2

if(i1(r,c)==1)

if (i1(r-1,c-1)+i1(r-1,c)+i1(r-1,c+1)+i1(r,c-1)+i1(r,c+1)+i1(r+1,c-1)+i1(r+1,c)+i1(r+1,c+1)==1)

crosspoints=crosspoints+1;

%disp(i1(r,c))

end

end

end

end

% feature extraction center of gravities of the vertically divided images

n1 = im(:, 1: xdiff/2);%splitting images

n2 = im(:, xdiff/2+1:xdiff);

sum1=0;

pix\_total=0;

%for the first half

for i=1:ydiff

pix\_sum=0;

for j=1:xdiff/2

if(n1(i,j)==0)

pix\_sum=pix\_sum+1;

pix\_total=pix\_total+1;

end

end

sum1=sum1+(pix\_sum\*i);

end

y1=sum1/pix\_total;

ry1=y1/ydiff;

sum1=0;

for i=1:xdiff/2

pix\_sum=0;

for j=1:ydiff

if(n1(j,i)==0)

pix\_sum=pix\_sum+1;

end

end

sum1=sum1+(pix\_sum\*i);

end

x1=sum1/pix\_total;

rx1=2\*x1/xdiff;

%for the secong half

sum1=0;

pix\_total=0;

for i=1:ydiff

pix\_sum=0;

for j=1:xdiff/2

if(n2(i,j)==0)

pix\_sum=pix\_sum+1;

pix\_total=pix\_total+1;

end

end

sum1=sum1+(pix\_sum\*i);

end

y2=sum1/pix\_total;

ry2=y2/ydiff;

sum1=0;

for i=1:xdiff/2

pix\_sum=0;

for j=1:ydiff

if(n2(j,i)==0)

pix\_sum=pix\_sum+1;

end

end

sum1=sum1+(pix\_sum\*i);

end

x2=sum1/pix\_total;

rx2=2\*x2/xdiff;

centroid = [ [rx1 ry1] [rx2 ry2] ];

% feature extraction - slope

m=xdiff;

m=m/2;

k=m+x2;

slope=(y2-y1)/(k-x1);

feat\_val = [ nsa aspect\_ratio hor\_proj crosspoints centroid slope];

end