**Automatic License Plate Recognition**

**PlateLocalization.m**

clc; % Clear command window.

clear all; % Delete all variables.

close all; % Close all figure windows except those created by imtool.

cd 'C:\Octave\Octave-4.2.1\share\octave\packages\image-2.6.1';

pkg load image;

cd 'H:\Octave';

% Read Image

I = imread ('img17.jpg');

figure(1);

imshow(I);

% Extract Y component (Convert an Image to Gray)

I = imresize( I , [NaN 450]);

Igray = rgb2gray(I);

[rows cols] = size(Igray);

figure(2);

imshow(Igray);

% Remove Noise through Median Filter

Igray = medfilt2( Igray, [1 1] );

figure(3);

imshow(Igray);

I = Igray;

difference = 0;

sum = 0;

total\_sum = 0;

difference = uint32(difference);

%% PROCESS EDGES IN HORIZONTAL DIRECTION

disp('Processing Edges Horizontally...');

max\_horz = 0;

maximum = 0;

for i = 2:cols

sum = 0;

for j = 2:rows

difference = abs(uint32(I(j, i) - I(j-1, i)));

if(difference > 20)

sum = sum + difference;

end

end

horz1(i) = sum;

% Find Peak Value

if(sum > maximum)

max\_horz = i;

maximum = sum;

end

total\_sum = total\_sum + sum;

end

average = total\_sum / cols;

figure(4);

% Plot the Histogram for analysis

subplot(3,1,1);

plot (horz1);

title('Horizontal Edge Processing Histogram');

xlabel('Column Number ->');

ylabel('Difference ->');

%% Smoothen the Horizontal Histogram by applying Low Pass Filter

sum = 0;

horz = horz1;

for i = 21:(cols-21)

sum = 0;

for j = (i-20):(i+20)

sum = sum + horz1(j);

end

horz(i) = sum / 41;

end

subplot(3,1,2);

plot (horz);

title('Histogram after passing through Low Pass Filter');

xlabel('Column Number ->');

ylabel('Difference ->');

%% Filter out Horizontal Histogram Values by applying Dynamic Threshold

disp('Filter out Horizontal Histogram...');

for i = 1:cols

if(horz(i) < average)

horz(i) = 0;

for j = 1:rows

I(j, i) = 0;

end

end

end

subplot(3,1,3);

plot (horz);

title('Histogram after Filtering');

xlabel('Column Number ->');

ylabel('Difference ->');

%% PROCESS EDGES IN VERTICAL DIRECTION

difference = 0;

total\_sum = 0;

difference = uint32(difference);

disp('Processing Edges Vertically...');

maximum = 0;

max\_vert = 0;

for i = 2:rows

sum = 0;

for j = 2:cols %cols

difference = abs(uint32(I(i, j) - I(i, j-1)));

if(difference > 20)

sum = sum + difference;

end

end

vert1(i) = sum;

%% Find Peak in Vertical Histogram

if(sum > maximum)

max\_vert = i;

maximum = sum;

end

total\_sum = total\_sum + sum;

end

average = total\_sum / rows;

figure(5)

subplot(3,1,1);

plot (vert1);

title('Vertical Edge Processing Histogram');

xlabel('Row Number ->');

ylabel('Difference ->');

%% Smoothen the Vertical Histogram by applying Low Pass Filter

disp('Passing Vertical Histogram through Low Pass Filter...');

sum = 0;

vert = vert1;

for i = 21:(rows-21)

sum = 0;

for j = (i-20):(i+20)

sum = sum + vert1(j);

end

vert(i) = sum / 41;

end

subplot(3,1,2);

plot (vert);

title('Histogram after passing through Low Pass Filter');

xlabel('Row Number ->');

ylabel('Difference ->');

%% Filter out Vertical Histogram Values by applying Dynamic Threshold

disp('Filter out Vertical Histogram...');

for i = 1:rows

if(vert(i) < average)

vert(i) = 0;

for j = 1:cols

I(i, j) = 0;

end

end

end

subplot(3,1,3);

plot (vert);

title('Histogram after Filtering');

xlabel('Row Number ->');

ylabel('Difference ->');

figure(6), imshow(I);

%% Find Probable candidates for Number Plate

j = 1;

for i = 2:cols-2

if(horz(i) ~= 0 && horz(i-1) == 0 && horz(i+1) == 0)

column(j) = i;

column(j+1) = i;

j = j + 2;

elseif((horz(i) ~= 0 && horz(i-1) == 0) || (horz(i) ~= 0 && horz(i+1) == 0))

column(j) = i;

j = j+1;

end

end

j = 1;

for i = 2:rows-2

if(vert(i) ~= 0 && vert(i-1) == 0 && vert(i+1) == 0)

row(j) = i;

row(j+1) = i;

j = j + 2;

elseif((vert(i) ~= 0 && vert(i-1) == 0) || (vert(i) ~= 0 && vert(i+1) == 0))

row(j) = i;

j = j+1;

end

end

[temp column\_size] = size (column);

if(mod(column\_size, 2))

column(column\_size+1) = cols;

end

[temp row\_size] = size (row);

if(mod(row\_size, 2))

row(row\_size+1) = rows;

end

%% Region of Interest Extraction

%Check each probable candidate

for i = 1:2:row\_size

for j = 1:2:column\_size

% If it is not the most probable region remove it from image

if(~((max\_horz >= column(j) && max\_horz <= column(j+1)) && (max\_vert >=row(i) && max\_vert <= row(i+1))))

%This loop is only for displaying proper output to User

for m = row(i):row(i+1)

for n = column(j):column(j+1)

I(m, n) = 0;

end

end

end

end

end

I = CropPlate(I);

figure(7);

imshow(I);

**CropPlate.m**

function I = CropPlate(I)

Ied = edge( I , 'Sobel');

stats = regionprops( Ied , 'BoundingBox' );

max\_area = 0;

for i = 1: length(stats)

pos = stats(i).BoundingBox;

area = pos(2) \* pos(3);

if( area > max\_area)

max\_area = area;

x = pos(1);

y = pos(2);

w = pos(3);

h = pos(4);

end

end

I = imcrop( I , [x y w h] );

end

**CharacterSegmentation.m**

close all;

figure(1);

imshow(I);

I = imresize(I , [100 NaN]);

figure(2);

imshow(I);

se = strel ('disk' , 1, 0 );

Idil = imdilate ( I , se );

Ierod = imerode ( I , se );

Isub = imsubtract ( Idil , Ierod );

Isub = mat2gray ( Isub );

Iconv = conv2 ( Isub , [1 1 1;1 1 1;1 1 1] );

figure(3);

imshow(Iconv);

I = mat2gray ( Iconv );

I = imadjust(I ,[0.2 0.4],[0 1],1);

I = round(I);

I = bwmorph(I , 'thin');

I = bwareaopen(I , 70);

figure(4);

imshow(I);

TemplateCreation;

load NewTemplates;

NewTemplates;

[r c] = size(I);

stats = regionprops( I , 'BoundingBox' );

k = 5;

word=[];

i = 0;

for j = 1:length(stats)

pos = stats(j).BoundingBox;

if ( pos(3) \* pos(4) < (r \* c)/10 )

if ( (pos(3) >= 12) && (pos(4) > 40) )

Icrop = imcrop( I , pos );

Icrop = imfill( Icrop, 'holes');

for n = 1 : 2

Icrop = bwmorph( Icrop, 'thin');

Icrop = bwmorph( Icrop, 'skel');

Icrop = bwmorph( Icrop, 'spur');

Icrop = bwmorph( Icrop, 'bridge');

end

Icrop = CropLetters(Icrop);

Icrop = imresize( Icrop , [42 24]);

figure(k++);

imshow(Icrop);

letter=TemplateMatching(Icrop,NewTemplates);

word=[word letter];

end

end

end

fid = fopen('noPlate.txt', 'wt'); % This portion of code writes the number plate

fprintf(fid,'%s\n',word); % to the text file, if executed a notepad file with the

fclose(fid); % name noPlate.txt will be open with the number plate written.

winopen('noPlate.txt');

**TemplateCreation.m**

function TemplateCreation

%CREATE TEMPLATES

%Letter

A=imread('char/A.bmp');B=imread('char/B.bmp');

C=imread('char/C.bmp');D=imread('char/D.bmp');

E=imread('char/E.bmp');F=imread('char/F.bmp');

G=imread('char/G.bmp');H=imread('char/H.bmp');

I=imread('char/I.bmp');J=imread('char/J.bmp');

K=imread('char/K.bmp');L=imread('char/L.bmp');

M=imread('char/M.bmp');N=imread('char/N.bmp');

O=imread('char/O.bmp');P=imread('char/P.bmp');

Q=imread('char/Q.bmp');R=imread('char/R.bmp');

S=imread('char/S.bmp');T=imread('char/T.bmp');

U=imread('char/U.bmp');V=imread('char/V.bmp');

W=imread('char/W.bmp');X=imread('char/X.bmp');

Y=imread('char/Y.bmp');Z=imread('char/Z.bmp');

Afill=imread('char/fillA.bmp');

Bfill=imread('char/fillB.bmp');

Dfill=imread('char/fillD.bmp');

Ofill=imread('char/fillO.bmp');

Pfill=imread('char/fillP.bmp');

Qfill=imread('char/fillQ.bmp');

Rfill=imread('char/fillR.bmp');

%Number

one=imread('char/1.bmp'); two=imread('char/2.bmp');

three=imread('char/3.bmp');four=imread('char/4.bmp');

five=imread('char/5.bmp'); six=imread('char/6.bmp');

seven=imread('char/7.bmp');eight=imread('char/8.bmp');

nine=imread('char/9.bmp'); zero=imread('char/0.bmp');

zerofill=imread('char/fill0.bmp');

fourfill=imread('char/fill4.bmp');

sixfill=imread('char/fill6.bmp');

sixfill2=imread('char/fill6\_2.bmp');

eightfill=imread('char/fill8.bmp');

ninefill=imread('char/fill9.bmp');

ninefill2=imread('char/fill9\_2.bmp');

letter=[A Afill B Bfill C D Dfill E F G H I J K L M N O Ofill P Pfill Q Qfill R Rfill S T U V W X Y Z];

number=[one two three four fourfill five six sixfill sixfill2 seven eight eightfill nine ninefill ninefill2 zero zerofill];

character=[letter number];

NewTemplates=mat2cell(character,42,[24 24 24 24 24 24 24 24 ...

24 24 24 24 24 24 24 ...

24 24 24 24 24 24 24 ...

24 24 24 24 24 24 24 ...

24 24 24 24 24 24 24 ...

24 24 24 24 24 24 24 ...

24 24 24 24 24 24 24]);

save ('NewTemplates','NewTemplates')

clear all

end

**CropLetters.m**

function I = CropLetters(I)

stats = regionprops( I , 'BoundingBox' );

max\_area = 0;

for i = 1: length(stats)

pos = stats(i).BoundingBox;

area = pos(2) \* pos(3);

if( area > max\_area)

max\_area = area;

x = pos(1);

y = pos(2);

w = pos(3);

h = pos(4);

end

end

I = imcrop( I , [x y w h] );

end

**TemplateMatching.m**

function letter= TemplateMatching(imgChar,NewTemplates)

C=[];

for n=1:50

sem=corr2(real(cell2mat(NewTemplates(1,n))),real(imgChar));

C=[C sem];

end

vd=find(C==max(C));

if vd==1 || vd==2

letter='A';

elseif vd==3 || vd==4

letter='B';

elseif vd==5

letter='C';

elseif vd==6 || vd==7

letter='D';

elseif vd==8

letter='E';

elseif vd==9

letter='F';

elseif vd==10

letter='G';

elseif vd==11

letter='H';

elseif vd==12

letter='I';

elseif vd==13

letter='3';

elseif vd==14

letter='K';

elseif vd==15

letter='L';

elseif vd==16

letter='M';

elseif vd==17

letter='N';

elseif vd==18 || vd==19

letter='O';

elseif vd==20 || vd==21

letter='P';

elseif vd==22 || vd==23

letter='Q';

elseif vd==24 || vd==25

letter='R';

elseif vd==26

letter='S';

elseif vd==27

letter='T';

elseif vd==28

letter='U';

elseif vd==29

letter='V';

elseif vd==30

letter='W';

elseif vd==31

letter='X';

elseif vd==32

letter='Y';

elseif vd==33

letter='2';

%\*-\*-\*-\*-\*

% Numerals listings.

elseif vd==34

letter='1';

elseif vd==35

letter='2';

elseif vd==36

letter='3';

elseif vd==37 || vd==38

letter='4';

elseif vd==39

letter='5';

elseif vd==40 || vd==41 || vd==42

letter='6';

elseif vd==43

letter='7';

elseif vd==44 || vd==45

letter='8';

elseif vd==46 || vd==47 || vd==48

letter='9';

else

letter='0';

end

end