**UE19EC317: Digital Image Processing**

Assignment

TITLE: CAPTURING OF LICENSE PLATE OF VEHICLES IN SINGLE PLANE TRAFFIC

Introduction to the topic:

Maintaining the order in traffic in a country like India which densely populated is not easy. The traffic has to be heavily monitored in order to maintain the order. This monitoring is usually done by motion sensing cameras and these cameras have to be constantly maintained by people. They are expensive but still end up producing images with degradation factors such as noise and blur. Sometimes we don’t end up having the budget for these high quality cameras. Our country certainly does not have the expenditure too squander over reliable, accurate imaging devices such as these, which still fail sometimes due to classic under-information.So, our project aims to provide a solution to this by capture the license plate of vehicles in the single lane traffic without the use of highly expensive motion sensing cameras. This would ensure that we can take a video of the required license plate, with even a meagre spectral imaging device, and the program would still create frames for it and we can have a higher chance at the accuracy of detection of the numbers, all in a completely cost-efficient manner.

Given below are the codes that we have used for the project.

THEORETICAL DETAILS:

The code consists of four files which help us in recognizing the license plate of the vehicle in one way lane traffic.

The algorithm we have followed here is, the final.m will process the video information of the given license plate as a source and generate frames accordingly.

The basic understanding behind the code functionality of the lic.m file is that it generates the final image to be analysed. It first converts the image to grayscale and does horizontal histogram edge processing then proceeds to process the edges of the image vertically by passing the vertical edge histogram through a low pass filter, finally get the filtered histogram and use it to generate the final image to be analyzed which is *new.jpg*. This new image will then be placed in the folder file of the optical recognition code- ocr.m which will use this *new.jpg* to display the input image with noise and then generate a text file identifying the numbers in the given license plate identified by the program with good accuracy. The create\_templates.m file aims to create certain templates to the numbers and letters assigned to the image electromagnetic values as analyzed by the program given in the read\_letter.m and the lines.m code. These templates facilitate easy assignment and identification in the final ocr.m code.

***CODE:***

>> Lines.m:

function [fl re]=lines(im\_texto)

% Divide text in lines

% im\_texto->input image; fl->first line; re->remain line

% Example:

% im\_texto=imread('TEST\_3.jpg');

% [fl re]=lines(im\_texto);

% subplot(3,1,1);imshow(im\_texto);title('INPUT IMAGE')

% subplot(3,1,2);imshow(fl);title('FIRST LINE')

% subplot(3,1,3);imshow(re);title('REMAIN LINES')

im\_texto=clip(im\_texto);

num\_filas=size(im\_texto,1);

for s=1:num\_filas

if sum(im\_texto(s,:))==0

nm=im\_texto(1:s-1, :); % First line matrix

rm=im\_texto(s:end, :);% Remain line matrix

fl = clip(nm);

re=clip(rm);

%\*-\*-\*Uncomment lines below to see the result\*-\*-\*-\*-

% subplot(2,1,1);imshow(fl);

% subplot(2,1,2);imshow(re);

break

else

fl=im\_texto;%Only one line.

re=[ ];

end

end

function img\_out=clip(img\_in)

[f c]=find(img\_in);

img\_out=img\_in(min(f):max(f),min(c):max(c));%Crops image

>> create\_templates.m:

%CREATE TEMPLATES

%Letter

A=imread('letters\_numbers\A.bmp');B=imread('letters\_numbers\B.bmp');

C=imread('letters\_numbers\C.bmp');D=imread('letters\_numbers\D.bmp');

E=imread('letters\_numbers\E.bmp');F=imread('letters\_numbers\F.bmp');

G=imread('letters\_numbers\G.bmp');H=imread('letters\_numbers\H.bmp');

I=imread('letters\_numbers\I.bmp');J=imread('letters\_numbers\J.bmp');

K=imread('letters\_numbers\K.bmp');L=imread('letters\_numbers\L.bmp');

M=imread('letters\_numbers\M.bmp');N=imread('letters\_numbers\N.bmp');

O=imread('letters\_numbers\O.bmp');P=imread('letters\_numbers\P.bmp');

Q=imread('letters\_numbers\Q.bmp');R=imread('letters\_numbers\R.bmp');

S=imread('letters\_numbers\S.bmp');T=imread('letters\_numbers\T.bmp');

U=imread('letters\_numbers\U.bmp');V=imread('letters\_numbers\V.bmp');

W=imread('letters\_numbers\W.bmp');X=imread('letters\_numbers\X.bmp');

Y=imread('letters\_numbers\Y.bmp');Z=imread('letters\_numbers\Z.bmp');

%Number

one=imread('letters\_numbers\1.bmp'); two=imread('letters\_numbers\2.bmp');

three=imread('letters\_numbers\3.bmp');four=imread('letters\_numbers\4.bmp');

five=imread('letters\_numbers\5.bmp'); six=imread('letters\_numbers\6.bmp');

seven=imread('letters\_numbers\7.bmp');eight=imread('letters\_numbers\8.bmp');

nine=imread('letters\_numbers\9.bmp'); zero=imread('letters\_numbers\0.bmp');

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

letter=[A B C D E F G H I J K L M...

N O P Q R S T U V W X Y Z];

number=[one two three four five...

six seven eight nine zero];

character=[letter number];

templates=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]);

save ('templates','templates')

clear all

>> OCR.m:

imagen =~im2bw(imagen,threshold);

% Remove all object containing fewer than 30 pixels

imagen = bwareaopen(imagen,30);

%Storage matrix word from image

word=[ ];

re=imagen;

%Opens text.txt as file for write

fid = fopen('text.txt', 'wt');

% Load templates

load templates

global templates

% Compute the number of letters in template file

num\_letras=size(templates,2);

while 1

%Fcn 'lines' separate lines in text

[fl re]=lines(re);

imgn=fl;

%Uncomment line below to see lines one by one

%imshow(fl);pause(0.5)

%-----------------------------------------------------------------

% Label and count connected components

[L Ne] = bwlabel(imgn);

for n=1:Ne

[r,c] = find(L==n);

% Extract letter

n1=imgn(min(r):max(r),min(c):max(c));

% Resize letter (same size of template)

img\_r=imresize(n1,[42 24]);

%Uncomment line below to see letters one by one

%imshow(img\_r);pause(0.5)

%-------------------------------------------------------------------

% Call fcn to convert image to text

letter=read\_letter(img\_r,num\_letras);

% Letter concatenation

word=[word letter];

end

%fprintf(fid,'%s\n',lower(word));%Write 'word' in text file (lower)

fprintf(fid,'%s\n',word);%Write 'word' in text file (upper)

% Clear 'word' variable

word=[ ];

%\*When the sentences finish, breaks the loop

if isempty(re) %See variable 're' in Fcn 'lines'

break

end

end

fclose(fid);

%Open 'text.txt' file

winopen('text.txt')

fprintf('For more information, visit: <a href= "http://www.matpic.com">www.matpic.com </a> \n')

clear all

>> read\_letter.m:

function letter=read\_letter(imagn,num\_letras)

% Computes the correlation between template and input image

% and its output is a string containing the letter.

% Size of 'imagn' must be 42 x 24 pixels

% Example:

% imagn=imread('D.bmp');

% letter=read\_letter(imagn)

global templates

comp=[ ];

for n=1:num\_letras

sem=corr2(templates{1,n},imagn);

comp=[comp sem];

end

vd=find(comp==max(comp));

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

if vd==1

letter='A';

elseif vd==2

letter='B';

elseif vd==3

letter='C';

elseif vd==4

letter='D';

elseif vd==5

letter='E';

elseif vd==6

letter='F';

elseif vd==7

letter='G';

elseif vd==8

letter='H';

elseif vd==9

letter='I';

elseif vd==10

letter='J';

elseif vd==11

letter='K';

elseif vd==12

letter='L';

elseif vd==13

letter='M';

elseif vd==14

letter='N';

elseif vd==15

letter='O';

elseif vd==16

letter='P';

elseif vd==17

letter='Q';

elseif vd==18

letter='R';

elseif vd==19

letter='S';

elseif vd==20

letter='T';

elseif vd==21

letter='U';

elseif vd==22

letter='V';

elseif vd==23

letter='W';

elseif vd==24

letter='X';

elseif vd==25

letter='Y';

elseif vd==26

letter='Z';

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

elseif vd==27

letter='1';

elseif vd==28

letter='2';

elseif vd==29

letter='3';

elseif vd==30

letter='4';

elseif vd==31

letter='5';

elseif vd==32

letter='6';

elseif vd==33

letter='7';

elseif vd==34

letter='8';

elseif vd==35

letter='9';

else

letter='0';

end

1. RESULTS:

















