

Text Recognition Applications

Al Rubaie Evan Madhi Hamzh

Department of Computer Science, University of Pitesti, Arges, Romania

Abstract: Recognition of written texts is one of the many subfields of pattern recognition. Recognition methods are based on driving a large number of symbols. These symbol images are grouped into databases that have mostly very high prices. In order to detect a symbol, we need to detect the direction of the cell where it is located. Although it seems simple, finally we get to solve a problem of pattern recognition to obtain driving data required for similar problem, the recognition of characters. Fortunately, the new problem is simpler and it can be approached using deterministic methods.

Keywords: Pattern recognition, text recognition, image detection.

I. IMAGE PROCESSING METHODS

1) Color filtering

In this paragraph, we will describe on short the techniques for transforming digital RGB images to grayscale or black and white. For converting the encoded intensity to RGB format into gray levels, we can use the following formulas:

- The arithmetic mean;
- Color rule.

After obtaining the image in shadows (levels) of gray, we can do the following conversion to black and white image. However, we have available several methods that differ by way of setting the threshold level between black and white; we shall discuss three of them.

- The first one, the simple one, consists in establishing a fixed threshold value in the mid range. Obviously, this simplistic approach may lead to inadequate results.
- The second one, more complicated sets the threshold intensity level between black and white by local analysis of the image and establishing depending on the intensity histogram of the threshold level.
- The third method employs a coplanar filter exploiting the co-planarity of the grayscale of neighboring pixels. The advantages of the method consist in removing noise, image a real smoothing and keeping sharp edges.

2) Gradient

The gradient is used to detect edges in an image. The gradient has intrinsic significance; intuitively, gradient indicates the direction of the steepest slope passing through the current point.

In case of digital pictures, gradient definition changes; meaning that we work with finite differences, therefore, the operations are carried out in a small window, for example 3x3 pixels or even 2x2 pixels. There are various methods for calculating the gradient, some of them departing pretty much from the rigorous mathematical definition thereof. Regarding the size of the gradient window calculation, it is noted that for the even sizes of its sides, the gradient value no longer achieved in the initial coordinates of the image, but between them.

3) Fourier transform (TF)

The Fourier transform was invented in 1807 by Jean Baptiste Joseph, Baron of Fourier. Its usefulness is given by the property of obtaining new features of a signal by turning it from space where it is located in the parametric area (space) of frequencies. The transform can be used for any periodic signal or not, analog or digital, with various variants of transformation for each case. Therefore, the Fourier transform achieves the frequency spectrum of a signal, and the reverse Fourier transform generates the signal in the frequencies spectrum.

Following the application of the FT on a signal, we achieve the amplitudes for the frequencies of the signal spectrum. The frequencies are given by the sine and cosine functions (or in exponential form, the imaginary exponentials). For even signals, the spectrum consists only of cosine functions and for odd signals, the spectrum contains only sine functions.

Returning to transform utility, it can be seen that it can be generally used for filtering signals and particularly digital images, for extracting the features that can be used to input other analysis and classification algorithms.

In terms of writing recognition, The Fourier transform can be used for image segmentation into rows and words.

4) Hough transform

Hough transform is used to detect lines in an image. Given a set of points in a two-dimensional Euclidean space with coordinates (x, y) , it is required to determine which of these points are collinear or not. To do this, we apply a transformation of the Euclidean space to the space of polar coordinates.

Because for every point (x, y) we have an infinity of pairs (r, θ) in the parameter space, we have to find a way to reduce the dimensionality of the problem for practical application. For this purpose, the problem is digitalized by digitalizing the parameter values. Thus, if the original image, in discrete form, had $N \times M$ pixels, we can digitalize the parameter space choosing, for example, $d\theta$

=1° and dr =1 pixel. The digitalization is done depending on the problem.

The next step consists in building an H matrix, where the line is the distance and the column is the angle. N and M are the size of the image. The matrix elements are initially 0. For all possible discrete values of r and θ, the corresponding lines are built and we have to count how many of the image points they pass. This number is the value of that cell.

Using this algorithm, finally, the matrix will include integers representing the number of points crossing each line. If there are cells with values exceeding or equal to three, it means that collinear points were found in the original image. Using this method, we can detect the orientation of lines in an image.

5) Segmentation

Segmentation is a very important step in the recognition process, as its success or failure depends upon it. It is performed at several levels, namely: segmenting the image into blocks of text, graphics and other artifacts, segmentation of text blocks into rows, rows into words and, most important, words into letters. It should be mentioned that the segmentation, beside the conventional processes to divide a graphic symbol into its constituent parts must also rebuild the symbols in these parts, symbols that must contain as much as possible all the parts belonging to them and none of the parts that do not belong to them, i.e. to separate each letters from its adjacent letters.

Since the process is very complex, a series of mathematical methods are used to achieve it. Among them, we can mention the following:

- Analytic geometry for converting bitmap symbols into vectors
- Statistics and probabilities – they are practically indispensable in almost every phase of the recognition process
- Artificial neural networks (ANN)
- BBN networks (Bayesian Belief Networks) – we have indicated the use of such networks in the segmentation process, as well as their benefits compared to simple Bayesian classifiers: decreasing the driving time by 55% and achieving a success rate of 86% in identifying segments' limits
- Metrics – these are used to remove the segments of Chinese characters

II. APPLICATION FOR TEXT RECOGNITION

text_recognition.m

```
% Text Recognition
% PROGRAMUL PRINCIPAL
% Image reading

imagen=imread('TEST_4.jpg');
```

```
% Image display
imshow(imagen);pause(1)

% Conversion to shadows of gray
if size(imagen,3)==3 % imagine RGB
    imagen=rgb2gray(imagen);
end

% Conversion to black and white
threshold = graythresh(imagen);
imagen =~im2bw(imagen,threshold);

% Removing all objects containing less than 30 pixels
imagen = bwareaopen(imagen,30);

%Storage matrix of the words in the image.
word=[ ];
re=imagen;

% We open the text.txt file for writing.
fid = fopen('text.txt', 'wt');

% We upload the templates
load templates
global templates

% We calculate the letters in the template
num_letras=size(templates,2);

while 1
    %The function 'lines' separates the lines in the text
    [fl re]=lines(re);
    imgn=fl;

    % Labeling and counting the connected components.
    [L Ne] = bwlabel(imgn);
    for n=1:Ne
        [r,c] = find(L==n);

        % Extracting the letter
        n1=imgn(min(r):max(r),min(c):max(c));

        % Resizing the letter to the template size
        img_r=imresize(n1,[42 24]);

        %Displaying the lters one by one
        imshow(img_r);pause(0.2)

        % Function to convert the image into text
        letter=read_letter(img_r,num_letras);

        % Letters concatenation
        word=[word letter];
    end

    % Writing in the file
    fprintf(fid,'%s\n',word);% Write 'word' in text file
    (upper)
    word=[ ];
end
```



```
if isempty(re)
    break
end
end
fclose(fid);

%We open the text.txt file
winopen('text.txt')

clear all

create_templates.m
%CREATE TEMPLATES
%Letter
A=imread('letters_numbers\A.bmp');B=imread('letters_nu
mbers\B.bmp');
C=imread('letters_numbers\C.bmp');D=imread('letters_nu
mbers\D.bmp');
E=imread('letters_numbers\E.bmp');F=imread('letters_nu
mbers\F.bmp');
G=imread('letters_numbers\G.bmp');H=imread('letters_nu
mbers\H.bmp');
I=imread('letters_numbers\I.bmp');J=imread('letters_numb
ers\J.bmp');
K=imread('letters_numbers\K.bmp');L=imread('letters_nu
mbers\L.bmp');
M=imread('letters_numbers\M.bmp');N=imread('letters_nu
mbers\N.bmp');
O=imread('letters_numbers\O.bmp');P=imread('letters_nu
mbers\P.bmp');
Q=imread('letters_numbers\Q.bmp');R=imread('letters_nu
mbers\R.bmp');
S=imread('letters_numbers\S.bmp');T=imread('letters_num
bers\T.bmp');
U=imread('letters_numbers\U.bmp');V=imread('letters_nu
mbers\V.bmp');
W=imread('letters_numbers\W.bmp');X=imread('letters_n
umbers\X.bmp');
Y=imread('letters_numbers\Y.bmp');Z=imread('letters_nu
mbers\Z.bmp');
%Number
one=imread('letters_numbers\1.bmp');
two=imread('letters_numbers\2.bmp');
three=imread('letters_numbers\3.bmp');four=imread('letter
s_numbers\4.bmp');
five=imread('letters_numbers\5.bmp');
six=imread('letters_numbers\6.bmp');
seven=imread('letters_numbers\7.bmp');eight=imread('lett
ers_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]);
save ('templates','templates')
clear all
```

lines.m

```
function [fl re]=lines(im_texto)
```

```
% Dividing the text into 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, :); % The first line of matrix
        rm=im_texto(s:end, :);
        fl = clip(nm);
        re=clip(rm);
        break
    else
        fl=im_texto;
        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));% Croppin
g the image
```

read_letter.m

```
function letter=read_letter(imagn,num_letras)
% Calculate the correlation between template and the
input image.
% and the result is a character string
```

```
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
```

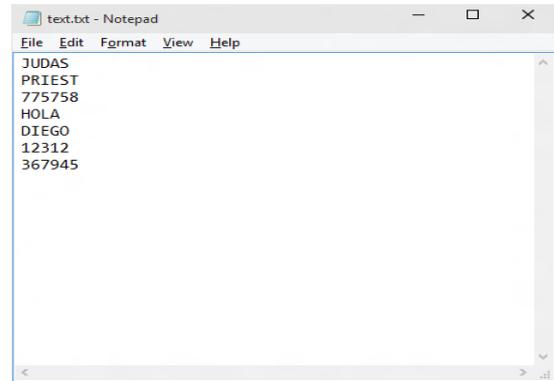
```

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
    
```

For the image:

JUDAS
 PRIEST
 775758
 HOLA
 DIEGO
 12312
 367945

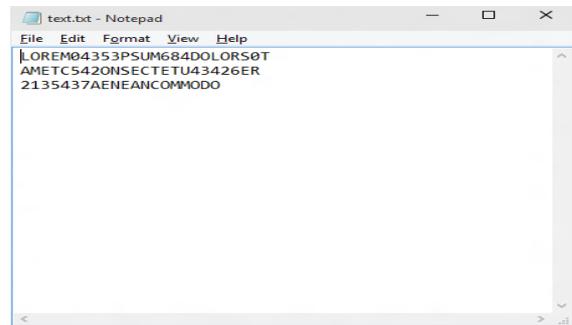
We have:



For the image:

LOREM I4353PSUM 684DOLOR SIT
 AMET C542ONSECTETU43426ER
 ADIP13ISCING ELIT52 5224
 2135437AENEAN COMMODO

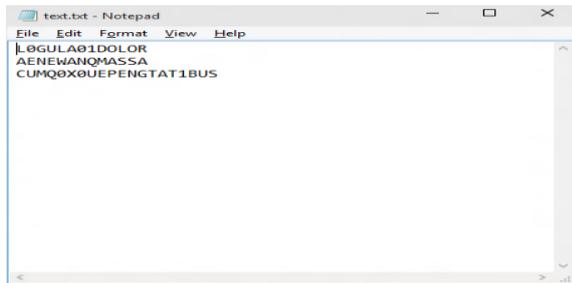
We have:



For the image:

LIGULA!| DOLOR.
 AENE*AN? MASSA.
 CUM Q%UE PEN@+ATIBUS

We have:



Babylon, College of Science Doctor - University of Pitesti
- Department of Computer Science
Hobby: Traveling, Reading

III. CONCLUSIONS

This paper contains an approach regarding text processing for model recognition. For each identification model of recognition accuracy algorithms were set based on advanced analysis techniques. The algorithm for text image recognition was developed using the analytical techniques. This algorithm improved the rate of text images recognition by using the information obtained from the evaluation of images text in applications text recognition.

REFERENCES

- [1] Constantin, D., State, L.: "A Comparative Analysis on a Class of Numerical Methods for Estimating the ICA Model", International Journal of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. III (2008), Suppl. issue: Proceedings of ICCCC 2008, pg. 218-222, Băile Felix, Oradea, Romania, 15-17 Mai, 2008.
- [2] Constantin, D., State, L.: "Successive Approximations - based Algorithm for Independent Component Analysis", Proceedings of the Third International Conference on Digital Telecommunications (ICDT 2008), IEEE Computer Society Press, Bucuresti, Romania, 29 Iunie - 5 Iulie, 2008.
- [3] Constantin, D., State, L.: "A Version of the FastICA Algorithm Based on the Secant Method combined with Simple Iterations Method", Proceedings of the 3rd International Conference on Image and Signal Processing 2008 (ICISP 2008), Image and Signal Processing, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Volume 5099/2008, Cherbourg-Octeville, Normandia, Franța, 1-3 Iulie, 2008.
- [4] Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis, John Wiley&Sons, 2001.
- [5] Hyvarinen, A., Oja, E.: "Independent component analysis: algorithms and applications", Elsevier Science, Neural Networks 13, 411-433, 2000.
- [6] Hyvarinen, A.: "Survey on Independent Component Analysis", Neural Computing Surveys, Helsinki University of Technology, Finland, 2 94-128, 1999.
- [7] Stone, J.V.: Independent Component Analysis, The MIT Press, Cambridge, Massachusetts London, England, 2004.

BIOGRAPHY



Name: Al Rubaie
Surname: Evan Madhi Hamzh
Date of birth: 25 October 1976, Iraq, Babylon
Address: Iraq, Babylon
Education: Secondary School "Al Taleeaa", Iraq, Babylon. University of