Tagged as

- C#
- Dev
- Intermediate

A C# Project in Optical Character Recognition (OCR) Using Chain Code

Hussein El Saadi, 22 Feb 2011 CPOL

4.86 (56 votes)

Rate this:

An article that looks to use chaing code to do optical character recognition

Download source code - 96.88 KB

Introduction: What is OCR?

OCR stands for optical character recognition i.e. it is a method to help computers recognize different textures or characters

Related Articles

OCR with Microsoft® Office

How To: Use Office 2007 OCR Using C#

Creating Optical Character Recognition (OCR) applications using Neural Networks

OCR with LEADTOOLS: The Better Choice

Introduction to Programming with LEADTOOLS .NET OCR - Multi-threaded OCR SDK for 32 and 64 Bit Development

OCR are some times used in signature recognition which is used in bank

And other high security buildings

In addition, texture recognition could be used in fingerprint recognition

OCR's are known to be used in radar systems for reading speeders license plates and lot other things

A Detailed Look on the OCR Implementation and its use in this Paper

The goal of Optical Character Recognition (OCR) is to classify optical patterns (often contained

A digital image) corresponding to alphanumeric or other characters. The process of OCR

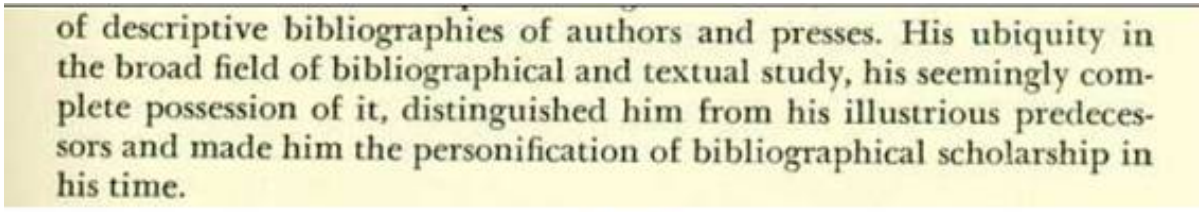
Involves several steps including segmentation, feature extraction, and classification. Each of

These steps is a field unto itself, and is described briefly here

Implementation of OCR.

One example of OCR is shown below. A portion of a scanned image of text, borrowed from the

Web, is shown along with the corresponding (human recognized) characters from that text.


of descriptive bibliographies of authors and presses. His ubiquity in the broad field of bibliographical and textual study, his seemingly complete possession of it, distinguished him from his illustrious predecessors and made him the personification of bibliographical scholarship in his time.

Of descriptive bibliographies of authors and presses. His ubiquity in the broad field of bibliographical and textual study, his seemingly complete possession of it, distinguished him from his illustrious predecessors and made him the personification of bibliographical scholarship in his time.

Figure 1: Scanned image of text and its corresponding recognized representation

A few examples of OCR applications are listed here. The most common for use OCR is the first

Item, people often wish to convert text documents to some sort of digital representation.

1. People wish to scan in a document and have the text of that document available in a word processor.
2. Recognizing license plate numbers
3. Post Office needs to recognize zip codes

Other Examples of Pattern Recognition

1. Facial feature recognition (airport security) – Is this person a bad-guy?
2. Speech recognition – Translate acoustic waveforms into text.
3. A Submarine wishes to classify underwater sounds – A whale. A Russian sub? A

Friendly ship?

The Classification Process

(Classification in general for any type of classifier) There are two steps in building a classifier:

Training and testing. These steps can be broken down further into sub-steps.

1. Training
    a. *Pre-processing* – Processes the data so it is in a suitable form for…
    b. *Feature extraction* – Reduce the amount of data by extracting *relevant*

        Information—usually results in a vector of scalar values. (We also need to

        NORMALIZE the features for distance measurements!)

    c. *Model Estimation* – from the finite set of feature vectors, need to estimate a model

        (Usually statistical) for each class of the training data

2. Testing
    a. *Pre-processing*
    b. *Feature extraction* – (both same as above)
    c. *Classification* – Compare feature vectors to the various models and find the

    Closest match. One can use a distance measure

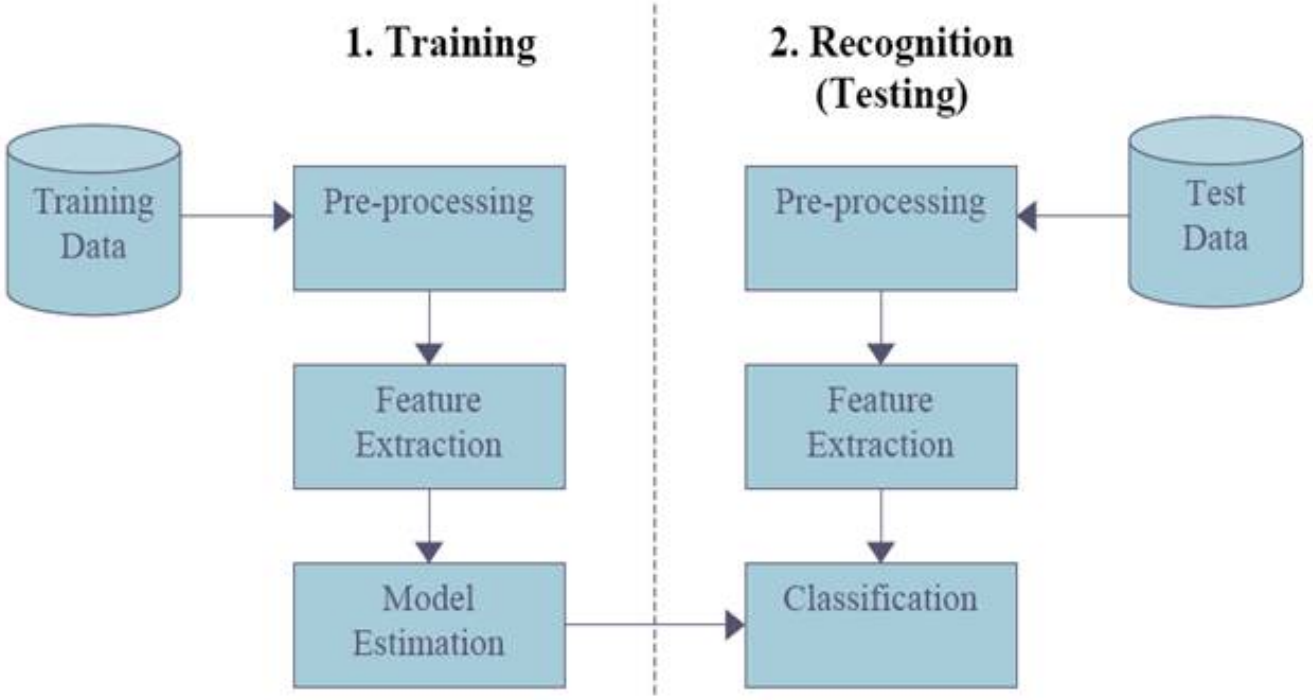


Figure 2: The pattern classification process

OCR – Pre-processing

These are the pre-processing steps often performed in OCR

- **Binarization** – Usually presented with a grayscale image, binarization is then simply a matter of choosing a threshold value.
- **Morphological Operators** – Remove isolated specks and holes in characters, can use the *majority* operator.
- **Segmentation** – Check connectivity of shapes, label, and isolate.

Segmentation is by far the most important aspect of the pre-processing stage. It allows the

Recognizer to extract features from each individual character. In the more complicated case of

Handwritten text, the segmentation problem becomes much more difficult as letters tend to be

Connected to each other.

OCR – Feature Extraction

Given a segmented (isolated) character, what are useful features for recognition?

1. Moment based features

Think of each character as a PDF. The 2-D moments of the character are:

$$m_{pq} = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} x^p y^q f(x,y)$$

From the moments, we can compute features like:

1. Total mass (number of pixels in a binarized character)
2. Centroid - Center of mass
3. Elliptical parameters
    i. Eccentricity (ratio of major to minor axis)
    ii. Orientation (angle of major axis)
4. Skewness
5. Kurtosis
6. Higher order moments
7. Hough and Chain code transform
8. Fourier transform and series

There are different methods for feature extraction or finding an image descriptor, these methods lie into two categories

1. one which uses the whole area of the image
2. an other that uses the contour or edges of the object

All the above methods uses the contour of the object to collect the object's features.

ALGORITHIM REQURIMENTS

The algorithm we needed for this OCR had to satisfy requirements

1. It must faithfully preserve the information of interest
2. Its must permit for compact storing and convenient retrieval
3. It must facilitate the required processing
4. Scaling invariant

5. Easy to implement

And so we decided to implement for this OCR in particular is the chain code

Which is also known as Freeman's chain code.

Freeman's chain code is one of the best and easiest methods for texture recognition

Freeman designed the chain-code in 1964 (also, the chain code is known to be a good method for image encoding but here we are using it as a method for feature extraction)

Although the chain code is a compact way to represent the contour of an abject yet is has some serious draw back when used as a shape descriptor

Chain-code Drawbacks

1. It works only on contoured shapes (which in our case means characters should be inputted in bold font)
2. it is so sensitive to noise as the errors are cumulative
3. The starting point of the chain code, and the orientation and scale of a contour affects the chain code.

Therefore, the chain correlation scheme, which can be used to match two chains, suffers from these drawbacks

N.B: How ever the scaling draw back was partially overcame

The Freeman's chain-code implementation process for character recognition?

First step as discussed before is the pre processing:

The only preprocessing step we will discuss in this paper is the edge detection or contouring.

First, what does a contoured object mean?

Well it means an object with edges only.

How to obtain a contoured object of the BOLD character object?

The concept of obtaining the edges was applied in reverse manner

I.e. if we can detect the filling and remove it then the remains are the edges.

So all we need is to find a unique property of the filling and apply this

Filling property:

- All eight surrounding pixels are black

Pseudo-code:

Collapse | Copy Code

```
        Begin
        WHILE (! end of image)
Search original image for black pixel
        If (the eight surrounding are black)
        Then
              This pixel is filling and we should remove it
Go to next-pixel
WHILEEND
END
```

So by implementing the above code in an image like this

Second step is the feature extraction process (using freeman's chain-code)

There are many methods to implement the chain-code and all of them lead to the idea of partitioning the target objects

And so the one we used is based on the idea of partitioning the object into tracks and sectors and then apply the chain-code in each sector for getting the pixels relations and saving it on a file.

How do we achieve this level of partitioning on the contoured object and extracting the feature vector it?

Achieving a sectors tracks partitioned object and extracting the feature vector from it involves applying the following steps

1. get the Centroid - Center of mass
2. find the longest radius
3. getting the track-step
4. virtually divide the object into tracks using the track-step(which is based on the default number of tracks used i.e. the same number of tracks must be used for both training and testing)
5. getting the sector step
6. divide those virtual tracks into equal sectors using the sector-step (which is based on the default number

of sectors used i.e. the same number of sectors must be used for both training and testing)
7. find relations between adjacent pixels
8. putting all the features together

The above steps will be discussed later in greater details

A DETAILED LOOK ON HOW TO achieve this level of partitioning of the contoured object.

1- GETTING THE CENTER OF MASS

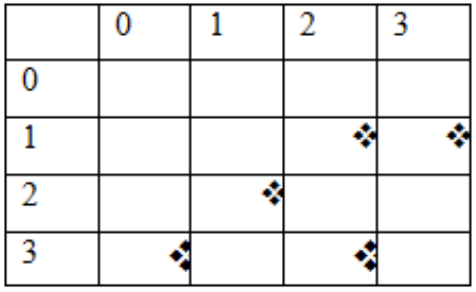
The center of mass of any texture (in our case character) is driven by the following equation:

Collapse | Copy Code

```
    Xc=∑x/ ∑∑f(x, y)
Yc= ∑y/ ∑∑f(x,y)
```

In English, this means that the X coordinate of the centroid is the sum of all the positions of x coordinate of all the pixels in the object

Divided by the number of pixel of the object



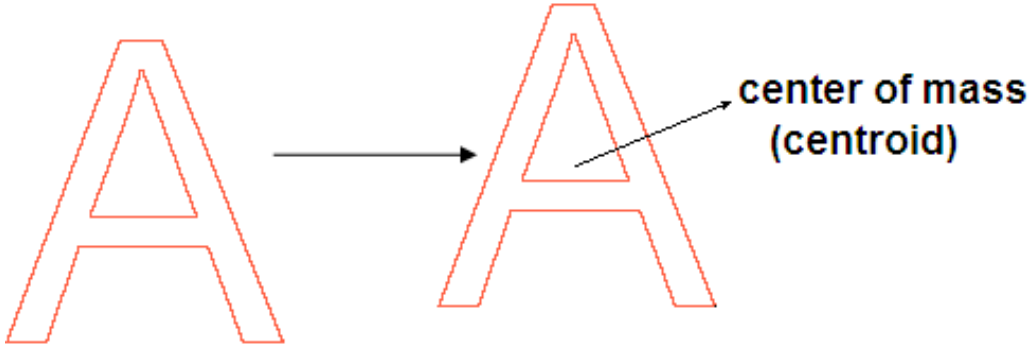
So the Xc for the above image= (0+1+2+2+3)/5

Therefore, the Y coordinate of the centroid is the sum of all the positions of y coordinate of all the pixels in the object divided by the number of pixel of the object

	0	1	2	3
0				
1			❖	❖
2		❖		
3	❖		❖	

So the Yc for the above image= (3+2+2+2+1)/5

And so by applying the above rule on a character we get



2- FINDING THE LONGEST RADIUS

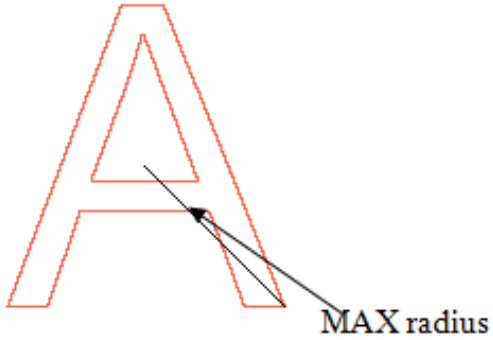
To get the longest radius we have to calculate the distance between the centriod (center of mass) and every other pixel of the contoured object and find the maximum length and this would be the longest radius

How to get the distance between to pixels?

Thanks to Pythagoras who invented Pythagoras theorem, which gives us the distance between any, two points and it states:

"The distance between any two points equals

Sqrt $((Xc-Xi)^2 + (Yc-Yi)^2)$"



3- GETTING THE TRACK STEP

What is the track-step?

The track step is the distance between any two adjacent tracks, which will be used to identify the pixels position i.e. in which track

How to get the track step?

The track step equals max-radius dived by the predefined number of tracks i.e. (Track_Step=M-radius/No' of tracks)

In our case, the number of tracks we used was five

4- VIRTUALLY DIVIDE THE OBJECT INTO TRACKS USING THE TRACK-STEP

(Which is based on the default number of tracks used i.e. the same number of tracks must be used for both training and testing)

This way we can identify in which track a pixel lies in.

5- GETTING THE SECTOR STEP

Based on the number of sectors you decide to use

The (SECTOR_STEP =360/No' of Sectors).

The sector step will be used to know under which sector does a pixel lie.

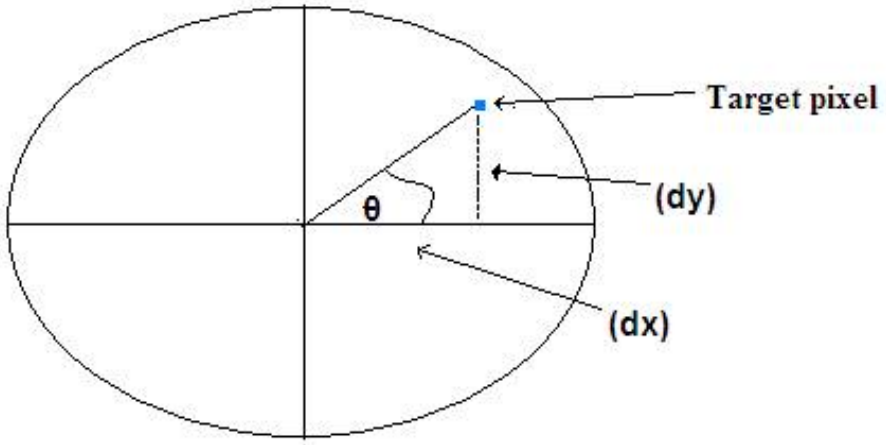
6- DIVIDE THOSE VIRTUAL TRACKS INTO EQUAL SECTORS USING THE SECTOR-STEP

(Which is based on the default number of sectors used i.e. the same number of sectors must be used for both training and testing);

In other words using the sector-step and the track-step we can identify under which sector and which track a pixel lies.

A detailed look on how to identify a pixel lies on which sector:

First, we have to get Θ, which is the angle between the pixel and the x-axis



Θi=tan-1(y-Yc/x-Xc)

Then pixel would lie at sector=Θ/sector-step

7- FINDING THE PIXEL RELATIONS

Finding pixel relation is by far the most important and easiest step in the feature extraction step since it some how describes the shape.

How do you extract relations?

To extract relations between pixels we follow this algorithm in

Pseudo-code

Collapse | Copy Code

```
        Begin
        For every pixel surrounding the target pixel
              Moving clock-wise from north direction
```

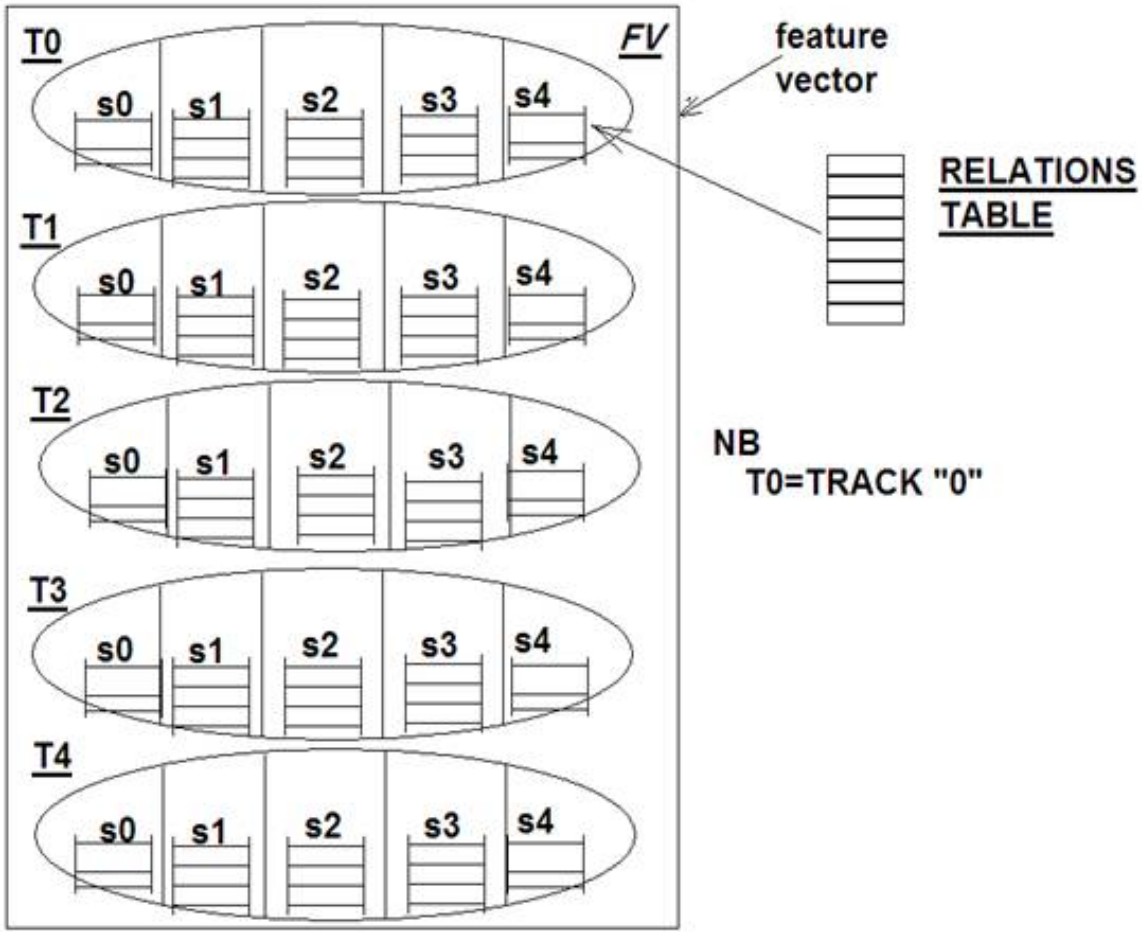
```
 If a pixel is, present surrounding the target pixel
 Then
   Store its position
   Go to next target pixel
End if
Next
END
```

By now, you should be able to identify for any pixel in which track and sector it lies on and its relation with its neighboring pixels.

By the way, those are the image features;

THE FEATURE VECTOR

The feature vector is simply the collection of the features of every pixel sorted by the track i.e. check the position of each pixel and add its properties to the feature vector. E.g. if we have two pixels lie in sector 1 and track 1 with the same relation of 4 then in cell4 in the relations table the one in t1 and s1 will be incremented twice since there are TWO pixels having this property



Freeman's chain code drawbacks that we were able to solve:

Actually, the only draw back that was solved was the scaling problem and it was partially solved (i.e. up scaling only).

How to over come the up scaling draw back of the chain code?

It has been solved by dividing the feature vector by the number of pixels in the object so this way the number of pixels (size of character) has no effect on the feature vector.

Classification

Classification is the process of identifing the unknown object

There are a number of classifiers availible that can be used such as

1. Neural networks
2. Support vecotor machines
3. K-nearest neighbor
4. Eculidian distance

In this article we discuss eculidian distance which is a variation from the knn

Simply the eculidian distance is calculating the distnace between the relations

Collapse | Copy Code

```
public double get_distence(cfeature_vector vec2)
{
        double x=0,y=0,z=0,zf=0;
        for (int i=0;i<6;i++)
        {
                for (int j=0;j<4;j++)
                {
                        for (int k=0;k<8;k++)
                        {
                        x=current_features.tracks[i].sectors[j].relations[k];
                        y=vec2.tracks[i].sectors[j].relations[k];
                                z=x-y ;
                                z*=z;
                                zf+=z ;
                        }
                }
        }
return zf ;
}
```

RECOGNITION RESULTS & CONLCUSIONS

The recognition rate for character images of the same font used (Arial) of up scaling is almost 100% correct

How ever, for down scaling the recognition rate is very poor

When we tested our OCR on hand written there where two VI observations that affects the recognition rate

1st people tend to use different fonts than the on it's trained

2nd objects with curves and like character "C" &"O"&"Q" ESPECIALLY were not recognized at all perhaps because most people used very bad handwriting

The handwritten test data we used where from two different sources

1. We asked people to draw the characters using the paintbrush and so we got so little number of volunteers actually one and my self.
2. the 2nd source was hand written letters on a piece of paper and we had approximately 7 samples of 3 different hand written letters

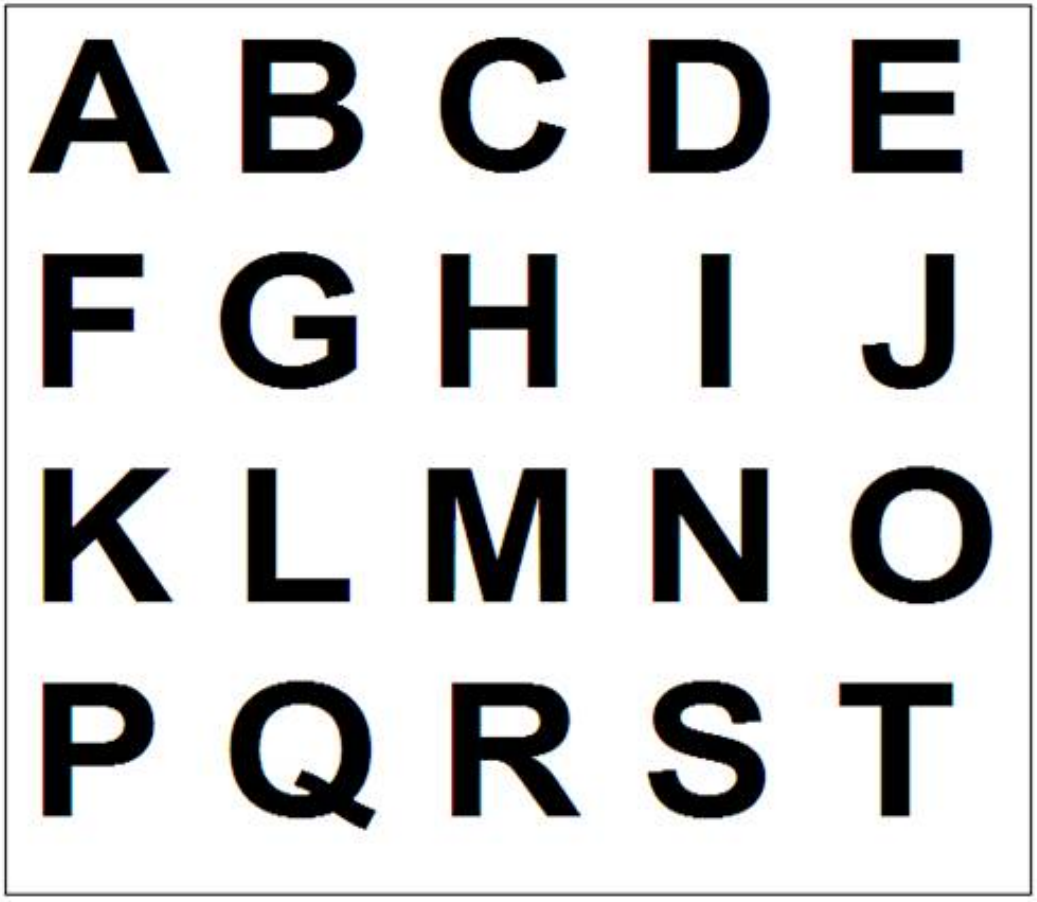
Correctness Rate for the Different Sources

For the first source the recognition rate came to its peek of 75% correctness (on a well-written letters <"neat writing"> but on the other samples the recognition rate was approximately 62%

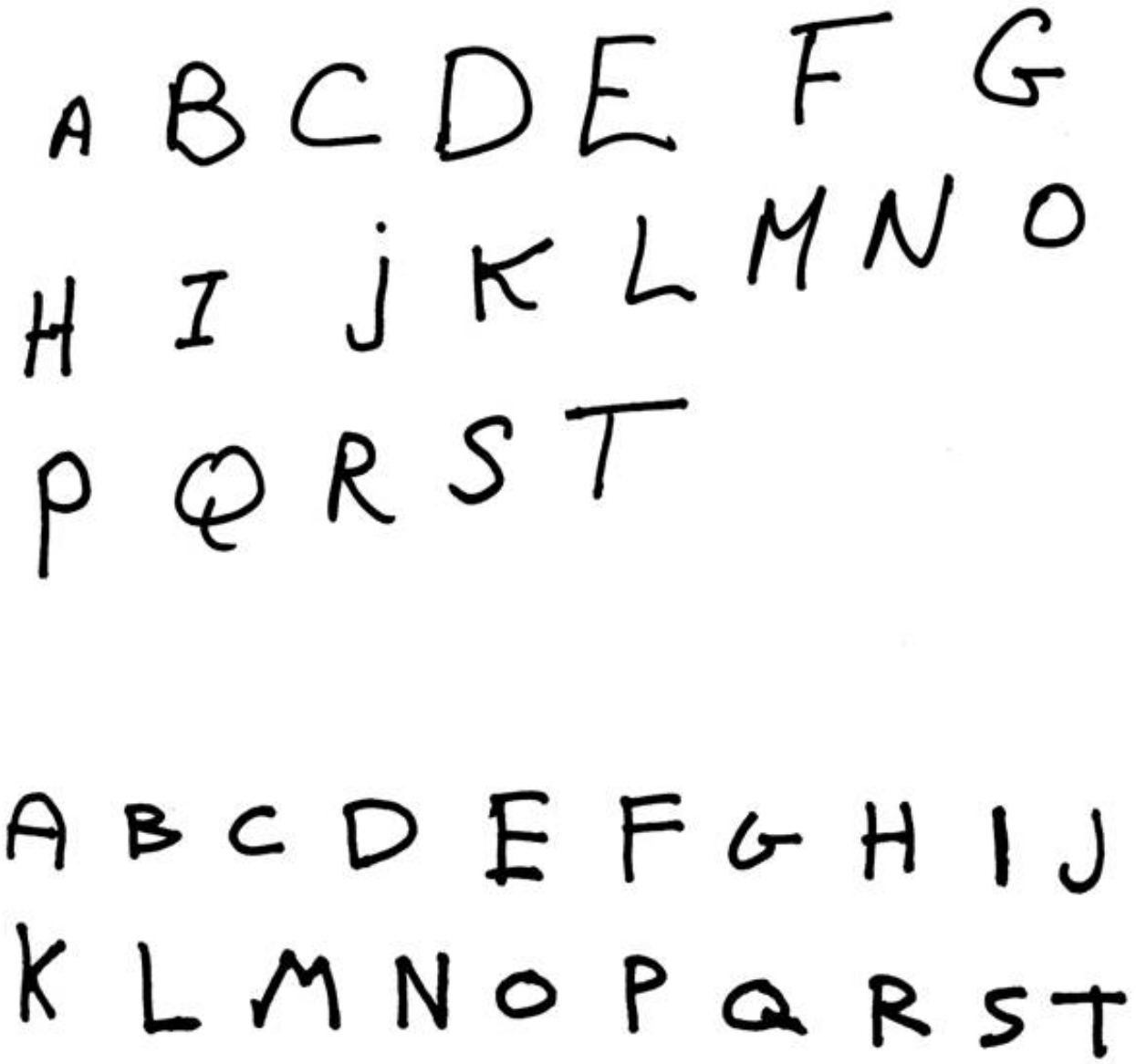
As for source, number two the recognition rate was poorer than what was expected, it was approximately 57% for most of the samples

THE FOLLOWING IS SOME OF THE SAMPLES USED IN TESTING PROCESS:

The following is teaching sample the engine was originally trained for:



THE FOLLOWING IS A HANDWRITING SAMPLE TAKEN TROUGH A SCANNER:



License

Share

About the Author

Hussein El Saadi

Egypt

No Biography provided

Comments and Discussions

You must Sign In to use this message board.

Search Comments Go

Profile popups Spacing Relaxed Noise Very Low Layout Normal Per page 50 Update

First Prev Next

How to run this code ?	Member 11069740	9-Sep-14 5:41
Re: How to run this code ?	Member 11432369	6-Feb-15 1:58
monotype cursiva	salimkadri	5-Mar-14 3:09
please reply	Member 10454183	11-Feb-14 17:21
pleasee help	Member 10454183	14-Jan-14 17:38
Re: pleasee help	Hussein El Saadi	9-Feb-14 12:41
please help	Member 10454183	14-Jan-14 17:26
My vote of 5	al13n	12-Jan-14 20:17
how to use the application??	Member 10361398	1-Dec-13 21:19
Please Help!	Member 10349626	21-Oct-13 5:06
Re: Please Help!	Hussein El Saadi	27-Oct-13 10:02
Re: Please Help!	Member 10349626	19-Nov-13 8:10
My Vote is 5	thaer.sqor	5-Mar-13 15:37
Help [modified]	thaer.sqor	5-Mar-13 15:33
Can i port this code to work with hardware !!	Sameh El-Hakim	12-Feb-13 12:15
Re: Can i port this code to work with hardware !!	Rohit Dubey from Hyderabad	15-Feb-13 6:45
Re: Can i port this code to work with hardware !!	Sameh El-Hakim	15-Feb-13 7:59
My Vote is 5	Tauqeer Ul Islam	5-Jan-13 4:12
My vote of 5	Grasshopper.iics	26-Oct-12 20:19

My vote of 1	programmerdon	24-Sep-12 7:28
help please	jebryns	2-Sep-12 23:03
How to train the system for separate characters	champike kegalle	12-Aug-12 20:17
My vote of 1	Mr Pham	3-Jul-12 0:23
Need help	Member 1771362	11-Jun-12 0:09
Please Help	michel chanaa	26-May-12 7:32
My vote of 5	manoj kumar choubey	11-Apr-12 3:54
My vote of 4	Hassanin	20-Mar-12 2:50
My vote of 5	a7med3laa	12-Mar-12 10:51
My vote of 5	Xmar	17-Jul-11 17:52
Confused - how to add more characters	Supervan2	29-Jun-11 4:53
Re: Confused - how to add more characters	Supervan2	29-Jun-11 11:56
Can this be modified to OCR dollar bills	kristopher maillard	23-Mar-11 12:36
Re: Can this be modified to OCR dollar bills	Hussein El Saadi	26-Mar-11 11:15
Re: Can this be modified to OCR dollar bills	saad324	3-Apr-13 0:58
Re:help!!!	saad324	3-Apr-13 0:58
how to run the prog. [modified]	nona444	19-Mar-11 0:58
Re: how to run the prog.he	Hussein El Saadi	22-Mar-11 12:14
Re: how to run the prog.he	Member 7836829	18-Apr-11 22:20

want more details...	Ahmed.lpo	15-Mar-11 12:56
Thanx	sajithdilhan	12-Mar-11 2:41
What does the demo code do?	ACanadian	1-Mar-11 5:11
Re: What does the demo code do?	Hussein El Saadi	22-Mar-11 12:16
download link is working now	Hussein El Saadi	25-Feb-11 3:24
My vote of 5	sean1332	23-Feb-11 5:47
Can't download source	nixite	23-Feb-11 1:24
Re: Can't download source	nixite	23-Feb-11 4:59
Re: Can't download source	Member 10923376	3-Jul-14 17:01
My vote of 5	Slacker007	23-Feb-11 1:13
Can't download Souce Code	Theingi Win	22-Feb-11 21:09
Re: Can't download Souce Code	knoami	22-Feb-11 23:03

Last Visit: 31-Dec-99 19:00 Last Update: 8-Feb-15 21:12 Refresh 1 2 Next »

General News Suggestion Question Bug Answer Joke Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.