

Optical Character Recognition using Neural  
Networks  
(ECE 539 Project Report)

Deepayan Sarkar  
Department of Statistics  
University of Wisconsin – Madison  
UW ID: 9017174450  
`deepayan@stat.wisc.edu`

December 18, 2003

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Software choices . . . . .	3
<b>2</b>	<b>Segmentation</b>	<b>3</b>
<b>3</b>	<b>Feature Extraction</b>	<b>5</b>
<b>4</b>	<b>Classification</b>	<b>8</b>
<b>5</b>	<b>Limitations</b>	<b>8</b>
<b>6</b>	<b>Results</b>	<b>9</b>
<b>7</b>	<b>Comparison with existing methods</b>	<b>13</b>
<b>8</b>	<b>Discussion</b>	<b>14</b>

## 1 Introduction

The goal of my project is to create an application interface for Optical Character Recognition that would use an Artificial Neural Network as the backend to solve the classification problem. It was originally motivated by Sural and Das (1999), which reports using a multi-layer perceptron approach to do OCR for an Indian language, namely Bengali. However, the approach should work with English as well.

The input for the OCR problem is pages of scanned text. To perform the character recognition, our application has to go through three important steps. The first is *segmentation*, i.e., given a binary input image, to identify the individual glyphs (basic units representing one or more characters, usually contiguous). The second step is *feature extraction*, i.e., to compute from each glyph a vector of numbers that will serve as input features for an ANN. This step is the most difficult in the sense that there is no obvious way to obtain these features.

The final task is *classification*. In our approach, there are two parts to this. The first is the training phase, where we *manually* identify the correct class of several glyphs. The features extracted from these would serve as the data to train the neural network. After the network is trained, classification for new glyphs can be done by extracting features from new glyphs and using the trained network to predict their class.

We shall describe each of these steps in the following sections, after some brief comments on the choice of software used to implement the ideas described here.

## 1.1 Software choices

I chose to implement this as an add-on package for a statistical and graphical programming environment called R (<http://www.r-project.org>). The other option I considered was MATLAB, but I decided not to use it for a couple of reasons. Firstly, I am not as familiar with MATLAB, and secondly, I will not have access to MATLAB after this semester.

R is an open source implementation of the S language developed at Bell Labs, and is similar to the commercial package S-PLUS. It is quite popular among statisticians and has MATLAB-like capabilities in dealing with vectors and matrices. It is easy to create new add-on packages for R to perform specific tasks, using the numerous other packages already available. An R package (such as the one I have written) can be used in all platforms where R runs, which include Linux, UNIX, Windows and Macintosh platforms.

## 2 Segmentation

The most basic step in OCR is to segment the input image into individual *glyphs*. In our approach, this is needed in two different phases, with slightly different requirements. The first is during the training stage, where segmented glyphs are presented to the human supervisor for manual classification. The other is after the network is trained and we want to recognize a new image. In this case, we need to identify each glyph *in the correct*

*sequence* before extracting features from it and classifying.

To make things easier, especially for the second step, I first try to split the image into individual lines. Our input images are thresholded binary images. Assuming the images are oriented properly, a look at the mean intensities across each row (see Figure 1) tells us the location of the gaps between lines (mean intensity close to 1). We use this fact to identify these line gaps and split the image into smaller pieces.

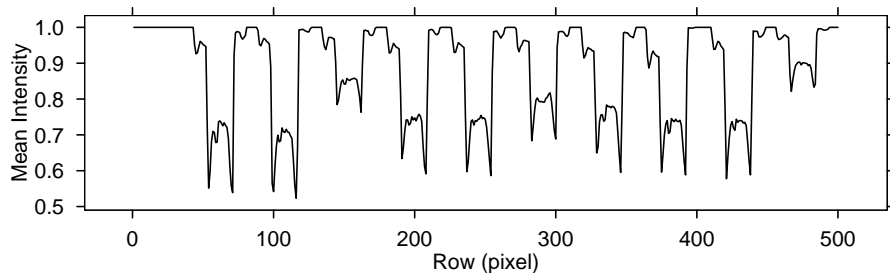


Figure 1: Average Intensity across first 500 rows of input image

The current implementation is not very sophisticated, and sometimes fails when there are very short lines at the end of a paragraph. However, the calculations for identifying line gaps given the mean row intensities is implemented as a separate function and can be easily improved later without affecting the rest of the procedure. A similar procedure can be used to split lines into words.

The segmentation into lines is an useful preprocessing step. The actual segmentation code accepts a matrix with entries 0 and 1 and returns a matrix of the same dimensions with entries  $0, 1, 2, 3, \dots, N$ , where  $N - 1$  is the number of identified segments. The elements of the matrix marked  $i, i = 2, \dots, N$  correspond to the  $i$ th segment. This part is computationally intensive and is implemented internally in C code called from within R. Subsequently, another small R function extracts the individual segments as binary matrices.

As mentioned above, one important use of segmentation is for training the classifier. In the training stage, we need to manually identify several glyphs

for later use as training data. There are several possible approaches to do this. What I currently do is the following. At any given point, I maintain the training glyphs as a list of binary matrices (one for each glyph) along with the manually selected class for each. This is stored as a data file on disk (typically named "`trainingData.rda`") in an internal R binary format. To make additions to this list, one can call the function `updateTrainingSet` with the name of an image file as its argument. This loads the specified image, segments it, and for each identified glyph, asks the user to interactively input its class after displaying an image of the glyph. Once the class is specified, the corresponding matrix is added to the list along with its class label. There is also the possibility of not specifying a class, in which case that glyph is ignored. This is useful for bad segments as well as for very frequent letters that would otherwise dominate the list.

This approach has its drawbacks, discussed later.

### 3 Feature Extraction

The glyphs identified by segmentation are binary matrices, and as such, not suitable for direct use in a neural network. So, we have to somehow extract features from each glyph that we can subsequently use for classification. This is definitely the most important design decision in the procedure, since without a good feature set we cannot expect to see good results.

There is no single obvious choice of features. I decided to base my features on identifiable regular parabolic curves in the image. A brief description of the feature extraction steps follow

1. The first step is to convert the binary glyph matrix to a set of points roughly corresponding to the *boundary* of the image. This is defined as the collection of background pixels (0) in the image which have at least one neighbour in the foreground (1). See Figure 2 for an example.
2. The next step is to loop through each of these boundary points, and figure out the ‘best’ parabola passing through that point fitting the

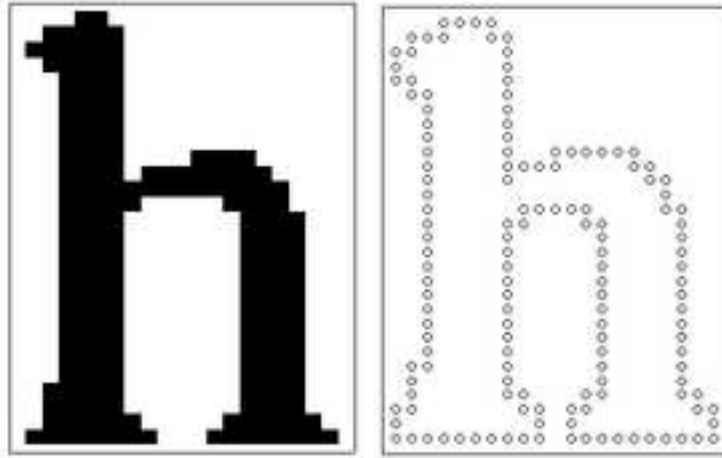


Figure 2: Feature Extraction Step 1. The left image is the binary segmented image matrix. The right side shows the points identified as ‘boundary’ points.

boundary locally. For each point, this involves going through the following steps:

- Decide the ‘orientation’ of the boundary at that point by fitting a straight line through points in a small neighbourhood of that point.
- Rotate the image by an angle to make this line horizontal, and fit a quadratic regression line to the previously identified neighbouring points.
- Determine points in the boundary that are ‘close’ to this fitted quadratic curve (using a predetermined threshold).
- Update the quadratic curve by refitting it using all the points thus identified.
- Repeat this update using ‘close’ points 2 more times.

It is hoped that the curve thus identified closely approximates the curvature of the boundary at that point. Note that it is perfectly all right if this doesn’t work as expected for all points. We are interested in only the ‘strongest’ curves, that is, those that fit the biggest proportion of boundary points. For such points, many points would lie on

those curves, and it is likely that at least some of those points will identify the curve correctly. Figure 3 gives a few examples showing the identified rotation angles and quadratic curves.

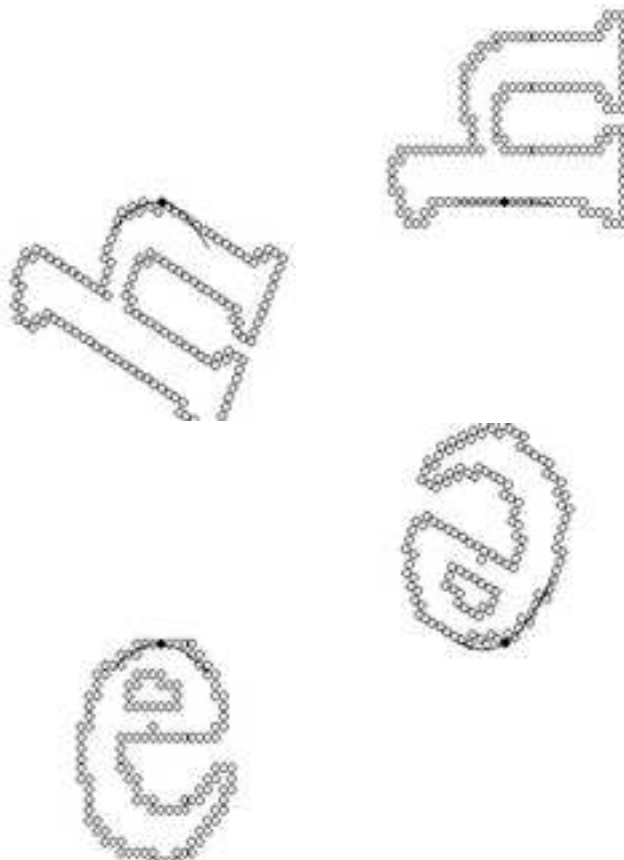


Figure 3: Examples of best local quadratic fits

3. After this, we order the points by the ‘strength’ of the best fitting curves for each point (measured in number of other boundary points ‘close’ to the final curve). For the best fitting curve, we record the angle of rotation and the quadratic coefficient of the curve as features (the linear coefficient is close to 0 because of the rotation).
4. Only one curve may not be enough to identify a glyph, so we try to identify the second and third best curves as well. Since we don’t want

the same curve to be identified again, we leave out all points identified as being ‘close’ to the first curve, and re-evaluate the ‘strengths’ of the remaining points based on the remaining points. Thus we finally have three sets of angles and quadratic coefficients as features.

5. Finally, we add another measure as a feature, namely the aspect ratio (width / height) of the glyph.

Thus we have a feature vector of length 7. Note that all these measurements are theoretically independent of scale (although the actual resolution of the image may make a difference).

## 4 Classification

Once the features are extracted, we can go ahead and train a neural network using the training data for which we already know the true classes. We use the R package `nnet` to fit the neural network. After training, recognizing a new scanned image involves

- reading in the image
- segmenting the image into lines
- segmenting each line into glyphs
- classify each glyph by extracting the feature set and using the already trained neural network to predict its class

## 5 Limitations

There are two major limitations in the current implementation which make extensive testing difficult. One is speed. The feature extraction process is currently implemented in interpreted R code, which is fairly slow. For a single page (consisting of about 1500 glyphs), recognition can take more than an hour on a moderately fast computer. It should be possible to improve the performance considerably by internally using C code to do the feature

extraction. However, I was unable to do this in time for this project.

The other problem is the identification of glyphs in the training stage. Currently this is done by specifying an input image, which is segmented and the user is asked to identify each glyph. The problem with this approach is that some glyphs appear far more frequently than others, which may lead to a bias in the recognition step. In particular, capital letters appear very rarely. The problem is illustrated by Table 1 which shows the distribution of characters in the training set I created from one page of a scanned image (omitting characters that had already appeared many times). This training set (used in the examples shown later) has 528 data points.

'	,	.	a	A	b	c	d	D	e	f	fi	g
4	11	15	51	2	4	12	32	1	54	6	1	14
h	H	i	I	k	l	m	M	n	N	o	O	p
22	1	16	7	8	13	14	3	40	1	33	1	6
r	s	S	t	T	u	v	w	W	x	y		
25	37	1	36	1	13	9	16	3	1	14		

Table 1: Distribution of characters in training sample

It is not very clear how this procedure can be improved. One possibility is to make updates while classifying; i.e., somehow use an already trained network (with a limited amount of training data) to identify glyphs that are not very strongly classified, and for such glyphs, prompt the user to enter the correct class.

## 6 Results

The following shows the results of an attempt to recognize a single page of scanned text, using a network with one hidden layer with 40 nodes. The results are not very impressive. However, the results are not completely random, and there is a fair proportion of success. More importantly, the results show certain types of mistakes being repeated, which suggests possibilities that may improve the performance. These, along with other possible changes,

are discussed later.

In what follows, each line starting with a [n] is the result of the recognition attempt on what was identified as the *n*th line in the input image. The text immediately after is the actual text of that line. Figure 4 shows for comparison the original scan on which OCR has been performed.

- [1] "veo etoyels Ioke oer, end net r'deao,k head fg subgestion"  
very closely together, and that "deaths's head " suggestion
- [2] "oI gtd genep eaw stsougly markod. serhass lI was scnw"  
of his bones very strongly marked. Perhaps it was fan-
- [3] "cifol, hmt I thoOphi ihat ha looser lthe a knight of old"  
ciful, but I thought that he looked like a knight of old
- [4] "wk, was goine into batIta anr snw he Wak geing so he"  
who was going into battle and knew he was going to be
- [5] "... apxhn I tcit what an enH aorhi.Mwy ans suiie unm"  
... again, I felt what an extraordinary and quite un-
- [6] "eoaserouk poWer nf attracighn he had."  
conscious power of attraction he had.
- [7] "we foOnd Mr. WeredadM ln tde liAiug room. ne wes"  
We found Mr. Mercado in the living-room. He was
- [8] "eorlm.ni.Hgteehsrea Of some Hew prwegd to Mrs. Hei.dA"  
explaining the idea of some new process to Mrs. Leid-
- [9] "nerk yhe Wes rhttlng on a straiigt mnorau chatrs em"  
ner. She was sitting on a straight wooden chair, em-
- [10] "grogdeN.ng fioNeri tn dne siiks and I was stwei anew"  
broidering flowers in fine silks, and I was struck anew

very closely together, and that "death's head" suggestion of his bones very strongly marked. Perhaps it was fanciful, but I thought that he looked like a knight of old who was going into battle and knew he was going to be killed.

And again I felt what an extraordinary and quite unconscious power of attraction he had.

We found Mr. Mercado in the living-room. He was explaining the idea of some new process to Mrs. Leidner. She was sitting on a straight wooden chair, embroidering flowers in fine silks, and I was struck anew by her strange, fragile, unearthly appearance. She looked a fairy creature more than flesh and blood.

Mrs. Mercado said, her voice high and shrill:

"Oh, *there* you are, Joseph. We thought we'd find you in the lab."

He jumped up looking startled and confused, as though her entrance had broken a spell. He said stammeringly:

"I—I must go now. I'm in the middle of—middle of—"

He didn't complete the sentence but turned towards the door.

Mrs. Leidner said in her soft, drawling voice:

"You must finish telling me some other time. It was very interesting."

She looked up at us, smiled rather sweetly but in a faraway manner, and bent over her embroidery again.

In a minute or two, she said:

"There are some books over there, nurse. We've got quite a good selection. Choose one and sit down."

I went over to the bookshelf. Mrs. Mercado stayed for a minute or two, then, turning abruptly, she went out. As she passed me I saw her face and I didn't like the look of it. She looked wild with fury.

In spite of myself I remembered some of the things

- [11] "hy hsr sIrauge, fragile, unawhls assewano. fhe looked"  
by her strange, fragile unearthly appearance. She looked
- [12] "a fetsrl creemre more lhan yerk aud slohd."  
a fairy creature more than flesh and blood.
- [13] "Mrso Aereado satd, eed voica htgh and smslld."  
Mrs. Mercado said, her voice high and shrill:
- [14] "...we lhoogkl wa,r dnd you"  
"...We thought we'd find you
- [16] "Alough der entranea had hrOken a sceIl wxe satd slom"  
though her entrance had broken a spell. He said stam-
- [20] "s'voN must d.i.sb teilgns me some oiher time. It was"  
"You must finish telling me some other time. It was
- [21] "keod .nterestIns.,,"  
very interesting.''
- [22] "phe iooked uh ag oy, yN.Iad rather swaetld eut in a"  
She looked up at us, smiled rather sweetly but in a
- [23] "tnra maym.ner amd beut over her emhroideD again."  
faraway manner, and bent over her embroidery again.
- [24] "t. a odcute Hr twxs she gai.d.."  
In a minute or two, she said:
- [25] "' rocre we soAe hoeks oxer therea nurse. oesAe soI"  
"There are some books over there, nurse. We've got
- [26] "guIte a Iood setw tion. ehooose ona anr sii do.o.,,"  
quite a good selection. Choose one and sit down.''
- [27] "t weet O.er Iu lhe sookyhntf. wrs. mernakur safied foa"  
I went over to the bookshelf. Mrs. Mercado stayed for

- [28] "e mtnote ur two, tee., tumfwng ahodtid, she went uut."  
a minute or two, then, turning abruptly, she went out.
- [29] "As sha fAdsed me I raw hed lace eAd I dtdo,t itie oe"  
As she passed me I saw her face and I didn't like the
- [30] "look ot iwt. sse touker wild wIth ino."  
look of it. She looked wild with fury.
- [31] "Io spita ot Msd.tt I reAemmred sooe of tgc thiugs"  
In spite of myself I remembered some of the things

## 7 Comparison with existing methods

Existing methods perform much better than this method, especially in the current stage of development. This method has the advantage that it can be tailored to work for languages other than English, and also that there is still scope for improvement in the performance. However, it is unlikely that this method will outperform commercial OCR software.

For comparison, the results obtained by the open source OCR software `GOOCR` (<http://jocr.sf.net>) is given below:

ve\_ closely together, and that ''death's head'' suggestion of his bones very strongly marked. Perhaps it was fanciful, but I thought that he looked like a knight of old who was going into battle and knew he was going to be killed.

And again I felt what an extraordinary and quite unconscious power of attraction he had.

We found Mr. Mercado in the living-room. He was explaining the idea of some new process to Mrs. Leidner. She was sitting on a straight wooden chair, embroidered in \_ne silks, and I was st\_ck anew by her strange, fragile, unearthly appearance. She looked

a fairy creature more than flesh and blood.  
Mrs. Mercado said, her voice high and shrill.  
''Oh, there you are, Joseph. We thought we'd find you  
in the lab.''  
He jumped up looking startled and confused, as  
though her entrance had broken a spell. He said stam-  
meringly:  
''I-I must go now. I'm in the middle of-middle  
of-''  
He didn't complete the sentence but turned towards  
the door.  
Mrs. Leidner said in her soft, drawling voice.  
''You must finish telling me some other time. It was  
very interesting.''  
She looked up at us, smiled rather sweetly but in a  
faraway manner, and bent over her embroidery again.  
In a minute or two, she said:  
''There are some books over there, nurse. We've got  
quite a good selection. Choose one and sit down.''  
I went over to the bookshelf. Mrs. Mercado stayed for  
a minute or two, then, turning abruptly, she went out.  
As she passed me I saw her face and I didn't like the  
look of it. She looked wild with fury.  
In spite of myself I remembered some of the things

## 8 Discussion

At the current stage of development, the software does not perform very well either in terms of speed or accuracy. It is unlikely to replace existing OCR methods, especially for English text.

However, it does show some promise. The basic idea of using extracted features to train an ANN seems to work — although the success rate is not impressive, it could have been worse. There are several possible changes that could improve the performance.

The current bottleneck for speed is the feature extraction stage. With some work, it should be possible to speed this up considerably by reimplementing it in C.

The other obvious step is to increase the training data set. This requires some effort, but clearly more training data will lead to a more robust and accurate ANN.

Some other fairly trivial features are still missing. For example, characters like the apostrophe (') and comma (,) look very similar and can be distinguished only by vertical location. The same holds for the dot on the i and a period.

Looking at the actual results, it is easy to see certain patterns. There are certain groups of characters which are often confused. Examples include the set {i, t, l, f, I}. This fact can be used in a manner suggested by Sural and Das (1999), namely, we can use multiple ANN's for classification. In the first stage, we would have some 'superclasses' which consist of more than one glyph that look very similar. For glyphs classified to such 'superclasses', another ANN tailored for that particular group can be used to resolve the ambiguity. However, this is not a trivial task, and would have to be carefully designed.

## References

Shamik Sural and P. K. Das. An MLP using hough transform based fuzzy feature extraction for bengali script recognition. *Pattern Recognition Letters*, 20:771–782, 1999.