Text Character Recognition: Identification of Hebrew Letters

Lavi Zamstein, dinoman@mil.ufl.edu
Michael Nechyba, nechyba@mil.ufl.edu
A. Antonio Arroyo, arroyo@mil.ufl.edu
Machine Intelligence Laboratory
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611-6200

Abstract

The concept of text letter recognition is definitely not a new one. The methods used here, however, are intended as an introductory exploration into the field of character recognition using print Hebrew letters as data. The two methods used and compared here are a pixel distance histogram and a uniform quantization histogram. The robustness of the two methods has been tested by applying them to both standard fonts and fancier fonts which contort or add parts onto the letters.

1 Hebrew Letters

There are a total of twenty-seven symbols, excluding vowel markings, in the Hebrew alphabet: twenty-two normal letters and five *sophit* (final) forms. For the purposes of this identification project, only the first six letters (Aleph, Bet, Gimel, Dalet, Hey, and Vav) were used to cut down on excessive complexity, since the process was intended as an introductory exploration into the field of character recognition.

Like in English, letters in Hebrew can have very differing appearances depending on the font. For character recognition tests, three different fonts were used. *Miriam*, a very plain-looking font, was used for the training set. *David*, another plain font, was used as the first test set and will henceforth be referred to as test. *Stam*, a very stylized font, was used as a second test set and will be referred to as fancy.

Figure 1
Training letters

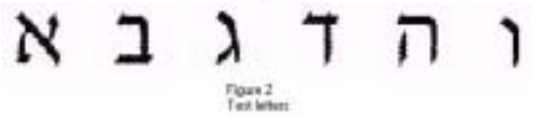

Figure 2
Test letters

Figure 3
Fancy letters

2 Previous Solutions

Other methods that have been used for successful OCR (Optical Character Recognition) include pixel neighborhoods, template matching, and structural classification.

The pixel neighborhoods method uses an idea similar to human recognition of letters. The letters are identified by general shape, and also by the shape of small groups of pixels, called neighborhoods. By comparing each neighborhood of pixels to one of 255 known shapes, the letter can be classified and identified from a list of known letters.

Template matching is a process which is only vaguely described by those

who have used it. It involves comparing similarities of individual pixels and trying to match them to an existing template of a letter.

Structural classification, also only vaguely described in previous research, uses individual parts of letters, such as straight lines, holes, or curves as features to compare to known letters.

3 General Classification Method

Two different classification methods were used to attempt this character identification problem. The first method, distance histograms, classifies letters based on the distances each pixel lies from the center of that letter. The second method, uniform quantization, splits the letter up into uniformly sized bins and compares the number of pixels in each bin to the training set.

For the methods here, each letter was split into three sizes: ten by ten pixels, twenty-five by twenty-five pixels, and fifty by fifty pixels. Since the fifty by fifty pixel images produced the best results, they are used for most of the outcome analysis.

4 Initial Processing

For both classification methods used here, the first step is to extract the black pixels from the images. Since the only concern is the black pixels, all the white pixels can safely be ignored. Once the black pixels of each letter in all three sets are identified, they are put into an array of pixel locations for further processing in each of the recognition methods.

5 Histogram Method

The first step in using the histogram method is to create the training bins for the histogram model.

In order to do this, the centroid of each letter needs to be calculated. This is done by taking the mean of all the black pixels in each letter along the x axis and the y axis, yielding the x and y coordinates of the centroid.

Once the centroid is calculated, and the distances of each black pixel along both the x and y axes are found, these are used to find the distance ratios, or the ratio of the distance of each pixel to the centroid and the maximum distance to the centroid. This results in a list of numbers ranging from 0 to 1 to form the histogram bins.

The histogram bins are then formed from these distance ratios, splitting the ratios into ten bins, each ranging over 0.1 (0-0.1, 0.1-0.2, etc.). The value in each bin is the number of pixels which fall into each ratio range. Once this is finished, the zero probabilities are removed, and the bins are normalized to form percentages. The end result of this is two histograms per letter, one along the x axis and one along the y axis.

Figure 4
Aleph histogram bin - x axis

Figure 5
Aleph histogram bin - y axis

Once the training histogram bins are constructed, the two test sets of data can be processed and classified. Just as with forming the histogram bins, both the test and fancy data sets need to be analyzed to find the distance ratios of each black pixel. For each set of pixels in a letter, probabilities are formed based on the histogram bin that each pixel in that set falls into. From there, the probabilities are calculated to determined which letter each of the test letters are most likely to be. Both the likelihood and the log likelihood were used, although the log likelihood graph is much easier to read to determine which letter has a higher choice probability.

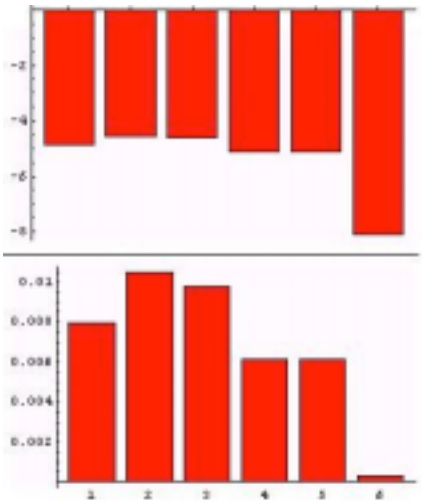

Figure 6
Top: log likelihood for test gimel with histograms
Bottom: likelihood for test gimel with histograms

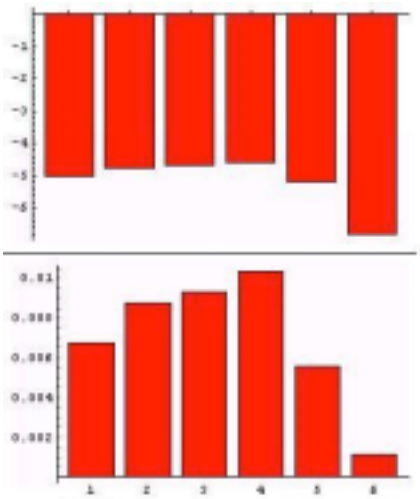

Figure 7
Top: log likelihood for fancy gimel with histograms
Bottom: likelihood for fancy gimel with histograms

6 Uniform Quantization Method

The initial process for the uniform quantization method is much simpler than for the histogram method. The image is split up into uniform bins of five by five pixels each. The zeroes are removed, and the bins are normalized, leaving a percentage probability for each bin.

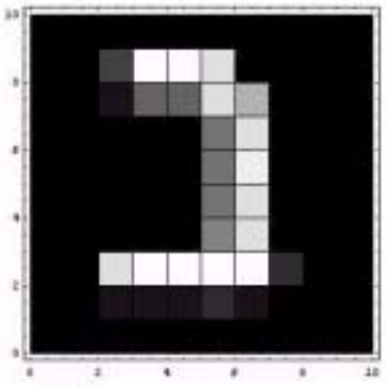

Figure 8
Uniform quantization bins for training bet

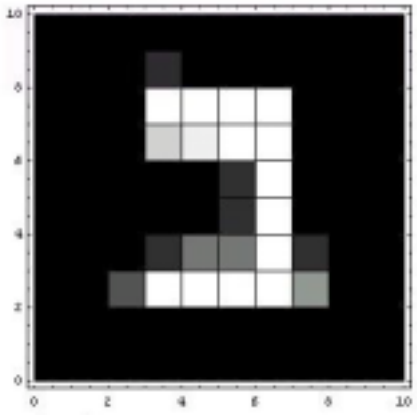

Figure 9
Uniform quantization bins for test bet

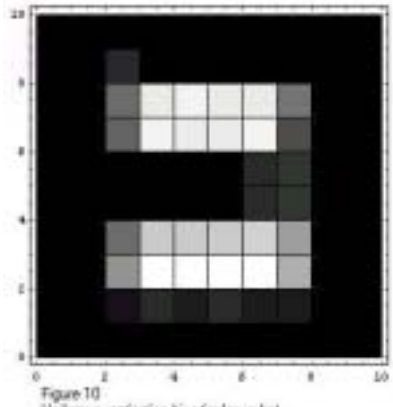

Figure 10
Uniform quantization bins for fancy bet

From there, in a similar process to the histogram identification, each of the test letters is split into the quantization bins and the overall likelihoods and log likelihoods are compared. In the case of the uniform quantization method, however, the likelihoods provide almost no information, since they are all relatively close to one another. In this case, the log likelihood provides the only useful information for identification.

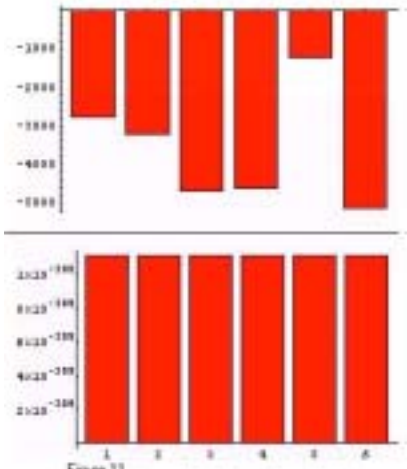

Figure 11
Top: log likelihood for test hey with quantization
Bottom: likelihood for test hey with quantization

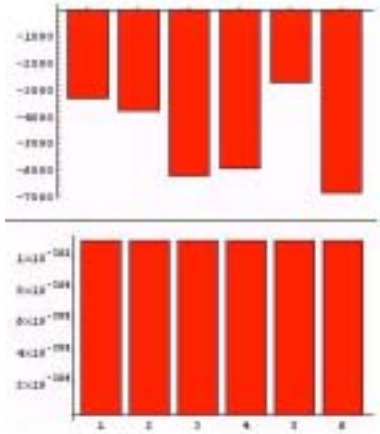

Figure 12
Top: log likelihood for fancy hey with quantization
Bottom: likelihood for fancy hey with quantization

7 Conclusions

Since the fifty by fifty pixel samples provided the most useful and consistent results, it is used for all final identifications.

Using the histogram method, only four of the six test letters were correctly identified, and only two of the six fancy letters were identified.

With the uniform quantization method, however, all six of the test letters were identified. Four of the six fancy letters were identified correctly.

These were, in fact, expected for the results. The two fancy letters that were not correctly identified using the uniform quantization method were significantly different in appearance from the corresponding letter in the training or test sets. After showing the letters to several people who were unfamiliar with Hebrew lettering, they were likewise unable to tell that the letters were the same. In that way, the uniform quantization method is a success.

The uniform quantization method was also expected to perform better than the histogram method. Since the histogram method used here operates only on the distance from the center along the x and y axes, and many of the letters when 'flattened' along each axis look very similar, it was to be expected that this classification method would have some difficulty in identifying the letters. In fact, the histogram method worked slightly better than expected by correctly identifying as many letters as it did.

8 References

OTIOT
www.kotev.co.il/otiot.htm

Unipen
unipen.nici.kun.nl

Pixel Neighborhoods
www.ccs.neu.edu/home/feneric/charred.html

Mathematica
www.wolfram.com