A Word Shape Analysis Approach to Recognition of Degraded Word Images [1]

Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari

Department of Computer Science
State University of New York at Buffalo
Buffalo, NY 14260, USA

email hotin@cs.buffalo.edu

Abstract

This paper presents a word shape analysis approach for word recognition that is independent of character segmentation. The algorithm receives a word image and a lexicon. A set of global and local shape features are extracted from the image and matched with words in the lexicon by a set of highly specialized classifiers. A ranking combination strategy is applied to produce a consensus ranking. The proposed method was applied to a test set of 827 images scanned on a postal OCR, with a lexicon of 186 words. A 96% correct rate within the top 10 choices was achieved.

1 Introduction

Visual word recognition has been an open problem that has motivated many important studies in document image analysis and pattern recognition. The motivation for finding a computational solution to word recognition is from the search for a robust methodology for machine recognition of text images of a wide range of font types and qualities. In this research, we are concerned with multi-font machine printed word images which are of very unstable quality.

Traditionally, word recognition is done by a three step process that includes character segmentation, character recognition, and contextual postprocessing [2]. This approach is appropriate for isolated characters, abbreviations, or well-printed text. However, character recognition is not very successful in domains with many degraded images and large variations in font style. It is observed that character segmentation is difficult for degraded images (Figure 1(a)). Premature recognition decisions on character identities may also create irrecoverable errors (Figure 1(b)).

In a recently proposed approach [8], the word is treated as a whole unit for shape analysis and recognition, without attempting to segment and recognize the individual characters. This is referred to as the word shape analysis approach. This approach avoids committing errors in character segmentation and premature character recognition. In this paper, we present a word recognition algorithm which is based on the word shape analysis approach.

The word recognition algorithm makes use of the assumption that the word in the image is contained in a fixed vocabulary. This is done so that the method can be supplemented by the

[1] This work is supported by the Office of Advanced Technology of the United States Postal Service.

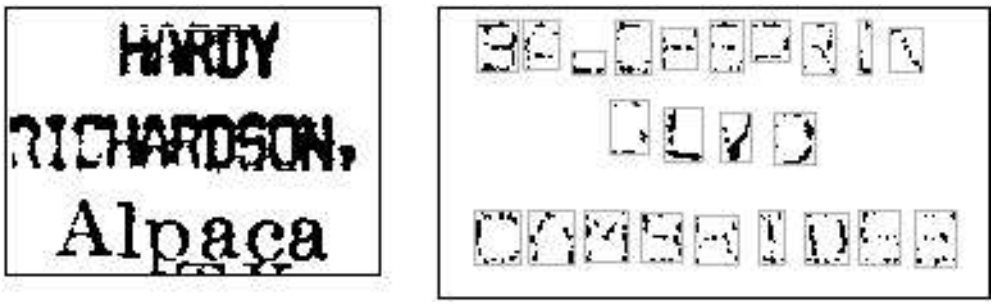


Figure 1: (a) word images difficult to segment, and (b) images with segmented characters that are difficult to recognize (words in the images: BALCHSPRIN, BLVD, CAMBRIDGE)

semantic domain [2], the syntax of a running text [7], or other constraints in the specific application domain. In postal applications, the contextual information in an address block provides useful constraints on the lexicon.

The inputs and outputs of our recognition system are outlined below:

Inputs: A word image and a lexicon.
Outputs: A ranking of the input lexicon such that the word in the image is ranked as close to the top as possible.

The words ranked near the top by the system are referred to as a *neighborhood* of the word in the image. These are the words that share similar shape characteristics as determined by the system. According to the performance of the system, one can determine the size of the neighborhood such that the true word will not be missed. Such a neighborhood can be treated as a set of hypotheses for the true identity of the word that are subject to further hypothesis testing [8] or selection in accordance with other contextual constraints.

2 A Word Shape Based Recognition Algorithm

Our design of the word shape based recognition algorithm is shown in Figure 2. An input word image is first passed to a preprocessing algorithm which performs underline removal, punctuation removal, density estimation, smearing, and reference line estimation. Global and local features are then extracted from the image and represented by a set of feature descriptors. The input lexicon is filtered using the computed global features. The filtered lexicon and the computed local feature descriptors are then input to a set of specialized classifiers. Each of these classifiers outputs a ranking of the filtered lexicon. The output rankings are then combined by a decision combination algorithm, which produces the final ranking of the lexicon.

3 Word Image Preprocessing

Some minimum preprocessing is applied to an input image before feature extraction. Underlines and punctuations are detected and removed. The density of the image, which is defined to be the number of black pixels divided by the image area, is computed as a measure of the image

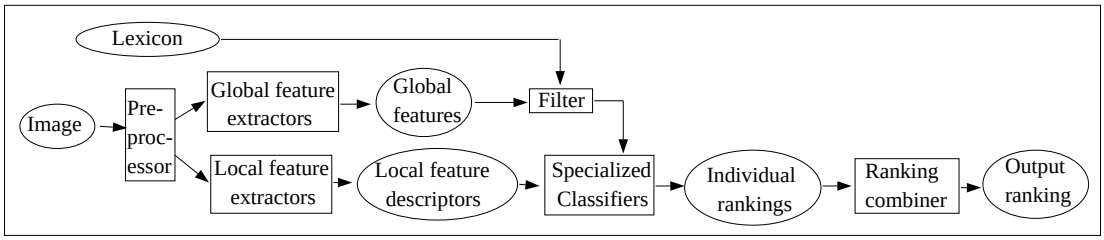


Figure 2: A Word Shape Based Recognition System.

quality. Faint images with exceptionally low density are rejected. Other images are enhanced by performing different levels of smearing and gap filling, according to the computed density level. Four reference lines, namely, the upper bound, lower bound, top line, and base line, are located within the image. These reference lines provide a frame of reference to describe the location of shape features.

4 Word Shape Feature Extraction

The shape of a word is characterized by global and local features. Global features describe wholistic characteristics of a word. An estimate of the *number of characters* in a word and the *letter case* are among the more important global features. Local features describe certain fine details of the letters composing the word, including the distribution of the strokes, curves, concavities, dots, holes, edges and endpoints. Studies of human reading performance provide some hints about the importance of certain features in recognition [13]. In our system, a heuristic technique is used to choose the features. The features are chosen so that visually similar words can be discriminated. The reliability of feature detection is also taken into consideration.

4.1 Extraction of global features

The global features we use include the case of the word (purely upper, lower or mixed), and the word length estimate (number of characters in the word). Words in the lexicon that do not match with the computed global features are ignored.

Computing Word Length The word length is estimated by a component merging method. Columns containing black pixels are labeled in the image. Neighboring columns are then merged until the size of the resultant component exceeds a threshold. The components are then re-examined. If it is determined that the width to height ratio exceeds that of a regular character, the component is split into two at a valley of the vertical projection profile. The number of obtained components is taken as a point estimate of word length.

Obtaining an exact word length estimate is equivalent to character segmentation in difficulty. In order to allow for errors, the point estimate obtained by the method described above is expanded to be a range by examining the difference between the size of each component and the average component size. This range is usually smaller for better quality images in constant pitch fonts, since the component sizes tend to be more regular for those images.

Computing Word Case Word case is estimated by two approaches. The first one compares the relative distances between the estimated reference lines. If there is sufficient height in the ascender or descender region, the word is determined to be in mixed case [2]. Another estimate is given by examining the heights of the processed connected components. If there is sufficient variation in height, the word is determined to be in mixed case. The final decision is made if the two estimates agree. Otherwise, the word case is left undetermined, and the feature prototypes of all the word cases are compared in the following stages.

4.2 Extraction of local features

Local features describe details of the shape of the letters composing the word. The relative location of local features makes up the shape of a word. The local features used can be described as in four sets. They are (1) stroke distribution, (2) edge features, (3) end point features, and (4) letter shape features. The stroke distribution is to capture the distribution of black pixels across the image, with each black pixel being labeled as belonging to a stroke of one of four different directions. The end points are points where the strokes terminate. The selected edges of the strokes provide an approximate *segmented skeleton* of the input image. The letter shape features are those perceptual features like holes, dots, ascenders, descenders, diagonal strokes and curves. These are perceptual features determined to be essential in human reading activity [13]. Each of these feature sets provides information on the shape of the word from different perspectives.

The local features are computed from the input image by a collection of feature extractors, mostly developed in related pattern recognition studies including [12] and [4].

Stroke Distribution The distribution of strokes is computed using the *local direction contribution* method suggested in [12]. At each black pixel in the image, the length of the current run in each of the four directions: (1) east-west, (2) northeast-southwest, (3) north-south, (4) northwest-southeast is computed. The pixel is labeled with the direction in which the *run length is a maximum*. That is, each black pixel is labeled as part of a stroke of one of the four directions. Figure 3(a)-(e) shows an example of such pixel labeling. The distribution is given by the count of labeled black pixels of each type at every area partition in the image. The counts are normalized by the number of black pixels in the image.

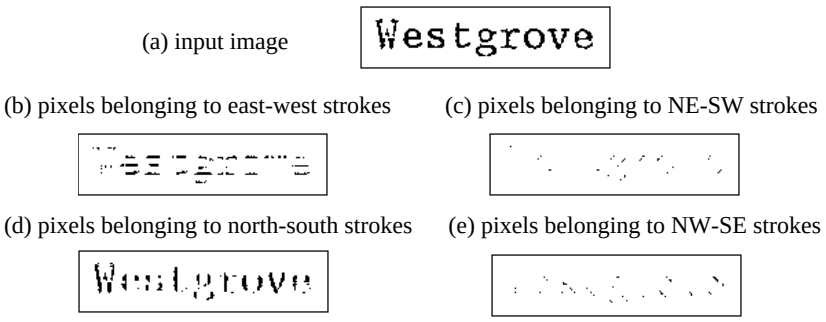


Figure 3: Example pixel labeling in stroke distribution calculation.

[2] In the application to words in address block images, purely lower case is not considered.

Edges The first step in computing the edges is to identify the edge pixels. This is done by checking, for every black pixel, if there is any horizontal or vertical *run* starting or ending at that pixel. In this process, each detected edge pixel is labeled as belonging to a horizontal edge or a vertical edge. Figure 4(a) shows the identified edge pixels in the image given in Figure 3(a). The edges are given by connecting the detected edge pixels of the same orientation. Since the edges appear in pairs, only one is needed to represent each stroke. Therefore the longer side of each pair is selected. Very short edges are treated as noises and ignored. The selected edges are shown in Figure 4(b). These selected edges are registered with their orientation (horizontal or vertical) and location.

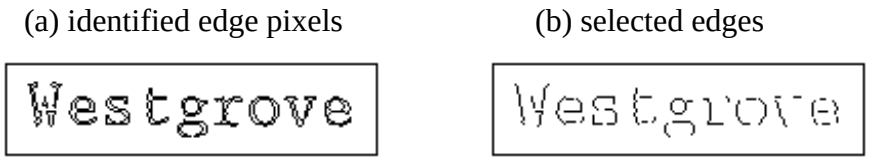


Figure 4: Example edge detection and selection.

End Points The end points are where the strokes start and terminate. This is computed by convolving a set of end point templates with the skeleton image. The skeleton is computed by consecutive applications of the thinning algorithms given by [1] and [9]. Figure 5 gives an example of skeletonization by these methods and the endpoints detected by convolution.

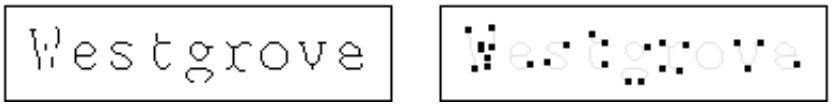


Figure 5: A skeleton image and the end points detected by convolution.

Letter Shape Features This set of features contains the more abstract perceptual features, as compared to the lower level edges and end points. These include *dots and holes*, computed by connected component analysis on the input image and the negated image; *ascenders, descenders, short vertical strokes, and horizontal strokes*, computed by run length analysis; *diagonal strokes*, computed by chain code analysis on the skeleton image; *curves*, computed by a multi-scale curvature analysis [4] on the chain codes obtained from the skeleton image; and *bridges between neighboring vertical and horizontal strokes*, computed by checking whether two neighboring strokes belong to the same connected component in the input image.

4.3 Synthesizing local feature prototypes

In order to match the features extracted from a word image to the features of the words in a lexicon, we need models or feature prototypes for the words in the lexicon. It is impractical to collect a large set of image samples for each word in the lexicon, therefore we adopt an alternative strategy, that is, to synthesize the words from a set of *character* samples of various fonts. The features of the synthesized words are then extracted and stored as prototypes.

5 Representation and Matching

As described before, most of the local features used are basically shape features of the *characters* composing the word. It is the *representation schemes*, i.e., the *descriptors* we use that enable us to match the features *independent of character segmentation*. We use descriptors that register the feature locations with reference to *a global frame*. This global frame consists of the four reference lines, dividing the vertical axis into the ascender region, the middle region, and the descender region, and ten equally-sized divisions along the horizontal axis. The middle vertical region is further divided into upper and lower halves. As a result, the image area is partitioned into 4 vertical regions, and 10 horizontal regions, i.e., divided into 40 cells. Extra white columns are removed before the area partitions are made. Figure 6 shows the area partitions given by such a frame.



Figure 6: An image after removal of extra white colomns and its area partitions.

Another concern in designing the descriptors is that *an appropriate distance (or similarity) function* is needed for matching each of these feature descriptors with its counterpart in the stored prototypes. The defined distance function must be appropriate for the type of the descriptor.

Feature Vector The stroke distribution is described by a 160-dimensional feature vector. The number of black pixels of each of the four types is counted in each of the 40 cells and normalized by the count of all black pixels. This results in a vector of 160 dimensions. These vectors are compared by computing the Euclidean distance.

Numeral Strings The feature vector description is subject to discretization error in representing the feature locations. For the edge, end point and letter shape features, the presence or absence of each feature is meaningful so that the descriptors should allow matching of corresponding features located in *neighboring* cells. In our design, these features are first categorized according to their vertical positions. For instance, a vertical edge may be labeled as an 'ascender region vertical edge,' a 'descender region vertical edge,' or a 'middle region vertical edge.' Then their locations are represented by numeral strings specifying the horizontal regions they are in. For instance, a string 'O: 256' describes that there are three holes in the image, and that they are located at the 2nd, 5th and 6th horizontal regions respectively. One numeral string is used for each of the typed features. There are 29 such numeral strings in total.

These numeral strings are compared by the *minimum edit distance*, computed by a *dynamic programming* algorithm given in [15]. The edit costs are derived from the reliabilities of detection measured on a training set. The substitution costs are the numerical differences between the digits in the strings. Hence a string '256' can be matched with another string '146' with a distance of 2 (i.e., $(2-1)+(5-4)+(6-6)$). Though there are some slight differences in their locations, these two sets of features can still be matched.

Symbol Strings The co-existence and relative locations of the local features of different types are also important for recognition. This information is captured by descriptors in the form of symbol strings, much like the strings used in syntactic pattern recognition systems. For instance, a string

'AOOD' is used to represent that an ascender is followed by two holes and then a descender. Some features are on top of others thus are better described in separate strings. In our system, 5 symbol strings are used to represent 5 different subsets of the local features. These symbol strings are also compared by minimum edit distances. The edit costs are derived from the reliabilities of detection, as well as substitution frequencies observed from a training set.

Figure 8 shows the computed global features and 35 local feature descriptors for an example image.

6 Classification

There are totally 35 descriptors (1 feature vector, 5 symbol strings, and 29 numeral strings) used to represent the local features. Each of these 35 descriptors has an associated distance measure. A nearest-neighbor classifier is designed using each of these 35 descriptors and distance measures. This results in 35 such classifiers, each is specialized in comparing one feature descriptor. Six composite distance measures are also introduced. They are sums of distances of 6 different feature subsets. As a total, there are 41 nearest-neighbor classifiers used in our system. Among these classifiers, 39 are distinctive from one another. The other two are introduced to study the effect of classifier correlation in designing a combination strategy.

Table 1 gives a summary of the local feature descriptors and the corresponding distance functions. Table 2 gives a summary of the local features used by the 41 classifiers.

Table 1: Summary of Feature Descriptors and Corresponding Distance Functions

Features	*Descriptor*	*Example*	*Distance Function*
stroke distribution	160 dimensional vector	[10 26 0 0 ....]	Euclidean distance
relative location of edges, end points, ascenders, descenders, holes, dots, curves etc.	symbol strings, one for each feature subset. each symbol represents a specific feature	$AOOAD$	minimum edit distance, with edit costs derived from reliability of detection, and frequency of substitutions
horizontal position of edges, end points, ascenders, descenders, holes, dots, curves etc.	digit strings, one for each feature. digits are positions w.r.t. the 10 width partitions	2334567	minimum edit distance, with edit costs derived from reliability of detection, and differences of digit values

6.1 Lexicon Filtering and Ranking

The input lexicon is first filtered by the global features. Only the words with word length falling within our estimated bounds, and with matching word case are retained in the filtered lexicon.

Each of the 41 local feature based classifiers produces a ranking of the filtered lexicon. The resultant 41 rankings are then combined by a union strategy [6]. A set of thresholds, derived by observing the classifier performance on a training set, is applied to these rankings so that a number of top decisions from each classifier are selected. The union of these selected decision is re-ranked using a *group consensus function*, which combines the individual rankings to derive a consensus ranking. The group consensus function we used is referred to as the *Borda count* [3]. For each particular word, the Borda count is the sum of the number of words ranked *below* it by each

Table 2: Summary of Local Features Used by the Classifiers

Classifier No.	*Features / Distances*
1	stroke distribution vector
2	symbol string for edge features
3	sum of symbol string distance for edge features (identical to 2)
4-11	location strings for edge features
12	sum of distances in 4-11
13	symbol string for endpoint features
14	sum of symbol string distance for endpoint features (identical to 13)
15-19	location strings for edge features
20	sum of distances in 15-19
21-23	symbol strings for letter shape features
24	sum of symbol string distances for letter shape features
25-40	location strings for letter shape features
41	sum of distances for 25-40

of the classifiers. The final ranking is given by ordering the lexicon words in descending Borda count. This ranking is the output of the word shape recognition system. The classifier combination algorithm is illustrated in Figure 7. Figure 9 shows the top 15 decisions by the 41 classifiers and the combined decisions for the example image shown in Figure 8.

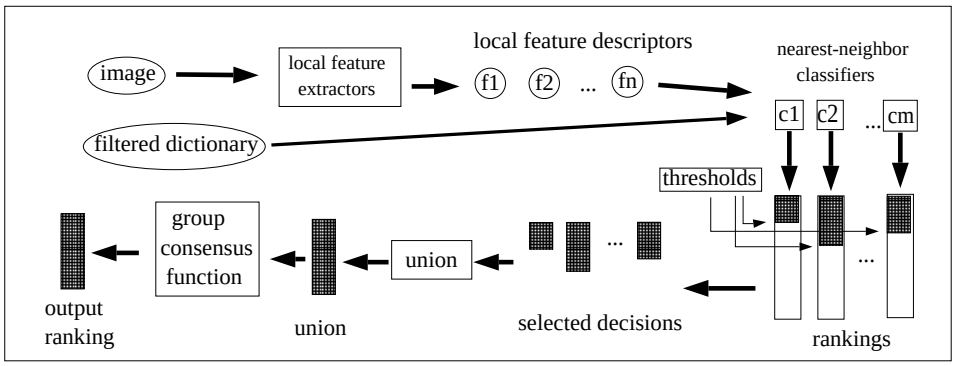


Figure 7: Combination of local feature based classifiers.

7 Experimental Results

Algorithm Training The recognition algorithm has been developed using a collection of images of *machine-printed* postal words obtained from live mail. They were scanned at roughly 200 pixels per inch and binarized. The font and quality of the images vary. The image database was divided into separate training sets and testing sets. The feature extractors were developed and modified by observing performance in a small initial training set of about 200 images. The distance functions (edit costs) were derived by running the feature extractors on a separate set of 2417 images. The stroke distribution vector was computed for characters in 169 font samples. Other feature descriptors were computed for characters in 10 font samples. Feature prototypes for the words in

the lexicon were obtained by synthesizing feature descriptors computed for the characters. Another set of 481 training images, different from those we used in developing the feature extractors and distance functions, was used to obtain the thresholds used in combining the local feature classifiers.

Algorithm Testing The recognition algorithm was tested on an image database of city names extracted from address block images. None of these images was used in the training stages. The input lexicon contained 186 words from a city name database. The same lexicon was used for all the images. There were totally 827 images in the test set. In the preprocessing stage, the program rejected 2 images because of faintness indicated by extremely low density.

Case estimation was correct for 814 (98.43%) images in the test set, where correctness means that the program did not make a wrong assignment, that is, either the case assignment was correct or the program left the word case undetermined. Word length computation was correct for 815 (98.55%) images. The word length estimate was given as an interval with lower and upper bounds. The average width of the interval is 1.98. For 804 (97.22%) images, the true word remained in the lexicon filtered by these global features, that is, the computed global features were correct. The average size of the *filtered* lexicon was 62.77 (words).

Local features were then extracted from the image and matched against the prototypes. After the classifiers produced the rankings, the neighborhood thresholds were applied to select a number of top decisions from each classifier. The union of these selected decisions was formed and re-ranked using the Borda count. Table 3 summarizes the recognition performance measured as the percentage correct (inclusion of the true word) in a various number of top choices in the final output ranking.

Table 3: Summary of Performance

Correct in No. of Top Choices	No. of Images	% of Total
1	749	90.57%
2	771	93.23%
3	775	93.71%
4	782	94.56%
5	788	95.28%
10	794	96.01%
15	796	96.25%
rejected	2	0.24%
total	827	100.00%

8 Conclusions and Future Work

A methodology for word recognition was presented that is based on word shape analysis without character segmentation and recognition. This method is used in a fixed vocabulary word recognition system and outputs a ranking of a given lexicon. The ranking specifies the order in which lexicon words are most likely to be found in the image. The objective of the method is to insure that a small number of words at the top of the ranking contain the word in the image.

The algorithm presented in this paper calculates both global and local features from a word image. The global features specify wholistic characteristics of the word. They are used to filter

the lexicon and remove entries from consideration that do not possess the specified global features. The local features include information about distribution of the strokes, edges, end points and other perceptual shape features. A set of detectors for local features are applied to each image, deriving 35 feature descriptors. Each descriptor is used to rank the lexicon filtered by the global features. These rankings are combined to yield an output ranking.

This technique was tested using word images segmented from address block images captured on a postal OCR. Experimentation is discussed where a typical lexicon contains 186 words. A 96% correct rate was achieved within the 10 best decisions for 827 test images.

The word shape analysis approach avoids committing errors in character segmentation and premature character recognition. However, it does not take advantage of the usefulness of isolated character information for some cases such as abbreviations and well-printed text. We plan to develop a robust word recognition algorithm by integrating a word shape based recognizer with an isolated character based recognizer. We also plan to refine the current word shape based algorithm, as well as conduct more studies on the strategies for classifier combination and the organization of multiple classifiers in a robust word recognition system.

Acknowledgements

The support of the Office of Advanced Technology of the United States Postal Service is gratefully acknowledged. Carl O'Connor of USPS as well as Gerardo Garcia and Gilles Houle of Arthur D. Little, Inc. provided helpful encouragement and useful criticisms in development of this project. Electrocom Automation supplied the address block and font sample images. Yan Li of SUNY at Buffalo developed the word length estimation module. Peter Cullen, Michal and Piotr Prussak, and Ralph Ames of SUNY at Buffalo helped in preparing the training and testing databases.

References

[1] H.S. Baird, K. Thompson, A VLSI architecture for binary image classification, *From the Pixels to the Features*, pre-proceedings, cost 13 workshop, France, August 22-23, 1988

[2] H.S. Baird, K. Thompson, Reading chess, *IEEE Transaction of Pattern Analysis and Machine Intelligence*, **PAMI-12**, 6, June 1990, 552-559.

[3] D. Black, *The Theory of Committees and Elections*, Cambridge University Press, London, 1958.

[4] K. Deguchi, Multi-scale curvatures for contour feature extraction, *Proceedings of the 9th International Conference on Pattern Recognition*, November, 1988, 1113-1115.

[5] R.O. Duda, P.E. Hart, *Pattern classification and scene analysis*, Addison-Wesley, New York, 1973.

[6] T.K. Ho, J.J. Hull, S.N. Srihari, Combination of structural classifiers, *Pre-Proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition*, New Jersey, June 13-15, 1990, 123-136.

[7] J.J. Hull, Inter-word constraints in visual word recognition, *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*, Montreal, Canada, May 21-23, 1986, 134-138.

[8] J.J. Hull, A computational theory and algorithm for fluent reading, *Proceedings of the Third IEEE Conference on Artificial Intelligence Applications*, Kissimmee, Florida, February 23-27, 1987, 176-181.

[9] M.S. Landy, Y. Cohen, G. Sperling, HIPS: A unix-based image processing system *Computer Vision, Graphics, and Image Processing*, **25**, 1984, 331-347.

[10] J. Mantas, An overview of character recognition methodologies, *Pattern Recognition*, **19**, 6, 1986, 425-430.

[11] J. Mantas, Methodologies in pattern recognition and image analysis - A brief survey, *Pattern Recognition*, **20**, 1, 1987, 1-6.

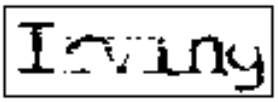
[12] S. Mori, K. Yamamoto, M. Yasuda, Research on machine recognition of handprinted characters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**, 4, July 1984, 386-405.

[13] H. Singer, R.B. Ruddell, *Theoretical Models and Processes of Reading*, International Reading Association, Inc., Newark, Delaware, 1985.

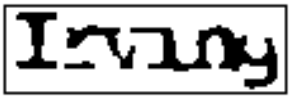
[14] S.N. Srihari, R.M. Bozinovic, A multi-level perception approach to reading cursive script, *Artificial Intelligence*, **33**, 2, October, 1987, 217-255.

[15] R.A. Wagner, M.J. Fischer, The string-to-string correction problem, *Journal of the ACM*, **21**, 1, January 1974, 168-173.

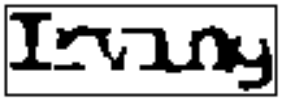
Input image



Preprocessed image



Packed image



Area Partitions



Estimated word length: 4 <= length <= 7
Estimated word case: Mixed

Stroke Distribution Vector

[10 width partitions]

24 22 0 0 0 0 0 0 0 0 0 [horizontal] [region 1]
0 1 30 35 17 17 0 11 1 0 [horizontal] [region 2]
24 27 24 0 0 24 27 12 21 29 [horizontal] [region 3]
0 0 0 0 0 0 0 0 1 19 [horizontal] [region 4]
0 1 0 0 0 0 0 0 0 0 [northeast-southwest] [region 1]
0 0 5 1 0 0 0 1 4 0 [northeast-southwest] [region 2]
0 0 0 0 2 0 2 0 0 0 [northeast-southwest] [region 3]
0 0 0 0 0 0 0 0 0 0 [northeast-southwest] [region 4]
12 6 0 0 0 0 0 0 0 0 [vertical] [region 1]
38 28 32 21 2 35 15 32 44 12 [vertical] [region 2]
23 10 4 11 26 30 21 29 46 34 [vertical] [region 3]
0 0 0 0 0 0 0 0 1 26 [vertical] [region 4]
1 0 0 0 0 0 0 0 0 0 [northwest-southeast] [region 1]
0 1 0 2 0 1 0 19 2 1 [northwest-southeast] [region 2]
0 1 0 7 4 0 6 0 17 4 [northwest-southeast] [region 3]
0 0 0 0 0 0 0 0 0 0 [northwest-southeast] [region 4]

End Point Features

$alallumluumlllludm$ [endpoint feature symbolic string]
a 01 [location : ascender region endpoints]
d 8 [location : descender region endpoints]
m 149 [location : middle region endpoints]
u 1348 [location : upper middle region endpoints]
l 01124778 [location : lower middle region endpoints]

Edge Features

$ulMlUuMlLulMlMuMmlMlM$
[edge feature symbolic string]
A $$ [location : ascending vertical edge]
D $$ [location : descending vertical edge]
M 0356789 [location : middle-centered short vertical edge]
U 1 [location : upper-centered short vertical edge]
L 4 [location : lower-centered short vertical edge]
u 0247 [location : upper horizontal edge]
m 8 [location : middle horizontal edge]
l 0245679 [location : lower horizontal edge]

Letter Shape Features

$ASBSTSBD$ [letter shape feature symbolic string 1]
$22L2RL1R1$ [letter shape feature symbolic string 2]
$P($ [letter shape feature symbolic string 3]
O $$ [location : holes]
o $$ [location : dots]
A 0 [location : ascenders]
S 567 [location : short vertical strokes]
D 9 [location : descenders]
T 7 [location : top bridges between verticals]
B 58 [location : bottom bridges between verticals]
1 79 [location : horizontal strokes]
2 025 [location : 2-aligned horizontal strokes]
3 $$ [location : 3-aligned horizontal strokes]
L 46 [location : left bridges between horizontals]
R 58 [location : right bridges between horizontals]
P 3 [location : positive slopes]
N $$ [location : negative slopes]
(8 [location : left bent curves]
) $$ [location : right bent curves]

Figure 8: Feature extraction example.

C 1 fv	C 2 edge sym	C 3 edge sym	C 4 A	C 5 D	C 6 M	C 7 U	C 8 L	C 9 u	C 10 m	C 11 l	C 12 edge sum	C 13 endptsym	C 14 endptsym
Irving	Lummi	Lummi	Caffee	Absher	Missabe	Canal	Absher	Prairi	Boling	Prairie	Solem	Levels	Levels
Benton	Pitzer	Pitzer	Canmer	Addison	Solem	Chopin	Adwolf	Redkey	Book	Sickles	Cogan	Harney	Harney
Corinth	Puunoa	Puunoa	Canton	Adwolf	Wabana	Cloyd	Airport	Shatley	Bouquet	Aflex	Swatara	Huxley	Huxley
Punaluu	Suite	Suite	Cayucos	Aflex	Addison	Cogan	Angeles	Tansi	Brook	Bakers	Sovran	Meyer	Meyer
Central	Dams	Dams	Coats	Airport	Bebee	Colony	Austin	Branc	Buda	Boehler	***Irving***	Knapps	Knapps
Canton	Lucas	Lucas	Cogan	Albert	Belden	Corinth	Bechyn	Brook	Calis	Buhler	Itasca	Shatley	Shatley
Decherd	Sano	Sano	Cornog	Albq	Benton	Cornog	Buskirk	Cedar	Canmer	Cardiff	Ojai	Randa	Randa
Daems	***Irving***	***Irving***	Corpers	Almena	Dolphin	Couch	Cogan	Central	Cornog	Central	Sano	Venango	Venango
Leetown	Dalas	Dalas	Cowpens	Almo	Dressen	Dabolt	Decherd	Cogan	Dalas	Decherd	Swansea	Kosoma	Kosoma
Emmaus	Couch	Couch	Gamma	Alstyne	Jaroso	Gamma	Dolphin	Cooley	Dallas	Duhring	Scenic	Huzzy	Huzzy
Dressen	Randa	Randa	Ganeer	Amber	Robbin	Garland	Gerry	Ennis	Dams	Jocasse	Strawn	Harvey	Harvey
Garland	Kosoma	Kosoma	Gerry	Amity	Wilburn	Grand	Grand	Forty	Dilts	Kendall	Branc	Branc	Branc
Okobojo	Duhring	Duhring	Grange	Amoco	Woden	Haltom	Gueydan	Gerry	Duhring	Madelia	Missabe	Eleanor	Eleanor
Berwyn	Syosset	Syosset	***Irving***	Anch	Amber	Island	Heceta	Holland	Ennis	Natalia	South	Moyie	Moyie
Boling	Bakers	Bakers	Itasca	Angeles	Boling	Keaa	Holland	Itasca	Erle	Oquaga	Lawn	Noxen	Noxen

C 15 a	C 16 d	C 17 m	C 18 u	C 19 l	C 20 endptsum	C 21 shape s1	C 22 shape s2	C 23 shape s3	C 24 s123 sum	C 25 O	C 26 o	C 27 A	C 28 S
Almena	Amity	Lesage	Eleanor	Anch	Eleanor	***Irving***	Wisdom	Almena	Coats	Airport	Absher	Amoco	Almo
Alstyne	Beaudry	Erle	Grange	Rhoades	Melrose	Couch	Coats	Almo	Colony	Almo	Addison	Bakers	Boling
Eleanor	Boling	Meet	Jaroso	Alstyne	Wares	Cornog	Colony	Alstyne	Forsan	Amity	Adwolf	Bays	Daems
Emmaus	Colony	Melrose	Oatmeal	Amber	Wier	Colony	Index	Amoco	Huzzy	Amoco	Aflex	Berwyn	Dams
Farmers	Cooley	Meyer	Powder	Index	Dressen	Yolyn	Sovran	Anch	Stahl	Anch	Airport	Book	Holland
Forsan	Cornog	Revenue	Puunoa	Kingdom	Kees	Sturgis	Cowpens	Angeles	Island	Austin	Albert	Branc	Island
Forty	Duhring	Newdale	Shindle	Moyie	Madre	Long	Solem	Ardel	Ulvah	Chopin	Albq	Brook	Kingdom
Harney	Forty	Steptoe	Branc	Rosburg	Mano	Just	Cozad	Buda	Wentz	Colony	Almena	Burrow	Mano
Harvey	Gerry	Boehler	Couch	Adwolf	Meyer	Hickory	Dalas	Dreyfus	Elfers	Corinth	Almo	Buskirk	Revenue
Heceta	Harney	Dressen	Erle	Albert	Newdale	Lawn	Dallas	Island	Eller	Couch	Alstyne	Canmer	Rosburg
Hovey	Harvey	Glennie	Garland	Driver	Revenue	Gerry	Ceder	Itasca	Solem	Ennis	Amber	Cayucos	Sabinal
Huzzy	Hickory	Jocasse	Kingdom	Heceta	Harney	Tansi	Kingdom	Lesage	Couch	Forty	Amity	Cogan	Salerno
Itasca	Hovey	Pendle	Kulli	Hokes	Jaroso	Mano	Powder	Lugo	Plano	Hickory	Amoco	Cornog	Sano
Jaroso	Hurley	Rolette	Newdale	Holland	Keaa	Sunny	Leetown	Sovran	Sturges	Hill	Anch	Corpers	Scenic
Jocasse	Huxley	Seattle	Sabinal	***Irving***	Masten	Almo	Corpers	Ulvah	Sovran	Huzzy	Angeles	Cowpens	Airport

C 29 D	C 30 T	C 31 B	C 32 1	C 33 2	C 34 3	C 35 L	C 36 R	C 37 P	C 38 N	C 39 (	C 40)	C 41 shapesum	Combined
Albq	Albq	Albq	Cedar	Central	Absher	Amber	Ottoman	Adwolf	Absher	Almena	Absher	Sovran	***Irving***
Beaudry	Alstyne	Amoco	Dolphin	Coastal	Addison	Ardel	Poydras	Alstyne	Addison	Almo	Addison	Couch	Meyer
Boling	Boling	Corinth	Eleanor	Colony	Adwolf	Cedar	Schofer	Buda	Adwolf	Alstyne	Adwolf	Island	Seattle
Cornog	Cayucos	Dodd	Kendall	Corpers	Aflex	Choctaw	Airport	Buhler	Aflex	Boling	Aflex	Forsan	Dressen
Duhring	Colony	Jaroso	Leetown	Cowpens	Airport	Chopin	Amber	Cayucos	Airport	Book	Airport	Lawn	Punaluu
Hickory	Cowpens	Keough	Albert	Dolphin	Albert	Kendall	Amoco	Cowpens	Albert	Branc	Albert	Strawn	Lesage
Irving	Garland	Lawn	Cardiff	Forsan	Albq	Lowdell	Angeles	Dreyfus	Albq	Brook	Albq	South	Central
Kenedy	Holland	Cayucos	Central	Kendall	Almena	Lucas	Ardel	Eleanor	Almena	Burrow	Almena	Lucas	Couch
Long	***Irving***	Cloyd	Corinth	Poydras	Almo	Moffit	Canal	Gueydan	Almo	Celilo	Almo	Long	Shatley
Mining	Kosoma	Couch	Cowpens	Rosburg	Alstyne	Morges	Cedar	Huxford	Alstyne	Cloyd	Alstyne	Dams	Redkey
Redkey	Lugo	Gamma	Earlham	Schofer	Amber	Newdale	Chopin	Huxley	Amber	Cornog	Amber	***Irving***	Decherd
Rosburg	Missabe	Harney	Knoll	Scholls	Amity	Sovran	Cloyd	***Irving***	Amity	Cozad	Amity	Canal	Leetown
Shatley	Reseda	Knapps	Mclean	Sovran	Amoco	Switch	Coastal	Keaa	Amoco	Dilts	Amoco	Rosburg	Steptoe
Weakly	Salerno	Knippa	Natrl	Bouquet	Anch	Wilburn	Corpers	Kemp	Anch	Dreyfus	Anch	Wier	Sano
Amity	Ship	Peridot	Ottoman	Caffee	Angeles	Wisdom	Cowpens	Lugo	Angeles	Duhring	Angeles	Cornog	Melrose

Figure 9: Top 15 decisions by the 41 classifiers and the combined decisions.