A Computational Model for Recognition of Multi-font Word Images

Abstract

A computational model for the recognition of multi-font machine-printed word images of highly variable quality is given. The model integrates three word recognition algorithms, each of which utilizes a different form of shape and context information. The approaches are character recognition based, segmentation based, and word-shape analysis based. The model overcomes limitations of previous solutions that focus on isolated characters. In an experiment using a lexicon of 33,850 words and a test set of 1,671 highly variable word images, the algorithm achieved a correct rate of 89% at the top choice and 95% in the top ten choices.

Keywords: character recognition, word recognition, pattern recognition, multiple classifiers, decision combination.

1 Introduction

Although visual word recognition has been extensively studied for several decades, the automatic recognition of printed words of multiple font types and varying quality remains an unsolved problem. It is well-known that global contextual knowledge provides useful information that may facilitate recognition. Global contextual knowledge includes domain knowledge, linguistic knowledge, and any information other than the visual shape of the word that helps determine its identity.

For recognizing a given word image, global contextual knowledge can be distilled into a lexicon, or list of words, so that the image need only be classified as one of the words in the lexicon. A higher rate of recognition can be expected if more than one choice is output. The result of recognition, ideally, is to rank the lexical entries according to a measure of belief that the word corresponds to the given image. Such a ranking provides a useful basis for further analysis and understanding of the text.

The computational task of word recognition is therefore defined as follows: given an image of a word and a lexicon that contains the word, determine a ranking of the lexicon such that the word in the image is ranked as close to the top as possible. A perfect solution would always assign the top rank to the true word.

A lexicon provides many contextual hints that facilitate recognition. For example, consider a word with six characters. Without a lexicon, such a word would be one of $26^6 = 308,915,776$ possible strings that are composed of characters in the English alphabet. Even a large, 100,000 word lexicon contains only 0.03% of all the possible strings, and hence contains many constraints. For example, if there are no words where the character 'p' is followed by the character 's', the 'ps' combination can be eliminated from the character decisions.

Recognition of a word from its image is based on visual information extracted from the image. Visual features may be extracted from the word as a whole object. Alternatively, a word may be segmented, or separated, into individual characters, each of which is recognized in isolation. The word's identity is then derived from the decisions made on individual characters.

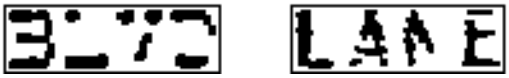
Methods to use shape information in a word image and context information from a lexicon can be divided into three major classes: *character recognition based* methods, *segmentation-based* methods, and *word-shape analysis* methods.

Character recognition based methods segment individual character images from a word image, extract features from each character and assign it to a character class, and finally achieve word recognition by post-processing character decisions using a lexicon (Bledsoe and Browning 1959; Rosenbaum and Hilliard 1975; Schuermann 1978). In segmentation-based approaches, features extracted from a character are not used to identify characters in isolation. Instead, the features of all characters are matched in the context of a word using feature models for the words in the lexicon (McClelland and Rumelhart 1981; Rumelhart and McClelland 1982). In word-shape analysis, words are described and recognized as whole units, without segmenting the word into characters. Features are extracted from a word image directly and matched with features of word prototypes (Ho et al. 1990b; Hull 1987; Hull 1988).

Each method is best for recognizing word images with certain degradation characteristics. Character recognition based methods are suited to word images whose characters are well-isolated and clearly printed. They are also good for images of short words, since short words have little useful contextual information. In some images, although the characters can be reliably segmented, it is difficult to recognize the individual characters in isolation. Such images are best recognized by segmentation-based methods. Images that are difficult to segment, and those with characters that would be deformed by size normalization, are more suitable for word shape analysis. Figure 1 shows example images and indicates the best recognition method for each.



(a) Word images that are best recognized by character recognition based methods.



(b) Word images that are best recognized by segmentation-based methods.



(c) Word images that are best recognized by word-shape based methods.

Figure 1: Word images of different types and the best recognition method for each.

Word images of multiple font types and variable quality caused by printing technique, paper quality, imaging technique, preprocessing method, and so on, may look like any of these examples. Since each of the three approaches excels only for images of a particular type, the performance of each approach over all types of images is limited. It can therefore be expected that all three classes of methods are needed to achieve satisfactory recognition performance over a wide range of images.

A computational model is proposed as a robust solution to the recognition of multi-font word images of highly variable quality (Ho 1992). The objective is to integrate the three word recognition approaches such that the weakness of one method may be complemented by the strength of another

method. Cooperation of the individual methods is achieved by the use of a group consensus function that combines and improves the decision performance of the individual methods.

2 A Computational Model for Word Recognition

The proposed computational model for word recognition consists of an activation control mechanism, a set of parallel classifiers, and a decision combination mechanism. The control mechanism uses information from the input image and the input lexicon. It activates suitable words in the lexicon based on a set of global features computed from the image and then selectively applies appropriate classifiers to the image. The parallel classifiers take three different approaches (character recognition based, segmentation based, and word-shape based) to recognize the word. Each computes a ranking of the activated words in the lexicon. The decision combination mechanism combines the rankings and produces a consensus ranking (Figure 2).

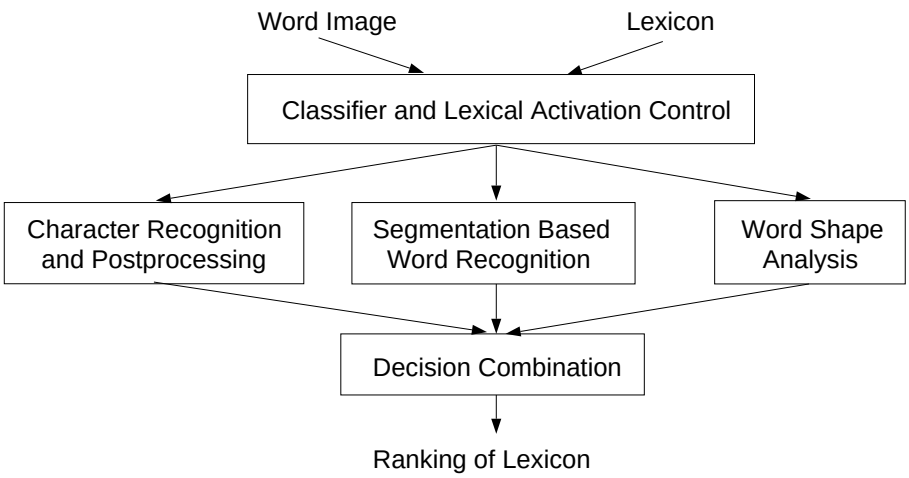


Figure 2: A computational model for word recognition using multiple approaches.

The classifiers range over a continuum in the scope of shape and context information that is used to recognize a word. At one extreme, the character based approach first recognizes individual characters as local units, and then modifies the decisions by lexical constraints. At the other extreme, the word-shape analysis approach recognizes a word as a whole unit, and uses contextual hints directly in feature description and matching. Thus, the integration of these methods operates on multiple scales, ranging from isolated characters to whole words, and thereby introduces a redundancy that is needed to tolerate variations in the appearance of a word image.

The control strategy uses both top-down and bottom-up information to activate the appropriate classifiers and words in an input lexicon. Top-down information includes any high level information and global context. For example, if a nonsense string like a serial number is expected, the classifiers that rely on a limited lexicon are not activated and only character recognition is applied. Bottom-up information is derived from a preliminary analysis of the word image. For example, if it is determined that the image is printed in a constant-pitch font and can be reliably

segmented, then high confidence can be placed in the character recognition decisions, and there is no need to activate word-shape based classifiers. Bottom-up information also includes global shape descriptions computed from the word image, such as an estimate of the word length. The control strategy selectively activates the words in the lexicon that match these descriptions.

When more than one classifier is activated, multiple rankings of a lexicon are produced. A decision combination mechanism that combines the rankings is applied. It consists of three combination functions, each of which computes a confidence score for each word in the lexicon. A combined ranking is derived from the confidence scores.

The rest of this paper describes each component of the model in detail. Sections 3 to 5 describe each of the three word recognition approaches. In each section, a general description of the approach is first presented, followed by an implementation example. For simplicity, the examples assume that the input images are machine-printed words and are binarized. Section 6 describes the decision combination functions. In Section 7, it is shown how the example techniques are integrated. Section 8 describes experimental results obtained by applying this algorithm to a set of word images extracted from live mail.

3 Character Recognition Based Approach

A character recognition based approach uses a segmentation algorithm to separate a word image into individual character images. A recognition algorithm is then applied to each character image. The character decisions are then postprocessed and a ranking of the input lexicon is derived (Figure 3).

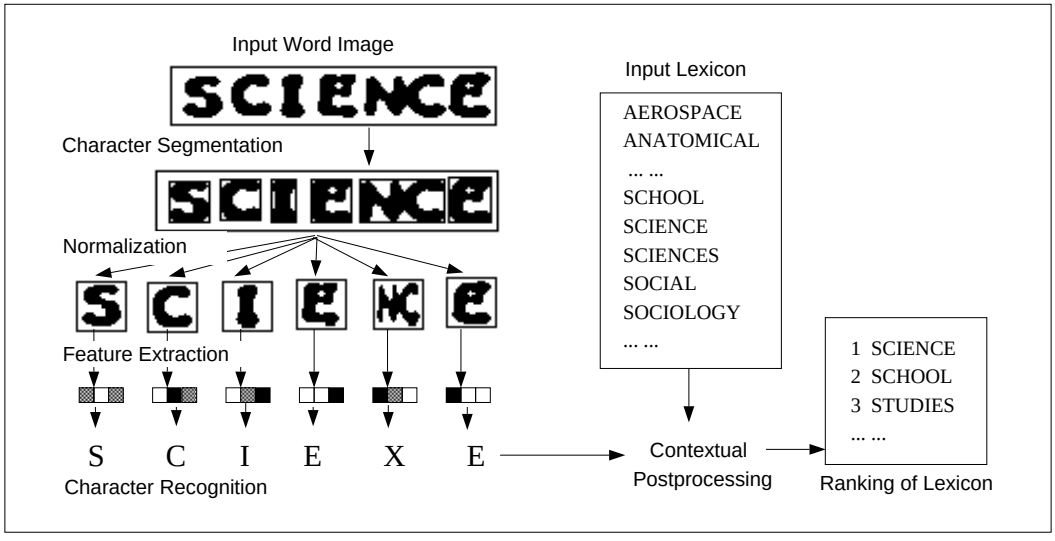


Figure 3: The character recognition approach.

A character segmentation procedure divides a word image into individual character images (Elliman and Lancaster 1990). Commonly used methods for character segmentation are based on analysis of the contours, analysis of vertical projection profiles, and extraction of connected components. Segmentation may be assisted by pitch and character size estimation (Tsuji and Asai

1984). Heuristic rules are often added to locate the best segmentation points. It is also suggested that segmentation may be coupled with recognition (Casey and Nagy 1982). The projection profile based technique is comparatively easy to implement and is taken as an example.

Many methods for character recognition have been studied. Examples of useful techniques include template matching, Bayesian classifiers, and structural methods (Mantas 1986; Mori et al. 1984). Recently, a fuzzy template matcher that uses weighted values of neighboring pixels in computing a distance between corresponding pixels in the input image and the stored templates has shown promise (Chen and Srihari 1991). It performs well on noisy images and is hence taken as an example.

The conventional approach uses only a single character recognizer whose decisions are postprocessed using a lexicon. Section 3.1 describes useful postprocessing algorithms in such an approach.

It is also observed that more than one character recognition algorithm may be applied to a character image. Reliability in recognition can be improved by using multiple recognizers and measuring their agreement. Valuable information is carried by the reliable character decisions, which may be used in contextual postprocessing using a lexicon. In Section 3.2 we propose a method that uses results of multiple character recognition algorithms.

3.1 Postprocessing Decisions From A Single Character Recognizer

Methods proposed to use a lexicon to correct character decisions include n-gram techniques (Riseman and Ehrich 1974), string editing methods (Wagner and Fischer 1974), as well as the Markov models and the Viterbi algorithm (Hull and Srihari 1982; Shinghal and Toussaint 1979). The n-gram techniques are used to remove invalid combinations of character decisions that are not contained in a lexicon. It does not give a ranking of the valid words. The Markov models and the Viterbi algorithm are inconvenient in processing decisions that may contain segmentation errors. The effectiveness of the string editing methods depends on a set of edit costs that need to be manually adjusted by observing results with a large training set.

Many character segmentation techniques can segment a majority of word images into the proper number of characters with at most one error (i.e. plus or minus one from the true word length). Therefore a matching algorithm that tolerates such an error level is useful. Similarly, most recognition algorithms are able to include the correct character class in the top two decisions. A matching algorithm may place special emphasis on the top two decisions, while not excluding the possibility that they are wrong. A heuristic string matching algorithm is developed based on these observations.

The algorithm takes the top two decisions for each character in a word, and constructs a set of strings using all top decisions and then by replacing each top decision in turn with the second choice. To avoid combinatorial complexity, only one replacement is made in each string. The strings are matched with the words in the lexicon that are of the same length, and those of one character longer or shorter. The matches are graded by the number of common characters in the two strings. A penalty is added to the matches that used the second choices. In matching words of unequal lengths, a character is removed from each position of the longer string in turn. A penalty is added to degrade the matches of words of unequal lengths. The overall score for a word is the maximum it received from all the matches. The lexicon is then ranked by the scores.

3.2 Postprocessing Decisions From Multiple Character Recognizers

More than one recognizer may be applied to a character extracted from a word image. Using multiple recognizers introduces a redundancy that improves the recognition reliability. Reliable decisions on characters are valuable to word recognition. They can be used by a special postprocessing algorithm that uses only the reliable decisions and ignores others.

An example technique is given as follows. Assume that a word image is segmented into isolated characters, which are then normalized to a fixed size. Six character recognizers are applied to the normalized characters. Four of them are nearest-neighbor classifiers using four different sets of features. The four feature sets are the pixel values, the Baird features (Baird et al. 1989) (defined on a set of 32 7×7 templates), the local stroke directions (Mori et al. 1984), and the weighted neighbors used in (Chen and Srihari 1991). The other two classifiers are Bayesian classifiers with the assumption that the feature values are independent of one another. The pixel features and the Baird features are used in these two classifiers.

Features used in these six classifiers are sensitive to different shape variations and image degradation. The pixel values contain the lowest level information in the image. They are most robust against noise but are least useful in representing a shape structure. The weighted neighbors take into account pixel values in a neighborhood of a fixed size, and are hence more robust against local shape variations. The Baird features are intended to detect some local shape attributes such as an edge of a particular orientation. The stroke directions are defined by the shape of a stroke. They are most invariant to the shape structure but are sensitive to defects such as the breaks in a stroke. Therefore, these methods give a range of descriptive power when used simultaneously. The advantages of different definitions of feature similarity are exploited by using both a nearest-neighbor classifier and a Bayesian classifier for pixel values and Baird features.

The agreement of the six classifiers is measured to obtain reliable character decisions. For each character, the top choice given by each classifier is taken. If all six classifiers agree on a decision, that decision is considered as the most reliable and assigned a score of 6. A score of 5 is assigned to a character if five classifiers agree, and 4 if four classifiers agree. Decisions agreed on by three or less classifiers are ignored. Figure 4 shows an example image together with the character decisions by the six classifiers.

As shown in Figure 4, even when all the six classifiers agree, there is still a chance that the decision is an error (see the character decision 'C' in the figure). Typically in those cases, a human looking at that character in isolation may make the same mistake. Therefore, contextual postprocessing with minor allowances for errors is still necessary.

```
Classifier 1:  TLNNYSCN
Classifier 2:  TENNY5CN
Classifier 3:  TENNYSCN
Classifier 4:  TLNNYSCN    6 classifiers agree:  T-NNY-CN
Classifier 5:  TENNYSCN    5 classifiers agree:  T-NNYSCN
Classifier 6:  TENNYSCN    4 classifiers agree:  TENNYSCN
```



Figure 4: An example word image (TENNYSON) and top character decisions by 6 classifiers.

A regular expression matcher is used to postprocess the reliable decisions. A set of character constraints are generated using the decisions agreed on by 6, 5, or 4 classifiers. The constraints are in two groups. One group is to match the strings formed by a combination of the reliable characters in the word, the other group is to match single reliable characters.

In the first group, all the reliable decisions in a word are matched in a single expression. There are three expressions in this group, derived from the decisions with confidence scores 6, 5, and 4, respectively. In a position where a reliable decision of that level is obtained, the decision is represented by an equivalence class containing the decided character, such as '[w|W]', '[x|X]', '[I|l]', and '[C|D|O|c|o]'. The purpose of using the equivalence class is to allow for common shape confusions. If the character heights relative to the base line are considered, some of these confusions such as [w|W] may be omitted to improve accuracy. A wild card character is placed in a position where a reliable decision of that level is missing. For instance, the string 'T-NNYSCN' in Figure 4 is represented by the expression 'T[A-Z|a-z]NNY[S|s][C|D|O|c|o]N', where '[A-Z|a-z]' means a wild card character. A different score is associated with each expression.

In the second group, each reliable character decision is matched in an individual expression. A number of wild card characters are placed to the left and the right of the character, according to the position of that character in the word image. The numbers are not exact, optional characters are used to allow for some fuzziness in the position of the decided character. For instance, in the example given in Figure 4, the decision 'S' is represented by the expression '....(.?|..)[S|s].(.?|..)', where each '.' is equivalent to '([A-Z]|[a-z])', and '(.?)' means that character may or may not be there, and '(.?|..)' means that there may be zero, one or two characters at that position. Again, a score is associated with each expression and the score varies with the reliability level of the character decision.

The regular expressions thus derived are matched to the words in the lexicon using the Unix utility 'awk' (Aho et al. 1980). A word in the lexicon that matches an expression receives the score associated with that expression. A word may match more than one expression. In such cases, the scores are accumulated. The words are then ranked by the accumulated scores.

4 Segmentation-Based Word Recognition

An alternative to the character recognition based techniques is to defer decisions about the character identity and to perform *segmentation-based word recognition.* This technique is suitable for word images where the characters can be reliably segmented, and better recognized together with other characters in the word.

In this approach, features are extracted from character images and matched at the word level (Figure 5). A word image is first segmented into individual character images, which are then normalized to a fixed size. Features are extracted from the normalized character images and represented by feature vectors. These character feature vectors are then concatenated to form a word feature vector, which is then matched with the prototypical feature vectors for the words in the lexicon. Hence the character features are compared in the context of a word.

This method is most useful for word images that are easy to segment but whose characters are difficult to recognize in isolation. An example image is given in Figure 6. This image is so broken that most of the shape features are lost, and the isolated characters are so degraded that even a human looking at them individually would have difficulty in guessing their identities. However, when the characters are placed together as in the original word, they reinforce one another and the word identity can then be determined.

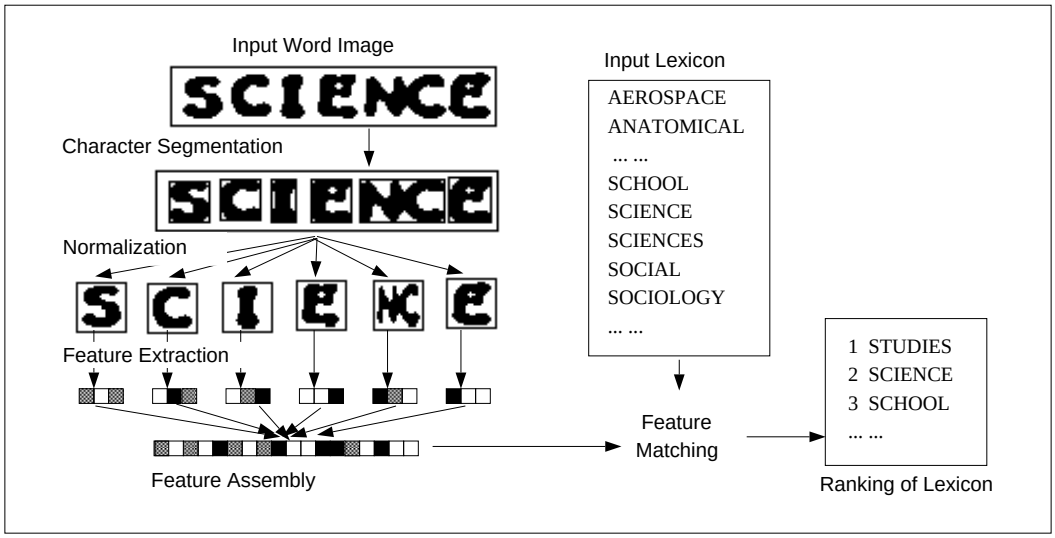


Figure 5: The segmentation-based word recognition approach.

Binary pixel values are an example set of features useful in this approach. Each segmented character is normalized to a 24x24 grid. The pixel values of each segmented character are then concatenated to form a word feature vector. Thus, a word segmented into 6 characters has a feature vector of 24x24x6 = 3,456 components.

The vector computed from an input image is matched to the prototype vectors for words in the lexicon. The prototype vectors for each word are synthesized with a set of sample font types.

The Hamming distance is used in matching these vectors. That is, the distance between two vectors is the number of different components in the two vectors. The computed distance is normalized by the number of characters compared. Note that the compositional nature of the feature vector allows for an optimization in distance computation. Once the character-level distances are computed and stored in a table, the word-level distances can be evaluated using that table.

Words in the lexicon that are of lengths equal to the number of extracted characters are compared directly. The words with lengths different by one from the segmentation are compared with one character removed at a time from the longer vector. The minimum distance among all the comparisons to the same word with various characters removed is taken as the distance between the input and that word. The distances from the input to the words with unequal lengths are penalized by a weight, which is determined experimentally. The words are then ranked by the adjusted distances.

5 Word-Shape Analysis

This approach attempts to describe and compare the shape of a word as a whole object (Figure 7). Features that describe the details of a word shape are extracted and represented by a feature vector. The feature vector is matched to prototype vectors of an input lexicon and a ranking of the lexicon is produced.

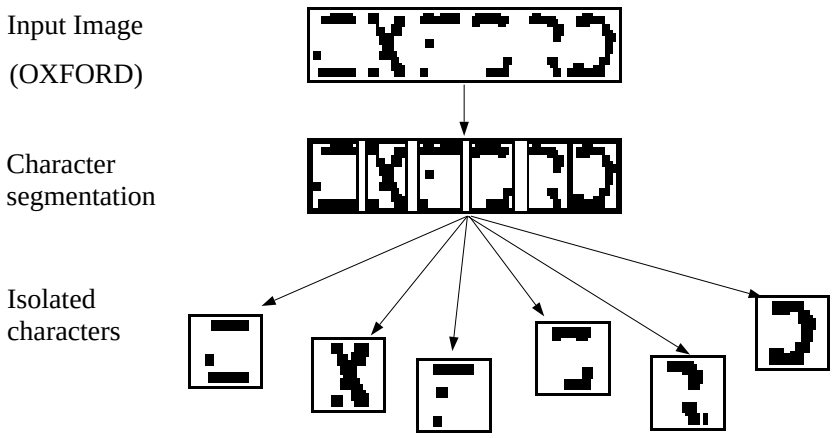


Figure 6: A word image and the segmented characters.

The advantage of this approach is that some errors in character segmentation and premature decisions on character identities are avoided. It is especially suitable for images that are difficult to segment into characters, or where the characters are distorted when they are extracted and normalized.

A global reference frame is needed to describe features detected across the whole word. This frame uses four reference lines including the upper and lower boundaries of the image, the top line, and the base line. The reference lines divide the vertical axis into the ascender region, the middle region, and the descender region. The middle vertical region is further divided evenly into upper and lower parts. Ten equal-sized divisions are made along the horizontal axis. As a result, the image area is partitioned into 4 vertical regions, and 10 horizontal regions, i.e., 40 cells. Figure 8 shows the area partitions given by this reference frame. Extra white spaces between adjacent characters are removed before area partitioning.

A set of features, referred to as the stroke direction distribution, are used to describe the shape of a word. This feature set captures the spatial distribution of black pixels across the image, with each black pixel labeled as belonging to a stroke in one of four directions. The stroke directions are computed using the *local direction contribution* method described in (Mori et al. 1984). At each black pixel in the image, the length of the current run in each of the four directions east-west, northeast-southwest, north-south, and 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 in one of the four directions. Figure 9(a)-(e) shows an example of such a pixel labeling. The stroke direction distribution is described by a 160-dimensional feature vector, which stores counts of black pixels of each of the four types in the 40 cells. The counts are normalized by the total number of black pixels in the image.

A word-shape feature vector extracted from an image is matched with similar feature vectors synthesized for the words in the lexicon. A city-block distance (Duda and Hart 1973) is computed between the input vector and all the prototypical vectors. A ranking of the lexicon is then derived

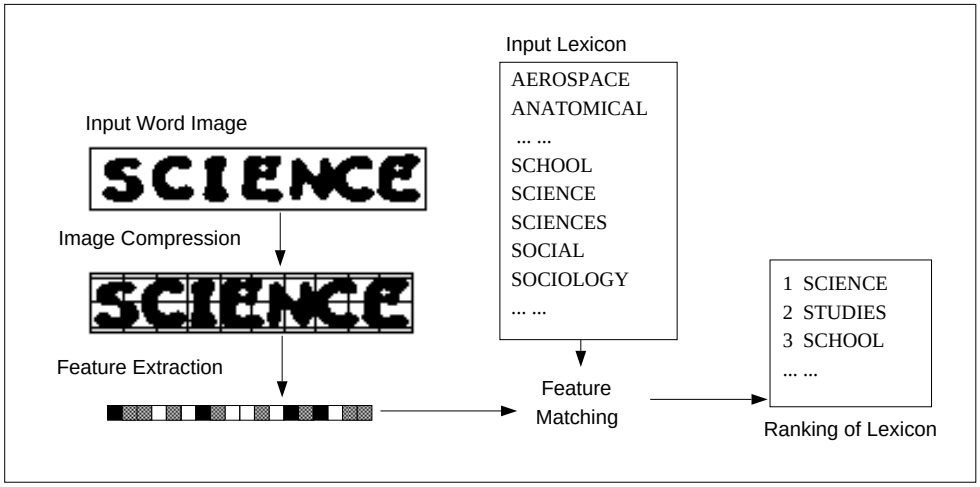


Figure 7: Word-shape analysis approach.



Figure 8: An image and its 40 area partitions.

by sorting the words in ascending order by their distances to the input vector.

The feature vectors for the words in a lexicon are derived as follows. A prototype word image is first created by appending the images of the characters in the word from a font sample. The feature vectors are then calculated from this image. A number of fonts are used to guarantee good performance.

Similar word-shape analysis techniques can be developed using other feature sets. An example is a set of features proposed in (Baird et al. 1989), which are defined by 32 feature templates, each of size 7×7. The features defined by these templates are detected by convolving the templates with a word image and thresholding the responses. Each non-zero response after thresholding represents a feature of a particular type that is detected at that position. The outputs are described by a 1280-dimensional feature vector, which stores counts of the 32 features detected in the 40 cells. The city-block distance metric is applied to match these feature vectors and derive a ranking of the lexicon.

6 Decision Combination

Ideally, a control strategy selects the best classifier for each input image, and only that classifier is applied. If such a dynamic selection is always successful, and if the selected classifier always gives a correct decision at the top choice, there is no need for decision combination. However, this is

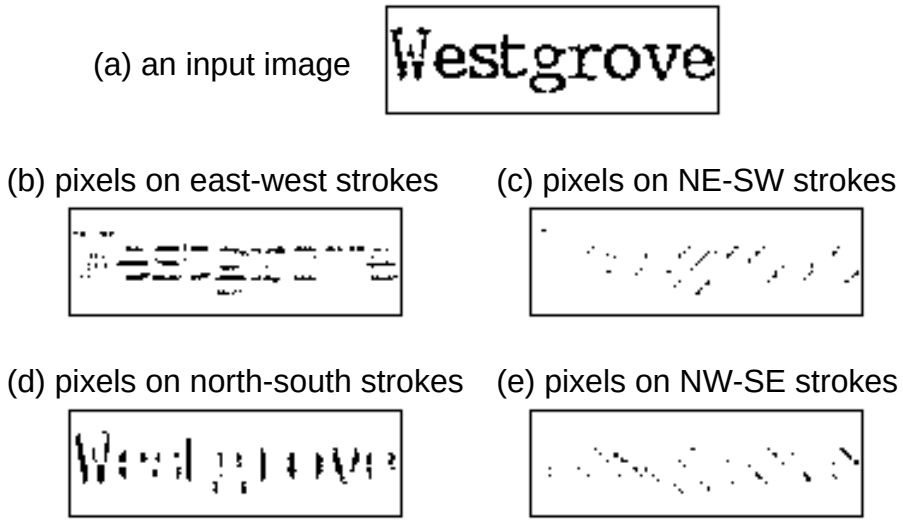


Figure 9: An example of the stroke direction distribution.

possible only in a limited number of cases where the image quality is exceptionally good. In other cases, it is very difficult to predict which classifier is the best for each input image. Moreover, the decision of an individual classifier is not always correct. In such cases, it is suggested that all the classifiers be applied in parallel to the image. As the classifiers are independent of one another, the redundancy in decisions may be used to improve the correctness (Ho et al. 1990a). This is the task for the decision combination mechanism.

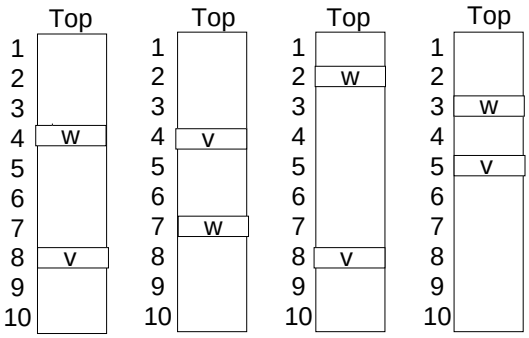
The decision combination mechanism uses the results of the activated classifiers to generate a consensus decision. A confidence score is computed for each word in a lexicon using the set of rankings produced by the independent classifiers. A consensus ranking is then generated by sorting the words by the computed confidence scores. Three combination functions are proposed for computing the confidence scores.

The first one is a highest rank method. A score is assigned to each word that is the highest rank among the ranks it receives from all the classifiers. Therefore, a word receives a high score as long as at least one classifier ranks it high. The combined ranking is given by sorting the words by the scores. Words ranked above a threshold in the combined ranking are taken as a neighborhood. This method is particularly useful to reduce a large lexicon to a small neighborhood, since it takes advantage of the best classifier for each image.

The second function is referred to as the Borda count (Black 1963). For a set of rankings on the same set of classes, the Borda count for each class is the sum of the number of classes ranked below that class by each classifier. The combined ranking is given by arranging the classes in descending Borda count. Intuitively, if a class is ranked near the top by more classifiers, its Borda count tends to be larger and will be closer to the top in the combined ranking. It is a measure of agreement among the classifiers.

The third function is a generalization of the Borda count by assigning weights to each ranking. The confidence score for each word is computed as a weighted sum of the ranks it receives from the

classifiers. The weight for each classifier is estimated using a regression analysis method that involves a *logit* (log-odds) transformation (Agresti 1990). Using a training set of images and their recognition results, the significance of contribution of each classifier can be estimated by the regression analysis. Figure 10 illustrates the use of the three combination functions with an example.



1. The highest rank method

H (w) = min (4, 7, 2, 3) = 2

H (v) = min (8, 4, 8, 5) = 4

Final ranking obtained by sorting the words by H in ascending order

2. The Borda count method

B (w) = (10 - 4) + (10 - 7) + (10 - 2) + (10 - 3)
= 6 + 3 + 8 + 7
= 24

B (v) = (10 - 8) + (10 - 4) + (10 - 8) + (10 - 5)
= 2 + 6 + 2 + 5
= 15

Final ranking obtained by sorting the words by B in descending order

3. The logistic regression method

L (w) = 0.23 X (10-4) + 0.16 X (10 - 7) + 0.41 X (10 - 2) + 0.35 X (10 - 3)
= 0.23 X 6 + 0.16 X 3 + 0.41 X 8 + 0.35 X 7
= 7.59

L (v) = 0.23 X (10 - 8) + 0.16 X (10 - 4) + 0.41 X (10 - 8) + 0.35 X (10 - 5)
= 0.23 X 2 + 0.16 X 6 + 0.41 X 2 + 0.35 X 5
= 3.99

Final ranking obtained by sorting the words by L in descending order

Figure 10: An example of ranking combination using three methods.

The three functions are applied in turn to the word rankings given by the individual classifiers for an input image. The lexicon is first reduced to a small neighborhood by the highest rank method. The ranks of the words in the neighborhood are then combined using the Borda count and the estimated logistic regression function. The regression method is more effective, but it needs a training set to estimate the weights for the classifiers whereas the Borda count does not. The choice about which method to apply depends on the conditions in a particular application.

7 An Implementation of the Model

An algorithm that implements the computational model has been developed. The algorithm consists of an activation control strategy, five parallel classifiers, and a decision combination module. A preprocessor is also used to enhance the quality of a word image. The lexical activation control is considered as a filtering stage that reduces the input lexicon. This implementation does not include dynamic selection of classifiers, thus all the classifiers are active. Five parallel classifiers are applied to all images that produce five independent rankings of the filtered lexicon. The rankings are then input to a decision combination module, which produces a final consensus ranking that is the output of the algorithm.

7.1 Lexical Activation by Global Features

Global features are the wholistic and simpler aspects of the shape of a word that can be easily and reliably measured. They are used to reduce the input lexicon to simplify subsequent classification tasks. The global features that are useful for this purpose include estimates of the word length and the word case (upper, lower, or mixed). If the estimates are accurate, only the words with lengths matching the estimated length need to be activated, and the words are converted to the estimated case.

It is difficult to estimate word length precisely, therefore an interval estimate is used instead. This is done by first performing character segmentation, and then relaxing the character count to be an interval, by examining the variations in sizes of the segmented characters.

Word case is estimated by examining variations in the sizes and the vertical alignment of the connected components. The result is confirmed by an analysis of the heights from the located top line and base line to the upper and lower boundaries of the image. If no agreement is obtained, the word case is left undetermined.

The words in the input lexicon that match the estimated word lengths are activated. They are converted to the estimated case or cases. These words are to be discriminated by the parallel classifiers.

7.2 Lexicon Ordering by Parallel Classifiers

Five parallel classifiers are used to order the activated words in the lexicon. The classifiers are based on the three approaches to word recognition that are described in Sections 3 to 5. The methods used in the five classifiers are summarized in Table 1. Each of the five classifiers produces a ranking of the lexicon that is filtered in the previous stage. The five independent rankings are input to the decision combination module.

7.3 Decision Combination

The rank order decisions by the five classifiers are combined using three combination functions. The highest rank method is first applied to reorder the activated words in the lexicon. An initial segment of the combined ranking is selected to be a neighborhood of the true word. The size of the neighborhood is selected taking the size of the lexicon into account. The top ten decisions from each classifier, if included in the neighborhood, are combined using the logistic regression function. The other words in the neighborhood are ordered by the Borda count function.

Table 1: Summary of methods used in the parallel classifiers

Classifier	*Method*
Classifier 1	A fuzzy character template matcher and a heuristic contextual postprocessing algorithm.
Classifier 2	Six character recognizers with decisions postprocessed by regular expression matching.
Classifier 3	A segmentation-based word recognizer with pixel values as features.
Classifier 4	A word-shape analyzer using stroke direction features.
Classifier 5	A word-shape analyzer using the Baird template features.

The final ordering is a concatenation of three rankings: first the ranking given by the logistic regression function, then the remaining words in the neighborhood that are ordered by the Borda count, and finally, the words outside the neighborhood that are ordered by the highest rank method. This final ordering is the output of the algorithm.

8 Experimental Results

The utility of the algorithm is demonstrated by an application to a set of machine printed word images obtained from live mail. The images were scanned on a postal optical character reader at 212 pixels per inch and binarized. The words are in unrestricted font types and are often very degraded. Figure 11 shows some example images in the data set.

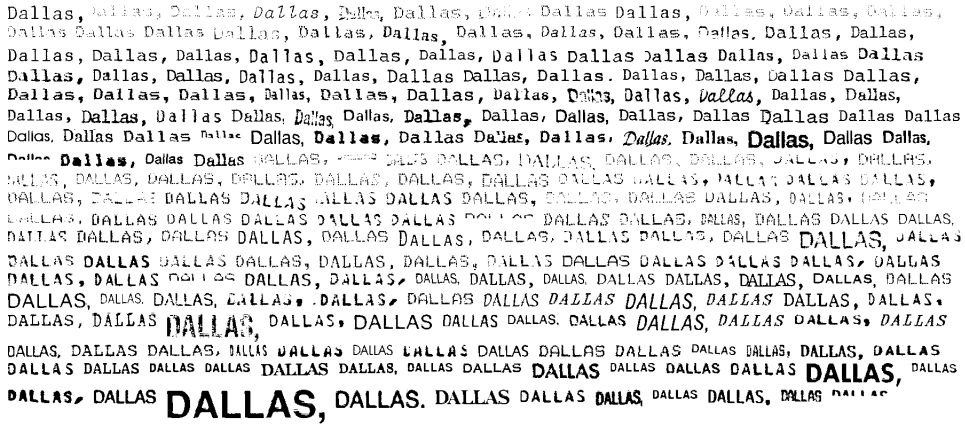


Figure 11: Examples of image degradation and font style variations included in the test set.

8.1 Training

The feature extractors were developed and modified by observing performance on a small initial training set of 200 images. The character recognizers were trained on a set of 19,151 sample characters extracted from 400 address blocks. The prototypes of stroke direction vectors used in word-shape

analysis were obtained by averaging those from 77 font samples. For the Baird feature vectors, 57 fonts were used. A set of 76 fonts was used to construct the prototype vectors for the segmentation-based method.

A set of 1,055 word images was used to generate decisions from the word classifiers. A lexicon of 33,850 postal words was used that included all the true words in the images. Each word in the lexicon is represented in both upper and mixed cases. In the application domain, purely lower case is not used. The results of this run were used to estimate the regression parameters for decision combination. All the five classifiers, as well as their combination by the highest rank, were used in the regression analysis. All the six rankings are determined to be contributing to the logit. The estimated weights were 0.1775, 0.2098, 0.3292, 0.0539, and 0.1880 for the five classifiers respectively, and 0.1864 for the ranking given by the highest rank method.

8.2 Testing

Another set of 1,671 images was used to test the algorithm. The same lexicon of 33,850 words was used. Table 2 summarizes the performance of the five word classifiers and the results at several stages in decision combination. In order to measure the performance of the classifiers using a lexicon with a fixed size, lexical activation (word length and word case filtering) is applied after recognition. Three subsets of sizes 1,000, 5,000, and 10,000 were also selected from the input lexicon. The objective was to determine the potential effect of a global contextual knowledge source that could provide reduced lexicons to word recognition and thereby improve its performance. Each of the subsets of the original lexicon contains all the true words in the images. The rest of the words in the subsets were randomly selected.

Table 2: Summary of performance on 1671 test images using a 33,850 word lexicon
(% Correct in Top N Decisions)

Descriptions	1	2	3	10	50	100	500
1. char recog with heuristic postprocessor	79.2	86.1	88.2	90.5	92.7	93.5	94.8
2. char recog with regular expression matcher	76.9	83.2	85.4	88.3	91.8	93.2	95.0
3. segmentation-based word recognition	74.8	84.1	86.3	90.5	93.4	94.6	95.5
4. word-shape using stroke direction features	42.4	53.7	59.5	72.1	81.9	84.9	90.8
5. word-shape using Baird features	58.9	70.0	74.5	82.9	88.5	90.2	93.2
6. combination of 1.-5. by highest rank	46.6	73.8	87.0	94.9	97.3	97.6	98.6
7. combination of 1.-6. by Borda count	83.1	88.1	90.7	94.6	96.3	97.0	98.6
8. combination of 1.-6. by logistic regression	88.4	91.2	92.7	95.1	97.4	98.1	98.6
9. results of 8. after case (upper,mixed) merging	88.7	91.4	92.9	95.2	97.6	98.5	98.9
10. results of 9. after word length filtering	88.9	91.6	92.9	95.3	97.8	98.4	98.7
11. results of 10. in a 10,000 word subset	92.6	94.0	94.9	97.0	98.6	98.8	98.9
12. results of 10. in a 5,000 word subset	94.1	95.0	96.0	97.8	98.7	98.9	98.9
13. results of 10. in a 1,000 word subset	95.5	97.1	97.9	98.7	98.9	98.9	98.9

Note that these results should not be interpreted as a fair evaluation of the effectiveness of each of the five classifiers. One reason is that many poor quality images were intentionally included in the image database to investigate the limitations of the methods. Another reason is that, the averaged vectors instead of the full set of prototypes for word shape analysis were used because of run-time efficiency constraints.

Besides general causes of errors due to the descriptive power of the features and effectiveness

of the similarity measures, there are special problems for each of the methods. Classifiers using character segmentation are vulnerable to segmentation errors caused by touching or severely broken characters. Character size normalization also caused some distortion in shape for unevenly degraded characters.

In word shape analysis, special problems are caused by errors in determining the 4x10 reference frame. These include errors in locating the reference lines due to breaks in the ascenders and descenders, or uneven faintness concentrated in the upper half or lower half of the image.

The effectiveness of the decision combination algorithms are well demonstrated. Using the highest rank method of decision combination, the lexicon was reduced to a neighborhood of 500 words with a 98.6% accuracy, which was not achieved by any of the individual classifiers. This method did not give a high correct rate at the top choice because of ties among the top 5 decisions. Better consensus rankings were achieved by the Borda count and the logistic regression method. There was a 9.2% gap between the top choice correct rate of the combination by logistic regression (8) and that of the best individual classifier (1). This shows that the integrated algorithm is significantly better than any of the individual classifiers in performance. The filtering stages could be useful in reducing run time but had no great impact on the rankings. The performance on the random subsets (lines 11,12,13) shows that other higher-level contextual constraints on the lexicon can be used to improve the final rankings significantly.

8.3 Run Time

The algorithm was not optimized at the time the test was performed. It was implemented in C, and the individual classifiers were invoked by system calls in a PERL script. The test was run on a SUN SPARC-2 running at 28 MIPS. The full algorithm ran at 16 minutes per word (4 minutes for 6 character recognizers, postprocessing and regular expression matching, 2.5 minutes for segmentation-based matching, 5.5 minutes for word shape analysis, and 4 minutes for external sorting and combination).

A recent re-implementation of the model using only three classifiers (classifiers 3,4 and another one similar to 1) runs at 2 minutes per word on the same machine. Efficiency was improved by code optimization, software integration and use of main memory for shape prototype storage. However, there is a 3% drop in the top choice correct rate when compared to line 9 in Table 2. This illustrates the tradeoff between speed, space and accuracy.

It should be noted that most parts of the algorithm can be conveniently executed in parallel. There is no interaction among the classifiers, so each can be run on a different processor. During the execution of each classifier, the most expensive steps are the distance computations, which can also be parallelized.

9 Summary and Conclusions

A computational model for word recognition has been proposed. It integrates three approaches, each of which uses shape and context information at a different level. The character recognition based approach analyzes individual character shapes and classifies them individually. The segmentation-based approach analyzes individual character shapes but determines character classes in the context of other character shapes. The word-shape based approach analyzes the overall shape of the word to assign it to a word class. A control strategy activates appropriate methods and suitable words in the lexicon, according to top-down information from global context, and bottom-up information

from an image. A decision combination mechanism combines the decisions given by the activated classifiers and outputs a consensus ranking.

The model is realized in an algorithm that uses five classifiers, each of which is based on one of the three different approaches to word recognition. The algorithm was tested using live mail images and postal words as a lexicon. In an experiment with 1,671 word images and a 33,850 word lexicon, the algorithm achieved a correct rate of 88.9% at the top choice and 95.3% in the top 10 choices. When the input lexicon is reduced to 1,000 words, a correct rate of 95.5% at the top choice and 98.7% in the top 10 choices was achieved. The performance of the algorithm is significantly better than each of the individual classifiers applied in isolation.

Future work includes further refinement of the control and decision combination strategies towards more flexible dynamic adaptation to both top-down and bottom-up constraints.

Acknowledgements The support of the Office of Advanced Technology of the United States Postal Service is gratefully acknowledged. Gary Herring and Carl O'Connor of USPS, as well as Gerardo Garcia and Gilles Houle of Arthur D. Little, Inc. provided helpful encouragement and useful criticisms. Jiah-Shing Chen provided the fuzzy template matcher for character recognition. Yan Li and Liang Li helped in implementing character segmentation, heuristic string matching and the segmentation-based method. Peter Cullen, Michal Prussak, Piotr Prussak and Ralph Ames assisted in the development of the database for the experiments. Edward Cohen provided useful comments on the manuscript. The authors are most appreciative for their help.

References

[1] Agresti A (1990), Categorical Data Analysis. John Wiley & Sons

[2] Aho AV, Kernighan BW, Weinberger PJ (1980), Awk - A Pattern Scanning and Processing Language (Second Edition). In: Unix User's Manual, Supplementary Documents, Regents of the University of California

[3] Baird HS, Graf HP, Jackel LD, Hubbard WE (1989), A VLSI Architecture For Binary Image Classification. In: From Pixels to Features, Simon JC (ed), North Holland, pp 275-286

[4] Black D (1963), The Theory of Committees and Elections. Cambridge University Press, London, 2nd ed.

[5] Bledsoe WW, Browning I (1959), Pattern Recognition and Reading by Machine. Proceedings of the Eastern Joint Computer Conference, 16:225-232

[6] Casey RG, Nagy G (1982), Recursive Segmentation and Classification of Composite Character Patterns. Proceedings of the 6th ICPR, Munich, pp 1023-1026

[7] Chen JS, Srihari SN (1991), A Fuzzy Correlation and Its Application to the Recognition of Low-Quality Characters. In preparation.

[8] Duda RO, Hart PE (1973), Pattern Classification and Scene Analysis. Addison-Wesley, New York

[9] Elliman DG, Lancaster IT (1990), A Review of Segmentation and Contextual Analysis Techniques for Text Recognition. Pattern Recognition, 23(3/4):337-346

[10] Ho TK, Hull JJ, Srihari SN (1990a), Combination of Structural Classifiers. Pre-Proceedings of the IAPR Syntactic and Structural Pattern Recognition Workshop, New Jersey, June 13-15, pp 123-136

[11] Ho TK, Hull JJ, Srihari SN (1990b), A Word Shape Analysis Approach to Recognition of Degraded Word Images. Proceedings of the Fourth USPS Advanced Technology Conference, pp 217-231

[12] Ho TK (1992), A Theory of Multiple Classifier Systems and Its Application to Visual Word Recognition. Doctoral Dissertation, SUNY at Buffalo, Department of Computer Science

[13] Hull JJ, Srihari SN (1982), Experiments in Text Recognition with Binary n-Gram and Viterbi Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4(5):520-530

[14] Hull JJ (1987), Hypothesis Testing in a Computational Theory of Visual Word Recognition. Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI), Seattle, Washington, pp 718-722

[15] Hull JJ (1988), A computational theory of visual word recognition. Doctoral Dissertation, SUNY at Buffalo, Department of Computer Science

[16] Mantas J (1986), An overview of character recognition methodologies. Pattern Recognition, 19(6):425-430

[17] McClelland JL, Rumelhart DE (1981), An interactive activation model of context effects in letter perception: Part 1. An account of the basic findings. Psychological Review, 88(5):375-407

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

[19] Riseman EM, Ehrich RW (1974), A Contextual Postprocessing System for Error Correction Using Binary N-grams. IEEE Transactions on Computers, C-23(5):480-493

[20] Rosenbaum WS, Hilliard JJ (1975), Multifont OCR Postprocessing System. IBM Journal of Research and Development, 19:398-421

[21] Rumelhart DE, McClelland JL (1982), An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. Psychological Review, 89(1):60-94

[22] Schuermann J (1978), A Multifont Word Recognition System for Postal Address Reading. IEEE Transactions on Computers, C-27(8):721-732

[23] Shinghal R, Toussaint GT (1979), Experiments in Text Recognition with the Modified Viterbi Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2):184-193

[24] Tsuji Y, Asai K (1984), Character Image Segmentation. SPIE Proceedings on Applications of Digital Image Processing VII, 405:2-9

[25] Wagner RA, Fischer MJ (1974), The string to string correction problem. Journal of ACM, 21(1):168-173