Pattern Recognition and Expert Systems

Ching Y. Suen, Stephen Malowany and Sam Adel

Centre for Pattern Recognition and Machine Intelligence
Concordia University, Suite GM-606
1455 de Maisonneuve Boulevard West
Montreal, Quebec H3G 1M8

Abstract

Both pattern recognition and expert systems have become important areas in applied computer science. This paper introduces these two subjects and describes how they may reinforce each other. It also presents some recent results on the application of expert system techniques to recognize totally unconstrained handwritten characters. The problems involved, the possibilities, and the challenges ahead, will be presented for discussion.

1. Introduction

Pattern recognition is a subject which is concerned with techniques and systems for the recognition of different types of objects and signals which may have different dimensions, e.g. 1, 2, 3 or more. Many pattern recognition applications have already been put into practice, e.g. optical character recognition, signature verification, speech recognition, fingerprint identification, detection of abnormalities in images, recognition of military targets and objects, etc. [see e.g. various journals and conference proceedings in this area]. Usually such a system must be trained before it can recognize the given type of objects or signals, e.g. by learning about their shapes and identifying features which can characterize them. Extraction of the salient features is one of the most difficult tasks in pattern recognition.

An *expert system* is a computer program that relies on a body of knowledge to perform a difficult task usually performed only by a human expert. A large number of successful applications of expert systems in different disciplines have already been reported, e.g. medical diagnosis, geological exploration, repair of computer networks, law enforcement, insurance and cost estimations, etc. [see e.g. Suen & Shinghal and various journals and conference proceedings in this area]. The principal power of an expert system is derived from its knowledge base obtained from human experts. It performs the task of an expert through an inference engine.

In order to connect the two subjects of pattern recognition and expert systems, let us first remind ourselves that many pattern recognition systems can be called expert systems because they are actually mimicking humans in performing their tasks, e.g. character recognition and speech recognition, which humans can do so well. However, it is extremely difficult to program a computer to mimick exactly how humans do these things. One reason is that they have been doing these things (e.g. recognizing friends' faces and their voices) so naturally and for so long that people find it very difficult to describe them to the computer, or even to other people. Indeed, many researchers have been trying to search for explanations of these phenomena, but without much success. As a result, many pattern recognition systems today remain imperfect or defective. Hence it is safe to say that a lot more has to be done before computers can truly perceive and recognize objects or signals. To illustrate this point, two examples will be given below. Both of them are related to computer recognition of handwritten characters, an area of intensive research carried out at CENPARMI.

2. Expert Systems for the Recognition of Handwritten Characters

There are infinite styles of handwriting and humans have no problems reading and recognizing them. Until now, computers using mathematical models have failed to match human performance in classifying handwritten characters. Classification of handwritten characters becomes especially difficult when computers are faced with confusing patterns. The following gives two examples of programming the computer to recognize unconstrained handwritten characters. The first one makes use of a graphical rule-based system, and the second one is a multi-expert system. Some results will also be presented to illustrate their effectiveness as well as the difficulties encountered by these systems.

2.1 Example 1 - Graphical Rule-based System

This section briefly presents a graphical rule-based system for recognizing isolated samples of handwritten digits. On-line pen stroke position data such as that acquired from a digitizer tablet or computer mouse is used as input. The system consists of four major components: a structural feature extractor, a rule-based classifier, a graphical user-interface (GUI), and a control module. The structural feature extractor segments the stroke data into sequentially connected portions from which geometric feature vectors are computed. These features are modeled after high-level structural primitives by which humans could describe the composition of characters. The feature vectors are then fed into a rule-based classifier where a heuristic rule base is applied by a conventional forward chaining inference engine. The classification proceeds by abstracting the quantitative feature vectors into *top level* qualitative elemental primitives. These primitives considered together are then matched against a knowledge base of digit templates. A list of possible digit identities with corresponding certainty factors is then concluded. The graphical user interface running under the X Window System provides ''point and click'' access to the system's underlying control module, data visualization of extracted features, animated display of digit samples, and the ability to draw new samples with the mouse. The control module simply interprets events generated by the user via the GUI, and dispatches commands to the various component processing modules.

2.2 Extraction of Structural Features

The goal of the structural feature extraction method used is to identify and describe characters using high-level primitives in much the same fashion as a human might do when presented with the image of a character and asked to elaborate on its structural composition. The feature extraction process consists of segmenting strokes into one or more descriptive subcomponents, which at the highest level, compose the character. For example, the character '5' may be described as having three major structural components; a horizontal bar, a vertical bar, and a curve with its concavity facing leftward. Relative positioning, size, connectivity, and directional information are also required to more precisely specify the inter-relationships among the major components. Since our objective is to identify characters based on their structure, the only temporal aspect exploited is the data point sequence.

Input data for a single character is represented as a linear array of coordinate pairs $C(i) = (x_i , y_i)$, where $i = 1,2,...,N$, and N is the number of points, including end of stroke indicators, which compose the character. For normalization purposes, the *size* of the input character is defined as the length of the diagonal of the character's bounding box.

Feature extraction is a four stage process. First, the input data is filtered to remove excess data points. Filtering consists of throwing out successive data points until a minimum inter-point distance is achieved. Raw feature segments are then identified from the stroke data. This involves sequential point by point processing to identify feature end points. A feature terminates when it violates any one of four specific constraints. These constraints ensure that features do not span multiple strokes, terminate at desirable vertices, rotate consistently in only a single direction, and have an arc of less than 180°. Feature vector values are then computed based on the identified raw feature stroke segments. Feature vectors are composed of the following eight structural and geometric elements: end point spread, apex point, apex depth, length, degree of curvature, direction of concavity, trajectory, rotation. Numeric values are normalized with respect to the size of the character. Directional values are given in terms of sixteen discrete, equal sub-ranges over 360°. Sequentially connected features of the same stroke are then considered for aggregation based on a set of four heuristics. This final stage joins neighboring features into larger, more representative features. The resulting high level features are then fed into the rule-based classifier.

2.3 Rule-Based Classification

Rule-based classification of digits is performed by an embedded run-time CLIPS expert system module. Input to the classifier consists of the set of feature vectors as calculated by the structural feature extractor. The features are considered in isolation, in connected pairs, and in connected triples to form top-level *primitives* representing common shapes such as bars, concavities, loops, S-curves, etc. These primitives are then labeled as being representative of one or more functional elements of some digit. For example, given a horizontal bar-shaped primitive in the upper region of the sample image, labels will be asserted indicating that this primitive could be considered as the the top bar of a '7', the top bar of '5', and the top bar of a '3' (for styles of 3's where the upper concavity is formed by two bars joined by a vertex in the upper-right region of the pattern).

The abstracted primitives, their labels, and their connectivity information, are then matched against an embedded knowledge base (KB) of structural digit templates expressed in terms of these top level primitives, resulting in classifications. Three sets of templates are used to represent digit prototypes consisting of one, two, and three primitives. Multiple templates exist for each numeral to account for variations in writing styles. The digit '1' for example has a template in all three sets because it can be drawn as a single vertical bar, a vertical bar with hat on top (like an upright harpoon), or a vertical bar with a hat and a horizontal bar at the base. The weighted proportion of the sample's features attributable towards a specific classification is used to calculate a degree of belief, or certainty factor associated with each classification.

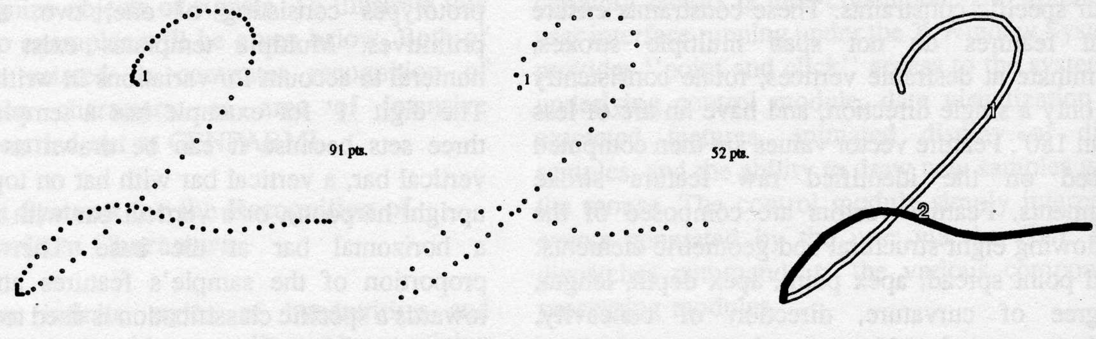
Training of the digit recognition rule base involved stepping through the training samples one by one, and manually editing rules such that the system would yield appropriate results.

2.4 Results

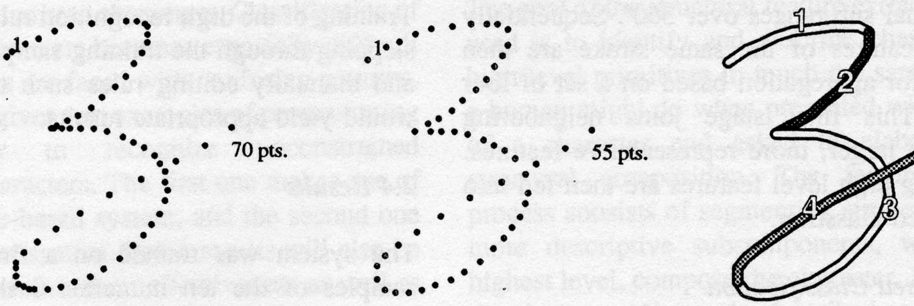
The system was trained on a first set of 1000 samples of the ten numerals 0 through 9. This balanced training set was obtained from ten individuals, each of whom contributed ten instances of each numeral. No special instructions or constraints were imposed regarding the size, placement, orientation, complexity, etc. of the characters. A balanced test set of 2300 samples obtained from 23 other individuals was used to evaluate the system. A few of these individuals were asked to draw difficult and confusing samples in a variety styles. A mixture of right and left-handed persons was used for both data sets.

Figure 1 illustrates some sample results of structural feature extraction obtained from two characters. For each sample, the left image shows the original raw data points, the center image shows the filtered data points, while the right image displays the stroke segments used to calculate the feature vectors. Below each row of images is a table listing the feature vector values obtained for each feature.

For the 1000-sample training set, the system achieved an overall recognition rate of 96.6%, rejection rate of 3.4%, and no substitutions, yielding a reliability rate of 100%. The overall average processing time is 97.24 msec. per sample.



Feature	Direction	Concavity	Rotation	Length	Curvature	Spread	Depth	End Points
1	lower-south-west	upper-south-west	positive	1.25	0.55	0.57	0.21	(0.38,0.74),(0.01,0.00)
2	upper-east	upper-south-east	positive	0.84	0.11	0.74	0.15	(0.01,0.00),(0.98,0.27)



Feature	Direction	Concavity	Rotation	Length	Curvature	Spread	Depth	End Points
1	upper-east	right-south	positive	0.43	0.04	0.41	0.05	(0.00,0.86),(0.62,0.98)
2	lower-south-west	lower-west	positive	0.41	0.05	0.39	0.05	(0.62,0.98),(0.21,0.60)
3	left-south	lower-west	positive	0.98	0.55	0.44	0.35	(0.21,0.60),(0.05,0.05)
4	lower-north-east	upper-east	positive	0.76	0.03	0.74	0.05	(0.05,0.05),(1.00,0.60)

Figure 1: Sample results of feature extraction for two characters.

On the 2300-member testing set, we observe an overall recognition rate of 81.3%, substitution rate of 1.17%, and rejection rate of 17.52%, for a reliability of 98.58%. The overall average processing time for the testing set is comparable to the training set at 85.74 msec. per sample.

2.5 Example 2 - Multi-expert Model

In this case, a multi-expert system is used to model the expertise of five human experts in identifying confusing cases of numerals (Figure 2). However, in order to get a better understanding of the design, implementation and performance issues of such a system, a complete system (involving simple and confusing cases) is constructed.

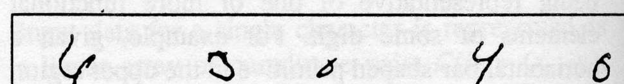


Fig. 2 **Samples of confusing cases of handwritten numerals.**

2.6 Knowledge Acquisition

The knowledge used in this expert system was gathered by the character recognition group of CENPARMI while they were working on the recognition of confusing cases of handwritten numerals. They selected 360 most difficult handwritten numerals from a database of 17,000 digits segmented from US zip codes written on envelopes by the general public. The data were provided by the US Postal Service for our research. These digits were pasted on a form in their original as shown in Figure 3. They were presented to 9 human subjects. Five of them were experts (researchers) in character recognition and the others were students.

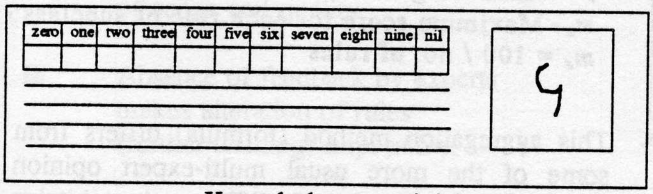


Fig. 3 **Knowledge acquisition form.**

The knowledge acquisition forms were filled by subjects without any discussion with one another or with those who conducted the experiment. They were required to write down how they arrived at the identities of the characters, with comments on the features used, and any clues which led to their decisions. These knowledgeable comments totalled 3,277, and were entered into a database. This database was subsequently reorganized so that all comments of the five experts with respect to a specific sample would appear under one another, making it easier to elicit knowledge for a specific case or class.

2.7 Contribution & Perfomance By Experts

It is well known that experts neither contributed nor performed similarly. Figure 4 and Table 1 show the number of comments written by each expert.

Expert	No. of Comments
1	1,358
2	718
3	310
4	408
5	433

Table 1

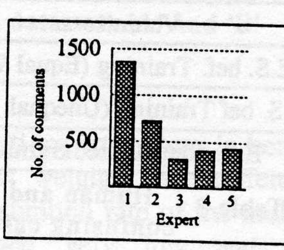


Fig. 4

Table 2 shows the performance of different experts in recognizing 360 confusing cases of numerals. The real identity of the samples was determined by the majority vote. A certainty of 70% is enough to recognize (or substitute) the identity of a numeral, below that it is a rejection.

Expert No.	Rec. %	Sub. %	Rej. %
1	36.1	1.7	62.3
2	63.9	4.2	31.8
3	16.9	0	83.1
4	62.5	6.2	31.3
5	58.3	4.2	37.5

Table 2 **Human experts perf. with 70% certainty.**

2.8 Knowledge Elicitation

Eliciting knowledge from a group of experts is a major task since it has the additional difficulty of integrating different opinions. A total of 3,227 comments, by five experts, regarding the features of only 10 characters (numerals 0 to 9) had to be checked for duplications, contradictions and other errors and finally put into a format usable by the inference engine (or classifier) of a multi-expert numeral recognition system.

We started the elicitation process by putting the relevant comments (features) pertaining to each class of numerals under its own category. However, very soon it was realized that, some of the comments (features) that were noted for a variation of a class would not hold for the other variations of the same class. Different variations of the same class of numeral might even have contradictory features. Therefore further classification of classes of numerals into their subclasses were evolved which was a matter of necessity rather than choice.

We continued the elicitation process and thereby we found forty eight subclasses of numerals in thirteen groups of subclasses. Examples of these subsclasses and their groups are shown in Figure 5.

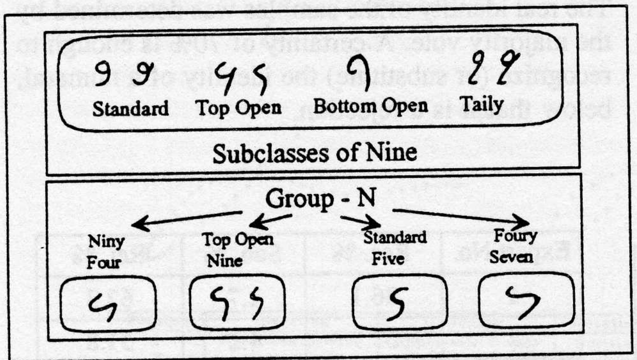


Fig. 5 Examples of subclasses and their groups.

2.9 Knowledge Base

Features (rules) for all subclasses after going through different stages of correction, refinement, dispersion and compilation were put together to form a knowledge base. This knowledge base embodies all the features (rules) of all subclasses which are necessary to verify if a sample is a subclass known to this knowledge base.

The created knowledge base consists of more than 350 rules (features) covering 13 groups of subclasses or 48 individual subclasses.

2.10 Inference Engine

In this multi-expert recognition system, the Group-determinator module and the classifiers for peculiar and confusing cases of numerals together form a background chaining inference engine. The inference engine begins its search after the feature extractor module completes the extraction of all the facts (features) from an input character.

Firing of rules of subclasses scores points for them and a comparison of their total scores determines the identity of an input character. A subclass scores more or less points depending on how many of its rules can be fired. The formula that determines the total score for any subclass is shown below.

$$S_s = \sum_{j \in u_s}^{u_s} e_j l_j m_s$$

Where:

S_s - Score of subclass, $0 \le S_s \le 100$

u_s - Set of fired rules of subclass s

e - Expert weight $0 \le e \le 1$

l - Line weight $0 \le l \le 1$

m_s - Maximum score for each rule of subclass,

$m_s = 100$ / no. of rules

This aggregation method (formula) differs from some of the more usual multi-expert opinion combination mechanism. It differs as it maintains expert hierarchies, does not override minority vote and does not force any kind of consensus (since there has not been any).

2.11 Testing of the Multi-Expert System

Three sets of samples of numerals (sets 'A', 'B' and 'T') were selected for training and testing of this numerals recogntion multi-expert system. Each set contained 2000 samples.

This multi-expert system (before and after training) was tested on the set of 360 confusing cases using equal and unequal expert weights. For comparison purposes, performance of experts, volunteers and multi-expert system are shown together in Table 3.

	Rec.%	Sub.%	Rej.%
Experts	59.2	0	40.8
Volunteers	46.4	6.1	47.5
E.S. bef. Training (Equal Wts.)	35.21	8.45	56.33
E.S. bef Training (Unequal. Wts.)	37.74	6.19	56.07
E.S. after limited training	46.47	11.83	41.69

Table 3 Human and E.S. performance with confusing cases of numerals.

The numbers in Table 3 indicate that human experts performed far better than the multi-expert system implemented. However, performance of the multi-expert system may be compared to that of the volunteer human subjects. The overall superior performance of human subjects can be explained on the basis of the following facts.

- Knowledge base of the multi-expert system is incomplete in terms of scope and quality.
- Absence of feedback by experts makes alteration of rules (further training) impossible.
- Bias in favour of experts' judgement, since identity label of the 360 samples are based on the majority vote of experts' classification.
- Noise in the binarized input samples.
- Relying on a single method for extracting features (facts).

The results of testing the multi-expert system on data set 'T' after training it on data sets 'A' and 'B', both with equal and unequal weights, are shown in Table 4.

	Rec.%	Sub.%	Rej.%
Trained on 'A' (Eql. Wts.)	71.3	1.95	26.75
Trained on 'A' (Uneql. Wts.)	71.8	2	26.2
Trained on 'A' & 'B' (Eql. Wts.)	76.2	1.85	21.95
Trained on 'A' & 'B' (Uneql. Wts.)	76.8	1.95	21.25

Table 4: Performance of E.S. on data set 'T' after training it on data sets 'A' and 'B'

Performance of the multi-expert system is better when experts and lines weights have different values. In all tests, substitution rate of the multi-expert system remained very low and its recognition rate improved considerably by training.

The Test results of this numeral recognition system indicates that the system maintained low substitution rates with different data sets and consistently improved on its recognition rate by further training.

4. Concluding Remarks

This paper has given a brief introduction about pattern recognition and expert systems. Two examples were given to illustrate how these two systems can work together to produce even more powerful systems. However, when we try to combine them, a number of problems have surfaced. Some of them are listed below for discussions as well as for further improvements and future research.

1. Knowledge acquisition: Experts must be given ample time to discover features for the pattern recognition tast and assign appropriate weights to them. Experts must distinguish key features from other features. More samples must be examined by human experts in order to establish a robust set of rules and build a complete recognition system. Experts and knowledge engineer(s) must be able to interact directly or indirectly.

2. Knowledge elicitation: Experts' weights may be determined on the basis of their performances with subclasses instead of classes of numerals. Experts must provide feedback during the knowledge elicitation phase. Expert code of those features which are observed by more than one expert must be indicative of the participant experts.

3. Preprocessing and Feature Extraction: The binarized matrix of an input character must represent the actual character properly. More than one method of feature extraction (fact finding) should be used.

4. Inference engine: Rules in the subclasses of the selected groups can be examined (fired) in parallel. The aggregation formula may be appended to include changes which are suggested in point no. 2 above.

5. Training and Testing: Larger data sets must be used for the training and testing of the system. Training of the system needs modification of the rules as well which can be done more appropriately with feedback from experts.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada, the FCAR program of the Ministry of Education of Quebec, and the National Networks of Centres of Excellence program of Canada. It is also partially supported by Bell Canada through a research contract.

References

[1] Proceedings of the 11th International Conference on Pattern Recognition, The Hague, Holland, August 1992.

[2] C. Y. Suen and R. Shinghal (eds.), Operational Expert System Applications in Canada, Pergamon Press, New York, Dec. 1991.

[3] Proceedings of the 1st International Congress on Expert Systems, Orlando, Dec. 1991.