On Combining Classifiers

Josef Kittler, *Member, IEEE Computer Society*, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas

Abstract—We develop a common theoretical framework for combining classifiers which use distinct pattern representations and show that many existing schemes can be considered as special cases of compound classification where all the pattern representations are used jointly to make a decision. An experimental comparison of various classifier combination schemes demonstrates that the combination rule developed under the most restrictive assumptions—the sum rule—outperforms other classifier combinations schemes. A sensitivity analysis of the various schemes to estimation errors is carried out to show that this finding can be justified theoretically.

Index Terms—Classification, classifier combination, error sensitivity.

1 Introduction

The ultimate goal of designing pattern recognition systems is to achieve the best possible classification performance for the task at hand. This objective traditionally led to the development of different classification schemes for any pattern recognition problem to be solved. The results of an experimental assessment of the different designs would then be the basis for choosing one of the classifiers as a final solution to the problem. It had been observed in such design studies, that although one of the designs would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggested that different classifier designs potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier.

These observations motivated the relatively recent interest in combining classifiers. The idea is not to rely on a single decision making scheme. Instead, all the designs, or their subset, are used for decision making by combining their individual opinions to derive a consensus decision. Various classifier combination schemes have been devised and it has been experimentally demonstrated that some of them consistently outperform a single best classifier. However, there is presently inadequate understanding why some combination schemes are better than others and in what circumstances.

The two main reasons for combining classifiers are efficiency and accuracy. To increase efficiency one can adopt multistage combination rules whereby objects are classified by a simple classifier using a small set of cheap features in combination with a reject option. For the more difficult objects more complex procedures, possibly based on different features, are used (sequential or pipelined [17], [7], or hierarchical [24], [16]). Other studies in the gradual reduction of the set of possible classes are [8], [6], [14], [21]. The combination of ensembles of neural networks (based on different initialisations), has been studied in the neural network literature, e.g., [11], [4], [5], [10], [15], [18].

An important issue in combining classifiers is that this is particularly useful if they are different, see [1]. This can be achieved by using different feature sets [23], [13] as well as by different training sets, randomly selected [12], [22] or based on a cluster analysis [3]. A possible application of a multistage classifier is that it may stabilize the training of classifiers based on a small sample size, e.g., by the use of bootstrapping [27], [19]. Variance reduction is studied in [30], [31] in the context of a multiple discriminant function classifier and in [35] for multiple probabilistic classifiers. Classifier combination strategies may reflect the local competence of individual experts as exemplified in [32] or the training process may aim to encourage some experts to achieve local decision making superiority as in the boosting method of Freund [28] and Shapire [29].

An interesting issue in the research concerning classifier ensembles is the way they are combined. If only labels are available a majority vote [14], [9] is used. Sometimes the use can be made of a label ranking [2], [13]. If continuous outputs like posteriori probabilities are supplied, an average or some other linear combination have been suggested [11], [23], [25], [33]. It depends on the nature of the input classifiers and the feature space whether this can be theoretically justified. An interesting study on these possibilities is given in [10], [26], [34]. If the classifier outputs are interpreted as fuzzy membership values, belief values or evidence, fuzzy rules [4], [5], belief functions and Dempster-Shafer techniques [9], [18], [20], [23] are used. Finally it is possible to train the output classifier separately using the outputs of the input classifiers as new features [15], [22], [36].

From the point of view of their analysis, there are basically two classifier combination scenarios. In the first scenario, all the classifiers use the same representation of the input pattern. A typical example of this category is a set of k-nearest

- *J. Kittler and J. Matas are with the Centre for Vision, Speech, and Signal Processing, School of Electronic Engineering, Information Technology, and Mathematics, University of Surrey, Guildford GU2 5XH, United Kingdom. E-mail: j.kittler@ee.surrey.ac.uk.*
- *M. Hatef is with ERA Technology Ltd., Cleeve Road, Leatherhead KT22 7SA, United Kingdom. E-mail: m.hatef@ee.surrey.ac.uk.*
- *R.P.W. Duin is with the Department of Applied Physics, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands. E-mail: bob@ph.tn.tudelft.nl.*

Manuscript received 17 June 1996; revised 16 Jan. 1998. Recommended for acceptance by J.J. Hull.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 106327.

neighbor classifiers, each using the same measurement vector, but different classifier parameters (number of nearest neighbors k, or distance metrics used for determining the nearest neighbors). Another example is a set of designs based on a neural network classifier of fixed architecture but having distinct sets of weights which have been obtained by means of different training strategies. In this case, each classifier, for a given input pattern, can be considered to produce an estimate of the same a posteriori class probability.

In the second scenario, each classifier uses its own representation of the input pattern. In other words, the measurements extracted from the pattern are unique to each classifier. An important application of combining classifiers in this scenario is the possibility to integrate physically different types of measurements/features. In this case, it is no longer possible to consider the computed a posteriori probabilities to be estimates of the same functional value, as the classification systems operate in different measurement spaces.

In this paper, we focus on classifier combination in the second scenario. We develop a common theoretical framework for classifier combination and show that many existing schemes can be considered as special cases of compound classification where all the representations are used jointly to make a decision. We demonstrate that under different assumptions and using different approximations we can derive the commonly used classifier combination schemes such as the product rule, sum rule, min rule, max rule, median rule, and majority voting. The various classifier combination schemes are then compared experimentally. A surprising outcome of the comparative study is that the combination rule developed under the most restrictive assumptions—the sum rule—outperforms other classifier combinations schemes. To explain this empirical finding, we investigate the sensitivity of various schemes to estimation errors. The sensitivity analysis shows that the sum rule is most resilient to estimation errors.

In summary, the contribution of the paper is twofold. First of all, we provide a theoretical underpinning of many existing classifier combination schemes for fusing the decisions of multiple experts, each employing a distinct pattern representation. Furthermore, our analysis of the sensitivity of these schemes to estimation errors enhances the understanding of their properties. As a byproduct, we also offer a methodological machinery which can be used for developing other classifier combination strategies and for predicting their behavior. However, it cannot be overemphasized that the problem of classifier combination is very complex and that there are many issues begging explanation. These include the effect of individual expert error distributions on the choice of a combination strategy, explicit differentiation between decision ambiguity, competence and confidence, and the relationship between dimensionality reduction and multiple expert fusion, with its implicit dimensionality expansion. Also, many practical decision making schemes are very complex, of sequential kind, with special rules to handle rejects and exceptions and it is currently difficult to envisage how the results of this paper could be made to bear on the design of such schemes. The theoretical framework and analysis presented is only a small step towards a considerably improved understanding of classifier combination which will be needed in order to harness the benefits of multiple expert fusion to their full potential.

The paper is organized as follows. In Section 2, we formulate the classifier combination problem and introduce the necessary notation. In this section, we also derive the basic classifier combination schemes: the product rule and the sum rule. These two basic schemes are then developed into other classifier combination strategies in Section 3. The combination rules derived in Sections 2 and 3 are experimentally compared in Sections 4 and 5. Section 6 investigates the sensitivity of the basic classifier combination rules to estimation errors. Finally, Section 7 summarizes the main results of the paper and offers concluding remarks.

2 Theoretical Framework

Consider a pattern recognition problem where pattern Z is to be assigned to one of the m possible classes $\{\omega_1, \ldots, \omega_m\}$. Let us assume that we have R classifiers each representing the given pattern by a distinct measurement vector. Denote the measurement vector used by the ith classifier by $\mathbf{x}_i$. In the measurement space each class ω_k is modeled by the probability density function $p(\mathbf{x}_i|\omega_k)$ and its a priori probability of occurrence is denoted $P(\omega_k)$. We shall consider the models to be mutually exclusive which means that only one model can be associated with each pattern.

Now, according to the Bayesian theory, given measurements $\boldsymbol{x}_i$, $i = 1, \ldots, R$, the pattern, Z, should be assigned to class ω_j provided the a posteriori probability of that interpretation is maximum, i.e.

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$P(\omega_j|\mathbf{x}_1, \ldots, \mathbf{x}_R) = \max_k P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_R) \tag{1}$$

The Bayesian decision rule (1) states that in order to utilize all the available information correctly to reach a decision, it is essential to compute the probabilities of the various hypotheses by considering all the measurements simultaneously. This is, of course, a correct statement of the classification problem but it may not be a practicable proposition. The computation of the a posteriori probability functions would depend on the knowledge of high-order measurement statistics described in terms of joint probability density functions $p(\mathbf{x}_1, \ldots, \mathbf{x}_R|\omega_k)$ which would be difficult to infer. We shall therefore attempt to simplify the above rule and express it in terms of decision support computations performed by the individual classifiers, each exploiting only the information conveyed by vector $\mathbf{x}_i$. We shall see that this will not only make rule (1) computationally manageable, but also it will lead to combination rules which are commonly used in practice. Moreover, this approach will provide a scope for the development of a range of efficient classifier combination strategies.

We shall commence from rule (1) and consider how it can be expressed under certain assumptions. Let us rewrite the a posteriori probability $P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_R)$ using the Bayes theorem. We have

$$P(\omega_k|\mathbf{x}_1,\ldots,\mathbf{x}_R) = \frac{p(\mathbf{x}_1,\ldots,\mathbf{x}_R|\omega_k)P(\omega_k)}{p(\mathbf{x}_1,\ldots,\mathbf{x}_R)} \quad (2)$$

where $p(\mathbf{x}_1, ..., \mathbf{x}_R)$ is the unconditional measurement joint probability density. The latter can be expressed in terms of the conditional measurement distributions as

$$p(\mathbf{x}_1,\ldots,\mathbf{x}_R) = \sum_{j=1}^{m} p(\mathbf{x}_1,\ldots,\mathbf{x}_R|\omega_j)P(\omega_j) \quad (3)$$

and therefore, in the following, we can concentrate only on the numerator terms of (2).

2.1 Product Rule

As already pointed out, $p(\mathbf{x}_1,\ldots,\mathbf{x}_R|\omega_k)$ represents the joint probability distribution of the measurements extracted by the classifiers. Let us assume that the representations used are conditionally statistically independent. The use of different representations may be a probable cause of such independence in special cases. We will investigate the consequences of this assumption and write

$$p(\mathbf{x}_1,\ldots,\mathbf{x}_R|\omega_k) = \prod_{i=1}^{R} p(\mathbf{x}_i|\omega_k) \quad (4)$$

where $p(\mathbf{x}_i \mid \omega_k)$ is the measurement process model of the ith representation. Substituting from (4) and (3) into (2) we find

$$P(\omega_k|\mathbf{x}_1,\ldots,\mathbf{x}_R) = \frac{P(\omega_k)\prod_{i=1}^{R} p(\mathbf{x}_i|\omega_k)}{\sum_j^m P(\omega_j)\prod_{i=1}^{R} p(\mathbf{x}_i|\omega_j)} \quad (5)$$

and using (5) in (1), we obtain the decision rule

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$P(\omega_j)\prod_{i=1}^{R} p(\mathbf{x}_i|\omega_j) = \max_{k=1}^{m} P(\omega_k)\prod_{i=1}^{R} p(\mathbf{x}_i|\omega_k) \quad (6)$$

or in terms of the a posteriori probabilities yielded by the respective classifiers

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$P^{-(R-1)}(\omega_j)\prod_{i=1}^{R} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m} P^{-(R-1)}(\omega_k)\prod_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \quad (7)$$

The decision rule (7) quantifies the likelihood of a hypothesis by combining the a posteriori probabilities generated by the individual classifiers by means of a product rule. It is effectively a severe rule of fusing the classifier outputs as it is sufficient for a single recognition engine to inhibit a particular interpretation by outputting a close to zero probability for it. As we shall see below, this has a rather undesirable implication on the decision rule combination as all the classifiers, in the worst case, will have to provide their respective opinions for a hypothesized class identity to be accepted or rejected.

2.2 Sum Rule

Let us consider decision rule (7) in more detail. In some applications it may be appropriate further to assume that the a posteriori probabilities computed by the respective classifiers will not deviate dramatically from the prior probabilities. This is a rather strong assumption but it may be readily satisfied when the available observational discriminatory information is highly ambiguous due to high levels of noise. In such a situation we can assume that the a posteriori probabilities can be expressed as

$$P(\omega_k|\mathbf{x}_i) = P(\omega_k)(1+\delta_{ki}) \quad (8)$$

where δ_{ki} satisfies $\delta_{ki} << 1$. Substituting (8) for the a posteriori probabilities in (7), we find

$$P^{-(R-1)}(\omega_k)\prod_{i=1}^{R} P(\omega_k|\mathbf{x}_i) = P(\omega_k)\prod_{i=1}^{R}(1+\delta_{ki}) \quad (9)$$

If we expand the product and neglect any terms of second and higher order, we can approximate the right-hand side of (9) as

$$P(\omega_k)\prod_{i=1}^{R}(1+\delta_{ki}) = P(\omega_k) + P(\omega_k)\sum_{i=1}^{R}\delta_{ki} \quad (10)$$

Substituting (10) and (8) into (7), we obtain a sum decision rule

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$(1-R)P(\omega_j) + \sum_{i=1}^{R} P(\omega_j|\mathbf{x}_i) =$$
$$\max_{k=1}^{m}\left[(1-R)P(\omega_k) + \sum_{i=1}^{R} P(\omega_k|\mathbf{x}_i)\right] \quad (11)$$

2.3 Comments

Before proceeding, in the next section, to develop specific classifier combination strategies based on decision rules (7) and (11), let us pause to elaborate on the assumptions made to derive the product and sum rules. We concede that the conditional independence assumption may be deemed to be unrealistic in many situations. However, three important points should be borne in mind before dismissing the results of the rest of the paper:

- For some applications, the conditional independence assumption will hold.
- For many applications, this assumption will provide an adequate and workable approximation of the reality which may be more complex. One could draw a parallel here between the Gaussian assumption frequently made even in situations where the class distributions patently do not obey the exponential law but still this simplification yields acceptable results.
- Finally, and perhaps most importantly, we shall see in the next section that all the derived classifier combination schemes based on this assumption are routinely used in practice. The analysis presented in the paper therefore provides a plausible theoretical underpinning of these combination rules and thereby draws attention to the underlying assumptions behind these schemes which the users may not be aware of.

As far as the sum rule is concerned, the assumption that the posterior class probabilities do not deviate greatly from the priors will be unrealistic in most applications. When

observations $\mathbf{x}_i$, $i = 1, ..., R$ on a pattern convey significant discriminatory information the sum approximation of the product in (10) will introduce gross approximation errors. However, we shall show in Section 6 that the injection of these errors will be compensated by a relatively low sensitivity of the approximation to estimation errors.

3 Classifier Combination Strategies

The decision rules (7) and (11) constitute the basic schemes for classifier combination. Interestingly, many commonly used classifier combination strategies can be developed from these rules by noting that

$$\prod_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \leq \min_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \leq \frac{1}{R}\sum_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \leq \max_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \quad (12)$$

The relationship (12) suggests that the product and sum combination rules can be approximated by the above upper or lower bounds, as appropriate. Furthermore, the hardening of the a posteriori probabilities $P(\omega_k|\mathbf{x}_i)$ to produce binary valued functions Δ_{ki} as

$$\Delta_{ki} = \begin{cases} 1 & \text{if } P(\omega_k|\mathbf{x}_i) = \max_{j=1}^{m} P(\omega_j|\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

results in combining decision outcomes rather than combining a posteriori probabilities. These approximations lead to the following rules:

3.1 Max Rule

Starting from (11) and approximating the sum by the maximum of the posterior probabilities, we obtain

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$(1-R)P(\omega_j) + R\max_{i=1}^{R} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m}\left[(1-R)P(\omega_k) + R\max_{i=1}^{R} P(\omega_k|\mathbf{x}_i)\right] \quad (14)$$

which under the assumption of equal priors simplifies to

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$\max_{i=1}^{R} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m}\max_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \quad (15)$$

3.2 Min Rule

Starting from (7) and bounding the product of posterior probabilities from above we obtain

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$P^{-(R-1)}(\omega_j)\min_{i=1}^{R} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m} P^{-(R-1)}(\omega_k)\min_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \quad (16)$$

which under the assumption of equal priors simplifies to

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$\min_{i=1}^{R} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m}\min_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \quad (17)$$

3.3 Median Rule

Note that under the equal prior assumption, the sum rule in (11) can be viewed to be computing the average a posteriori probability for each class over all the classifier outputs, i.e.,

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$\frac{1}{R}\sum_{i=1}^{R} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m}\frac{1}{R}\sum_{i=1}^{R} P(\omega_k|\mathbf{x}_i) \quad (18)$$

Thus, the rule assigns a pattern to that class the average a posteriori probability of which is maximum. If any of the classifiers outputs an a posteriori probability for some class which is an outlier, it will affect the average and this in turn could lead to an incorrect decision. It is well known that a robust estimate of the mean is the median. It could therefore be more appropriate to base the combined decision on the median of the a posteriori probabilities. This then leads to the following rule:

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$\underset{i=1}{\overset{R}{med}}\, P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m}\underset{i=1}{\overset{R}{med}}\, P(\omega_k|\mathbf{x}_i) \quad (19)$$

3.4 Majority Vote Rule

Starting from (11) under the assumption of equal priors and by hardening the probabilities according to (13), we find

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$
$$\sum_{i=1}^{R}\Delta_{ji} = \max_{k=1}^{m}\sum_{i=1}^{R}\Delta_{ki} \quad (20)$$

Note that for each class ω_k the sum on the right hand side of (20) simply counts the votes received for this hypothesis from the individual classifiers. The class which receives the largest number of votes is then selected as the consensus (majority) decision.

All the above combination schemes and their relationships are represented in Fig. 5.

4 Experimental Comparison of Classifier Combination Rules: Identity Verification

The first experiment is concerned with the problem of personal identity verification. Three different sensing modalities of biometric information are used to check the claimed identity of an individual: frontal face, face profile, and voice. The verification methods using these biometric sensing modalities have been developed as part of the European Union project in Advance Communication Technologies and Services M2VTS as described in [41], [44], [43]. The design of the verification modules and their performance testing has been carried out using the M2VTS database [42] made up of about eight seconds of speech and video data for 37 clients taken five times (five shots) over a period of one month. The image resolution is 286×350 pixels.

4.1 Frontal Face

The face verification system used in the experiments is described in detail in [41]. It is based on robust correlation of a frontal face image of the client and the stored face template corresponding to the claimed identity. A search for the optimum correlation is performed in the space of all valid geometric and photometric transformations of the input image to obtain the best possible match with respect to the template. The geometric transformation includes translation, rotation and scaling, whereas the photometric transformation corrects for a change of the mean level of illumination. The search technique for the optimal transformation parameters is based on random exponential perturbations. Accordingly, at each stage the transformation between the test and reference images is perturbed by a random vector drawn from an exponential distribution and the change is accepted if it leads to an improvement of a matching criterion. Computational efficiency is achieved by means of random sampling based on Sobel sequences which allow faster convergence as compared to uniform sampling.

The score function adopted rewards a large overlap between the transformed face image and the template, and the similarity of the intensity distributions of the two images. The degree of similarity is measured with a robust kernel. This ensures that gross errors due to, for instance, hair style changes do not swamp the cumulative error between the matched images. In other words, the matching is benevolent, aiming to find as large areas of the face as possible supporting a close agreement between the respective gray-level profiles of the two images. The gross errors will be reflected in a reduced overlap between the two frames which is taken into account in the overall matching criterion. The system is trained very easily by means of storing one or more templates for each client. Each reference image is segmented to create a face mask which excludes the background and the torso as these are likely to change over time. The testing is performed on an independent test data composed of 37 clients and 37×36 impostors.

4.2 Face Profile

The verification approach involves a comparison of a candidate profile with the template profile of the claimed identity. The candidate image profile is extracted from the face profile images by means of color-based segmentation. The similarity of the two profiles is measured using the Chamfer distance computed sequentially [44]. The efficiency of the verification process is aided by precomputing a distance map for each reference profile. The map stores the distance of each pixel in the face profile image to the nearest point on the profile. As the candidate profile can be subject to translation, rotation and scaling, the objective of the matching stage is to compensate for such geometric transformation. The parameters of the compensating transformation are determined by minimizing the chamfer distance between the template and the transformed candidate profile. The optimization is carried out using a simplex algorithm which requires only the distance function evaluation and no derivatives. The convergence of the simplex algorithm to a local minimum is prevented by a careful initialization of the transformation parameters. The translation parameters are estimated by comparing the position of the nose tip in the two matched profile. The scale factor is derived from the comparison of the profile heights and the rotation is initially set to zero. Once the optimal set of transformation parameters is determined, the user is accepted or rejected depending on the relationship of the minimal chamfer distance to a prespecified threshold.

The system is trained on the first three shots. One profile per client per shot is stored in the training set. From the three profiles for each client a single reference profile is selected by pairwise comparison of the profile images. The profile yielding the lowest matching distance to the other two images is considered as the best representative of the triplet. The trained system is tested on Shot 4 profiles. As there are 37 users in the M2VTS database the testing involves 37 correct authentication matches and 37×36 imposter tests. The acceptance threshold is selected from the Receiver Operating Characteristic so as to produce equal error rate (false rejection and false acceptance).

4.3 Voice

The personal identity verification based on voice employs a text dependent approach described in [43]. It is assumed that the audio input signal representing the uttered sequence of digits from zero to nine can be segmented into individual words. Both, the segmentation of the speech data and the claimed identity verification is accomplished using speech and speaker recognition methods based on Hidden Markov Models. The audio signal is first transformed into a multivariate time series of linear predictive cepstral coefficients. During training, digit HMMs are trained using segmented speech data from three shots of the M2VTS database. The digit models have the same structure, with the number of states being digit specific. The models allocate one state per phoneme and one state per transition between phonemes. A single Gaussian mixture is used to model the distribution of the cepstral coefficient vectors within one state.

Two models are acquired for each digit: the client model, and the world model. The latter, which is common to all users, captures the variability of the uttered sound in a large database. The verification of a claimed identity is based on a score computed as the sum over the individual digits of the log likelihood ratio of the claimed model and the world model normalized by the number of cepstral coefficient frames. The score is mapped on the interval zero-one using a sigmoid function. The performance is assessed using an independent test set.

4.4 Experimental Results

The equal error rates obtained using the individual sensing modalities are shown in Table 1. The table shows that the lowest rate of 1.4 percent was achieved using voice based verification. The face profile verification produced an equal error rate of 8.5 percent whereas the frontal face method yielded 12.2 percent. The soft decisions output by the three verification systems were then combined using the various classifier combination strategies discussed in Section 3.

The validity of the conditional independence assumption was tested by computing the average within class corre-

TABLE 1
EQUAL ERROR RATES

method	EER (%)
frontal	12.2
profile	8.5
speech	1.4
sum	0.7
product	1.4
maximum	12.2
median	1.2
minimum	4.5

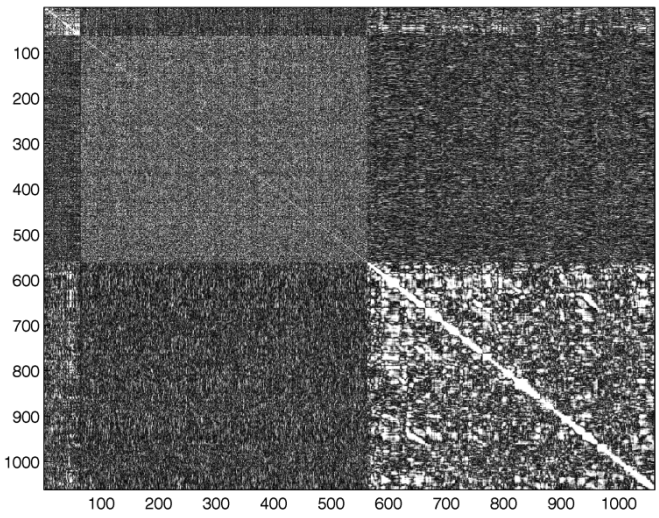


Fig. 1. Correlation of face profile, frontal face, and speech data.

lation matrix for the data used in decision making. Since the overall dimensionality of the data exceeds tens of thousands, it is impossible to present a full view of the correlations between the measurements of the respective modalities. However, by adopting a visual representation of the correlation matrix, we will be able to look at the correlations at least in a representative subspace of this highly dimensional feature space. This subspace was created by taking as features 500 samples of the face image gray levels taken at prespecified spatial locations. Each profile image was represented by 60 sample points evenly distributed along the profile. The sampling is registered with respect to the tip of the nose and the sampling interval normalized by the nose length. The profile landmarks needed for the registration and normalization can be easily detected. For the speech data, we took the first 100 frames from each of the first five cepstral coefficients. The utterances for each client were first time warped using a client specific template. This created a client representation subspace of 1,060 dimensions. In particular, the face profile variables occupy the first 60 dimensions, followed by 500 frontal face image samples, and finally 5×100 speech measurements. The average within class correlation matrix was computed by removing the class conditional mean of each variable. The resulting vectors of deviations from the means were used to compute the elements of the average within class covariance matrix. These were then normalized by dividing each ijth element by the product of standard deviations of the ith and jth component of the vector of deviations. This normalisation process produced average within class correlations taking values in the interval [–1, 1]. For display purposes, we have taken the absolute value of these correlation coefficients. The result of this representation of variable correlations is a matrix with all elements on the diagonal equal to unity (displayed as gray level 255) and the strength of correlation between one and zero mapped onto the gray-level scale 255 to 0. The correlation matrix is shown in Fig. 1.

The correlation matrix exhibits a block diagonal structure, which suggests that the observations generated by each modality are class conditionally dependent. The correlation are particularly strong between the features of the face profiles and similarly between those of the speech utterances. They are weaker for the features of the face image. Owing to the random spatial sampling of the face image, the spatial ordering of the successive features is destroyed and consequently the correlation matrix block corresponding to the facial data has a random structure (with the exception of the diagonal elements). Note that the correlations between features from different modalities are considerably weaker than within modality correlations. This applies in particular to the correlations between the frontal face and the other two modalities. There is a small subset of the face profile variables for which the correlations are not insignificant but on the whole the conditional independence assumption may be considered to hold.

Next, the three biometric modalities were combined using the fusion strategies discussed in Section 3. The results presented in Table 1 show the benefits of classifier combination. It is interesting to note that the sum rule outperformed all the other combination strategies and also the individually best expert.

5 EXPERIMENTAL COMPARISON OF CLASSIFIER COMBINATION RULES: HANDWRITTEN DIGIT RECOGNITION

As a second domain to assess the classifier combination strategies, we used the problem of handwritten character recognition. The task is to recognize totally unconstrained handwritten numerals. Samples are images of isolated numeric characters taken from addresses on the letter envelopes provides by the U.S Postal Service.

The database used is the CEDAR-CDROM produced by the Center of Excellence for Document Analysis and Recognition, at the State University of New York, Buffalo. Images are scanned from dead-letter envelopes provided by the U.S. Postal Service. We used the BR and BS sets of the database that consist of bitonal isolated images of numeric characters. BR set contains 18,468 samples and is used as a training set while BS set (2,213 samples) served as a test set.

Four types of classifiers are first applied to perform the classification individually. We used structural [38], Gaussian, Neural Network, and Hidden Markov Model classifiers [40].

5.1 Character Representation

Four different representations are used as follows:

1) Pixel-level representation: in the Gaussian classifier case the bitonal image of each numeric character is scaled into 10×10 gray-level image. The character is thus represented by a 100-dimensional vector in which each dimension is the gray level of the corresponding pixel.
2) Complex object representation: this is used in the case of the structural classifier. The bitonal image is first skeletonized using some thinning process. The skeleton of the character is then decomposed into a number of primitives. Each primitive being either a line, curve or a loop is parameterized using a number of unary measurements such as the size, direction, etc. In addition, a number of binary measurements are extracted to describe the geometrical relations between each primitive and its neighbors. A more detailed description of this representation is presented in [39].
3) In the HMM classifier, the 2D image is represented as two 1D signals by using the horizontal and vertical profiles. The profile consists of the binary pattern of the image across the horizontal/vertical line. This pattern is quantized into 16 vectors in the codebook. Each pattern is therefore given the index of the closest vector in the codebook. Further, the center of gravity of each line in the profile is calculated and also quantized (to 13 levels). The feature space thus consists of two indices one for the pixel pattern and the other for the center of gravity. More details on this representation can be found in [40].
4) The pixel representation in Item 1 is used as a starting point to derive a distinct character description by the hidden layer of the neural network employed as one of the classifiers.

5.2 Classification

5.2.1 Structural Classifier

Handwritten characters have natural structures as they are generally composed of number of smaller elements with certain topological relations. To recognize a character, we need to identify its basic primitives and the particular structural relations between them.

The binary image is first skeletonized, then decomposed into number of primitives where junctions and reflection points serve as breaking points. Both symbolic and numeric attributes are used to describe the structure of the character. Firstly, primitives are categorized into one of three types using a discretizing criterion: zero-line, one-curve, or two-loop. The connectivity between the primitives is encoded to reflect the topological structure of the character. The character code consists of the code of each primitive which in turn consists of the type of the primitive, the number of the neighbors on the first end point and their types, and the number of the neighbors on the second endpoint and their types. For example the code:

$$(1, 200, 0), (2, 10, 10), (0, 210, 12), (0, 210, 0)$$

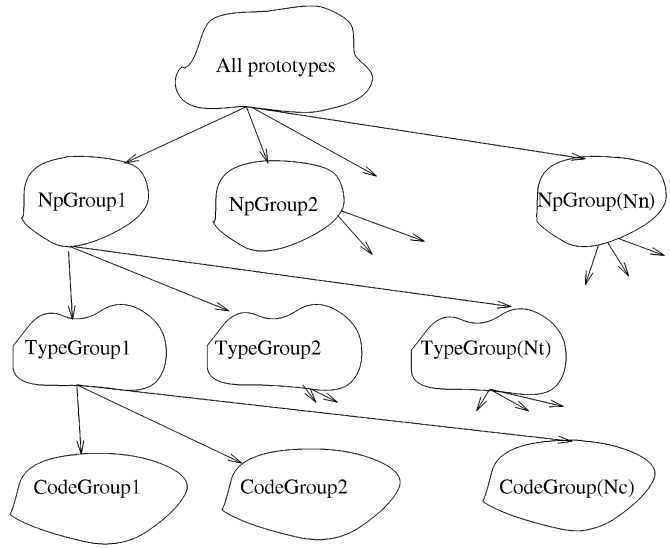


Fig. 2. The prototype tree structure.

represents a character consisting of four primitives. The first primitive is a curve (1) connected to two primitives in the first end point, both of them being lines (200). The other endpoint is not connected to any primitive (0).

Numeric information is also used to characterize unary attributes of primitives and relations and binary relations between primitives. The length of the primitive, its direction, the degree of curvature are some of the unary measurements used. Example of the binary measurements used are the direction of the line connecting the centers of the primitive and its neighbor as well as the direction of the line connecting the terminal point of the primitive and the center of the neighbor. Each class is represented by one or more prototypes. In our scheme prototypes are generated from the training samples. Samples of each class are divided into small groups by means of levels of clustering. The first is to group all samples with the same number of primitives in a cluster. Each cluster is called Np-Group and is further divided according to the types of the primitives. For example, samples that consist of a curve and two lines are grouped together. Each such group or cluster is called type-Group and further divided into a number of clusters each containing samples that have the same structural code. Cluster in this level is called code-Group. Finally, each code-Group is further divided using the dynamic clustering algorithm [37] where each of the clusters produced is called dist-Group. The mean, variance and the actual range around the mean are calculated for each of the unary and binary measurements to characterize the particular cluster. The prototypes of all classes are saved in this multilevel tree structure (Fig. 2).

5.2.2 The Classification Scheme

An unknown sample is tested first at the root of the prototype tree to decide the right Np-group. In the next level, it is checked to select the right type-Group and eventually it reaches the appropriate code-Group. If no code-Group is found, the class of the sample is reported as unknown. Otherwise, the sample will be checked against all prototypes in

the code-Group to find the closest candidate(s) to the sample. First, the probabilistic relaxation algorithm [38] is used to find the correspondence between the primitives in the sample and those in the prototype. Then, a distance measure is used to quantify the similarity between the sample and each candidate. It is pertinent to point out that a meaningful measure can be defined because each sample is compared only to prototypes that have the same number of primitives as well as connectivity (adjacency matrix). This means that they have the same number of dimensions. Moreover, after finding the correspondence between the primitives in the sample and the prototype through the matching process, the attribute vectors associated with the sample and prototypes respectively can be considered as belonging to the same space. This facilitates the task of finding a metric for measuring the distance between them. We used the Euclidean distance first, but due to the fact that this distance does not take into account second-order statistics of each measurement, the results were not satisfactory. On the other hand, using distance measure that exploits second order statistics, such as the Mahalanobis distance, requires a large number of samples in each cluster. Due to the large variability in the structure of the handwritten characters there are prototypes that contain only a few samples in the training set which makes the estimate of these statistics unsatisfactory. Consequently, we chose a modified Euclidean distance whereby the difference in each dimension between the measurement vector in the sample and that in prototype is penalized if it exceeds a certain value. The value is chosen to be some weight multiplied by the standard deviation of that particular measurement. The modified distance therefore is:

$$d_{me} = \left(\sum \left\{ F\left(\left| x_i - m_i \right| \right) \right\}^2 \right)^{\frac{1}{2}} \tag{21}$$

where:

$$F(y) = \begin{cases} \kappa |y| & \text{if } |y| > \Theta \sigma_i \\ |y| & \text{otherwise} \end{cases} \tag{22}$$

and m_i is the mean of the ith feature while σ_i is its standard deviation. Θ is the threshold constant. κ is a penalizing weight. The values of Θ and κ are selected experimentally.

An estimate of the a priori probability is then computed as follows:

$$P(\omega_i|\mathbf{x}) = \frac{e^{-dme_i^2} P(\omega_i)}{\sum_k e^{-dme_k^2} P(\omega_k)} \tag{23}$$

where $P(\omega_i)$ is the a posteriori probability estimated from the number of samples in each cluster that generated the prototype. The sample is then assigned the class that has the maximum $P(\omega_k|\mathbf{x})$. Note that when no prototype matches the sample structure it is assigned zero a posteriori probability for all classes.

5.2.3 Gaussian Classifier

The classes in the feature space are assumed to possess a normal distribution:

$$p(\mathbf{x}|\omega_i) = (2\pi)^{-\frac{d}{2}} \left| \sum_i \right|^{-\frac{1}{2}} e^{-(\mathbf{x}-\mathbf{m_i})^T \Sigma_\mathbf{i}^{-1} (\mathbf{x}-\mathbf{m_i})} \tag{24}$$

where $\mathbf{m_i}$ is the mean vector and $\mathbf{\Sigma_i}$ is the covariance matrix of class i. They are estimated in the training phase from the training data set. d is the number of dimensions in the feature space.

The a posteriori probability is then calculated:

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i) P(\omega_i)}{\sum_k p(\mathbf{x}|\omega_k) P(\omega_k)} \tag{25}$$

5.2.4 Hidden Markov Models Classifier

Hidden Markov Models (HMMs), a popular method of statistical representation in speech processing is based on the representation of an object as a random process that generates a sequence of states. The model consists of a number of states with their probabilities as well as probabilities associated with the transition from one state to another.

The character in this classifier is scanned vertically and horizontally to generate the corresponding vertical and horizontal profiles. The vertical profile consists of the rows in the binary image while the horizontal profile consists of the columns. Each state represents a pattern of binary pixels in each line along the profile. The number of possible patterns (states) can be numerous. For example, in a 32×32 binary image there are 2^{32} possible combinations. To reduce the number of possible states, the training patterns are clustered and the centroid of each cluster serves as a reference vector in a code book (Vector Quantization). An unknown sample is compared to each reference in the codebook and assigned the index of the closest one. The codebook is generated using the k-means clustering algorithm with k = 16, resulting in a 16-vector codebook. In the clustering process some distance measure is required to determine how close a sample is to its cluster in order to decide that it should be kept in the cluster or moved to another (closer) one. Hamming distance is a natural choice when dealing with binary vectors. The Hamming distance, however, is known to be sensitive to the shift between two binary patterns. Slight shifts are inevitable in a problem like character recognition. A Shift Invariant Hamming distance (the minimum Hamming Distance between two discrete vectors when they are allowed to slide on each other) is used. The same advantageous property of shift invariance can be undesirable in some cases. For example, the profile of letter "q" and "d" would appear to have the same codebook index. Therefore, another measure is used to distinguish between such instances. The center of gravity of line is calculated and then subtracted from the running average of the last three lines. The relative center of gravity is in turn quantized to 13 levels. The state representation is thus reduced to a pair of numbers—one represents the pixel pattern index and the other is the relative center of gravity.

The discrete hidden Markov models are generated using the Baum-Welch reestimation procedure while a scoring mechanism based on the Viterbi algorithm is used in the test phase. The scoring result reflects the likelihood of the sample to be generated by the class model. These score values are used as the soft-level assignment of the classifier (as posteriori probabilities estimates).

TABLE 2
THE CLASSIFICATION RATE FOR EACH CLASSIFIER

Individual classifier	Classification rate %
Structural:	90.85
Gaussian:	93.93
Neural Net:	93.2
HMM:	94.77

TABLE 3
THE CLASSIFICATION RATE USING DIFFERENT COMBINING SCHEMES

Combining rule	Classification rate %
Majority Vote:	97.96
Sum rule:	98.05
Max rule:	93.93
Min rule:	86.00
Product rule:	84.69
Median rule:	98.19

5.2.5 Neural Network Classifier

Our next classifier is a feed forward neural network (Multilayer Perceptron) trained as a pattern classifier. The momentum Back-propagation algorithm is used to train the network. The network consists of 100 nodes in the input layer (corresponding to the 100 dimensions in the feature space), 25 nodes in the hidden layer, and 10 nodes in the output layer. Each node in the output layer is associated with one class and its output O_i, with [zero to one] range, reflects the response of the network to the corresponding class ω_i. To facilitate a soft-level combination the responses are normalized and used as estimates of the a posteriori probability of the classes as

$$P(\omega_i|\mathbf{x}) = \frac{O_i}{\sum_k O_k} \tag{26}$$

5.3 The Combination Scheme

In this expert fusion experiment, due to staff changes, we were unable to compute the within class correlation matrix for the different representations used. We can, therefore, only hope that the distinct representations used by the individual experts satisfy the assumption of class conditional independence at least approximately. Six different combination schemes are applied under the assumption of equal priors and their results are compared. These schemes can be divided into two groups according to the format of the individual classifiers used by the combiner. Hard-level combination uses the output of the classifier after it is hard-thresholded (binarized). Soft-level combination on the other hand uses the estimates of a posteriori probability of the class by each classifier. The majority vote combiner is a representative of the first category while the five different operators are the soft-level combiners. Table 2 shows the results of classification of the individual classifiers while the results of different combining schemes are shown in Table 3.

Note that the worst results are achieved when using the *product* rule which are similar to the performance of the *min* rule. The results using these two rules are worse than any of the individual classifiers as well, and the reason is that if any of the classifiers reports the correct class a posteriori probability as zero, the output will be zero, and the correct class cannot be identified. Therefore, the final result reported by the combiner in such cases is either a wrong class (worst case) or a reject (when all of the classes are assigned zero a posteriori probability). Another interesting outcome of our experiments is that the *Sum* rule as well as the *median* rule have the best classification results. The *majority vote* rule is very close in performance to the mean and median rules. The *Max* rule is still better than any of the individual classifiers, with the exception of the HMM classifier.

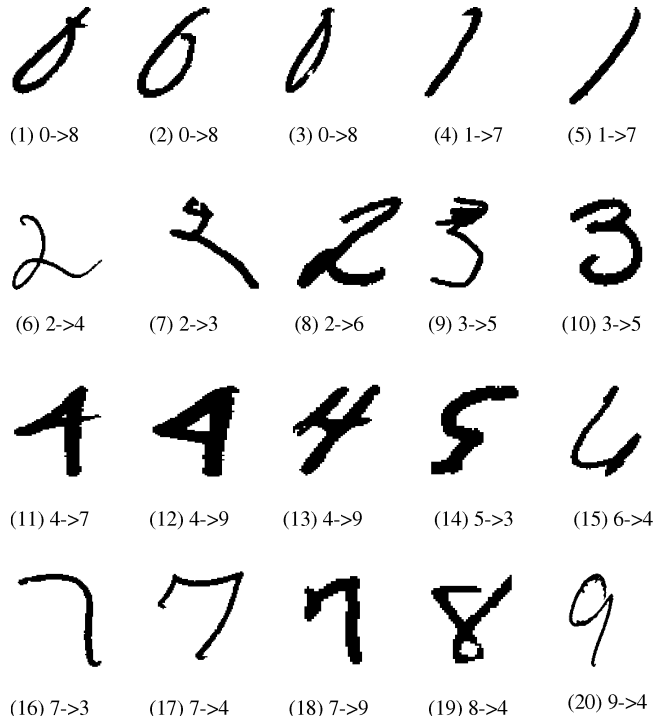


Fig. 3. Samples misclassified by the HMM classifier and corrected by the sum-rule combiner.

5.4 Analysis of the Results

We analysed the results in more detail to see how the performance of the system improves through decision combination. HMM classifier that yields the best classification rate among individual classifiers is chosen as a reference.

Twenty examples of the samples misclassified by the HMM classifier and corrected by the sum-rule combiner are shown in Fig. 3. The numbers below each character represent the true class and that assigned by the HMM classifier, respectively. Although the HMM classifier scored quite well in the overall classification, it seemed to have failed to classify samples that otherwise look easy to recognize.

Table 4 contains the corresponding samples with the a posteriori probabilities estimated by each classifier. The table shows a clear difference in the values assigned to some of the samples by different classifiers. While one of the classifiers is 100 percent sure about the classification of the sample (the probability estimate is 1.0), the HMM classifier is 100 percent sure that it is not the true class (its estimate is zero). Note that 66 of the 107 of the misclassified samples are corrected by the simple sum rule combiner.

An important requirement for a combiner that uses the output of the individual classifiers is that the classifiers

TABLE 4
SAMPLES MISCLASSIFIED BY HMM CLASSIFIER

	True class	HMM decision	Structural	Neural Net.	Gaussian	HMM
1	0	8	0.95	0.99	1.00	0.00
2	0	8	0.71	0.05	1.00	0.00
3	0	8	1.00	0.56	0.04	0.00
4	1	7	0.17	1.00	1.00	0.00
5	1	7	0.58	1.00	1.00	0.12
6	2	4	0.1	0.99	1.00	0.00
7	2	6	1.00	0.99	1.00	0.25
8	2	3	0.96	0.00	1.00	0.00
9	3	5	0.73	0.95	1.00	0.00
10	3	5	0.1	1.00	1.00	0.00
11	4	7	1.00	0.91	1.00	0.39
12	4	9	0.1	1.00	1.00	0.00
13	4	9	1.00	1.00	1.00	0.15
14	5	3	0.71	0.86	0.99	0.00
15	6	4	0.97	1.00	1.00	0.00
16	7	3	1.00	0.92	0.00	0.00
17	7	4	0.1	0.97	1.00	0.00
18	7	9	0.70	0.74	0.97	0.00
19	8	4	0.75	0.59	1.00	0.17
20	9	4	0.98	0.98	1.00	0.00

True class, class assigned by the HMM classifier and the a posteriori probabilities estimated by each classifier.

should not be strongly correlated in their "misclassification." That is, classifiers should not agree with each other when they misclassify a sample, or at least they should not assign the same incorrect class to a sample. This requirement can be satisfied to a certain extent by

1) using different representations for the object (different feature sets) and
2) using a different classification principle for each of the individual classifiers.

Using different representations (feature sets) leads, in many cases, to a reduction in the correlation between the outputs of individual classifiers, since there is almost always less correlation between the input vectors using different representations than when using the same set of features. Different classifiers usually use different assumptions about the structure of the data and the stochastic model that generates it. This leads to a different estimate of the a posteriori probabilities especially around the Bayes decision boundaries.

It is also pertinent to look at the samples that are misclassified by the combiner to see whether there was full correlation between all the classifiers in their decision. Thirty samples out of the 43 misclassified samples are correctly classified by at least one classifier. Fig. 4 displays some of the misclassified samples by the sum-rule combiner. In Fig. 4a, the samples are not recognized by any of the individual classifiers. In Figs. 4b, 4c, 4d, and 4e, samples are correctly classified by the classifier indicated below each sample.

6 ERROR SENSITIVITY

A somewhat surprising outcome of the experimental comparison of the classifier combination rules reported in Sections 4 and 5 is that the sum rule (11), which has been developed under the strongest assumptions, namely, those of

- conditional independence of the respective representations used by the individual classifiers and
- classes being highly ambiguous (observations enhance the a priori class probabilities only slightly)

appear to produce the most reliable decisions. In this section, we shall investigate the sensitivity of the product rule (7) and the sum rule (11) to estimation errors. We shall show that the sum rule is much less affected by estimation errors. This theoretically established behavior is consistent with the experimental findings.

In the developments in Sections 2 and 3, we assumed that the a posteriori class probabilities $P(\omega_j|\mathbf{x}_i)$, in terms of which the various classifier combination rules are defined, are computed correctly. In fact, each classifier i will produce only an estimate of this probability, which we shall denote $\hat{P}(\omega_j|\mathbf{x}_i)$. The estimate deviates from the true probability by error e_{ji}, i.e.,

$$\hat{P}(\omega_j|\mathbf{x}_i) = P(\omega_j|\mathbf{x}_i) + e_{ji} \tag{27}$$

It is these estimated probabilities that enter the classifier combination rules rather than the true probabilities.

Let us now consider the effect of the estimation errors on the classifier combination rules. Substituting (27) into (7) we have

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$

$$P^{-(R-1)}(\omega_j)\prod_{i=1}^{R}\left[P(\omega_j|\mathbf{x}_i) + e_{ji}\right] =$$

$$\max_{k=1}^{m} P^{-(R-1)}(\omega_k)\prod_{i=1}^{R}\left[P(\omega_k|\mathbf{x}_i) + e_{ki}\right] \tag{28}$$

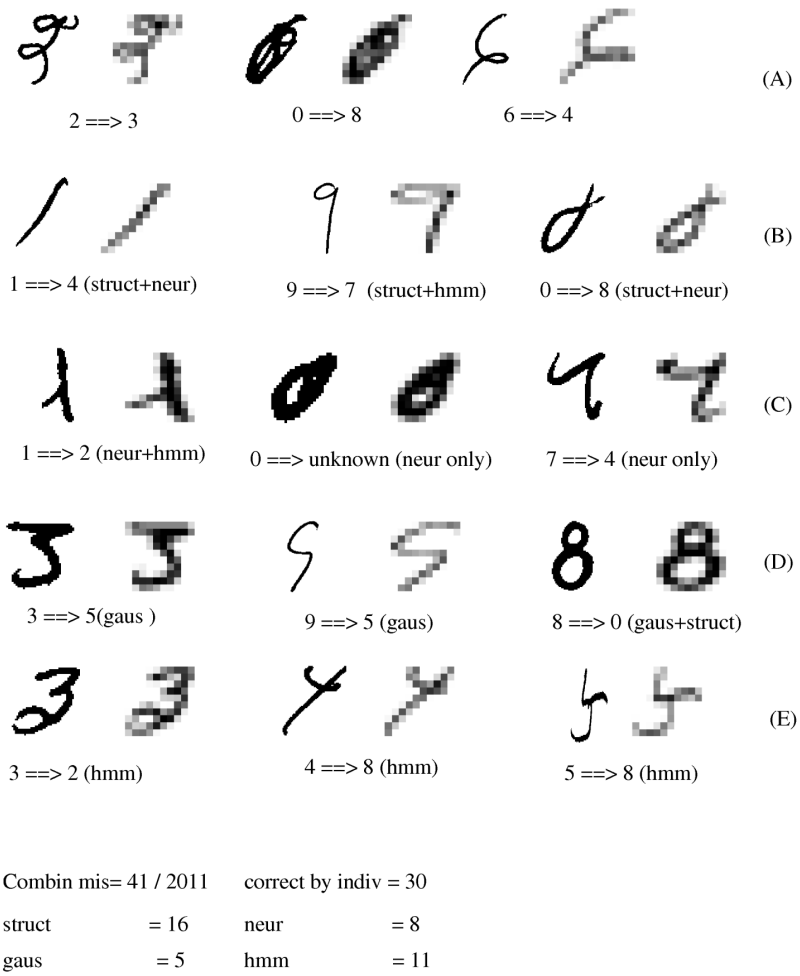


Fig. 4. Samples misclassified by the sum-rule combiner. (a) Samples not classified correctly by any individual classifier. (b) Samples classified correctly by the structural classifier. (c) By the Neural Network classifier. (d) By the Gaussian classifier. (e) By the HMM classifier.

Under the assumption that $e_{ki} << P(\omega_k|\mathbf{x}_i)$ which is rather strong and may not represent the worst case scenario, and further assuming that $P(\omega_k|\mathbf{x}_i) \neq 0$ we can rearrange the product term as

$$\prod_{i=1}^{R}\left[P(\omega_k|\mathbf{x}_i)+e_{ki}\right]=\left[\prod_{i=1}^{R}P(\omega_k|\mathbf{x}_i)\right]\prod_{i=1}^{R}\left[1+\frac{e_{ki}}{P(\omega_k|\mathbf{x}_i)}\right] \quad (29)$$

which can then be linearized as

$$\prod_{i=1}^{R}\left[P(\omega_k|\mathbf{x}_i)+e_{ki}\right]=\left[\prod_{i=1}^{R}P(\omega_k|\mathbf{x}_i)\right]\left[1+\sum_{i=1}^{R}\frac{e_{ki}}{P(\omega_k|\mathbf{x}_i)}\right] \quad (30)$$

Substituting (30) into (28) we get

$$\textit{assign} \quad Z\rightarrow\omega_j \quad \textit{if}$$

$$\left[P^{-(R-1)}(\omega_j)\prod_{i=1}^{R}P(\omega_j|\mathbf{x}_i)\right]\left[1+\sum_{i=1}^{R}\frac{e_{ji}}{P(\omega_j|\mathbf{x}_i)}\right]=$$

$$\max_{k=1}^{m}\left[P^{-(R-1)}(\omega_k)\prod_{i=1}^{R}P(\omega_k|\mathbf{x}_i)\right]\left[1+\sum_{i=1}^{R}\frac{e_{ki}}{P(\omega_k|\mathbf{x}_i)}\right] \quad (31)$$

Comparing (7) and (31) it is apparent that each term (class ω_k hypothesis) in the *error free* classifier combination rule (7) is affected by error factor

$$\left[1+\sum_{i=1}^{R}\frac{e_{ki}}{P(\omega_k|\mathbf{x}_i)}\right] \quad (32)$$

A similar analysis of the sum rule (11) commences with

$$\textit{assign} \quad Z\rightarrow\omega_j \quad \textit{if}$$

$$(1-R)P(\omega_j)+\sum_{i=1}^{R}\left[P(\omega_j|\mathbf{x}_i)+e_{ji}\right]=$$

$$\max_{k=1}^{m}\left\{(1-R)P(\omega_k)+\sum_{i=1}^{R}\left[P(\omega_k|\mathbf{x}_i)+e_{ki}\right]\right\} \quad (33)$$

which can be rewritten as

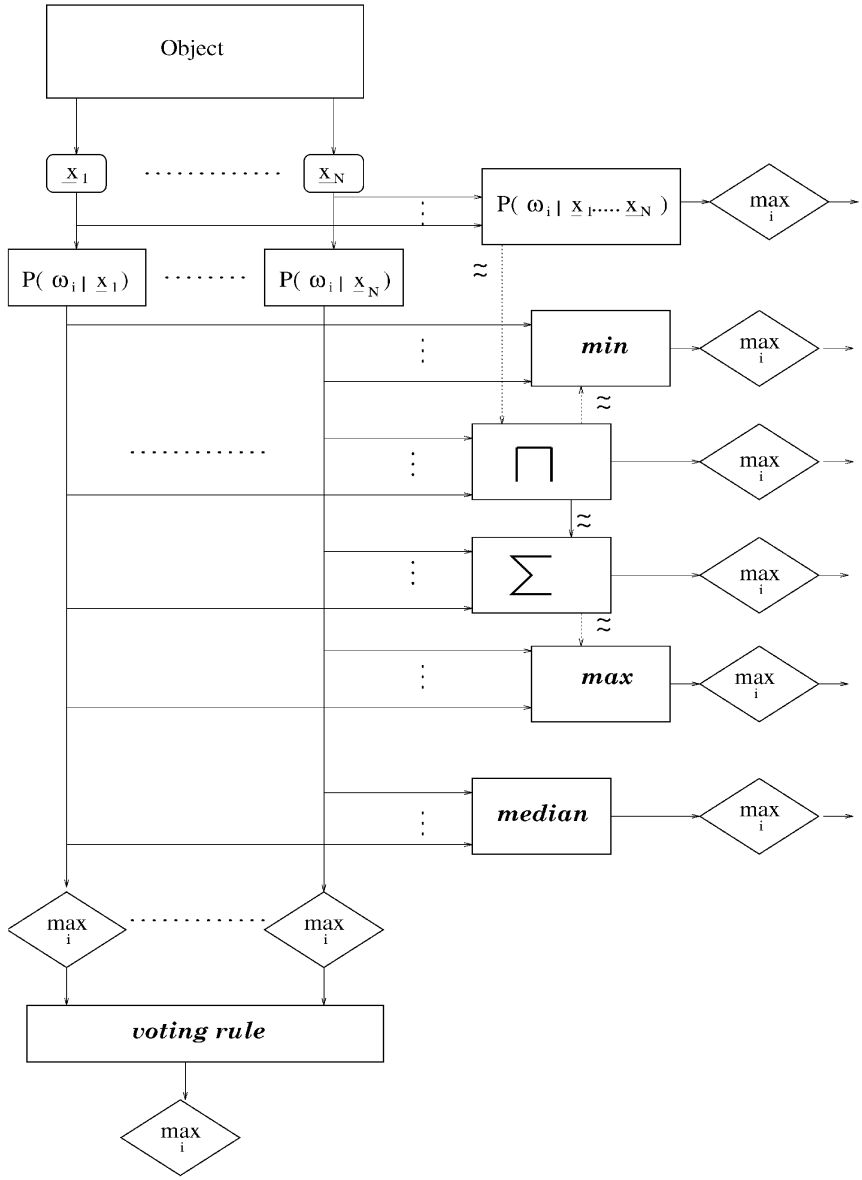


Fig. 5. Classifier combination schemes.

$$\text{assign} \quad Z \rightarrow \omega_j \quad \text{if}$$

$$(1-R)P(\omega_j) + \left[\sum_{i=1}^{R} P(\omega_j|\mathbf{x}_i)\right]\left[1 + \frac{\sum_{i=1}^{R} e_{ji}}{\sum_{i=1}^{R} P(\omega_j|\mathbf{x}_i)}\right] =$$

$$\max_{k=1}^{m}\left\{(1-R)P(\omega_k) + \left[\sum_{i=1}^{R} P(\omega_k|\mathbf{x}_i)\right]\left[1 + \frac{\sum_{i=1}^{R} e_{ki}}{\sum_{i=1}^{R} P(\omega_k|\mathbf{x}_i)}\right]\right\} \quad (34)$$

A comparison of (11) and (34) shows that each term in the *error free* classifier combination rule (11) is affected by error factor

$$\left[1 + \frac{\sum_{i=1}^{R} e_{ki}}{\sum_{i=1}^{R} P(\omega_k|\mathbf{x}_i)}\right] \quad (35)$$

Comparing error factors (32) and (35), it transpires that the sensitivity to errors of the former is much more dramatic than that of the latter. Note that since the a posteriori class probabilities are less than unity, each error e_{ki} in (32) is amplified by $\frac{1}{P(\omega_k|\mathbf{x}_i)}$. The compounded effect of all these amplified errors is equivalent to their sum. In contrast, in the sum rule, the errors are not amplified. On the contrary, their compounded effect, which is also computed as a sum, is scaled by the sum of the a posteriori probabilities. For the most probable class, this sum is likely to be greater than one which will result in the dampening of the errors. Thus, the sum decision rule is much more resilient to estimation er-

rors and this may be a plausible explanation of the superior performance of this combination strategy that we observed experimentally in Sections 4 and 5, or at least a contributing factor to it. It follows, therefore, that the sum classifier combination rule is not only a very simple and intuitive technique of improving the reliability of decision making based on different classifier opinions but it is also remarkably robust.

7 Conclusions

The problem of combining classifiers which use different representations of the patterns to be classified was studied. We have developed a common theoretical framework for classifier combination and showed that many existing schemes can be considered as special cases of compound classification where all the pattern representations are used jointly to make a decision. We have demonstrated that under different assumptions and using different approximations we can derive the commonly used classifier combination schemes such as the product rule, sum rule, min rule, max rule, median rule, and majority voting. The various classifier combination schemes were compared experimentally. A surprising outcome of the comparative study was that the combination rule developed under the most restrictive assumptions—the sum rule—outperformed other classifier combinations schemes. To explain this empirical finding, we investigated the sensitivity of various schemes to estimation errors. The sensitivity analysis has shown that the sum rule is most resilient to estimation errors and this may provide a plausible explanation for its superior performance.

Acknowledgments

This work was supported by the Science and Engineering Research Council, UK (GR/K68165) and by the European Union ACTS Project M2VTS. The authors would like to thank Andrew Elms for making available the classification results obtained using his HMM character recognizer and Kenneth Jonsson and Medha Pandit for providing the frontal face image and the voice data, respectively. We are also indebted to Stephane Pigeon for providing the face profile data and verification results and to Gilbert Maitre for making available the voice-based verification decisions, which were then used in fusion experiments.

References

[1] K.M. Ali and M.J. Pazzani, "On the Link Between Error Correlation and Error Reduction in Decision Tree Ensembles," Technical Report 95-38, ICS-UCI, 1995.

[2] S.C. Bagui and N.R. Pal, "A Multistage Generalization of the Rank Nearest Neighbor Classification Rule," *Pattern Recognition Letters*, vol. 16, no. 6, pp. 601-614, 1995.

[3] J. Cao, M. Ahmadi, and M. Shridhar, "Recognition of Handwritten Numerals With Multiple Feature and Multistage Classifier," *Pattern Recognition*, vol. 28, no. 2, pp. 153-160, 1995.

[4] S.B. Cho and J.H. Kim, "Combining Multiple Neural Networks by Fuzzy Integral for Robust Classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 25, no. 2, pp. 380-384, 1995.

[5] S.B. Cho and J.H. Kim, "Multiple Network Fusion Using Fuzzy Logic," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 497-501, 1995.

[6] D.A. Denisov and A.K. Dudkin, "Model-Based Chromosome Recognition Via Hypotheses Construction/Verification," *Pattern Recognition Letters*, vol. 15, no. 3, pp. 299- 307, 1994.

[7] H. El-Shishini, M.S. Abdel-Mottaleb, M. El-Raey, and A. Shoukry, "A Multistage Algorithm for Fast Classification of Patterns," *Pattern Recognition Letters*, vol. 10, no. 4, pp. 211-215, 1989.

[8] M.C. Fairhurst and H.M.S. Abdel Wahab, "An Interactive Two-Level Architecture for a Memory Network Pattern Classifier," *Pattern Recognition Letters*, vol. 11, no. 8, pp. 537-540, 1990.

[9] J. Franke and E. Mandler, "A Comparison of Two Approaches for Combining the Votes of Cooperating Classifiers," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, Conf. B: Pattern Recognition Methodology and Systems, vol. 2, pp. 611-614, 1992.

[10] L.K. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1,001, Oct. 1990.

[11] Hashem and B. Schmeiser, "Improving Model Accuracy Using Optimal Linear Combinations of Trained Neural Networks," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 792-794, 1995.

[12] T.K. Ho, "Random Decision Forests," *Third Int'l Conf. Document Analysis and Recognition*, pp. 278-282, Montreal, 14-16 Aug. 1995.

[13] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66-75, Jan. 1994.

[14] F. Kimura and M. Shridhar, "Handwritten Numerical Recognition Based on Multiple Algorithms," *Pattern Recognition*, vol. 24, no. 10, pp. 969-983, 1991.

[15] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," *Advances in Neural Information Processing Systems 7*, G. Tesauro, D.S. Touretzky, and T.K. Leen, eds. Cambridge, Mass.: MIT Press, 1995.

[16] M.W. Kurzynski, "On the Identity of Optimal Strategies for Multistage Classifiers," *Pattern Recognition Letters*, vol. 10, no. 1, pp. 39-46, 1989.

[17] P. Pudil, J. Novovicova, S. Blaha, and J. Kittler, "Multistage Pattern Recognition With Reject Option," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, Conf. B: Pattern Recognition Methodology and Systems, vol. 2, pp. 92-95, 1992.

[18] G. Rogova, "Combining the Results of Several Neural Network Classifiers," *Neural Networks*, vol. 7, no. 5, pp. 777-781, 1994.

[19] M. Skurichina and R.P.W. Duin, "Stabilizing Classifiers for Very Small Sample Sizes," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, Vienna, 1996.

[20] V. Tresp and M. Taniguchi, "Combining Estimators Using Non-Constant Weighting Functions," *Advances in Neural Iinformation Processing Systems 7*, G. Tesauro, D.S. Touretzky, and T.K. Leen, eds. Cambridge, Mass.: MIT Press, 1995.

[21] C.H. Tung, H.J. Lee, and J.Y. Tsai, "Multi-Stage Pre-Candidate Selection in Handwritten Chinese Character Recognition Systems," *Pattern Recognition*, vol. 27, no. 8, pp. 1,093-1,102, 1994.

[22] D.H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-260, 1992.

[23] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418-435, 1992.

[24] J.Y. Zhou and T. Pavlidis, "Discrimination of Characters by a Multi-Stage Recognition Process," *Pattern Recognition*, vol. 27, no. 11, pp. 1,539-1,549, 1994.

[25] J. Kittler, M. Hatef, and R.P.W. Duin, "Combining Classifiers," *Proc. 13th Int'l Conf. Pattern Recognition*, vol. 2, Track B, pp. 897-901, Vienna, 1996.

[26] J. Kittler, J. Matas, K. Jonsson, and M.U. Ramos Sánchez, "Combining Evidence in Personal Identity Verification Systems," *Pattern Recognition Letters*, pp. 845-852, 1997.

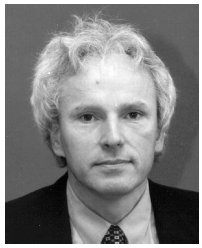
[27] L. Breiman, "Bagging Predictors," Technical Report 421, Dept. of Statistics, Univ. of California at Berkeley, 1994.

[28] Y. Freund and R.E. Shapire, "Experiments With a New Boosting Algorithm," *Proc. 13th Int'l Conf. Machine Learning*, 1996.

[29] R.E. Shapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *Proc. 14th Int'l Conf. Machine Learning*, 1997.

[30] K. Tumer and J. Ghosh, "Analysis of Decision Boundaries in Linearly Combined Neural Classifiers," *Pattern Recognition*, vol. 29, pp. 341-348, 1996.

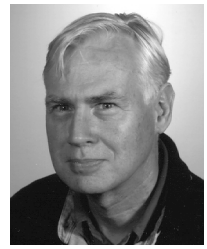
[31] K. Tumer and J. Ghosh, "Classifier Combining: Analytical Results and Implications," *Proc. Nat'l Conf. Artificial Intelligence*, Portland, Ore., 1996.
[32] K.S. Woods, K. Bowyer, and W.P. Kergelmeyer, "Combination of Multiple Classifiers Using Local Accuracy Estimates," *Proc. CVPR '96*, pp. 391-396, 1996.
[33] J. Kittler, A. Hojjatoleslami, and T. Windeatt, "Weighting Factors in Multiple Expert Fusion," *Proc. British Machine Vision Conf.*, Colchester, England, pp. 41-50, 1997.
[34] J. Kittler, A. Hojjatoleslami, and T. Windeatt, "Strategies for Combining Classifiers Employing Shared and Distinct Pattern Representations," *Pattern Recognition Letters*, to appear.
[35] J. Kittler, "Improving Recognition Rates by Classifier Combination: A Theoretical Framework," *Frontiers of Handwriting Recognition 5*, A.G. Downton and S. Impedovo, eds. World Scientific, pp. 231-247, 1997.
[36] T.S. Huang and C.Y. Suen, "Combination of Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90-94, Jan. 1995.
[37] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, N.J.: Prentice Hall, 1982.
[38] M. Hatef and J. Kittler, "Constraining Probabilistic Relaxation With Symbolic Attributes," *Proc. Sixth Int'l Conf. Computer Analysis of Images and Patterns*, V. Hlavac and R. Sara, eds., pp. 862-867, Prague, 1995.
[39] M. Hatef and J. Kittler, "Combining Symbolic With Numeric Attributes in Multiclass Object Recognition Problems," *Proc. Second Int'l Conf. Image Processing*, vol. 3, pp. 364-367, Washington, D.C., 1995.
[40] A.J. Elms, "A Connected Character Recogniser Using Level Building of HMMs," *Proc. 12th IAPR Int'l Conf. Neural Networks,* Conf. B: Pattern Recognition Methodology and Systems, vol. 2, pp. 439-441, 1994.
[41] J. Matas, K. Jonsson, and J. Kittler, "Fast Face Localisation and Verification," A. Clark, ed., *British Machine Vision Conf.,* pp. 152-161, BMVA Press, 1997.
[42] S. Pigeon and L. Vandendrope, "The M2VTS Multimodal Face Database (Release 1.00)," J. Bigun, G. Chollet, and G. Borgefors, eds., *Audio- and Video-Based Biometric Person Authentication*, pp. 403-409. Springer, 1997.
[43] D. Genoud, G. Gravier, F. Bimbot, and G. Chollet, "Combining Methods to Improve the Phone Based Speaker Verification Decision," *Proc. Int'l Conf. Speech and Language Processing*, vol 3, pp. 1,756-1,760, Philadelphia, 1996.
[44] S. Pigeon and L. Vandendrope, "Profile Authentication Using a Chamfer Matching Algorithm," J. Bigun, G. Chollet, and G. Borgefors, eds., *Audio- and Video-Based Biometric Person Authentication*, pp. 185-192. Springer, 1997.



Josef Kittler graduated from the University of Cambridge in electrical engineering in 1971, where he also obtained his PhD in pattern recognition in 1974, and the ScD degree in 1991. He joined the Department of Electronic and Electrical Engineering of Surrey University in 1986 where he is a professor in charge of the Centre for Vision, Speech, and Signal Processing. He has worked on various theoretical aspects of pattern recognition and on many applications including automatic inspection, ECG diagnosis, remote sensing, robotics, speech recognition, and document processing. His current research interests include pattern recognition, image processing, and computer vision.

He has co-authored a book with the title *Pattern Recognition: A Statistical Approach*," published by Prentice-Hall. He has published more than 300 papers. He is a member of the editorial boards of *Pattern Recognition Journal*, *Image and Vision Computing*, *Pattern Recognition Letters*, *Pattern Recognition and Artificial Intelligence*, and *Machine Vision and Applications*.

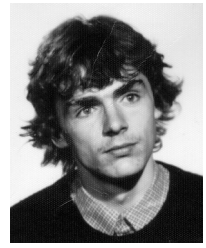
Mohamad Hatef received the BSc degree in electrical engineering from the University of Baghdad, Iraq in 1982. After graduation, he worked automatic control. He joined the University of Surrey to pursue postgraduate studies in 1991. Since 1995, he has been with ERA Technology, where he works on image and video compression.



Robert P.W. Duin studied applied physics at Delft University of Technology in the Netherlands. In 1978, he received the PhD degree for a thesis on the accuracy of statistical pattern recognizers. In his research, he included various aspects of the automatic interpretation of measurements, learning systems, and classifiers. Around 1980, he integrated the Delft Image Processor, a reconfigurable pipelined machine, in an interactive research environment for image processing. In connection with this, he initiated several projects on the comparison and evaluation of parallel architectures for image processing and pattern recognition. At the moment, his main research interest is the comparison of neural networks with the traditional pattern recognition classifiers for learning. In 1994, he stayed as a visiting professor at the University Teknologi Malaysia. In 1995, he spent his sabbatical leave at the University of Surrey in the Vision, Speech and Signal Processing Group.

He held official positions in both, the Dutch Society for Pattern Recognition and Image Processing as well as the International Association for Pattern Recognition (IAPR). He has been a member of the organizing committees of international conferences on signal processing (Eusipco 86) and pattern recognition (ICPR 92), as well as of the scientific committees of many conferences on pattern recognition, image processing, and computer vision.

At present, he is an associate professor of the Faculty of Applied Sciences of Delft University of Technology. He leads several projects on pattern recognition and neural network research, sponsored by both, the Dutch government and the industry. He is the author of a large number of scientific papers and has served as an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Pattern Recognition Letters,* and for *Pattern Analysis and Applications*. Several PhD students are coached by him. His teaching includes undergraduate, graduate, and postgraduate courses on statistics, pattern recognition, neural networks, and image processing.



Jiri (George) Matas received the MSc degree (with honors) in electrical engineering from the Czech Technical University in Prague, Czechoslovakia, in 1987, and a PhD degree from the University of Surrey, UK in 1995. He currently is a research fellow both with the Centre for Vision, Speech, and Signal Processing at the University of Surrey and the Centre for Machine Perception at the Czech Technical University. His interest include pattern recognition and computer vision.