Neural Networks for Intelligent Multimedia Processing

SUN-YUAN KUNG, FELLOW, IEEE, AND JENQ-NENG HWANG, SENIOR MEMBER, IEEE

This paper reviews key attributes of neural processing essential to intelligent multimedia processing (IMP). The objective is to show why neural networks (NN's) are a core technology for the following multimedia functionalities: 1) efficient representations for audio/visual information, 2) detection and classification techniques, 3) fusion of multimodal signals, and 4) multimodal conversion and synchronization. It also demonstrates how the adaptive NN technology presents a unified solution to a broad spectrum of multimedia applications. As substantiating evidence, representative examples where NN's are successfully applied to IMP applications are highlighted. The examples cover a broad range, including image visualization, tracking of moving objects, image/video segmentation, texture classification, face-object detection/recognition, audio classification, multimodal recognition, and multimodal lip reading.

Keywords— *Access-control security, ACON/OCON networks, active contour model, adaptive expectation-maximization, audio classification, audio-to-visual conversion, audio/visual fusion, decision-based neural network, face recognition, hidden Markov models, image segmentation, image/video representation, independent components analyses, intelligent multimedia processing, linear/nonlinear fusion network, mixture of experts, motion-based video segmentation, multilayer perceptron, multimodal lip reading, neural-network technology, object classification, object detection, object tracking, personal authentification, principal components analyses, RBF networks, self-organizing feature map, subject-based retrieval, texture classification, time-delay neural network, vector quantization, video indexing and browsing.*

I. Introduction

Multimedia technologies will profoundly change the way we access information, conduct business, communicate, educate, learn, and entertain [21], [95], [142]. Multimedia technologies also represent a new opportunity for research interactions among a variety of media such as speech, audio, image, video, text, and graphics. As digitization and encoding of image/video have become more affordable, computer and Web data base systems are starting to store voluminous image/video data. Consequently, a massive amount of visual information on-line has come closer to a reality. It promises a quantum elevation of the level of tomorrow's world in entertainment and business. As data-acquisition technology advances rapidly, however, we have now substantially fallen behind in terms of technologies for indexing and retrieving visual information in large archives.

Manuscript received August 12, 1997; revised December 15, 1997. The Guest Editor coordinating the review of this paper and approving it for publication was K. J. R. Liu.

S.-Y. Kung is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: kung@princeton.edu).

J.-N. Hwang is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: hwang@ee.washington.edu).

Publisher Item Identifier S 0018-9219(98)03525-7.

A. Challenges of Multimedia Information Processing

Consider only a small sample of the challenging problems:

- how to find some desired pictures from a large archive of photographs;
- how to search a specific clip from a video archive consisting of tons of tapes;
- how to support visual-based interactive learning systems servicing a broad spectrum of customers including homemakers, students, and professionals;
- how to create and maintain multimedia data bases capable of providing information in multiple medias, including text, audio, image, and video.

For example, it would be desirable to have a tool that efficiently searches the Web for a desired picture (or video clip) and/or audio clip by using as a query a shot of multimedia information [139], [152]. Nowadays, some popular queries might look like: *"Find frames with 30% blue on top and 70% green in bottom"* or *"Find the images or clips similar to this drawing."* In contrast to the above similarity-based queries, it was argued that a so-called "subject-based" query [152] might be more likely to be used by users, e.g., a query request such as *"Find Reagan speaking to the Congress."* The subject-based query offers a more user-friendly interface but it also introduces a greater technical challenge, which calls for advances in two distinctive research frontiers [21].

- *Computer networking technology*: Novel communication and networking technologies are critical for a multimedia data base system to support interactive dynamic interfaces. A truly integrated media system must

connect with individual users and content-addressable multimedia data bases. This will involve both logical connection to support information sharing and physical connection via computer networks and data transfer.

- *Information processing technology*: To advance the technologies of indexing and retrieval of visual information in large archives, multimedia content-based indexing would complement well the text-based search. On-line and real-time visual information retrieval and display systems would provide popular services to professionals, such as business traders, researchers, and librarians, as well as general users, such as students and homemakers. Such systems must successfully combine digital video and audio, text animation, graphics, and knowledge about such information units and their interrelationships in real time.

This paper mainly addresses emerging issues closely related to the research frontier on *information-processing technology*.

As speech, image, and video are playing increasingly dominant roles in multimedia information processing, content-based retrieval has a broad spectrum of applications. Quick and easy access of large speech/image/video data bases must be incorporated as an integral part of many near-future multimedia applications. Future multimedia technologies will need to handle information with an increasing level of *intelligence,* i.e., automatic extraction, recognition, interpretation, and interactions of multimodal signals. This will lead to what can be called *intelligent multimedia processing* (IMP) technology.

Indeed, the technology frontier of information processing is shifting from coding (MPEG-1 [100], MPEG-2 [101], and MPEG-4 [102]) to automatic recognition—a trend precipitated by a new member of the Moving Picture Experts Group (MPEG) family, MPEG-7 [103], [104], which focuses on "multimedia content description interface." Its research domain will cover techniques for object-based tracking/segmentation, pattern detection/recognition, content-based indexing and retrieval, and fusion of multimodal signals. For these, neural networks can offer a very promising horizon.

B. Why Neural Processing for Multimedia Applications

The main reason why neural networks (NN's) are perceived as a critical core technology for IMP hinges upon their adaptive learning capability, [3], [66], which enables machines to be taught to interpret possible variations of the same object or pattern, e.g., scale, orientation, and perspective. More specifically, they have the following attributes.

- Neural networks offer unsupervised clustering (i.e., no specific target label/response is provided for any input) and/or supervised learning mechanisms (the input and corresponding target label/response are both given) for recognition of objects that are deformed or with incomplete information. Ultimately, a neural information engine can be "trained" to see or hear, to recognize objects or speech, or to perceive human gestures.
- Neural networks are powerful pattern classifiers and encompass many similarities with statistical pattern-recognition approaches. They appear to be most powerful and appealing when explicit *a priori* knowledge of underlying probability distributions is unknown. By their *training by example* property, neural network classifiers may be properly trained to provide outputs that nonparametrically approximate *a posteriori* class probabilities [123].
- Neural networks offer a universal approximation capability, i.e., they are able accurately to approximate unknown systems based on sparse sets of noisy data. In this context, some neural models have also effectively incorporated statistical signal-processing and optimization techniques.
- Temporal neural models, which are specifically designed to deal with temporal signals (see Section II-D), further expand the application domain in multimedia processing, particularly audio, speech, and audio-visual integration and interactions.
- A hierarchical network of neural modules will be vital to facilitate a search mechanism used in a voluminous, or Web-wide, data base. Typically, a tree network structure would be adopted so that kernels that are common to all the models can be stored as the root of the tree. The leaves of the tree would correspond to the individual neural modules, while the paths from root to leaf correspond to the kernels involved.

Consequently, neural networks have recently received increasing attention in many multimedia applications. Here, we list just a few examples:

a) *human perception:* facial expression and emotion categorization [125], human color perception [130], multimedia data visualization [2], [124];

b) *computer–human communication:* face recognition [83], lip-reading analysis [23], [24], [80], [112], human–human and computer–human communication [106];

c) *multimodal representation and information retrieving:* hyperlinking of multimedia objects [74], queries and search of multimedia information [91], three-dimensional (3-D) object representation and motion tracking [138], image sequence generation and animation [94].

More concrete application examples will be discussed in the subsequent sections.

C. Focal Technical Issues Addressed in this Paper

This paper will focus on vital technical issues in the research frontier on information-processing technology, par-

ticularly those closely related to IMP. More specifically, this paper will demonstrate why and how neural networks are a core technology for efficient representations for audio/visual information (Section II-A), detection and classification techniques (Section II-B), fusion of multimodal signals (Section II-C), and multimodal conversion and synchronization (Section II-D). Let us first offer some motivations as well as a brief explanation of the key technical points.

1) Efficient Representations for Audio/Visual Information: An efficient representation of the information can facilitate many useful multimedia functionalities, such as object-based indexing and access. To this end, it is vital to have sophisticated preprocessing of the image or video data. For many multimedia applications, preprocessing is usually carried out on the input signals to make the subsequent processing modeling and classification tasks easier. [For example, segmentation of two-dimensional (2-D) or 3-D images and video for content-based coding and representation in the context of the MPEG or Joint Photographic Experts Group standards]. The more sophisticated representation obtained by preprocessing, the less sophisticated classifier would be required. Hence, a synergistic balance (and eventually interaction) between representation and indexing needs to be explored.

An efficient representation of a vast amount of multimedia data can often be achieved by adaptive data clustering or model representation mechanisms, which happen to be the most promising strength of many well-established unsupervised neural networks, for example, self-organizing feature map and principal component analysis (PCA) neural networks. The evolution from conventional statistical clustering and/or contour/shape modeling to these unsupervised NN's will be highlighted in Section II-A.

Some of these NN's have been incorporated for various feature-extraction, moving-object-tracking, and segmentation applications. Illustrative samples for such preprocessing examples are provided in Section III-A.

2) Detection and Classification for Audio/Visual Data Bases: As most digital text, audio, and visual archives will exist on various servers all over the world, it will become increasingly difficult to locate and access the information. This necessitates automatic search tools for indexing and access. Detection and classification constitute a very basic tool for most search and indexing mechanisms. Detection of a (deformable) pattern or object has long been an important machine-learning and computer-vision problem. The task involves finding specific (but locally deformable) patterns in images, e.g., human faces. What are critically needed are powerful search strategies to identify contents on speech or visual clues, possibly without the benefit of textual information. These will have important commercial applications including automatic teller machine access control, surveillance, and video-conferencing systems.

Several *static* (i.e., no feedback connections are used in the network and can only respond to each individual input one at a time without memory) supervised NN's, which are useful for detection and classification, will be covered in Section II-B. Built upon these NN's, many NN content-based image search systems have been developed for various applications. On the horizon are several promising tools that allow users to specify image queries by giving examples, drawing sketches, selecting visual features (e.g., color, texture, shape, and motion), and arranging the spatio-temporal structure of features. Some exemplar NN systems will be presented in Section III-D. They serve to demonstrate the fact that unsupervised and supervised NN models are useful means for developing reliable search mechanisms.

3) Multimodal Media Fusion. Combine Multiple Sources: Multimedia signal processing is more than simply a collage of text, audio, images, and video. The correlation between audio and video can be utilized to achieve more efficient coding and recognition. New application systems and thus new research opportunities arise in the area of fusion and interaction among these medias.

Humans perform most perception and recognition tasks based on joint processing of the input multimodal data. The biological NN's of humans handle multimodal data through visual, auditory, and sensory mechanisms via some form of adaptive-processing (learning/retrieving) algorithms, which remain largely mysterious to us. Motivated by the nature of biological NN processing, fusion NN models, combining information from multiple sensor/data sources, are being pursued as a universal data-processing engine for multimodal signals. Linear fusion networks and nonlinear fusion networks are discussed in Section II-C.

Audio-visual interaction can be used for personal authentification and verification. A visual/auditory fusion network for such an application is discussed in Section III-B.

4) Multimodal Conversion and Synchronization: One of the most interesting interactions among different media is that between audio and video. In multimodal speech communication, audio-visual interaction has a significant role, as evidenced by the "McGurk effect" [96]. It shows that human perception of speech is bimodal in that acoustic speech can be affected by visual cues from lip movements. For example, one experiment shows that when a person *sees* a speaker saying /ga/ but *hears* the sound /ba/, the person perceives neither /ga/ nor /ba/ but something close to /da/. In a video-conferencing application, it is conceivable that the video frame rate is severely limited by the bandwidth and is by far very inadequate for lip-synchronization perception. One solution is to warp the acoustic signal to make it synchronized with the person's mouth movements, which will be useful for dubbing in a studio and other nonreal-time applications. There are a class of *temporal* neural models (i.e., feedback connections are used to keep track of temporal correlation of signals) that can facilitate the conversion and synchronization processes. Prominent temporal NN models and popular statistical approaches will be reviewed in Section II-D. In addition, an application

example of audio-to-visual conversion will be presented in Section III-C1. Verbal communication has been efficiently achieved by combining speech recognition and visual interpretation of lip movements (or even facial expressions or body language). As another example, an NN-based lip-reading system via audio and visual integration will be presented in Section III-C2. Other potential applications include dubbing of movies, segmentation of video scenes, and human–computer interfaces.

D. Organization of this Paper

Section II reviews some of the key NN's and their relationship with statistical pattern-recognition techniques, then highlights their usefulness to IMP applications. Built upon these NN models, exemplar IMP applications will be illustrated in Section III. Last, some open technical issues and promising application trends will be suggested in the concluding section.

II. Neural and Statistical Approaches to Multimedia Data Representation, Classification, and Fusion

We will discuss in this section a variety of statistical learning techniques adopted by NN's. By these techniques, machines may be taught automatically to interpret and represent possible variations of the same object or pattern. Some of these NN's (e.g., self-organization feature map) can be perceived as a natural evolution from traditional statistical-clustering and parameter-estimation techniques [e.g., vector quantization (VQ) and expectation-maximization (EM)]. These NN's can also be incorporated into traditional pattern-recognition techniques (e.g., active contour model) to enhance the performance.

A. Multimedia Data Representation

From the learning perspective, neural networks are grouped into *unsupervised learning* and *supervised learning* networks. Static features extraction is often inadequate for an adaptive environment, where users may require adaptive and dynamic feature-extraction tools. Unsupervised neural techniques are very amenable for dynamic feature extraction. The self-organization feature map (SOFM) is one representative of an unsupervised NN, which combines the advantage of statistical data clustering (such as VQ and PCA) and local continuity constraint (as imposed in the active contour model search). We will provide a quick overview of these techniques and their application examples in this section.

1) VQ: K-mean and VQ commonly are used interchangeably. K-mean [37] can be treated as a special means for implementing VQ [43]. K-mean and VQ classify input patterns based on the *nearest neighbor rule*. Given a data set $\mathbf{V} = \{\mathbf{v}_i \mid i = 1, \cdots, N\}$ to be grouped into K representative patterns, $\hat{\mathbf{V}} = \{\mathbf{v}'_r \mid r = 1, \cdots, K\}$. The nearest neighbor rule for classifying a $\mathbf{v}$ is to assign it the class associated with $\mathbf{v}'_r$. K-mean and VQ have simple learning rules, and the classification scheme is straightforward. Mathematically, the objective is to minimize

$$E(h; V) = \sum_{i,r} h_r(\mathbf{v}_i)|\mathbf{v}_i - \mathbf{v}'_r|^2 \tag{1}$$

where $h_r(\mathbf{v}_i) = 1$ for the members only (i.e., $\mathbf{v}_i$ is closest to $\mathbf{v}'_r$ among all K members in $\hat{\mathbf{V}}$), otherwise $h_r(\mathbf{v}_i) = 0$. Approximately, the criterion also leads to a best compression in coding. It was shown that with an unlimited number of patterns, the error rate is never worse than twice the Bayes rate [37].

The K-mean algorithm [90] provides a simple mechanism for minimizing the sum of squared error with K clusters. The fundamental K-mean and VQ learning rule can be summarized as follows [85]:

Step 0) *Given:* A set of training data and K initial cluster centroids.

Step 1) Cluster the training data using the centroids of current iteration based on the (weighted) Euclidean distance, as given in (1). If the average distortion is small enough, quit.

Step 2) Replace the old centroids by the centroids of clusters obtained in Step 1). Go to Step 1).

a) Application examples: Daugman [32] and Rice [122] used the minimum Hamming distance rule for classifying iris patterns and hand-vein structural patterns, respectively. Pentland *et al.* [97], [113], [140] projected the face images onto the eigenface subspace and determined the classes of the input patterns by applying nearest neighbor search using a Euclidean distance metric. In [16], a VQ-like NN was applied for object recognition by evidence accumulation. Cox *et al.* [31] proposed an algorithm for face recognition called the *mixture-distance* algorithm, which uses VQ to estimate both the true pattern-generative process (the "platonic" process) and the process that generates the vectors we ultimately observed (the "observation" process). A recognition rate of 95% on a data base of 685 individuals was reported.

2) EM Method. In the case of clustering data with known parametric distribution without knowing the specific values of distribution parameters, the EM method serves as a powerful tool for estimating these distribution parameters and thus results in the purpose of data clustering. The EM algorithm is a well-established iterative method for maximum likelihood estimation (MLE) [34], [150] and for clustering a mixture of Gaussian distributions. While the EM algorithm can be perceived as a soft version (i.e., no hard decision on the data clustering) of the VQ algorithms, it is more computationally intensive. It offers several attractive attributes for IMP applications.

- It offers an option of "soft" classification.
- It offers a "soft pruning" mechanism. This is important because features with low probability should not be

allowed to have too much influence on the training of the class parameters.

- It naturally accommodates model-based clustering formulation.
- It allows incorporation of prior information.
- The EM training algorithm yields probabilistic parameters, which are instrumental for media fusion. For linear media fusion, EM plays a role in training the weights on the fusion layer. This will be elaborated on later.

Suppose the data in the set $\mathbf{V} = \{\mathbf{v}_i \mid i = 1, \cdots, N\}$ are independently identically distributed samples, with a mixture-of-Gaussian (say, K Gaussians) type likelihood function $p(\mathbf{v}_i)$

$$p(\mathbf{v}_i) = \sum_{r=1}^{K} P(\Theta_r) p(\mathbf{v}_i \mid \Theta_r) \tag{2}$$

where Θ_r represents the rth cluster and $P(\Theta_r)$ denotes the *prior* probability of cluster r. By definition, $\sum_{r=1}^{K} P(\Theta_r) = 1$.

The most common clustering is via either a radial basis function (RBF) or a more general elliptic basis function. In the latter case, the component density $p(\mathbf{v}_i \mid \Theta_r)$ is a Gaussian distribution, with the model parameter of the rth cluster $\Theta_r = \{\mu_r, \mathbf{\Sigma}_r\}$ consisting of the mean vector and full-rank covariance matrix.

By incorporating an additional entropy term for the purpose of inducing the membership fuzziness [46], [154], the EM algorithm can be interpreted as minimizing (with respect to $\{\Theta_r, h_r(\mathbf{v}_i), \forall i, r\}$) the following effective energy function:

$$\sum_{i,r} h_r(\mathbf{v}_i) s_r(\mathbf{v}_i, \mu_r, \Sigma_r) + \sigma_T \sum_{i,r} h_r(\mathbf{v}_i) \log h_r(\mathbf{v}_i) \tag{3}$$

where $s_r(b, \mu_r, \Sigma_r)$ denotes a weighted squared error

$$s_r(b, \mu_r, \Sigma_r) = (b - \mu_r)\Sigma_r^{-1}(b - \mu_r)^T$$

where $h_r(\mathbf{v}_i)$ denotes a "fuzzy" membership function defined as the probability of $\mathbf{v}_i$'s belonging to the rth cluster given a prior model, i.e., $\Pr(\mathbf{v}_i \in \Theta_r \mid \mathbf{v}_i, \Theta)$.

There are two steps in EM iterations.

1) The *E step* involves searching the best cluster probability h_r in order to optimize E while fixing the model parameter $\Theta = \{\Theta_r, \forall r\}$.
2) The *M step* involves searching the best model parameter $\Theta = \{\Theta_r, \forall r\}$, which optimizes E, while fixing the cluster probability $h_r(\mathbf{v}_i), \forall i$.

The fuzzy membership function can be derived as

$$h_r(\mathbf{v}_i) \propto e^{-s_r(\mathbf{v}_i, \mu_r, \Sigma_r)/\sigma_T} \tag{4}$$

and $\sum_{r=1}^{K} h_r(\mathbf{v}_i) = 1$. The EM scheme can be seen as a probabilistic (and more general) extension of K-mean clustering. In other words, K-mean clustering is a special case of the EM scheme. [Note that (3) would be reduced into (1) when σ_T approaches zero]. By examining (4), this leads to a hard-decision clustering (i.e., with cluster probabilities equal to either one or zero). This demonstrates that σ_T plays the same role as the temperature parameter in the simulated annealing method. By the same token, it is a common practice to use some sort of annealing temperature schedule, i.e., starting with a higher σ_T and then gradually decreasing σ_T to a lower value as iterations progress.

a) Application examples: EM has a broad range of applications. The data set $\mathbf{V}$ used in EM could be input–output training pairs in supervised learning or input data in unsupervised learning [154]. When the EM iteration converges, it is ideal to yield the MLE of the data distribution. For image and multimedia applications, EM has been reported to deliver excellent performance in several image-segmentation applications [39], [78], [115], [146]. An application example for motion-based video segmentation will be discussed in the Section III-A4.

3) Active Contour Model: The Snake Algorithm: The snake algorithm is another way of searching a small set of representative data with distinctive characteristics among a large set of potential candidates. The original snake models were first introduced by Kass *et al.* [61]. A snake is an open or closed elastic curve represented by a set of control points. Finding contours of distinct features (specified by the user's *a priori* in an energy formulation) is done by deforming and moving the elastic curve gradually from an initial shape residing on the image toward the positions where distinct features are to be extracted. This deformation process is guided by iteratively searching for a nearby local minimum of an energy function, which consists of the internal energy (a smoothness constraint of the snake curve: tension and bending) and the external energy that indicates the degree of matching for features (such as high image intensity for bright regions or large gradient strength for edges).

Mathematically, a snake is a deformable curve whose shape is controlled by the internal spline energy (smoothness constraint imposed on the curve) and the external feature energy (defined by the distinct features). More specifically

$$E_{\text{snake}} = \sum_{i=1}^{N} (E_{\text{int}}(i) + E_{\text{ext}}(i)) \tag{5}$$

$$E_{\text{int}}(i) = \alpha_i \left\| \mathbf{v}'_i - \mathbf{v}'_{i-1} \right\|^2 + \beta_i \left\| \mathbf{v}'_{i-1} - 2\mathbf{v}'_i + \mathbf{v}'_{i+1} \right\|^2 \tag{6}$$

where N is the number of snake points, $\mathbf{v}'_i = (x_i, y_i)$ is a coordinate of the ith snake point, α_i is a constant imposing the tension constraint between two adjacent snake points, β_i is a constant imposing the bending constraint among three neighboring snake points, and $E_{\text{ext}}(i)$ is usually some sort of image gradient function if edge detection is the goal.

In a 2-D application, the following equations [61] were iteratively solved to find a local minimum of the snake energy function given in (5):

$$\begin{aligned} \mathbf{A}\mathbf{x} + \mathbf{f}_x(\mathbf{x}, \mathbf{y}) &= 0 \\ \mathbf{A}\mathbf{y} + \mathbf{f}_y(\mathbf{x}, \mathbf{y}) &= 0 \end{aligned} \tag{7}$$

where the pentadiagonal matrix $\mathbf{A}$, whose band is a function of α and β, imposes constraints on the relationship among five neighboring snake points; vectors $\mathbf{x} = [x_1, \cdots, x_N]^T$ and $\mathbf{y} = [y_1, \cdots, y_N]^T$ are coordinates of N snake points, and vectors $\mathbf{f}_x(\mathbf{x}, \mathbf{y})$ and $\mathbf{f}_y(\mathbf{x}, \mathbf{y})$ denote the partial derivatives of external energy on each snake point, i.e., $f_x(x_i, y_i) = \partial E_{\text{ext}}(x_i, y_i)/\partial x$ and $f_y(x_i, y_i) = \partial E_{\text{ext}}(x_i, y_i)/\partial y$.

The formulation of a "good" external energy function is difficult because real-world images are often too noisy and/or too complex to expect low-level image-processing techniques to generate a usable energy profile. Thanks to its nonlinear mapping and generalization capability, the feed-forward multilayer neural networks [86], [126], [127] can be used to generate systematically the external energy profile through data training [25].

a) Application examples: The snake algorithms have been widely used in many signal/image applications. More specifically, in [61], the original snake model is used to track the movements of the human mouth. In [76], a snake model is combined with a Euclidean distance transform to simulate the grass-fire transform for shape description. In [40] and [61], a stereo energy is used in a snake model to improve the detection of the matched features. In [155], lines of fingerprint images are extracted by interpolating global curves from short splines through a 2-D tangent field. In [135], a Kalman filter is used with a snake model to track the dynamics of a moving object. The active contour model incorporated with "balloon force" can also be used to radially push outward the contour nodes and can serve as a good mechanism for tracking the 2-D heart contour frame by frame [49]. A slight modification of the balloon tracking by considering region-based external force has also been proposed [65].

4) SOFM: The basic idea of constructing an SOFM is to incorporate into the competitive learning (clustering) rule some degree of sensitivity with respect to the neighborhood or history. This provides a way to avoid totally unlearned neurons and helps enhance certain topological property that should be preserved in the feature mapping (or data clustering).

Suppose that an input pattern has n features and is represented by a vector $\mathbf{x}$ in an n-dimensional pattern space. The network maps the input patterns to an output space. The output space in this case is assumed to be one-dimensional or 2-D arrays of output nodes, which possess a certain topological order. The question is how to cluster these data so that the ordered relationship can be preserved. Kohonen proposed to allow the centroids (represented by output nodes of an SOFM) to interact laterally, leading to the SOFM [63], [64], which was originally inspired by a biological model.

The most prominent feature is the concept of excitatory learning within a neighborhood around the winning neuron. The size of the neighborhood slowly decreases with each iteration. A version of *training rule* is described below.

1) First, a winning neuron is selected as the one with the shortest Euclidean distance (nearest neighbor)

$$\|\mathbf{x} - \mathbf{w}_i\|$$

between its weight vector and the input vector, where $\mathbf{w}_i$ denotes the weight vector corresponding to the ith output neuron.

2) Let i^* denote the index of the winner and let I^* denote a set of indexes corresponding to a defined neighborhood of winner i^*. Then the weights associated with the winner and its neighboring neurons are updated by

$$\Delta \mathbf{w}_j = \eta(\mathbf{x} - \mathbf{w}_j)$$

for all the indexes $j \in I^*$ and η is a small positive learning rate. The amount of updating may be weighted according to a preassigned "neighborhood function" $\Lambda(j, i^*)$

$$\Delta \mathbf{w}_j = \eta \Lambda(j, i^*)(\mathbf{x} - \mathbf{w}_j) \tag{8}$$

for all j. For example, a *neighborhood function* $\Lambda(j, i^*)$ may be chosen as

$$\Lambda(j, i^*) = \exp\left(-|\mathbf{r}_j - \mathbf{r}_{i^*}|^2/2\sigma^2\right) \tag{9}$$

where $\mathbf{r}_j$ represents the position of the neuron j in the output space. The convergence of the feature map depends on a proper choice of η. One plausible choice is that $\eta = 1/t$, where t denotes the iteration number. The size of neighborhood (or σ) should decrease gradually.

3) The weight update should be immediately succeeded by the normalization of $\mathbf{w}_i$.

In the *retrieving phase,* all the output neurons calculate the Euclidean distance between the weights and the input vector, and the winning neuron is the one with the shortest distance.

By updating all the weights connecting to a neighborhood of the target neurons, the SOFM enables the neighboring neurons to become more responsive to the same input pattern. Consequently, the correlation between neighboring nodes can be enhanced. Once such a correlation is established, the size of a neighborhood can be decreased gradually, based on the desire of having a stronger identity of individual nodes.

a) Application examples: There are many examples of successful applications of SOFM's. More specifically, the SOFM network was used to evaluate the quality of a saw blade by analyzing its vibration measurements, which ultimately determines the performance of a machining process [9]. The major advantage of SOFM's is the unsupervised learning capability, which makes them ideal for machine health monitoring situations, e.g., novelty detection is then possible on-line or classes can be labeled to give diagnosis [45]. A good system-configuration algorithm produces the

Table 1 Comparison Between Extractions of PCA and ICA

	PCA	ICA
components extracted	principal component	super-Gaussian component
basic learning rule	$\mathbf{w}(t) = +\beta[\mathbf{x}(t)y(t) - \mathbf{m}(t)y(t)^2]$	$\Delta\mathbf{w}(t) = +\beta[\mathbf{v}(t)y^3(t) - \mathbf{w}(t)y(t)^4]$
optimization function	output variance	single-output kurtosis

required performance and reliability with maximum economy. Actual design changes are frequently kept to a minimum to reduce the risk of failure. As a result, it is important to analyze the configurations, components, and materials of past designs so that good aspects may be reused and poor ones changed. A generic method of configuration evaluation based on an SOFM has been successfully reported [105]. The SOFM architecture with activation retention and decay in order to create unique distributed response patterns for different sequences has also been successfully proposed for mapping arbitrary sequences of binary and real numbers, as well as phonemic representations of English words [57]. By using a selective learnable SOFM, which has the special property of effectively creating spatially organized internal representations and nonlinear relations of various input signals, a practical and generalized method was proposed in which effective nonlinear shape restoration is possible regardless of the existence of distortion models [44]. For many other examples of successful applications, the interested reader may see, e.g., [29], [72], and [131].

5) Principal and Independent Components Analyses (ICA's): PCA provides an effective way to find representative components of a large set of multivariate data. The basic learning rules for extracting principal components follow the Hebbian rule and the Oja rule [111]. The PCA problem can be viewed as an unsupervised learning network following the traditional Hebbian-type learning. The basic network is one where the neuron is a simple linear unit with

$$y(t) = \mathbf{w}(t)^T\mathbf{x}(t). \tag{10}$$

To enhance the correlation between the input $\mathbf{x}(t)$ and the extracted component $y(t)$, it is natural to use a Hebbian-type rule

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \beta\mathbf{x}(t)y(t). \tag{11}$$

The above Hebbian rule is impractical for PCA, taking into account the finite-word-length effect, since the training weights will eventually overflow (i.e., exceed the limit of dynamic range) before the first component totally dominates and the other components sufficiently diminish. An effective technique to overcome the overflow problem is to ensure that the weight vectors remain normalized after updating, leading to the following Oja rule:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \beta\big[\mathbf{x}(t)y(t) - \mathbf{w}(t)y(t)^2\big]. \tag{12}$$

A lateral neural network (called APEX) was found useful for the extraction of *multiple* principal components [33], [67]. The structure incorporates lateral connections into the network. The structure, together with an orthogonalization learning rule, helps ensure that the "orthogonality" is preserved between multiple principal components. A numerical analysis on their learning rates and convergence properties have also been established.

ICA has played a prominent role in many communication and signal/image-processing application domains [8], [15], [27]. It has recently received a lot of interest from the neural-network community [110]. Just like the PCA, the output of an ICA network is meant for extracting an independent component. A *hybrid mixture* is a mixture of super-Gaussian, Gaussian, and sub-Gaussian independent components (IC's). It was shown in [69] and [70] that the kurtosis of a single-output process $y(t) = \mathbf{m}^T\mathbf{x}$ has the very desirable property that most extrema points correspond to pure IC's. In fact, all the positive local maxima (with respect to negative local minima) can yield super-Gaussian (resp. sub-Gaussian) IC's from any mixture [69].

It is interesting to note that PCA and ICA can share a learning network. For example, the APEX lateral network could be used for either PCA or ICA. Table 1 summarizes a comparison between extractions of PCA and ICA.

Briefly speaking, after prewhitening of the input $\mathbf{x}(t)$, which yields a new observation space $\mathbf{v}$, the following learning rule [69]

$$\Delta\mathbf{w}(t) = \pm\beta\big[\mathbf{v}(t)y^3(t) - \mathbf{w}(t)y(t)^4\big] \tag{13}$$

can be applied to extract super-Gaussian (via "+" rule) and sub-Gaussian (via "−" rule) IC's. (A numerical analysis indicated that renormalization of the weight vector $\|\mathbf{w} = 1\|$ will be needed for sub-Gaussian extraction.) This learning rule has a very similar appearance as the Oja rule (12). The learning rule and an APEX-like orthogonalization (deflation) technique form the basis of an ICA neural network called kurtosis-based independent component network (KuicNet). Some KuicNet application examples for real speech/image data processing are presented in [70], and its extension to blind deconvolution/equalization is found in [35].

a) Application examples: ICA has recently found interesting applications in signal processing (e.g., interference removal, blind source separation, cocktail party problem) and communication (e.g., blind deconvolution, multipath, multichannel equalization) [8], [15]. PCA has been successfully applied in much broader signal/image processing and compression applications. Due to space constraints, only a few samples can be highlighted here. For example, in

Section III-A4, a motion-based video segmentation makes use of PCA for initial clustering of the selected feature blocks. A PCA-based eigenface technique for a face-recognition algorithm was studied in [7]. Its performance was compared with the "Fisherface" method based on tests on the Harvard and Yale face data bases.

The lip-reading system of Bregler and Konig [10], combining both audio and visual features, adopted PCA to guide the snake search (the so-called active shape models [28]) on grayscale video for the visual front end of their lip-reading system. There are two ways of performing PCA. The contour-based PCA is directly based on the snake-searched points (form a vector using the searched snake points and project into the fewer numbers of principal components). The area-based PCA is directly based on the gray-level matrix surrounding the lips. Instead of reducing the dimensionality of the visual features, as performed by the contour-based Karhunen–Loeve transform, one can also reduce the variation of mouth shapes by summing fewer principal components to form the contours (with the same dimension as the original features). It was concluded that a gray-level matrix contains more information for classifying visemes. Another attempt in the PCA-based lip-motion modeling is to express the PCA coefficients as a function of a limited set of articulatory parameters that describe the external appearance of the mouth [81]. These articulatory parameters were directly estimated from the speech waveform based on a bank of (time-delay) NN's.

B. Multimedia Data Detection/Classification

In many application scenarios, e.g., optical character recognition (OCR), texture analysis, face detection, etc., many prior examples of a targeted class or object are available for training, while the *a priori* class probability distribution is unknown. These training examples may be best exploited as valuable teacher information in supervised learning models. In general, detection/classification based on supervised learning models by far outperforms those via unsupervised clustering techniques. (See Section III-A3.) That is why supervised neural networks are generally adopted for detection/classification applications.

1) Multilayer Perceptron (MLP): MLP is one of the most popular neural network models. Usually the basis function of each neuron (the perceptron) is the linear basis function, and the activation is modeled by a sigmoidal function. Structure wise, MLP is a spatially iterative neural network with several layers of *hidden neuron units* between the input and output neuron layers. The most commonly used learning scheme for the MLP is the *back-propagation* algorithm [126]. The weight updating for the hidden layers adopts the mechanism of back-propagated corrective signal from the output layer. It has been shown that the MLP, given flexible network/neuron dimensions, offers an asymptotic approximation capability. It was demonstrated in [147] that two-layer (one hidden only) perceptrons should be adequate as universal approximators of any nonlinear functions.

Let us assume an L-layer feed-forward neural network (with N_l units at the lth layer). Each unit—say, the ith unit at the $(l+1)$th layer—receives the weighted inputs from other units at the lth layer to yield the net input $u_i(l+1)$. The net input value $u_i(l+1)$, along with the external input $\theta_i(l+1)$, will determine the new activation value $a_i(l+1)$ by the *nonlinear activation* function $f_i(l+1)$. From algorithmic point of view, the processing of this multilayer feed-forward neural network can be divided into two phases: *retrieving* and *learning*.

a) Retrieving phase: Suppose that the weights of the network are known. In response to the input (test pattern) $\{a_i(0), i = 1, \cdots, N_0\}$, the system dynamics in the retrieving phase of an L-layer MLP network iterates through all the layers to generate the responding (retrieval) response $\{a_i(L), i = 1, \cdots, N_L\}$ at the output layer

$$u_i(l+1) = \sum_{j=1}^{N_l} w_{ij}(l+1)a_j(l) + \theta_i(l+1)$$
$$a_i(l+1) = f_i(u_i(l+1)) = f_i(l+1) \tag{14}$$

where $1 \le i \le N_{l+1}$, $0 \le l \le L-1$, and f_i is nondecreasing and differentiable (e.g., sigmoid function [126]). For simplicity, the external inputs $\{\theta_i(l+1)\}$ are often treated as special modifiable synaptic weights $\{w_{i,0}(l+1)\}$, which have clamped inputs $a_0(l) = 1$.

b) Learning phase: The learning phase of this L-layer MLP network follows a simple gradient descent approach. Given a pair of input/target training patterns $\{a_i(0), i = 1, \cdots, N_0\}$, $\{t_j,\ j = 1, \cdots, N_L\}$, the goal is to iteratively (by presenting a set of training pattern pairs many times) choose a set of $\{w_{ij}(l),\ \forall l\}$ for all layers so that the squared error function E can be minimized

$$E = \frac{1}{2}\sum_{i=1}^{N_L}(t_i - a_i(L))^2. \tag{15}$$

To be more specific, the iterative gradient descent formulation for updating each specific weight $w_{ij}(l)$ given a training pattern pair can be written as

$$w_{ij}(l) \Longleftarrow w_{ij}(l) - \eta\frac{\partial E}{\partial w_{ij}(l)} \tag{16}$$

where $(\partial E)/(\partial w_{ij}(l))$ can be computed effectively through a numerical chain rule by back propagating the error signal from the output layer to the input layer.

c) Discriminative learning of MLP: The discriminative learning of an MLP distinguishes itself from the traditional back-propagation learning by adopting a different cost function. The presence of the discriminative cost function has a profound impact on the learning capability and performance of the network and usually results in better performance. The discriminative learning [60], [62] was proposed specifically for pattern-recognition problems, aiming at achieving a minimum classification error rate. Based on a given set of training samples, the objective criterion is defined by the classification rule in a functional form and is

optimized by numerical search algorithms. Under the back-propagation learning framework for an M-class recognition task, the discriminant functions $\{y_i(\mathbf{x}; \mathbf{W}) = a_i(L), i = 1, 2, \cdots, M\}$, which are the neural network outputs and indicate the classification posterior probabilities $P(i \mid \mathbf{x})$ [123], are first calculated, where $\mathbf{W}$ denotes the parameter set of the classifier (i.e., the feed-forward network weights $\{w_{ij}(l)\}$ in the lth layer) and the training sample $\mathbf{x}$ is known to belong to one of M classes. For each input $\mathbf{x}$, the classifier makes its decision by choosing the largest of the discriminant functions evaluated on $\mathbf{x}$. A misclassification measure for this data is then defined as follows:

$$d_i(\mathbf{x}) = -y_i(\mathbf{x}; \mathbf{W}) + \left[\frac{1}{M-1} \sum_{j, j \neq i}^{M} y_i(\mathbf{x}; \mathbf{W})^{\alpha} \right]^{\frac{1}{\alpha}} \quad (17)$$

where α is a positive number.

Finally, the minimum error objective is formulated as a differentiable function of the misclassification measure. More specifically, the error objective function E_k of the kth class is defined as

$$E_k(\mathbf{x}; \mathbf{W}) = E_k(d_k(\mathbf{x})) = \frac{1}{1 + e^{-\tau d_k}}, \quad \tau > 0. \quad (18)$$

Note that a positive $d_k(\mathbf{x})$ leads to a penalty, which is a count of classification error, while a negative $d_k(\mathbf{x})$ implies a correct classification.

Once the discriminative cost function is defined, the learning process can be carried out in a straightforward manner, i.e., for each training datum belonging to the kth class

$$w_{ij}(l) \Longleftarrow w_{ij}(l) - \eta \frac{\partial E_k}{\partial w_{ij}(l)}. \quad (19)$$

Due to the popularity of MLP, it is not possible to exhaust all the numerous IMP applications using MLP's. For examples, Sung and Poggio [134] used MLP for face detection, and Huang [51] used it as preliminary channels in an overall fusion network. More details about using MLP's for multimodal signal will be discussed in the audio/visual processing section.

2) RBF and One Class in One Network (OCON): Another type of feed-forward network is the RBF network. Each neuron in the hidden layer employs an RBF (e.g., a Gaussian kernel) to serve as the activation function. The weighting parameters in the RBF network are the centers, widths, and heights of these kernels. The output functions are the linear combination (weighted by the heights of the kernels) of these RBF's. It has been shown that the RBF network has the same asymptotic approximation power as an MLP [116].

The conventional MLP adopts an all-class-in-one-network (ACON) structure, where all the classes are lumped into one supernetwork. The supernet has the burden of having to simultaneously satisfy all the teachers, so the number of hidden units tends to be large. Empirical results confirm that the convergence rate of ACON degrades drastically with respect to the network size because the training of hidden units is influenced by (potentially conflicting) signals from different teachers [66].

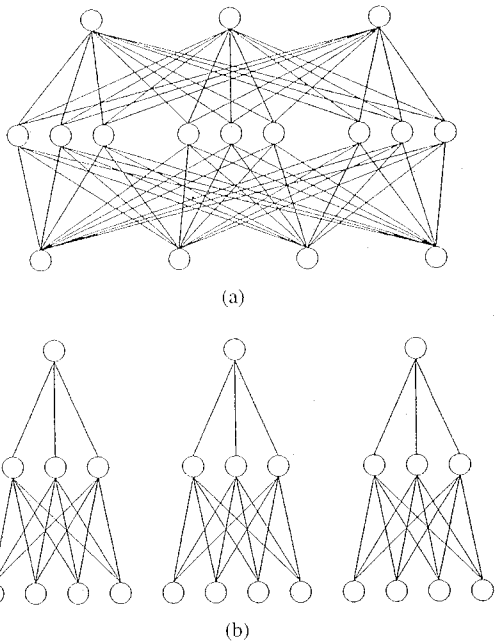


Fig. 1. (a) An ACON structure. (b) An OCON structure.

In contrast, it is natural for the RBF to adopt another type of the network structure, i.e., the one-class-in-one-network (OCON) structure, where one subnet is designated to one class only. The difference between these two structures is depicted in Fig. 1. Each subnet in the OCON network specializes in distinguishing its own class from the others, so the number of hidden units is usually small. In addition, OCON structures have the following features.

- Locally, unsupervised learning may be applied to determine the initial weights for individual subnets. The initial clusters can be trained by VQ or K-mean clustering techniques. If the cluster probabilities are desired, the EM algorithm can be applied to achieve maximum likelihood estimation for each *class conditional likelihood density.*
- The OCON structure is suitable for *incremental training,* i.e., network upgrading upon adding/removing memberships [66], [68].
- The OCON network structure supports the notion of distributed processing. It is appealing to smart-card biometric systems. An OCON-type classifier can store personal discriminant codes in individual class subnets, so the magnet strip in the card only needs to store the network parameters in the subnet that has been designated to the card holder.

a) Application examples: In [13], Brunelli and Poggio proposed a special type of RBF network called the "HyperBF" network for successful face-recognition applications. In [87], the associated audio information is exploited for video scene classification. Several audio fea-

tures have been found to be effective in distinguishing audio characteristics of different scene classes. Based on these features, a neural-net classifier can successfully separate audio clips from different television programs.

3) Linear Fusion Network: A decision-based neural network (DBNN) [68] has two variants: one hard-decision model and a probabilistic model. A DBNN has a modular OCON network structure: one subnet is designated to represent one object class. For multiclass classification problems, the outputs of the subnets (the *discriminant functions*) will compete with each other, and the subnet with the largest output values will claim the identity of the input pattern.

a) Decision-based learning rule: The learning scheme of the DBNN is decoupled into two phases: *locally unsupervised* and *globally supervised* learning. The purpose is to simplify the difficult estimation problem by dividing it into several localized subproblems. Thereafter, the fine-tuning process would involve minimal resources.

- *Locally unsupervised learning: VQ or EM clustering method*: Several approaches can be used to estimate the number of hidden nodes, or the initial clustering can be determined based on VQ or EM clustering methods.
  - — In the hard-decision DBNN, the VQ-type clustering (e.g., K-mean) algorithm can be applied to obtain initial locations of the centroids.
  - — For the probabilistic (P)DBNN, the EM algorithm can be applied to achieve maximum likelihood estimation for each class conditional likelihood density. (Note that once the likelihood densities are available, the posterior probabilities can be easily obtained).
- *Globally supervised learning*: Based on this initial condition, the decision-based learning rule can be applied to further fine-tune the decision boundaries. In the second phase of the DBNN learning scheme, the objective of the learning process changes from maximum likelihood estimation to *minimum classification error*. Interclass mutual information is used to fine-tune the decision boundaries (i.e., the *globally supervised* learning). In this phase, DBNN applies the reinforced-antireinforced learning rule [68], or discriminative learning rule [68], to adjust network parameters. Only *misclassified patterns* are involved in this training phase.

b) Reinforced-antireinforced learning rules: Suppose that the mth training pattern $\mathbf{x}^{(m)}$ is known to belong to class Ω_i and that the leading challenger is denoted $j = \arg\max_{j \neq i} \phi(\mathbf{x}^{(m)}, \mathbf{w}_j)$. The learning rule is

Reinforced Learning:

$$\mathbf{w}_i^{(m+1)} = \mathbf{w}_i^{(m)} + \eta \nabla \phi(\mathbf{x}^{(m)}, \mathbf{w}_i)$$

Antireinforced Learning:

$$\mathbf{w}_j^{(m+1)} = \mathbf{w}_j^{(m)} - \eta \nabla \phi(\mathbf{x}^{(m)}, \mathbf{w}_j).$$

c) Application examples: DBNN is an efficient neural network for many pattern-classification problems, for example, OCR and texture classification [66] and face- and palm-recognition problems [79], [84].

4) Mixture of Experts (MOE): MOE learning [56] has been shown to provide better performance due to its ability to solve a large, complicated task effectively by smaller and modularized trainable networks (i.e., experts), whose solutions are dynamically integrated into a coherent one using the trainable gating network. For a given input $\mathbf{x}$, the posterior probability of generating class $\mathbf{y}$ given $\mathbf{x}$ using K experts is computed by

$$P(\mathbf{y} \mid \mathbf{x}, \mathbf{v}\{\theta_i\}) = \sum_{i=1}^{K} g_i(\mathbf{v}) P(\mathbf{y} \mid \mathbf{x}, \theta_i) \tag{20}$$

where $\mathbf{y}$ is a binary vector. For example, if we consider two classes for a classification problem, then $\mathbf{y}$ is [1 0] and [0 1]. We also define ϕ to be a combined parameter vector $[v, \{\theta_i\}]$:, where v parameter vector determines the gating network's probabilistic weighting outputs $\{g_i\}$ for all the expert network outputs and θ_i is the parameter vector for the ith expert network ($i = 1, \cdots, K$), which generates the output $P(\mathbf{y} \mid \mathbf{x}, \theta_i)$.

The gating network can be a nonlinear neural network (for example, an MLP) or linear neural network (for example, a linear perceptron). To obtain the linear gating network output, the softmax function is utilized [12]

$$g_i = \exp(b_i) \Big/ \sum_{j=1}^{K} \exp(b_j) \tag{21}$$

where $b_i = v_i \mathbf{x}^T \cdot \{v_i\}$ denotes the weights of the ith neuron of the gating network when the gating network is a linear perceptron.

The learning algorithm for the MOE is based on the maximum likelihood principle to estimate the parameters (i.e., choose parameters for which the probability of the training set given the parameters is largest). The gradient ascent algorithm can be used to estimate the parameters.

Assume that the training data set is $\{\mathbf{x}^{(t)}, \mathbf{y}^{(t)}\}$, $t = 1, \cdots, N$. First, we take the logarithm of the product of N densities of $P(\mathbf{y} \mid \mathbf{x}, \phi)$

$$l(\mathbf{y}, \mathbf{x}, \phi) = \sum_t \sum_i \log \Big[g_i^{(t)} P(\mathbf{y}^{(t)} \mid \mathbf{x}^{(t)}, \theta_i) \Big]. \tag{22}$$

Next, we maximize the log likelihood by gradient ascent. The learning rule for the weight vector v_i in a linear gating network is obtained as follows [see (21) and (22)]:

$$\Delta v_i = -\rho \frac{\partial l(\mathbf{y}, \mathbf{x}, \phi)}{\partial v_i} = \rho \sum_t \Big(h_i^{(t)} - a_i^{(t)} \Big) \mathbf{x}^{(t)T} \tag{23}$$

where ρ is a learning rate and

$$h_i = \frac{g_i P(\mathbf{y} \mid \mathbf{x}, \theta_i)}{\sum_j (g_j P(\mathbf{y} \mid \mathbf{x}, \theta_j)}.$$

The MOE [56] is a modular architecture in which the outputs of a number of "experts," each performing a

classification task in a particular portion of the input space, are combined in a probabilistic way by a "gating" network, which models the probability that each portion of the input space generates the final network output. Each local expert network performs multiway classification over K classes by using either K independent binomial models, each modeling only one class, or one multinomial model for all classes. The MOE gives two advantages over traditional nonlinear function approximators such as the MLP: 1) a statistical understanding of the operation of the predictor and 2) provision of information about the performance of the predictor in the form of likelihood information and local error bars.

MOE has an explicit relationship with statistical pattern-classification methods. Given a pattern, each expert network estimates the pattern's conditional *a posteriori* probability on local areas, and the outputs of the gating network represent the probabilities that its corresponding expert subnet produces the correct answer. The final output is the weighted sum of the estimated probabilities from all the expert networks.

MOE and PDBNN have a lot of similarities. Both are based on the principle of "divide and conquer," in which a large, difficult problem is broken into many smaller, more tractable problems. Moreover, both use the EM algorithm in their learning schemes. However, there are substantial differences too. Each expert network in the MOE estimates the conditional posterior probabilities for all the pattern classes. The output of a local expert is ready to make classification decision for a particular local area. This characteristic suggests that the *interclass communication* in the MOE exists even down to the local network level. As to the PDBNN, each neuron estimates the *class* conditional likelihood density. Therefore, the classification decision cannot be made until the final subnet output is formed. This delay in decision making implies that there is no interclass communication in the PDBNN until the final level. MOE can be extended to a multilayer tree structure, which is known as hierarchical mixture of experts [59].

a) Application example: The MOE model was applied to time-series analysis with well-understood temporal dynamics. It produced significantly better results than single networks. Furthermore, it discovered the regimes correctly. It allows the users to characterize the subprocesses through their variances and avoids overfitting in the training process [92]. A Bayesian framework for inferring the parameters of an MOE model based on ensemble learning by variational free energy minimization was successfully applied to sunspot time-series prediction [144]. Integrating pretrained expert networks with constant sensitivity, which is defined as the percentage of abnormal objects being correctly classified as abnormal, into an MOE configuration enables each trained expert to be responsive to specific subregions of the input spaces with minimum ambiguity and thus produces better performance in automated cytology screening applications [54]. By applying a likelihood splitting criterion to each expert in the hierarchical mixture of experts (HME), Waterhouse and Robinson [143] first grew the HME tree adaptively during training. Then, by considering only the most probable path through the tree, they pruned branches away, either temporarily or permanently in case of redundancy. This improved HME showed significant speed-ups and more efficient use of parameters over the standard fixed HME structure in discriminating for artificial applications as well as prediction of parameterized speech over short time segments [145]. The HME architecture has also been applied to text-dependent speaker identification [22].

C. Neural Networks for Multimodal Media Fusion

In many multimedia applications, it is useful to have a versatile multimedia fusion network, where sensor information is laterally combined to yield improved classification. Neural networks offer a natural solution for sensor or media fusion. This is because of their capability for nonlinear and nonparametric estimation in the absence of complete knowledge on the underlying models or sensor noises.

The problem of combining the classification power of several classifiers is of great importance to various applications. First, for several recognition problems, numerous types of media could be used to represent and recognize patterns. In addition, for those applications that deal with high-dimensional feature data, it makes sense to divide a feature vector into several lower dimensional vectors before integrating them for final decision (i.e., divide and conquer).

The outputs of the NN classifiers represent class memberships. To combine the information present in individual channels efficiently, we need to introduce another layer of network called the *fusion layer*. The parameters of the fusion layer can be adaptively updated by the outputs of the individual channel classifiers. Several types of fusion layers have been proposed [1].

- *Linear Fusion:* Information fusion may be based on linear combination of outputs weighted by some proper confidence parameters. It is largely motivated by the following statistical and computational reasons.
  - — It can make use of the popular Bayesian formulation.
  - — It can facilitate adoption of EM training of the confidence parameters.
- *Nonlinear Fusion:* In general, a fusion scheme that nonlinearly combines decisions from the participating channels could be adopted.

1) Linear Fusion Network: The multiclassifier DBNN consists of several "classifier channels," each of which receives an input vector from different media separately.

In [77], two channel fusion schemes based on the probabilistic DBNN are proposed. Following a Bayesian formulation, the outputs of channels are linearly combined by some confidence weightings [denoted as $P(C \mid \omega)$ in Fig. 2(a)]. The weighting factor is assigned based on the *confidence* the

corresponding channel has on its recognition result. Since DBNN generates probabilistic outputs, it is natural to design the channel weightings to have probability properties. The overall configuration of a multichannel fusion network is depicted in Fig. 2, where the score functions from two channels are combined after some proper preweightings.

- The *class-dependent channel fusion scheme* deploys one PDBNN for each sensor channel. Each PDBNN receives only the patterns from its corresponding sensor. The *class* and *channel* conditional likelihood densities ($p(\mathrm{x} \mid \omega_i, C_j)$) are estimated. The outputs from different channels are combined together in the weighted sum fashion. The *weighting parameters* $P(C_j \mid \omega_i)$ represents the confidence of the channel C_j producing the correct answer for the object class ω_i. $P(C_j \mid \omega_i)$ can be trained by the EM algorithm, and its value remains *constant* during the identification process (remember that the values of the weighting parameters in the HME are functions of the input pattern). Fig. 2(a) illustrates the structure of the class-dependent channel fusion scheme. The scheme considers the data distribution as the mixture of the likelihood densities from various sensor channels. This is a simplified density model.

  If the feature dimension is very large and the number of training examples is relatively small, the direct estimation approaches can hardly obtain good performance due to the curse of dimensionality. For this kind of problem, since the class-dependent fusion scheme greatly reduces the number of parameters, it could achieve better estimation results.

- Another fusion scheme is *data-dependent channel fusion*. Fig. 2(b) shows the structure of this scheme. Like the class-dependent fusion method, each sensor channel has a PDBNN classifier. The outputs of the PDBNN's are transformed to the posterior probabilities by softmax functions. In this fusion scheme, the channel weighting $P(C_j \mid \mathrm{x})$ is a function of input pattern x. Therefore, the importance of the individual channel may vary if the input pattern is different.

In the class-dependent channel fusion scheme [see Fig. 2(a)], the weighting factors correspond to the confidence $P(C_k \mid \omega_i)$ for classifier channel C_k. Here, $P(C_k \mid \omega_i)$ represents the indicator on the confidence on channel k when the test pattern is originated from the ω_i class. (By definition, $\sum_{k=1}^{K} P(C_k \mid \omega_i) = 1$, so it has the property of a probability function). Suppose that there are K channels in the subnet ω_i. The probability model of the PDBNN-based channel fusion network can be described as follows:

$$p(\mathrm{x}(t) \mid \omega_i) = \sum_{k=1}^{K} P(C_k \mid \omega_i) p(\mathrm{x}(t) \mid \omega_i, C_k)$$

where $p(\mathrm{x}(t) \mid \omega_i, C_k)$ is the discriminant function of subnet i in channel k, and $p(\mathrm{x}(t) \mid \omega_i)$ is the combined discriminant function for class ω_i. The channel confidence

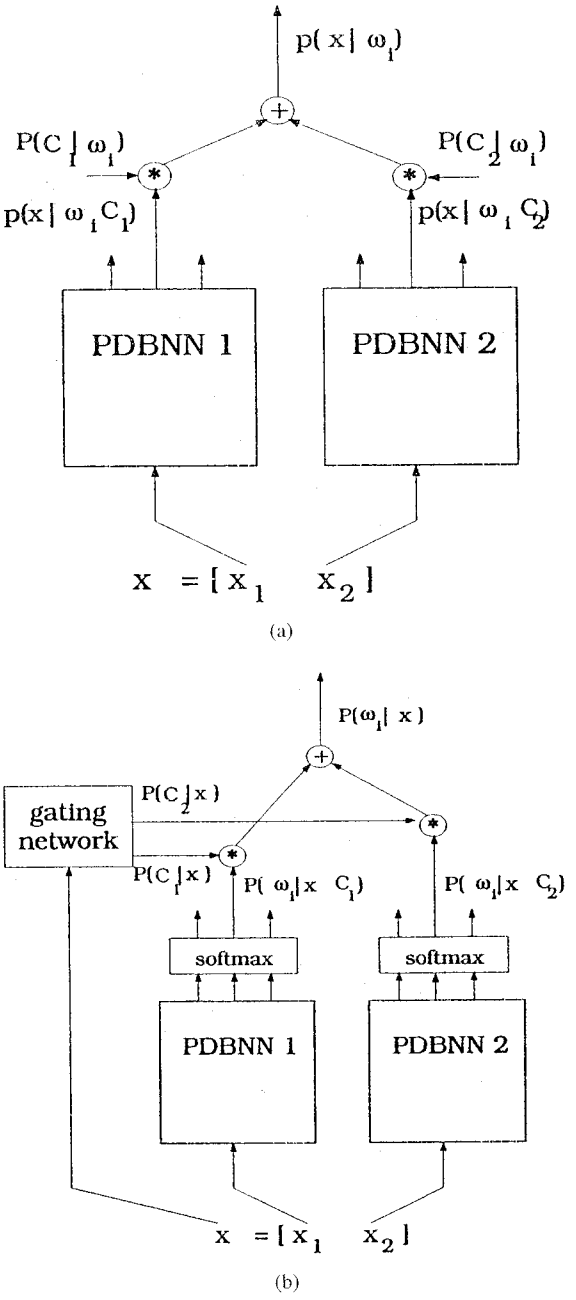


Fig. 2. A media fusion network: linear fusion of probabilistic DBNN classifiers. (a) For the applications where there are several sensor sources, a class-dependent channel fusion scheme can be applied for classification. $P(C_j \mid \omega_i)$ is a trainable parameter. Its value is fixed during the retrieving time. (b) Data-dependent channel fusion scheme. In this scheme, the channel weighting parameters are functions of the input pattern x ($P(C_j \mid \mathrm{x})$).

$P(C_k \mid \omega_i)$ can be learned by the following. Define $\alpha_k = P(C_k \mid \omega_i)$. At the beginning, assign $\alpha_k = 1/K, \forall k = 1, \cdots, K$. At step j

$$h_k^{(j)}(t) = \frac{\alpha_k^{(j)} p(\mathrm{x}(t) \mid \omega_i, C_k)}{\sum_l \alpha_l^{(j)} p(\mathrm{x}(t) \mid \omega_i, C_l)}$$
$$\alpha_k^{(j+1)} = \frac{1}{N} \sum_{t=1}^{N} h_k^{(j)}(t). \tag{24}$$

Once the NN is trained, then the fusion weights will remain constant during the retrieving phase.

In the data-dependent channel fusion [see Fig. 2(b)], instead of using the likelihood of observing $\mathbf{x}(t)$ given a class ($p(\mathbf{x}(t) \mid \omega_i, C_k)$) to model the discriminant function of each cluster, we shall use the posterior probabilities of electing a class given $\mathbf{x}(t)$ ($p(\omega_i \mid \mathbf{x}(t), C_k)$). For this version of a multichannel network, a new confidence $P(C_k \mid \mathbf{x}(t))$ is assigned, which stands for the confidence we have on channel k when the input pattern is $\mathbf{x}(t)$. Accordingly, the probability model is also modified to become

$$P(\omega_i \mid \mathbf{x}(t)) = \sum_{k=1}^{K} P(C_k \mid \mathbf{x}(t)) P(\omega_i \mid \mathbf{x}(t), C_k)$$

where $P(\omega_i \mid \mathbf{x}(t), C_k) = P(\omega_i \mid C_k) p(\mathbf{x}(t) \mid \omega_i, C_k) / p(\mathbf{x}(t) \mid C_k)$, and the confidence $P(C_k \mid \mathbf{x})$ can be obtained by the following equations:

$$P(C_k \mid \mathbf{x}(t)) = \frac{P(C_k) p(\mathbf{x} \mid C_k)}{\sum_l P(C_l) p(\mathbf{x}(t) \mid C_l)}$$

where $p(\mathbf{x}(t) \mid C_k)$ can be computed as $p(\mathbf{x}(t) \mid C_k) = \sum_i P(\omega_i \mid C_k) p(\mathbf{x}(t) \mid \omega_i, C_k)$ and $P(C_k)$ can be learned by (24) (but replace $p(\mathbf{x}(t) \mid \omega_i, C_k)$ with $p(\mathbf{x}(t) \mid C_k)$). The term $P(C_k)$ can be interpreted as "the general confidence" we have on channel k. Unlike the class-dependent approach, the fusion weights need to be computed for each testing pattern during the retrieving phase. Notice that this data-dependent fusion scheme can be considered as the combination of PDBNN and MOE [56].

a) Application example: The class-dependent channel fusion scheme has been observed to have very good classification performance on vehicle- and face-recognition problems [77]. The experiment used six car models from different view angles to create the training and testing data base. Around 30 images (each with size 256×256 pixels) were taken for each car model from various viewing directions. There are in total 172 examples in the data set. Two classifier channels were built from two different feature-extraction methods: one uses intensity information and the other edge information. The fusion of two channels (with 94% and 85% recognition rate each) can yield a near perfect.

The fusion model was compared with a single network classifier. The input vectors of these two networks were formed by cascading the intensity vector with the edge vector. Therefore, the input vector dimension becomes $144 \times 2 = 288$. The network is the RBF-typed DBNN. The experimental result shows that the performance is worse than the fusion network (about 95.5% recognition rate).

2) Nonlinear Fusion Network: Neural nets offer a natural approach to nonlinear information fusion. Huang *et al.* [51] proposed a sensor fusion technique that makes use of a new neural model to combine data autonomously extracted from different sources. The fusion is based on a "cooperative/competitive" approach for training a layer of the generalized McCulloch–Pitts neurons. Ideally, one output of each MLP classifier will be one and others will be zero. So the weights are trained to simultaneously excite one output neuron and modulate the activity of others. The learning procedure adjusts the weights so as to adjust the contribution of the information from each sensor to the final decision based on training data. When the modulating input dominates, the sigmoidal function is effectively flattened; otherwise, the modulating input acts to increase the nonlinearity of the sigmoidal function (thus enhancing its sensitivity to excitatory input).

Once trained, the outputs of a particular MLP provide estimates of the class membership of new patterns presented at the input to the system. Fusion of the information represented by these outputs is accomplished with a layer of modulated neurons. The activity of each of these neurons is dependent upon both excitation and modulation signals derived from the outputs of the MLP classifiers.

D. Temporal Models for Multimodal Conversion and Synchronization

The class of neural networks that are most suitable for applications in multimodal conversion and synchronization is made up of the so-called *temporal* neural networks. Unlike the feed-forward type of artificial neural networks, temporal networks allow connections both ways between a pair of neuron units, and sometimes feedback connections from a unit to itself. Let us elaborate further on this difference. From the perspective of connection patterns, neural networks can be grouped into two categories: *feed-forward* networks, in which graphs have no loops, and *recurrent* networks, where loops occur because of feedback connections. Feed-forward networks are *static*, that is, a given input can produce only one set of output values rather than a sequence of data. Thus, they carry no memory. In contrast, many *temporal* neural networks employ some kind of *recurrent* network structure. Such an architectural attribute enables the information to be temporally memorized in the networks without being washed away at the presentation of next data.

A simple extension to the existing feed-forward structure to deal with temporal sequence data is the *partially recurrent network* [sometimes called a *simple recurrent network* (SRN)]. An SRN has mainly feed-forward connections, enhanced with a carefully chosen set of feedback connections. In most cases, the feedback connections are prefixed and not trainable. Thanks to recurrence, it can remember cues from the past and yet does not appreciably complicate the training procedure. The most widely used SRN's are Elman's network and Jordan's network [38], [58]. The time-delay neural network (TDNN) is a further extension to cope with the shift-invariance properties required in speech recognition. It is achieved by making time-shifted copies of the hidden units and linking their corresponding weights to the output layer [141]. Several fully recurrent NN architectures with the corresponding learning algorithms

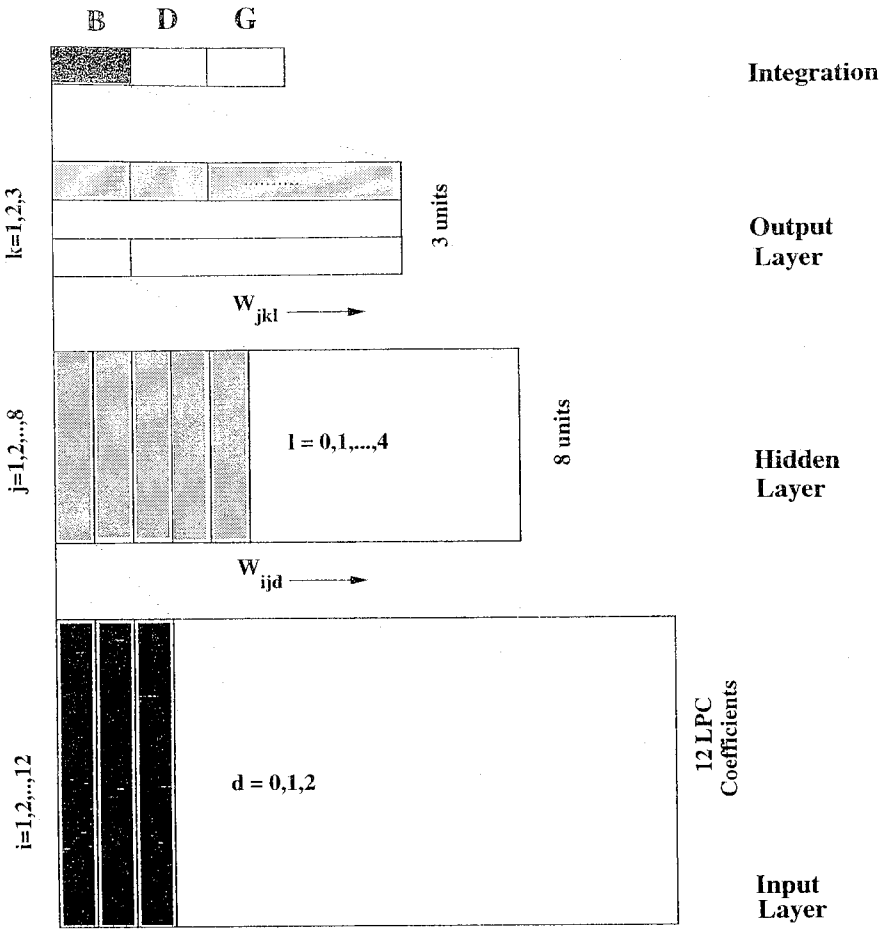


Fig. 3. The architecture of a TDNN.

are available, such as real-time recurrent learning (RTRL) networks [148] and back-propagation through time (BPTT) networks [50], [126]. The computational requirement of these and several variants is very high. Among all the recurrent networks, BPTT's performance is best unless on-line learning is required, in which case RTRL is needed instead. But for many applications involving temporal sequence data, an SRN or a TDNN may suffice and is much less costly than RTRL or BPTT.

A hidden Markov model (HMM) [117]–[119] is a doubly stochastic process with an underlying stochastic process that is not observable (i.e., hidden) but can only be observed through another set of stochastic processes that produce the sequence of observed symbols [118]. The trellis diagram realization of an HMM can be considered as a BPTT network expanding in time since its connections (transition probabilities) are carrying the information about the environment and it consists of a multilayer network of simple units activated by the weighted sum of the unit activations at the previous iteration. In addition, the learning technique used in HMM's has a close algorithmic analogy with that used in the BPTT networks [52].

1) TDNN: Fig. 3 shows the TDNN architecture [141] for a three-class temporal sequence recognition task. A TDNN is basically a feed-forward multilayer (four layers) NN with time-delay connections at the hidden layer to capture varying amounts of contexts. The basic unit in each layer computes the weighted sum of its inputs and then passes this sum through a nonlinear sigmoid function to the higher layer. The TDNN classifier shown in Fig. 3 has an input layer with 16 units, a hidden layer with eight units, and an output layer with three units (one output unit represents one class).

When a TDNN is used for speech recognition, the speech utterance is partitioned frame by frame (e.g., 30 ms frame with 15 ms advance). Each frame is transformed to 16 coefficients serving as input to the network. Every three frames with time delay zero, one, and two are input to the eight time-delay hidden units, i.e., each neuron in the first hidden layer now receives input (via 3×16 weighted connections) from the coefficients in the three-frame window. The eight-unit hidden layer is delayed five times to form a 40-unit layer. At the second hidden layer, each unit looks at all five copies of the delayed eight-unit hidden blocks of the first hidden layer. Last, the output is obtained by integrating the evidence from the second hidden layer over time and connecting it to its output unit. This procedure can be formalized as the following equation:

$$y_{\text{class}} = \frac{1}{T-6} \sum_{t=7}^{T} b_{\text{class}}^{(t)} \tag{25}$$

$$b_c^{(t)} = S\left(\sum_{j=0}^{7}\sum_{l=0}^{4} w_{cjl}^{H} S\left[\sum_{i=0}^{15}\sum_{d=0}^{2} w_{ijd}^{I} x_i^{(t-l-d)} + \theta_j^{I}\right] + \theta_c^{H}\right) \tag{26}$$

where T is the total number of frames, $\mathbf{x}$ is the input, $\{b_c^{(t)}\}$ are the outputs of the c class at the second layer at different time instances, and $S(\cdot)$ is the sigmoid function. The network structure forces the input layer to be shift invariant, so that the absolute time of the event is not important.

Like an MLP, a TDNN is also trained by the back-propagation learning rule [141]. Suppose the input to TDNN is a vector $\mathbf{x}$. Then the updating of the weights $\mathbf{w}$ can be described as

$$\mathbf{w} \Leftarrow\Leftarrow \mathbf{w} - \eta \frac{\partial E}{\partial \mathbf{w}}$$

where

$$E = E(\{\mathbf{w}\}, \{\mathbf{x}\}) = \frac{1}{2}\sum_{c=1}^{C}(\mathbf{t}_c - \mathbf{y}_c(\mathbf{x}))^2.$$

Therefore, by training, the local short-duration features in a speech signal can be formed at the lower layer and more complex longer duration features formed at the higher layer. The learning procedure ensures that each of the units in each layer has its weights adjusted in a way that improves the network's overall performance [53].

After the TDNN has learned its internal representation, it performs recognition by passing input speech over the TDNN neurons and selecting the class that has the highest output value. Section III-C2 will present an example employing such a TDNN model to audio/visual synchronization in the lip-reading application.

2) HMM's: The basic theory of Markov chains has been known to mathematicians and engineers for nearly 80 years, but it is only in the past decade [5], [6] that it has become the predominant approach to speech recognition. It is beyond the scope of this paper to describe the HMM's in full detail. Instead, a brief outline will be given, and more details can be found in [75] and [117]–[119].

a) Model descriptions: An HMM is a doubly stochastic process with an underlying stochastic process that is not observable (that is why it is called *hidden*) but can only be observed through another set of stochastic processes that produce a sequence of observed symbols. An HMM can be thought as a collection of states connected by transitions. Three sets of model parameters for HMM are transition probabilities $\mathbf{A} = \{a_{ij}\}$, output observation probabilities $\mathbf{B} = \{b_j(\mathbf{o}_t)\}$, and the initial state probabilities $\pi = \{\pi_j\}$. Given a sequence of temporal observations $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T\}$, the typical problems associated with HMM's are the following.

1) Given $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T\}$ and the model $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, how to compute $\Pr(\mathbf{O} \mid \lambda)$, i.e., the probability that the observation sequence was produced by model λ. This evaluation problem is solved by the *forward-backward algorithm*. In an isolated word-recognition task, each isolated word has its own HMM model. An unknown speech pattern is recognized as the word whose model gives the highest probability $\Pr(\mathbf{O} \mid \lambda)$.
2) Given $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T\}$ and the model $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, how to choose a best state sequence $\mathbf{I} = \{i_1, i_2, \cdots, i_T\}$, which is optimal in some meaningful sense. The solution to this problem, e.g., Viterbi algorithm, can be extended to solve continuous speech-recognition tasks where the path search is among all isolated words.
3) Given $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T\}$, how to adjust the model parameters $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ to maximize $\Pr(\mathbf{O} \mid \lambda)$. This is called HMM training. The Baum–Welch reestimation algorithm is the most popular HMM learning method.

According to the representation of the observation probabilities $\{b_j(\mathbf{o}_t)\}$, HMM's fall into two major categories: *discrete-density HMM's* and *continuous-density HMM's*. In a discrete HMM, VQ is used to represent each frame of speech as a symbol from a finite set. On the other hand, in continuous HMM's, observations are treated as continuous multidimensional vectors. For example, with a Gaussian mixture density [117], an observation probability is described by the mean vectors, the covariance matrices, and the mixture weights of several Gaussians.

The HMM approach provides a framework that includes an efficient decoding method for use in recognition (the *forward* algorithm and the *Viterbi* algorithm) and an automatic supervised training method (the *forward-backward* and the *Baum–Welch* algorithm). Since HMM theory does not specify the structure of implementation hardware, and current HMM algorithms require very high precision and large amounts of memory (particularly in speaker-independent, large-vocabulary, continuous-speech recognition tasks), it is necessary to compare it to the well-known neural-network models, which offer the potential of providing massive parallelism and user adaptation with low precision implementations.

Last, we note that when HMM is applied to classification, it always adopts the OCON structure depicted in Fig. 1. That means one HMM will be designated to one class, e.g., for digital numeric OCR application, ten HMM's would be used—one for each digital number. Furthermore, for speech-recognition applications, it is common to adopt a special left-to-right HMM, i.e., the state transition connections only flow in (left-to-right) unidirection. Section III-C1 will present an example applying such a left–right HMM to audio-visual conversion.

III. Neural Networks for IMP Applications

Neural networks have played a very important role in the development of multimedia application systems [21],

[55], [109], [142]. Their usefulness ranges from low-level preprocessing to high-level analysis and classification. A complete multimedia system consists of many of the following information-processing stages, for which neural processing offers an efficient and unified core technology.

- *Visualization, tracking, and segmentation:*
  - — Neural networks have been found useful for some visualization applications, such as optimal image display [71] and color constancy and induction [30]. See Section III-A1.
  - — Feature-based tracking is crucial to motion analysis and motion/shape reconstruction problems. Neural networks can be applied to motion tracking schemes for feature- and object-level tracking [22]. See Section III-A2.
  - — Segmentation is a critical task for both image and video processing. The object boundary detection can use a hierarchical technique by adopting pyramid representation of images for computation efficiency [18], [88]. Active contour (e.g., snake) can also take advantage of NN's adaptive learning capability for continuous and fast tracking the region of interest [25]. Both unsupervised and supervised neural networks may be adopted for object boundary detection methods based on a variety of cues, including motion, intensity, edge, color, and texture. Several application examples will be elaborated in Section III-A3.
- *Detection and recognition:*
  - — Neural networks can be applied to machine-learning and computer-vision problems with applications to detection and recognition of a specific object class. Examples are on-line OCR applications [14], [41], signature verification [4], currency recognition [136], and structure from motion [73].
  - — Neural networks can facilitate detection or recognition of high-level features such as human faces in pictures or a certain object shapes under inspection. See Section III-B1.
  - — Multimodality recognition and authentification will have useful applications to network security and access control. One example using a (nonlinear) fusion network for personal identification will be highlighted in Section III-B2.
- *Multimodal coding, conversion, and synchronization:*
  - — Multimodal coding, conversion, and synchronization will remain a challenging research task. Static MLP networks for multimodal facial-image coding driven by speech and phoneme was already studied in [99].
  - — A more recent application of a temporal model (HMM) to audio-visual conversion will be discussed in Section III-C1.
  - — A temporal NN model (e.g., TDNN) for multimodality synchronization, integrating audio and visual signal for lip reading, will be elaborated on in Section III-C2.
- *Video/image content indexing and browsing:* It is important to have technical capabilities to quickly access audio-visual objects, manipulate them, and present them in a highly flexible way. For video content selection, extracting and utilizing proper information content inherent in video clips may lead to efficient search schemes for many disciplines:
  - — object- and subject-based video indexing/data base;
  - — video skimming and browsing;
  - — content-based retrieval.

  Again, neural processing presents a promising approach for these tasks. Several neural-network image/video browser and data base systems will be highlighted in Section III-D.

A. Image and Video Visualization and Segmentation

The task of feature extraction is critical to search schemes, as an efficient representation of the information can facilitate many subsequent multimedia functionalities, such as feature- or object-based indexing and access. Efficient representation of multimedia data can be achieved by neural clustering mechanisms. The general objectives are 1) to extract the most salient features to make classification tasks easier and 2) to extract representation of media information needed at various levels of abstraction.

1) Neural Network for Optimal Visualization: For many image-processing applications (e.g., medical), a display that maximizes diagnostic information would be very desirable. Neural networks have been successfully applied for optimal visualization, so that information can be more noticeably displayed. Note that raw data may contain more bits than what can be displayed in an ordinary computer monitor. For example, a magnetic resonance image contains 12-bit data, while most monitors only have 8 bits. To map 12-bit data to an 8-bit display, the appearance of the image hinges upon a proper selection of window width/center, which is a typical representation of image dynamic range in the medical field. An NN-based system [71] is used to estimate the window width/center parameters for optimal display.

To reduce the input dimension of NN, a feature vector of an input image is first extracted via PCA transformations. Then a competitive layer (unsupervised) neural network is applied to label the feature vector into several (say, six) possible classes with their confidence measures. For each class, both nonlinear and linear adaptive estimators are used to best calibrate window width/center. A nonlinear

estimator, while very efficient in reaching local optimum, is vulnerable to drastic and very unreasonable failures. To alleviate such concern, a safety net is provided via a linear estimator. A final data fusion scheme outputs the optimal window width/center parameters by combining the results from all possible classes with appropriate weighting of the confidence measures. It is also worth mentioning that the total system is equipped with on-line training capability. For more details, see [71].

2) True Motion Tracking (TMT): Neural techniques for motion estimation have been under investigation. In [39], a motion-estimation algorithm based on the EM technique was proposed. First, the motion field is represented by a model characterized by a series of the motion coefficients. Smoothness of motion is imposed in the assumption. Then the EM-based iterative algorithm is adopted to estimate the image motion coefficients from noisy measurements.

In [22], a feature TMT for object-based motion tracking was proposed. Based on a neighborhood relaxation neural model, it can effectively find true motion vectors of the prominent features of an object. By prominent feature, we mean that a) any region of an object contains a good number of blocks, whose motion vectors exhibit certain consistency and b) only true motion vectors for a few blocks per region are needed. Therefore, at the outset, it would disqualify some reference blocks that are deemed unreliable to track. The method adopts a multicandidate prescreening to provide some robustness in selecting motion candidates. Furthermore, assuming that the true motion field is piecewise continuous, the method calculates the motion of a feature block after consulting all of its neighboring blocks' (tentative) motions. This precaution allows a singular and erroneous motion vector to be corrected by its surrounding motion vectors, (yielding an effect very much like median filtering). As demonstrated by a good number of MPEG-2 and MPEG-4 benchmark sequences, the tracking results exhibit very satisfactory accuracy and reliability. One "foreman" example is shown in Fig. 4. The tracker has also found useful application to motion-based video segmentation [22] (see Section III-A4).

3) Image Segmentation via Texture Classification:

a) Texture classification by unsupervised EM algorithm: The EM algorithm was applied to texture classification for image segmentation application in [115]. The classification is accomplished by the following procedure. For each pixel location, determine the texture class (or model) that would yield the highest conditional probability for the neighborhood of that pixel. The preliminary result of such a procedure usually has a "noisy" appearance. This problem may be alleviated by imposing some spatial homogeneity. More precisely, for a pixel to be assigned to a specific class, it must also at the same time have high conditional likelihood with respect to several neighborhoods. According to the experimental report [115], the total error rate is below 5% for segmenting a four-square-region image. Note, however, that this rate was accomplished via unsupervised learning, which may yield a higher error rate when compared with some supervised classifiers discussed subsequently. It is well known that substantial improvement may be gained by supervised learning when this option is available and utilized. It is also interesting to note that the overall error rate could be greatly reduced (to 1%) by applying a simple low-pass filter to perform the spatial averaging.

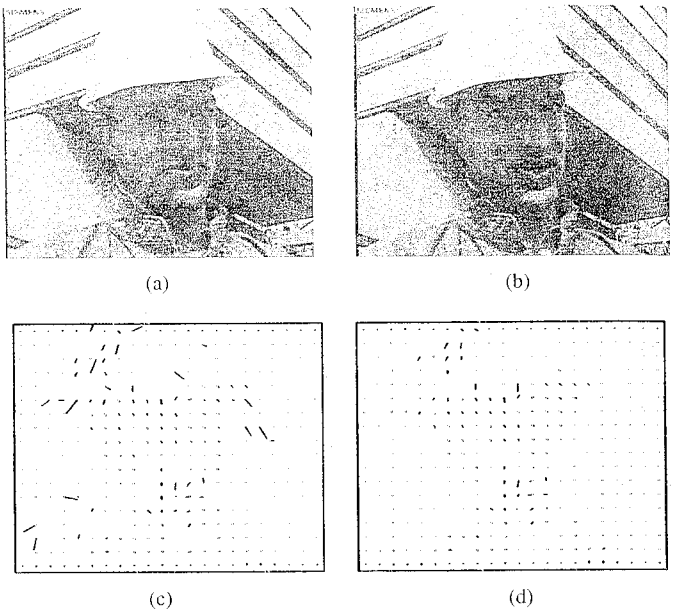


Fig. 4. (a), (b) Two frames of the "foreman" sequence. (c) Motion vectors found by the original full-search block-matching algorithm. (d) Motion vectors obtained by the neural method via neighborhood relaxation.

b) Texture classification by supervised neural models: In [137], a new texture feature, fuzzy texture spectrum, for texture classification was proposed. It is based on the relative gray levels between pixels. A vector of fuzzy values will be used to indicate the relationship of the gray levels among the neighboring pixels. The fuzzy texture spectrum can be considered as the distribution of the fuzzy differences between the neighboring pixels. The success of the texture classification of a given set of images hinges upon the designs of texture features and the classifiers. The feature used is an improved variant of the reduced texture spectrum. The feature appears to be less sensitive to the noise and the changing of the background brightness in texture images. Twelve Brodatz texture images were used in the simulations to show the effectiveness of the new texture feature. It was reported that with a DBNN classifier, the rate of classification error can be reduced to 0.2083%.

4) Motion-Based Video Segmentation via TMT, PCA, and EM: Robust scene segmentation is a prerequisite for object-based video processing. Various approaches to this complex task have been proposed, including classification of motion flow, color/texture-based segmentation, and dominant-motion extraction dividing a scene into moving and stationary regions. In [78], an object-based video-segmentation method combining all these approaches is

proposed for high-performance video compression. The object-oriented motion segmentation algorithm, which segments a video scene into different motion regions where each region can represent one independently moving object, is a candidate to be used in object-based (or even knowledge-based) video-processing schemes.

To get a more robust and accurate extraction of the moving objects from the video scene, initial motion clustering is performed upon a selected set of motion features of the associated feature blocks tracked by a TMT (see Section III-A2). The feature blocks are represented by the principal components (PC's) of their *position* and *velocity* (see Section II-A5). An example of video-scene segmentation containing two moving books under a moving camera is shown in Fig. 5. Fig. 5(a) shows the distribution of the feature blocks in the PC-coordinate. The unsupervised EM clustering scheme is then adopted to cluster the feature blocks. The results are shown in Fig. 5(b)–(d), corresponding to the background, left book, and right book, respectively. The motion parameters for each of the clustered feature blocks may be estimated and used as the initial condition for the final segmentation process.

The final step must involve the classification *f all* the blocks in the entire frame. The segmentation and the corresponding motion parameters are iteratively updated by a *model-based EM algorithm*. Briefly, the proposed model-based EM minimizes an energy function of the form (the blocks are labeled by b)

$$E(\mathbf{A}, h; V) = \sum_{b,r} h_r(b) s_r(b, A_r) + \sigma_T^2 \sum_{b,r} h_r(b) \log h_r(b) - \sigma_T^2 \sum_{b,r} h_r(b) \log \pi_r(b). \quad (27)$$

The first term represents the external (error) energy function, so that each cluster (say, the rth cluster) would be best fit to a given motion (say, affine) model denoted by A_r. The second term stands for the *entropy function,* which encourages a softer classification. The third term captures the *channel priors,* which allows the spatially neighboring blocks to have influence on the classification of the targeted block. This is the basis of the so-called multicue fusion, as it forces the classification to take into account the intensity/texture continuity (i.e., image cues), resulting in a smoother segmentation. Fig. 5(e)–(g) demonstrates that the three object regions are successfully extracted.

B. Personal Authentification and Recognition

Neural networks have been recognized as an established and mature tool for many pattern-classification problems. Particularly, they have been successfully applied to face-recognition applications. By combining face information with other biometric features such as speech, feature fusion should not only enhance accuracy but also provide some fault tolerance, i.e., it could tolerate temporary failure of one of the bimodal channels.

1) Face Detection and Recognition: For many visual monitoring and surveillance applications, it is important to determine human eye positions from an image or an image sequence containing a human face. Once the human eye positions are determined, all of other important facial features, such as positions of the nose and mouth, can easily be determined. The basic facial geometry information, such as the distance between two eyes, nose and mouth size, etc., can further be extracted. This geometry information can then be used for a variety of tasks, such as to recognize a face from a given face data base.

There are many successful neural-network examples for face detection and recognition. Brunelli and Poggio have adopted an RBF network for face recognition [13]. Pentland *et al.* [97], [113], [140] use eigenface subspace to determine the classes of the face patterns. Eigenface and Fisherface recognition algorithms were studied and compared in [7]. Cox *et al.* [31] proposed *mixture-distance* VQ network for face recognition and reached a 95% rate on a large (685 persons) data base. In [77] and [82], neural networks have been successfully applied to find such patterns with specific applications to detecting human faces and locating eyes in the faces.

2) Personal Authentification by Fusing Image and Speech: The fusion net has been applied to person recognition (see Fig. 6). (The diagram was originally from [51].) The system recognizes a person's identity by combining information from two (image and speech) channels.

- *Image channel:* Noisy face images, each containing a 64 × 64, 8-bit grayscale image, serve as one source of information for classification. The images were decomposed into 13 channels by four-level biorthogonal wavelet kernels.
- *Speech channel:* A noisy segment of speech, consisting of the spoken name of the same person. Speech segments, digitized at 8 kHz, were decomposed into eight channels using a length-eight wavelet kernel.

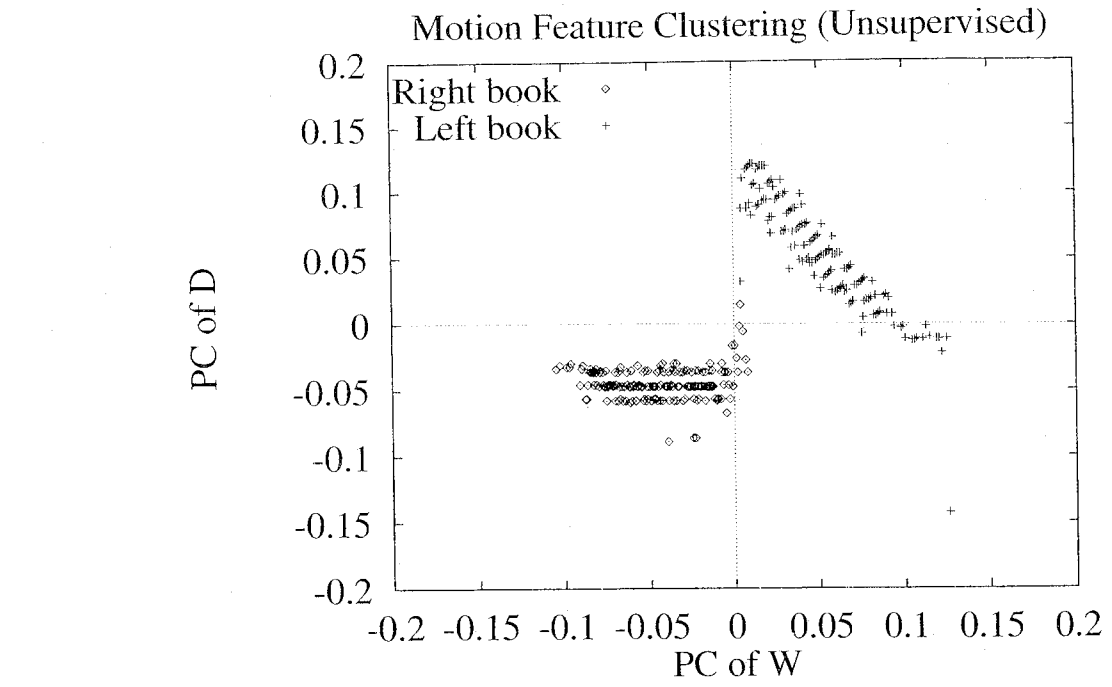
The combined use of two channels yields a performance that is much improved over using either channel alone [51].

C. Audio-to-Visual Conversion and Synchronization

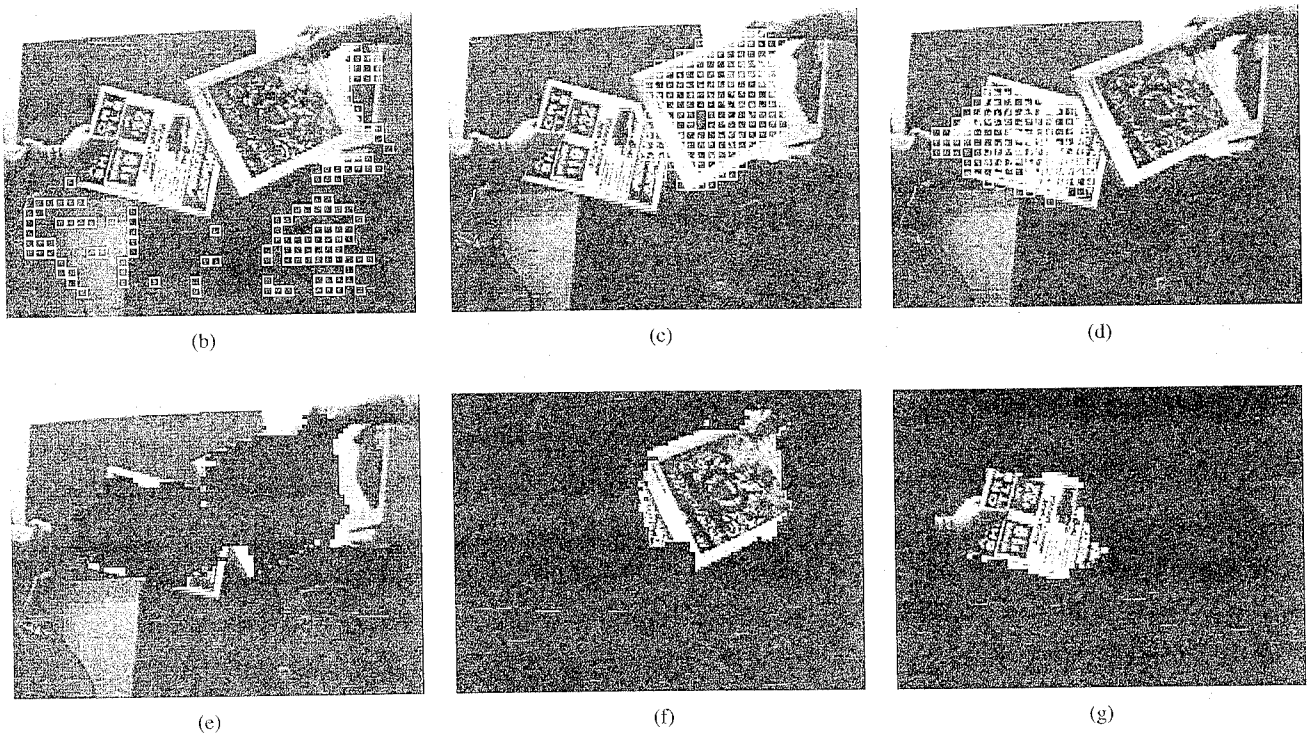
There exist already a few application examples applying temporal neural models to conversion and/or synchronization. Included in this subsection are one example using an HMM for audio-to-visual conversion and another using TDNN for lip-reading application.

1) HMM for Audio-to-Visual Conversion: Recent multimedia results exploit the audio-visual interaction, which includes speech-assisted lip synchronization and joint audio-video coding [23]. The goal of speech-driven facial animation is to synthesize realistic video sequences from acoustic speech.

In [23] and [121], this conversion process was accomplished with HMM's. The correlation between audio and video was exploited for speech-driven facial animation. One problem addressed is that frame skipping due to limited



(a)



(b) (c) (d)

(e) (f) (g)

Fig. 5. Motion-based video segmentation result of sequence with *2-Books* with *moving camera*. (a) PC distribution of the feature blocks obtained by a TMT. (b)–(d): Feature blocks clustered by EM into three clusters. (e)–(g): Final segmentation by a multicue, model-based EM.

bandwidth commonly introduces artifacts such as jerky motion and loss of lip synchronization in talking-head video. Therefore, lip synchronization becomes an important issue in video telephony and video conferencing. An important enabling technology was proposed for bimodal speech processing by mapping from audio speech [e.g., linear predictive coding (LPC) cepstra] to lip movements (visual parameters) [23]. This approach contains two stages. In the first stage, the acoustics must be classified into one of a number of groups. The second stage maps each acoustic group into a corresponding visual output.

A (typical) left–right audio-visual HMM was adopted. Consider estimating a single visual parameter $\mathbf{v}$ from the corresponding multidimensional acoustic parameter $\mathbf{a}$.

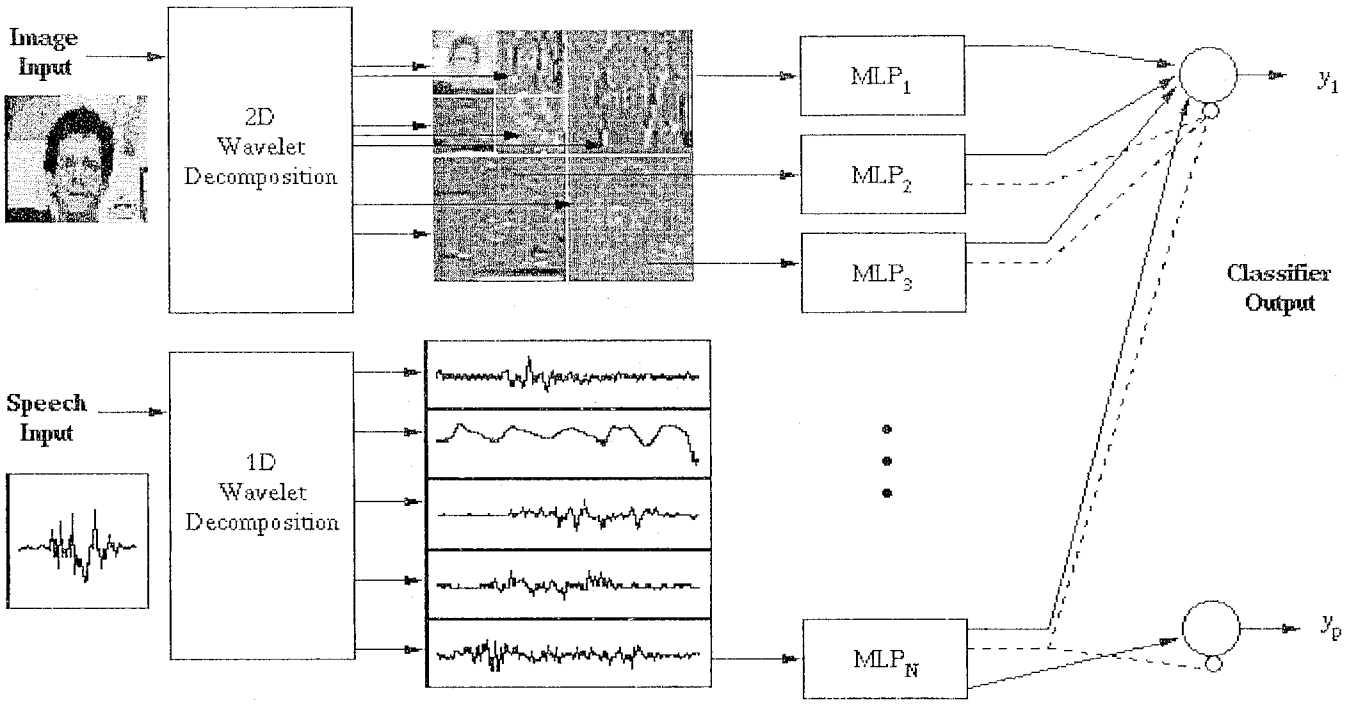


Fig. 6. The nonlinear fusion net, proposed by Huang *et al.*, is based on the McCulloch–Pitts neural net, where information from image and speech channels is combined.

Defining the combined observation to be $\mathbf{O} = [\mathbf{a}, \mathbf{v}]^T$, the audio-to-visual conversion process using HMM's can be treated as a "missing data" problem. More specifically, a continuous-density HMM was trained with a sequence of $\mathbf{O}$ for each word in the vocabulary. In the training phase, the Gaussian mixtures in each state of the HMM are modeling the joint distribution of the audio-visual parameters. When presented with a sequence of acoustic vectors that correspond to a particular word, conversion can be made by using the HMM to segment the sequence of acoustic parameters into the optimal state sequence using the Viterbi algorithm. More exactly, when presented with a sequence of acoustic vectors $\{\mathbf{a}\}$ that correspond to a particular word, the maximum likelihood estimator for the associated visual parameter vectors $\{\mathbf{v}\}$ is equal to the conditional expectation, which can be derived from the optimal state sequence using the Viterbi algorithm of the HMM.

In [23] and [121], audio-visual parameter sets were used to train one static MLP and one temporal model (HMM). In the former approach, an OCON structure was adopted (see Fig. 1). One static MLP network per word was used to estimate the visual height, which was considered to be the most salient feature. Each MLP has six nodes in the first hidden layer, eight nodes in the second hidden layer, and one node in the output layer. On the other hand, the temporal HMM that was trained had seven states, three mixtures per state, and diagonal covariance matrices. The results seem to indicate that the temporal model seems to posses a better capability (than the static MLP) in breaking the word up into phonetically meaningful states. This demonstrates the power of the temporal model for synchronization applications.

2) Audio and Visual Integration for Lip-Reading Applications: Although signal-processing theory of automatic speech recognition is well advanced, practical applications lag because the speech signals to be processed are usually contaminated with background noise in adverse environments such as offices, automobiles, aircraft, and factories. To improve the performance of the speech-recognition system, the following approaches can be used: 1) compensate the noise in the acoustic speech signals prior to or during the recognition process [98] or 2) use multimodal information sources, such as semantic knowledge and visual features, to assist acoustic speech recognition. The latter approach is supported by the evidence that humans rely on other knowledge sources, such as visual information, to help constrain the set of possible interpretations [149].

Due to the maturity of digital video technology, it is now feasible to incorporate visual information in the speech-understanding process (lip reading). These new approaches offer effective integration of visually derived information into the state-of-the-art speech-recognition systems so as to gain an improved performance in noise without suffering degraded performance on clean speech [129]. Other important evidence as to the use of lip reading in human speech perception is offered by the auditory-visual blend illusion or the McGurk effect [96].

Three mechanisms about the means by which the two disparate (audio and visual) streams of information are

integrated have been proposed [133]. First, vision is used to direct the attention, which commonly occurs in situations such as crowded rooms where several people are talking at once. Second, visual information provides redundancy to the audio information. Last, visual information complements the audio information, especially when listening conditions are poor. Most current research efforts concentrate on the third mechanism of integration. A complete audio/visual lip-reading system can be decomposed into three major components [129]:

1) *audio/visual information preprocessing:* explicit feature extraction from audio and visual data;
2) *pattern-recognition strategy:* hidden Markov modeling, pattern matching with dynamic or linear time warping, and various forms of neural networks;
3) *integration strategy:* decision from audio and visual signal recognition.

a) Audio/visual information preprocessing: Audio information processing has been well discussed in the speech-recognition literature [119]. Briefly, the digitized speech is commonly sampled at 8 kHz. The sampled speech is preemphasized, then blocked and Hamming windowed into frames with a fixed time interval (say, 32 ms long) and with some overlap (say, 16 ms). For each frame, an N-dimensional feature vector is extracted (e.g., 12-order LPC cepstral coefficients, 12-order delta cepstral coefficients, 12-order delta-delta coefficients, a log-energy coefficient, a delta-log-energy coefficient, and a delta-delta-log-energy coefficient).

There are two major types of visual features useful for lip reading: contour-based and area-based features. The active contour models [61] are a good example of contour-based features, which have been applied to object contours found in many image analysis problems [25], [26]. PCA of a gray-level image matrix, a typical area-based method, has been successfully used for principle feature extraction in pattern-recognition problems [93], [140]. Most early systems used explicit contour feature extraction. Petajan [114] extracted contour features from binary thresholded mouth images. This approach was also used by Goldschen [42]. Deformable template approaches to obtain contour features, such as snake, have been the dominating methods for contour feature extraction [10], [47], [120]. Chiou and Hwang made the first attempt in using neural networks to guide the search of the deformable template for lip-reading application [24]. These methods attempt to measure physical aspects of the mouth directly, and such measurements are invariant to changes in lighting, camera distance, and orientation. Area-based techniques have primarily been based on neural networks [132], [153]. These area-based features are directly derived from the gray-level matrix surrounding the lips and capture more detailed information surrounding the mouth, including the cheek and chin. However, purely area-based approaches tend to be very sensitive to changes in position, camera distance, rotation, and speaker.

b) Pattern-recognition strategies: Most lip-reading systems used similar pattern-recognition strategies as the traditional speech recognition, such as dynamic time warping [114] and HMM's [24], [128]. Neural-network architectures have also been extensively explored, such as the static feed-forward back-propagation networks used by Yuhas *et al.* [153], the TDNN's used by Stork *et al.* [132], the multistage TDNN's used in [36], and the HMM recognizer with neural networks for observation probability calculation [11].

The speech data used in Yuhas' experiments were captured from a male speaker under a well-lit condition. They are based on the National Television Systems Committee video with 30 frames/s. Nine different phonemes were recognized. A reduced subimage (20×25) centered around the mouth was automatically identified for visual features, which were converted into the corresponding "clean" audio short-term cepstrum magnitude envelope (STSAE) by a feed-forward back-propagation network. The resulting cepstrum were weighted averaged with the noisy cepstrum directly derived from the audio signals. The weighting between the visual converted STSAE and the audio STSAE is determined based on the environment's signal-to-noise ratio (SNR). Another feed-forward neural network collected the sequence of the combined STSAE as the inputs and performed the recognition of vowels.

The work presented by Stork *et al.* [132] used a TDNN for recognizing the combined audio/visual speech data for five speakers. In their experiments, a video-only (VO) TDNN was used to recognize the visual speech inputs, which were acquired every 10 ms. From the 10-ms visual frame, five features (noise-chin separation, vertical separation of mouth opening, horizontal separations estimated from upper and lower lips, and horizontal separation of mouth opening) were estimated and were recognized by the VO TDNN, which ultimately produced the classification posterior probabilities $P(C \mid V)$, where C represents one of the ten spoken letters. Similarly, an audio-only (AO) TDNN was used to recognize the audio speech inputs, which again were acquired every 10 ms. From the 10-ms audio frame, 14 mel-scale coefficients (from 0 Hz to 5 kHz) were estimated and were recognized by the AO TDNN, which also ultimately produced the classification posterior probabilities $P(C \mid A)$. The resulting classification posterior probability $P(C \mid V, A)$ is approximated as $P(C \mid V, A) = P(C \mid V)P(C \mid A)$. The performance of this combined VO and AO TDNN network outperforms that of using a single video-audio (VA) TDNN, which receives the concatenated video and audio features (19 dimensions) as inputs.

The See Me, Hear Me project [36] developed at Carnegie-Mellon University extended the idea of using two separate (VO and AO) TDNN's in performing continuous letter recognition encountered in continuous spelling tasks. The

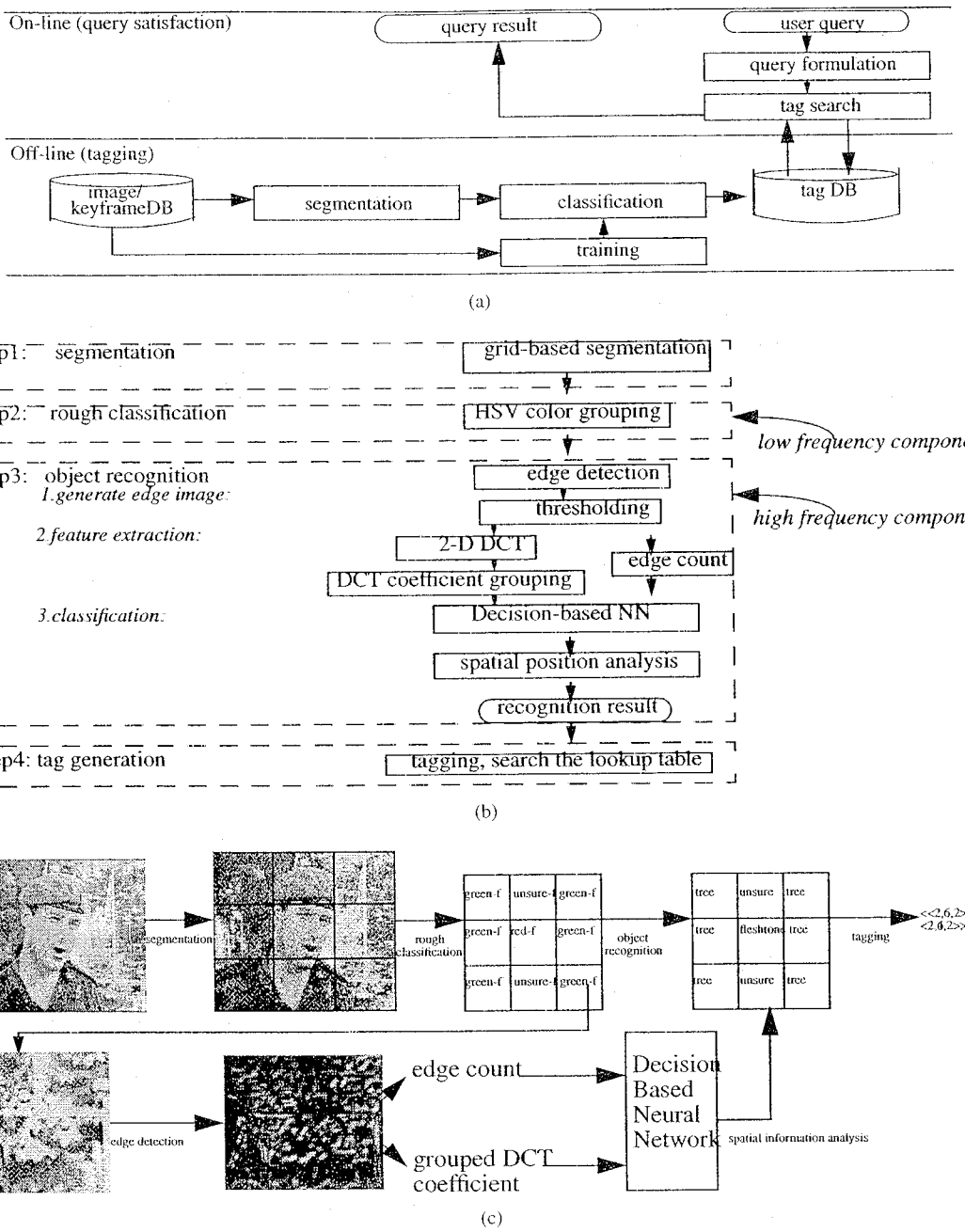


Fig. 7. A subject-based indexing system. (a) Visual search methodology. (b) Tagging procedure. (c) Tagging illustration.

audio features consist of 16 mel-scale Fourier coefficients obtained at 10-ms frame rate. The visual features were formed from the PCA transform with reduced dimensionality (32 only) out of 24 × 16 smoothed eigenlips. The two TDNN's are actually used for recognizing the corresponding phoneme (out of 62) and viseme (out of 42), which were then combined statistically for recognition of the continuous letter sequence based on the dynamic time-warping algorithm.

The project presented in [10] also combines the acoustic and visual features for effective lip reading. Instead of using neural networks as the temporal sequence classifier, this project adopted the HMM's and used a multilayer perceptron to calculate the observation probabilities $\{P(phone \mid audio, visual)\}$. The system combines the ten-order PCA transform coefficients (and/or the delta-features) from a gray-level eigenlip matrix (instead of the PCA from the snake points) from the video data and nine of the acoustic features from audio data [48]. They used a discriminatively trained MLP to compute the observation probabilities (the likelihood of the input speech data given the state of a subword model) needed by the Viterbi algorithm. Theoretically, the MLP provides the posterior probabilities, instead of the likelihood, which can be easily converted to likelihood according to Bayes' rule using the prior probability information. This bimodal hybrid

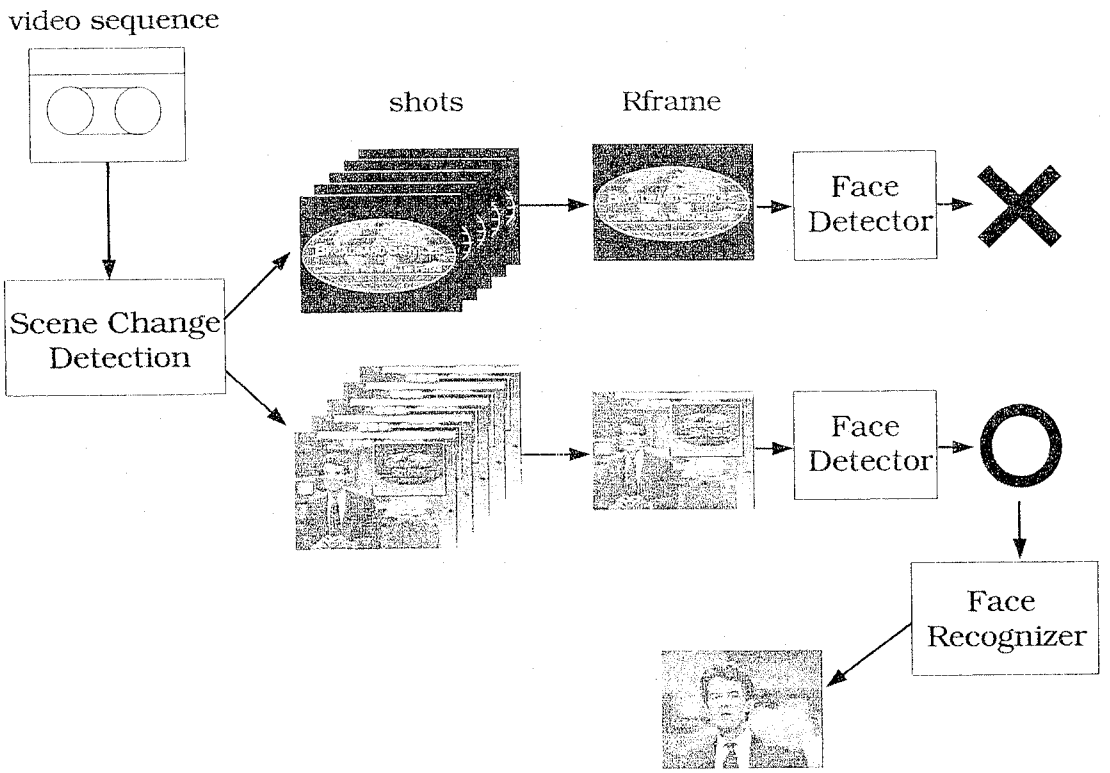


Fig. 8. Probabilistic DBNN face-based video browsing system. A scene change algorithm divides the video sequences into several shots. A face detector examines all the representative frames to see if they contain human faces. If they do, the face detector passes the frame to a face recognizer to find out whose face it is.

speech-recognition system has already been applied to a multispeaker spelling task, and work is in progress to apply it to a speaker-independent spontaneous speech-recognition system, the Berkeley Restaurant Project.

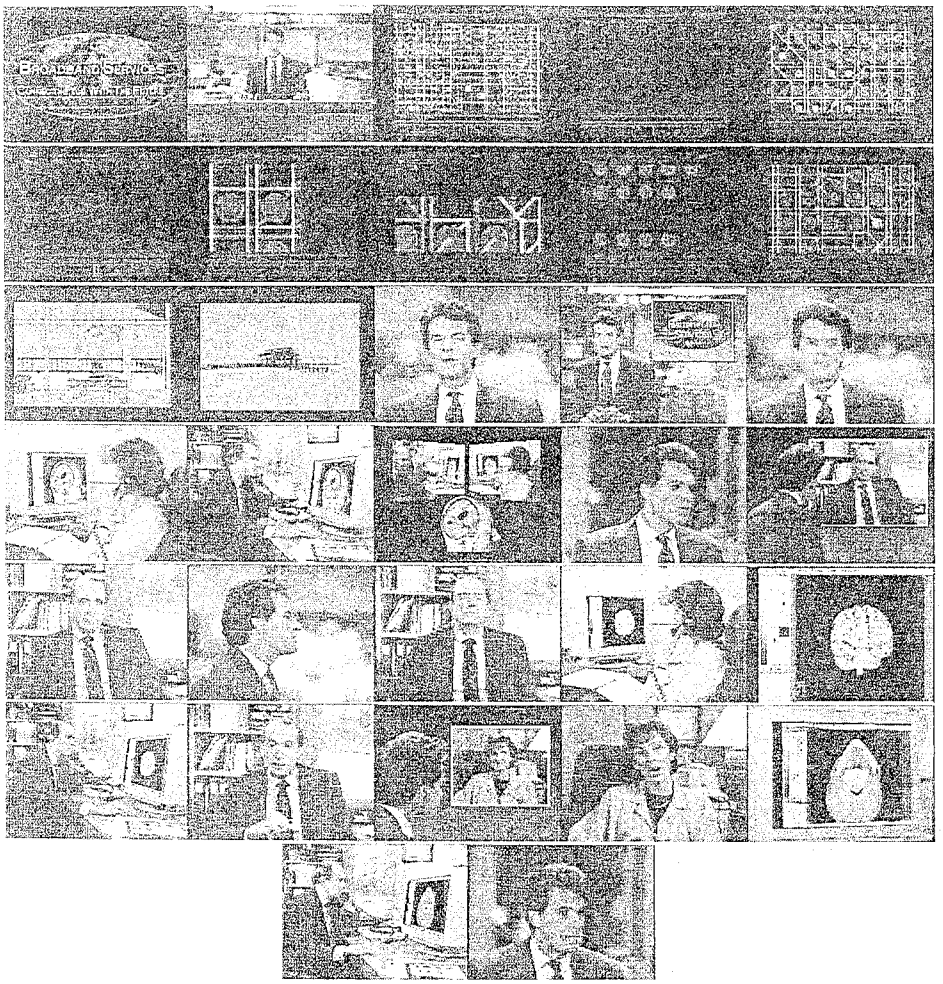
c) Decision integration: As discussed in the previous section, the audio and visual features can be combined into one vector before pattern recognition. Then, the decision is solely based on the result of the pattern recognizer. In the case of some lip-reading systems, which perform independent visual and audio evaluation, some rule is required to combine the two evaluation scores into a single one. Typical examples included the use of heuristic rules to incorporate knowledge of the relative confusability of phonemes in the evaluation of two modalities [114]; others used multiplicative combination of independent evaluation scores for each modality. These postintegration methods possess the advantages of conceptual and implementational simplicity as well as give the user the flexibility to use just one of the subsystems if desired.

D. Video Browsing and Content-Based Indexing

Digital video processing has recently become an important core information-processing technology. The MPEG-4 audio/visual coding standards tend to allow content-based interactivity, universal accessibility, and a high degree of flexibility and extensibility. To accommodate voluminous multimedia data, researchers have long suggested the content-based indexing and retrieval paradigm. Content-based intelligent processing is so critical because it offers a very broad application domain, including video coding, compaction, object-oriented representation of video, content-based retrieval in the digital library, video mosaicing, video composition (hybrid of natural and synthetic scenes), etc. [19].

1) Subject-Based Retrieval for Image and Video Data Bases: An NN-based tagging algorithm is proposed for subject-based retrieval for image and video data bases [152]. Object classification for tagging is performed off-line using DBNN. A hierarchical multiresolution approach is used, which helps cut down the search space of looking for a feature in an image. The classification is performed in two phases. In the first phase, color is used, and in the second, texture features are applied to refine the classification (both via DBNN). The general indexing scheme and tagging procedure are depicted in Fig. 7. The system [152] allows a customer to search the image data base by semantic subject. The images are not manipulated directly in the on-line phase. Each image is classified into a series of predefined subjects off-line using color and texture features and neural-network techniques. Queries are answered by searching over the tag data base. Unlike previous approaches, which directly manipulate images on-line using templates or low-level image parameters, the system tags the images off-line, which greatly enhances performance.

Compared to most of the other existing content-based retrieval systems, which only support similarity-based re-



(a)

(b)

Fig. 9. (a) Representative frames of news report sequence. (b) Representative frames of anchorman found by PDBNN face recognizer.

trieval, this system supports subject-based retrieval, allowing the system to retrieve by visual objects. The difference between subject- and similarity-based retrieval lies in the necessity for visual object recognition. Therefore, previous low-level models are not suitable for subject-based retrieval. Novel models are needed for subject-based retrieval, which could be utilized in film- and television-program-oriented digital video data bases. Neural networks provide a natural effective technology for intelligent information processing.

The tagging procedure includes four steps. In the first step, each image is cut into 25 equal-size blocks. Each block may contain single or multiple objects. In the second step, color information is employed for an initial classification, where each block is classified into one of the following families in the HSV color space: black, gray, white, red, yellow, green, cyan, blue, and magenta. In the next step, texture features are applied to refine the classification using DBNN if the result of color classification is a nonsingleton set of subject categories. Each block may be further classified into one of the following categories: sky, foliage, fleshtone, blacktop, white-object, ground, light, wood, unknown, and unsure. Last, an image tag generated from the lookup table using the object-recognition results is saved in the tag data base. The experimental results on the Web-based implementation show that this model is very efficient for a large film or television-program-oriented digital video data base.

2) Face-Based Video Indexing and Browsing: A video indexing and browsing scheme based on human faces is proposed by Lin *et al.* [82], [83]. The scheme is implemented by applying the face detection and recognition techniques. In many video applications, browsing through a large amount of video material to find the relevant clips is

an extremely important task. The video data base indexed by human faces provides users the facility to acquire video clips about the person of interest efficiently. For example, a film-study student may conveniently extract the clips of his favorite actor/actress from a movie archive to study his/her performance, and a TV news reporter may quickly find out from a news data base the clips containing images of a particular politician in order to edit headline news (see Figs. 8 and 9).

The scheme contains three steps (see Fig. 8). The first step of our face-based video browser is to segment the video sequence by applying a scene change detection algorithm. Scene change detection gives an indication of when a new shot starts and ends. Each segment created by scene change detection can be considered as a story unit of this sequence. After video sequence segmentation, a probabilistic DBNN face detector [83] is invoked to find the segments (shots) that most possibly contain human faces. From every video shot, we take its representative frame (Rframe) and feed it into face detector. Those representative frames from which the detector gives high face detection confidence scores are annotated and serve as the indexes for browsing.

This scheme can also be very helpful to algorithms for constructing hierarchies of video shots for video-browsing purposes. One such algorithm [151], for example, proposes using global color and luminance information as similarity measures to cluster video shots in an attempt to build video-shot hierarchies. Their similarity metrics enable very fast processing of videos. In their demonstration, however, some shots featuring the same anchorman fail to be grouped together due to insufficient image content understanding. For this type of application, we believe that the existence of similar objects, and human objects in particular, should provide a good similarity measure. As reported in [17], this scheme successfully classifies these shots to the same group.

IV. Conclusion

In this paper, we have focused on the main attributes of neural networks relevant to their application to intelligent multimedia applications. It is obvious that the space limitation prohibits a more exhaustive coverage on the subjects. More illustrative examples can be found in [109], [142], and numerous signal-processing journals.

Many critical research topics remain yet to be solved. From the commercial system perspective, there are many promising application-driven research problems. These include analysis of multimodal scene change detection, facial expressions and gestures, fusion of gesture/emotion and speech/audio signals, automatic captioning for the hearing impaired or second-language television audiences, multimedia telephone, and interactive multimedia services for audio, speech, image, and video contents.

From a long-term research perspective, there is a need to establish a fundamental and coherent theoretical ground for intelligent multimedia technologies. A powerful preprocessing technique, capable of yielding salient object-based video representation, would provide a healthy footing for on-line, object-oriented visual indexing. This suggests that a synergistic balance and interaction between representation and indexing must be carefully investigated. Another fundamental research subject needing our immediate attention is modeling and evaluation of perceptual quality in multimodal human communication. For content-based visual query, incorporating user feedback in the interactive search process will be also a challenging but rewarding topic.

In conclusion, future telecommunication will place a major emphasis on media integration for human communication. Multimedia systems can achieve their potential only when they are truly integrated in three key ways: integration of content, integration with human users, and integration with other media systems [108]. Therefore, the following technologies will emerge to lead future multimedia research [107]:

1) technologies for generating any kind of cyberspace;
2) technologies for warping into cyberspace;
3) technologies for manipulating objects in cyberspace;
4) technologies for communicating with residents of cyberspace.

To sum up, the research and application opportunities in intelligent multimedia processing are truly boundless. It is now up to us to explore further their vast benefits and enormous potential.

References

[1] M. A. Abidi and R. C. Gonzalez, *Data Fusion in Robotics and Machine Intelligence*. Boston, MA: Academic, 1992.

[2] E. Andre, G. Herzog, and T. Rist, "From visual data to multimedia presentations," in *Proc. Inst. Elect. Eng. Colloquium Grounding Representations: Integration of Sensory Information in Natural Language Processing, Artificial Intelligence and Neural Networks*, London, UK, May 1995, pp. 1:1–3.

[3] M. Arbib, *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA: MIT Press, 1995.

[4] R. Bajaj and S. Chaudhury, "Signature verification using multiple neural networks," *Pattern Recognit.*, vol. 30, no. 1, pp. 1–7, 1997.

[5] J. K. Baker, "The DRAGON system—An overview," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 23, pp. 24–29, Feb. 1975.

[6] L. E. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes," *Inequalities*, pp. 3:1–8, 1972.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 711–720, July 1997.

[8] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind convolution," *Neural Computation*, vol. 7. Cambridge, MA: MIT Press, 1995.

[9] H. Brandt, H. W. Lahmann, and R. Weber, "Quality control of saw blades based on neural networks and laser vibration measurements," in *Proc. SPIE 2nd Int. Conf. Vibration Measurements by Laser Techniques: Advances and Applications*, Ancona, Italy, 1996, vol. 2868, pp. 119–124.

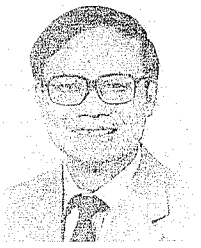
[10] C. Bregler and Y. K. Cao, "Eigenlips for robust speech recognition," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP'94)*, Adelaide, Australia, 1994, pp. II 669–672.

[11] C. Bregler, S. M. Omohundro, and Y. Konig, "A hybrid approach to bimodal speech recognition," in *Proc. 28th Asilomar*

Conf. Signals, Systems, and Computers, Pacific Grove, CA, 1994, pp. 572–577.
[12] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neuro-computing: Algorithms, Architectures and Applications*, F. Fogelman-Soulie and J. Hérault, Eds. Berlin, Germany: Springer-Verlag, 1991, pp. 227–236.
[13] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1042–1052, 1993.
[14] I. M. Ahmad and M. Shridhar, "A hierarchical neural network architecture for handwritten numeral recognition," *Pattern Recognit.*, vol. 30, no. 2, pp. 289–294, 1997.
[15] J.-F. Cardoso, "Blind signal separation: A review," *Proc. IEEE*, to be published.
[16] G. A. Carpenter and W. D. Ross, "ART-EMAP: A neural network architecture for object recognition by evidence accumulation," *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 805–814, 1995.
[17] Y. Chan, S. H. Lin, Y. P. Tan, and S. Y. Kung, "Video shot classification using human faces," in *Proc. IEEE Int. Conf. Image Processing 1996*, Lausanne, Switzerland, pp. 843–846.
[18] V. Chandrasekaran, M. Palaniswani, and T. M. Caelli, "Range image segmentation by dynamic neural network architecture," *Pattern Recognit.*, vol. 29, no. 2, pp. 315–329, 1996.
[19] S. F. Chang, "Content-based indexing and retrieval of visual information" *IEEE Signal Processing Mag.*, pp. 45–48, July 1997.
[20] K. Chen, D. Xie, and H. Chi, "Text-dependent speaker identification using hierarchical mixture of experts," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 32, no. 3, pp. 396–404, May 1996.
[21] T. Chen, A. Katsaggelos, and S. Y. Kung, Eds., "The past, present, and future of multimedia signal processing," *IEEE Signal Processing Mag.*, July 1997.
[22] Y.-K. Chen, Y. Lin, and S. Y. Kung, "A feature tracking algorithm using neighborhood relaxation with multi-candidate pre-screening," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Sept. 1996, vol. II, pp. 513–516.
[23] T. Chen and R. Rao, "Audio-visual interaction in multimedia communication," in *Proc. ICASSP'98* Munich, Germany, Apr. 1997, vol. 1, pp. 179–182.
[24] G. I. Chiou and J. N. Hwang, "Image sequence classification using a neural network based active contour model and a hidden Markov model," in *Proc. Int. Conf. Image Processing*, Austin, TX, Nov. 1994, pp. III:926–930.
[25] ——, "A neural network based stochastic active contour model (NNS-SNAKE) for contour finding of distinct features," *IEEE Trans. Image Processing*, vol. 4, pp. 1407–1416, Oct. 1995.
[26] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1131–1141, Nov. 1993.
[27] P. Comon, "Independent component analysis: A new concept," *Signal Process.*, vol. 36, pp. 287–314, 1994.
[28] T. F. Cootes and C. J. Taylor, "Active shape models—Smart snakes," in *Proceedings of the British Machine Vision Conference*. Berlin, Germany: Springer-Verlag, 1992, pp. 266–275.
[29] J. M. Corridoni, A. del Bimbo, and L. Landi, "3D object classification using multi-object Kohonen networks," *Pattern Recognit.*, vol. 29, no. 6, pp. 919–935, 1996.
[30] C. M. Courtney, L. H. Finkel, and G. Buchsbaum, "A multistage neural network for color constancy and color induction," *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 972–985, 1995.
[31] I. J. Cox, J. Ghosn, and P. Yianilos, "Feature-based face recognition using mixture distance," NEC Research Institute, Princeton, NJ, Tech. Rep. 95-09, 1995.
[32] L. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1148–1161, Nov. 1993.
[33] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks*. New York: Wiley, 1996.
[34] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," in *J. Royal Statist. Soc.*, pp. 1–38, 1976.
[35] S. Douglas and S. Y. Kung, submitted for publication.
[36] P. Duchnowski, U. Meier, and A. Waibel, "See me, hear me: Integrating automatic speech recognition and lipreading," in *Proc. ICSLP'95*, Yokohoma, Japan, 1995, pp. 547–550.
[37] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[38] J. L. Elman, "Finding structure in time," *Cognitive Sci. 14*, pp. 179–211, 1990.
[39] C. Fan, N. Namazi, and P. Penafiel, "New image motion estimation algorithm based on the EM technique," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 348–352, Mar. 1996.
[40] P. Fua and A. J. Hanson, "Optimization framework of feature extraction; Applications to semiautomated and automated feature extraction," in *Proc. DARPA Image Understanding Workshop*, May 1989, pp. 676–694.
[41] K. Fukushima and N. Wake, "Handwritten alphanumerical character recognition by the neocognition," *IEEE Trans. Neural Networks*, vol. 2, no. 3, pp. 355–365, 1991.
[42] A. J. Goldschen, O. N. Garcia, and E. Petajan, "Continuous optical automatic speech recognition by lipreading," in *Proc. 28th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, 1994, pp. 572–577.
[43] R. M. Gray, "Vector quantization," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 1, pp. 4–29, Apr. 1984.
[44] D. H. Han, H. K. Sung, and H. M. Choi, "Nonlinear shape restoration based on selective learning SOFM approach," *J. Korean Inst. Telematics Electronics*, vol. 34C, no. 1, pp. 59–64, Jan. 1997.
[45] T. Harris, "Kohonen neural networks for machine and process condition monitoring," in *Proc. Int. Conf. Artificial Neural Nets and Genetic Algorithms*, Ales, France, Apr. 1995, pp. 3–4.
[46] R. J. Hathaway, "Another interpretation of the EM algorithm for mixture distributions," *Statist. Prob. Lett.*, vol. 4, pp. 53–56, 1986.
[47] M. E. Hennecke, K. V. Prasad, and D. G. Stork, "Using deformable templates to infer visual speech dynamics," in *Proc. 28th Ann. Asilomar Conf.*, Pacific Grove, CA, Nov. 1994, vol. 1, pp. 578–582.
[48] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-RLP speech analysis technique," in *Proc. ICASSP'92*, San Francisco, CA, 1992, pp. I 121–124.
[49] I. L. Herlin and N. Ayache, "Features extraction and analysis methods for sequences of ultrasound images," in *Proc. 2nd European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992, pp. 43–57.
[50] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991, ch. 7, pp. 163–196.
[51] T. S. Huang, C. P. Hess, H. Pan, and Z.-P. Liang, "A neuronet approach to information fusion," in *Proc. IEEE First Workshop Multimedia Signal Processing*, Y. Wang, A. Reibman, B. H. Juang, and S. Y. Kung, Eds., Princeton, NJ, June 1997, pp. 45–50.
[52] J. N. Hwang, J. A. Vlontzos, and S. Y. Kung, "A systolic neural network architecture for hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1967–1979, Dec. 1989.
[53] J. N. Hwang and H. Li, "A limited feedback time delay neural networks," in *Int. Joint Conf. Neural Networks*, Nagoya, Japan, Oct. 1993, pp. I 271–274.
[54] J. N. Hwang and E. Lin, "Mixture of discriminative learning experts of constant sensitivity for automated cytology screening," *1997 IEEE Workshop for Neural Networks for Signal Processing*, Amelia Island, FL, Sept. 1997, pp. 152–161.
[55] J. N. Hwang, S. Y. Kung, M. Niranjan, and J. C. Principe, Eds., "The past, present, and future of neural networks for signal processing," *IEEE Signal Processing Mag.*, Nov. 1997.
[56] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.
[57] D. L. James, "SARDNET: A self-organizing feature map for sequences," in *Proc. Advances in NIPS 7*, Denver, CO, Nov. 1994, pp. 577–584.
[58] M. I. Jordan and R. A. Jacobs, "Learning to control an unstable system with forward modeling," in *Proc. Advances in NIPS'90*, 1990, pp. 325–331.
[59] ——, "Hierarchies of adaptive experts," *Advances in Neural Information Systems*. Los Altos, CA: Morgan Kaufmann, vol. 4, pp. 985–992, 1992.
[60] B. H. Juang and S. Katagiri, "Discriminative learning for

minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.
[61] M. A. Kass and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, pp. 321–331, 1988.
[62] S. Katagiri, C. H. Lee, and B. H. Juang, "Discriminative multilayer feedforward networks," *Proc. 1991 IEEE Workshop Neural Networks for Signal Processing*, Princeton, NJ, 1991, pp. 11–20.
[63] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybern.*, vol. 43, pp. 59–69, 1982.
[64] ——, *Self-Organization and Associative Memory*, 2nd ed. Berlin, Germany: Springer-Verlag, 1984.
[65] D. Kucera and R. W. Martin, "Segmentation of sequences of echocardiographic images using a simplified 3-D active contour model with region-based external forces," to be published.
[66] S. Y. Kung, *Digital Neural Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[67] S. Y. Kung, K. I. Diamantaras, and J. S. Taur, "Adaptive principal component EXtraction (APEX) and applications," *IEEE Trans. Signal Processing*, vol. 42, pp. 1202–1217, May 1994.
[68] S. Y. Kung and J. S. Taur, "Decision-based neural networks with signal/image classification applications," *IEEE Trans. Neural Networks*, vol. 6, pp. 170–181, Jan. 1995.
[69] S. Y. Kung, "Independent component analysis in hybrid mixture: KuicNet learning algorithm and numerical analysis," in *Proc. Int. Symp. Multimedia Information Processing*, Taipei, Taiwan, Dec. 1997, pp. 368–381.
[70] S. Y. Kung and C. Mejuto, "Extraction of independent components from hybrid mixture: KuicNet learning algorithm and applications," in *Proc. ICASSP'98*, Seattle, WA, 1998.
[71] S.-H. Lai and M. Fang, "Robust and automatic adjustment of display window width and center for MR images," Siemens Corporate Research, Princeton, NJ, SCR Invention 97E7464, 1997.
[72] L. Lampinen and E. Oja, "Distortion tolerant pattern recognition based on self-organizing feature extraction," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 539–547, 1995.
[73] R. Laganiere and P. Cohen, "Gradual perception of structure from motion: A neural approach," *IEEE Trans. Neural Networks*, Langer, vol. 6, no. 3, pp. 736–748, 1995.
[74] K. and F. Bodendorf, "Flexible user-guidance in multimedia CBT-applications using artificial neural networks and fuzzy logic," in *Proc. Int. ICSC Symp. Intelligent Industrial Automation and Soft Computing*, Mar. 1996, pp. B9–13.
[75] K.-F. Lee, "Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system," Ph.D. dissertation CMU-CS-88-148, Carnegie-Mellon University, Pittsburgh, PA, 1988.
[76] F. Leymarie and M. D. Levine, "Simulating the grassfire transform using an active contour model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 56–75, Jan. 1992.
[77] S. H. Lin, "Biometric identification for network security and access control," Ph.D. dissertation, Dept. of Electrical Engineering, Princeton University, Princeton, NJ, 1996.
[78] Y. Lin, Y.-K. Chen, and S. Y. Kung, "A principal component clustering approach to object-oriented motion segmentation and estimation," *J. VLSI Signal Process. Syst.*, vol. 17, pp. 163–188, Nov. 1997.
[79] S.-H. Lin, S. Kung, and L.-J. Lin, "A probabilistic DBNN with applications to sensor fusion and object recognition," in *Proc. 5th IEEE Workshop on Neural Networks for Signal Processing*, Cambridge, MA, Aug. 1995, pp. 333–342.
[80] F. Lavagetto, "Converting speech into lip movements: A multimedia telephone for hard of hearing people," *IEEE Trans. Rehab. Eng.*, p. 114, Mar. 1995.
[81] F. Lavagetto, S. Lepsoy, C. Braccini, and S. Curinga, "Lip motion modeling and speech driven estimation," in *Proc. ICASSP'97*, Munich, Germany, Apr. 1994, pp. 183–186.
[82] S.-H. Lin, Y. Chan, and S. Y. Kung, "A probabilistic decision-based neural network for location deformable objects and its applications to surveillance system and video browsing," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing 1996*, Atlanta, GA, pp. 3554–3557.
[83] S.-H. Lin, S. Y. Kung, and L. J. Lin, "Face recognition/detection by probabilistic decision-based neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 114–132, Jan. 1997.
[84] S.-H. Lin, S. Y. Kung, and M. Fang, "A neural network approach for face/palm recognition," in *Proc. 5th IEEE Workshop Neural Networks for Signal Processing*, Cambridge, MA, Aug. 1995, pp. 323–332.
[85] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[86] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 4, nos. 4/22, p. 153, 1987.
[87] Z. Liu, J. Huang, Y. Wang, and T. Chen, "Extraction and analysis for scene classification," in *Proc. IEEE First Workshop Multimedia Signal Processing*, Y. Wang, A. Reibman, B. H. Juang, and S. Y. Kung, Eds., Princeton, NJ, June 1997, pp. 343–348.
[88] S. W. Lu and A. Szeto, "Hierarchical artificial neural networks for edge enhancement," *Pattern Recognit.*, vol. 26, no. 8, pp. 1149–1164, 1993.
[89] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, pp. 746–748, Dec. 1976.
[90] M. MacQueen, "Some methods for classification and analysis of multivariate observation," in *Proceedings of the Fifth Berkeley Symp. Mathematical Statistics and Probabilities*, vol. 1, L. M. LeCun and J. Neyman, Eds. Berkeley, CA: Univ. of California Press, 1967, pp. 281–297.
[91] T. Mandl and H. C. Womser, "Soft computing—Vague query handling in object oriented information systems," in *Proc. HIM'95*, Konstanz, Germany, 1995, pp. 277–291.
[92] M. Mangeas and A. S. Weigend, "First experiments using a mixture of nonlinear experts for time series prediction," in *Proc. 1995 World Congress Neural Networks*, Washington, DC, July 1995, vol. 2, pp. 104–109.
[93] K. Mase and A. Pentland, "Automatic lipreading by optical-flow analysis," *Syst. Comput. Jpn.*, vol. 22, no. 6, pp. 67–76, 1991.
[94] Y. Matsuyama and M. Tan, "Multiply descent cost competitive learning as an aid for multimedia image processing," in *Proc. 1993 Int. Joint Conf. Neural Networks*, Nagoya, Japan, Oct. 1993, pp. 2061–2064.
[95] M. McLuhan, *Understanding Media*. New York: McGraw-Hill, 1964.
[96] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, 1976.
[97] B. Moghaddam and A. Pentland, "Face recognition using view-based and modular eigenspaces," in *Proc. SPIE Automatic Systems for the Identification and Inspection of Humans*, 1994, p. 2257.
[98] S. Y. Moon and J. N. Hwang, "Robust speech recognition based on joint model and feature space optimization of hidden Markov models," *IEEE Trans. Neural Networks*, vol. 8, pp. 194–204, Mar. 1997.
[99] S. Morishima, K. Aizawa, and H. Harashima, "An intelligent facial image coding driven by speech and phoneme," in *Proc. ICASSP*, Glasgow, UK, 1989, p. 1795.
[100] "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s," Draft Standard ISO 11172-2, Nov. 1991.
[101] "MPEG-2 video coding standard," Draft Standard ISO 13818, Nov. 1994.
[102] "Special issue on MPEG-4 video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Feb. 1997.
[103] "Second draft of MPEG-7 applications document," ISO/IEC JTC1/ SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG97/ N2666, Oct. 1997.
[104] "Third draft of MPEG-7 requirements," ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG97/N2606, Oct. 1997.
[105] T. Murdoch and N. Ball, "Machine learning in configuration design," in *Artificial Intell. Eng. Design, Anal. Manufact.*, vol. 10, no. 2, pp. 101–113, Apr. 1996.
[106] Y. Nakagawa, E. Hirota, and W. Pedrycz, "The concept of fuzzy multimedia intelligent communication system (FuMICS)," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, Sept. 1996, pp. 1476–1480.
[107] R. Nakatsu, "Media integration for human communication" *IEEE Signal Processing Mag.*, pp. 36–37, July 1997.
[108] P. L. Nikias, "Riding the new integrated media systems wave" *IEEE Signal Processing Mag.*, pp. 32–33, July 1997.
[109] *Proceedings of IEEE Workshops: Neural Networks for Signal Processing*, vols. I–VII. New York: IEEE Press, 1991–1997.

[110] *Proceedings of IEEE Workshops. Neural Networks for Signal Processing*, vol. VII, J. L. Giles, N. Morgan, and E. Wilson, Eds. New York: IEEE Press, 1997.
[111] E. Oja, "Principal component analysis, minor components, and linear neural networks," *Neural Networks*, vol. 5, no. 6, pp. 927–935, Nov.–Dec. 1992.
[112] A. Pedotti, G. Ferrigno, and M. Redolfi, "Neural network in multimedia speech recognition," in *Proc. Int. Conf. Neural Networks and Expert Systems in Medicine and Healthcare*, Plymouth, UK, Aug. 1994, pp. 167–173.
[113] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 1994, pp. 84–91.
[114] E. Petajan, B. Bischoff, D. Bodoff, and N. Brooke, "An improved automatic lipreading system to enhance speech recognition," in *Proc. ACM SIGCHI*, 1988, pp. 19–25.
[115] K. Popat and R. W. Picard, "Cluster-based probability model and its applications to image and texture processing," *IEEE Trans Image Processing*, vol. 6, pp. 268–284, Feb. 1997.
[116] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.
[117] L. R. Rabiner and B. H. Juang, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Tech. J.*, vol. 64, no. 6, pp. 1211–1233, July 1985.
[118] ——, "An introduction to hidden Markov models," *IEEE Acoust., Speech, Signal Processing Mag.*, pp. 4–16, Jan. 1986.
[119] ——, *Fundamentals of Speech' Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[120] R. R. Rao and R. M. Mersereau, "Lip modeling for visual speech recognition," in *Proc. 28th Ann. Asilomar Conf.*, Pacific Grove, CA, Nov. 1994, vol. 1, pp. 587–590.
[121] R. R. Rao, R. M. Mersereau, and T. Chen, "Using HMM's for audio-to-visual conversion," in *Proc. IEEE First Workshop Multimedia Signal Processing*, Y. Wang, A. Reibman, B. H. Juang, and S. Y. Kung, Eds., Princeton, NJ, June 1997.
[122] J. Rice, "A quality approach to biometric imaging," in *Proc. Inst. Elect. Eng. Colloquium Image Processing for Biometric Measurement*, Apr. 1994, pp. 4/1–4/5.
[123] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.
[124] J. Risch, R. May, J. Thomas, and S. Dowson, "Interactive information visualization for exploratory intelligence data analysis," in *Proc. IEEE 1996 Virtual Reality Annual Int. Symp.*, Santa Clara, CA, Apr. 1996, pp. 230–238.
[125] L. Rothkrantz, V. R. Van, and E. Kerckhoffs, "Analysis of facial expressions with artificial neural networks," in *Proc. Eur. Simulation Multiconf.*, Prague, Czech Republic, June 1995, pp. 790–794.
[126] D. E. Rumelhart, G. E. Hinton, and R. J. William, "Learning internal representation by error propagation," in *Parallel Distributed Processing: Explorations in the Micro-Structure of Cognition*, vol. 1, *Foundations*. Cambridge, MA: MIT Press, 1986.
[127] T. J. Sejnowski and C. R. Rosenberg, *NETtalk: A Parallel Network that Learns to Read Aloud*, Johns Hopkins Univ., Baltimore, MD, Tech. Rep., 1986.
[128] P. L. Silsbee, "Sensory integration in audiovisual automatic speech recognition," in *Proc. 28th Ann. Asilomar Conf.*, Pacific Grove, CA, Nov. 1994, vol. 1, pp. 561–565.
[129] P. L. Silsbee and A. C. Bovik, "Computer lipreading for improved accuracy in automatic speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 337–351, Sept. 1996.
[130] V. Shastri, L. C. Rabelo, and E. Onjeyekwe, "Device-independent color correction for multimedia applications using neural networks and abductive modeling approaches," in *Proc. 1996 IEEE Int. Conf. Neural Networks*, Washington, DC, June 1996, pp. 2176–2181.
[131] N. Srinvasa and R. Sharma, Eds., "SOIM: A self-organizing invertable map with applications in active vision," *IEEE Trans. Neural Networks*, vol. 8, no. 3, pp. 758–773, 1997.
[132] D. G. Stork, G. Wolff, and E. Levine, "Neural network lipreading system for improved speech recognition," in *Proc. IJCNN*, 1992, pp. 285–295.
[133] Q. Summerfield, "Some preliminaries to a comprehensive account of audio-visual speech perception," in *Hearing by Eye: The Psychology of Lip-Reading*, B. Dodd and R. Campbell, Eds. London: Lawrence Erlbaum, 1987, pp. 97–113.
[134] K. Sung and T. Poggio, "Learning human face detection in cluttered scenes," *Comput. Anal. Image Patterns*, pp. 432–439, 1995.
[135] R. Szeliski and D. Terzopoulos, "Physically-based and probabilistic models for computer vision," in *Proc. SPIE Geometric Methods in Computer Vision*, July 1991, vol. 1570, pp. 140–152.
[136] F. Takeda and S. Omatu, "High speed paper currency recognition by neural networks," *IEEE Trans. Neural Networks*, vol. 6, no. 1, pp. 73–77, 1995.
[137] J. S. Taur and C. W. Tao, submitted for publication.
[138] E. Y. H. Tseng, J. N. Hwang, and F. Sheehan, "Three-dimensional object representation and invariant recognition using continuous distance transform neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 141–147, Jan. 1997.
[139] L. H. Tung, I. King, and W. S. Lee, "Two-stage polygon representation for efficient shape retrieval in image databases," in *Proc. 1st Int. Workshop Image Databases and Multi-Media Search*, Amsterdam, The Netherlands, 1996, pp. 146–153.
[140] M. Turk and A. Pentland, "For recognition," *J. Cognitive Neurosci.*, vol. 3, pp. 71–86, 1991.
[141] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 328–339, Mar. 1989.
[142] *Proceedings of IEEE Workshops on Multimedia Signal Processing*, Y. Wang, A. Reibman, F. Juang, T. Chen, and S. Y. Kung, Eds. Princeton, NJ: IEEE Press, 1997.
[143] S. R. Waterhouse and A. J. Robinson, "Constructive algorithms for hierarchical mixtures of experts," in *Proc. Advances in Neural Information Processing 8*, Denver, CO, Nov. 1995, pp. 584–590.
[144] S. R. Waterhouse, D. MacKay, and A. J. Robinson, "Bayesian methods for mixtures of experts," in *Proc. Advances in Neural Information Processing 8*, Denver, CO, Nov. 1995, pp. 351–357.
[145] S. R. Waterhouse and A. J. Robinson, "Non-linear prediction of acoustic vectors using hierarchical mixtures of experts," in *Proc. Advances in Neural Information Processing Systems 7*, Denver, CO, Nov. 1994, pp. 835–842.
[146] Y. Weiss and E. H. Adelson, "Motion estimation and segmentation using a recurrent mixture of experts architecture," in *Proc. IEEE Workshop Neural Network for Signal Processing*, Boston, MA, Aug. 1995.
[147] H. White, "Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings," *Neural Networks*, vol. 3, pp. 535–549, 1990.
[148] R. J. William and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1994.
[149] W. A. Woods, "Language processing for speech understanding," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds. Los Altos, CA: Morgan Kaufman, 1990, pp. 519–533.
[150] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithms for Gaussian mixtures," Massachusetts Institute of Technology AI Lab, Cambridge, MA, Tech. Rep. A.I. Memo 1520, 1995.
[151] M. M. Yeung, B. L. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *Proc. SPIE Multimedia Computing and Networking*, 1995.
[152] H. H. Yu and W. Wolf, "A hierarchical, multi-resolution method for dictionary-driven content-based image retrieval," in *Proc. Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997, pp. 823–826.
[153] B. P. Yuhas, M. H. Goldstein, T. J. Sejnowski, and R. E. Jenkins, "Neural networks models for sensory integration for improved vowel recognition," *Proc. IEEE*, vol. 78, pp. 1658–1668, Oct. 1990.
[154] A. L. Yuille, P. Stolorz, and J. Utans, "Statistical physics, mixtures of distributions, and the EM algorithm," *Neural Computation*, vol. 6, pp. 334–340, 1994.
[155] S. Zucker, C. David, A. Dobbins, and L. Iverson, "The organization of curve detection: Coarse tangent fields and fine spline coverings," in *Proc. 2nd Int. Conf. Computer Vision*, Tampa, FL, Dec. 1988, pp. 568–577.



Sun-Yuan Kung (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

In 1974, he was an Associate Engineer with Amdahl Corporation, Sunnyvale, CA. From 1977 to 1987, he was a Professor of electrical engineering-systems at the University of Southern California, Los Angeles. In 1984, he was a Visiting Professor at Stanford University and the Delft University of Technology, The Netherlands. In 1994, he was a Toshiba Chair Professor at Waseda University, Tokyo, Japan. Since 1987, he has been a Professor of electrical engineering at Princeton University, Princeton, NJ. His research interests include spectrum estimations, digital signal/image processing, very-large-scale-integration (VLSI) array processors, neural networks, and multimedia signal processing. Since 1990, he has been Editor-In-Chief of the *Journal of VLSI Signal Processing*. He has authored more than 300 technical publications and three books, most recently *Principal Component Neural Networks* (New York: Wiley, 1996). He has edited numerous reference and proceedings books, most recently *Application-Specific Array Processors* (New York: IEEE Computer Society Press, 1991). He was the Keynote Speaker for the First International Conference on Systolic Arrays, Oxford, England, in 1986.

He has been an Associate Editor of several IEEE TRANSACTIONS. He was the first Associate Editor in the areas of VLSI (1984) and neural networks (1991) of IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was a member of the Administration Committee of the IEEE Signal Processing Society (SPS) (1989–1991). He was a founding member of IEEE-SPS Technical Committees on VLSI Signal Processing and on Neural Networks. He currently is member of the Technical Committee on Multimedia Signal Processing. He was a founding member and General Chairman of various IEEE conferences, including IEEE Workshops on VLSI Signal Processing in 1982 and 1986, the International Conference on Application Specific Array Processors in 1990 and 1991, IEEE Workshops on Neural Networks and Signal Processing in 1991 and 1992, and the first IEEE Workshop on Multimedia Signal Processing in 1997. He became an IEEE-SPS Distinguished Lecturer in 1994. He received the 1996 IEEE SPS's Best Paper Award for his publication on principal component neural networks and the 1992 IEEE SPS Technical Achievement Award for his contributions on "parallel processing and neural network algorithms for signal processing."

Jenq-Neng Hwang (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the National Taiwan University, Taipei, in 1981 and 1983, respectively. He received the Ph.D. degree from the University of Southern California, Los Angeles, in 1988.

In 1985, he was a Research Assistant with the Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California. From 1987 to 1989, he was a Visiting Student at Princeton University, Princeton, NJ. In 1989, he joined the Department of Electrical Engineering, University of Washington, Seattle, where he has been an Associate Professor since 1994. He has published more than 100 journal and conference papers and book chapters in the areas of signal/image/video processing, statistical data analysis, computational neural networks, parallel algorithm design, and very-large-scale-integration (VLSI) array architecture. He is on the editorial board of the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*. He was the General Cochair of the International Symposium on Artificial Neural Networks in December 1995. He is the Program Cochair of the 1998 International Conference on Acoustics, Speech, and Signal Processing.

Dr. Hwang received the 1995 Best Paper Award (with S.-R. Lay and A. Li) from the IEEE Signal Processing Society (SPS) in the area of neural networks for signal processing. He was Secretary of the Neural Systems and Applications Committee of the IEEE Circuits and Systems Society from 1989 to 1991. He was a member of the SPS's Design and Implementation of Signal Processing Systems Technical Committee and a founding member of the SPS's Multimedia Signal Processing Technical Committee. Currently, he is Chairman of the SPS's Neural Networks Signal Processing Technical Committee and is the society's representative to the IEEE Neural Network Council. He was an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1992 to 1994 and currently is an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He was the Conference Program Chair of the 1994 IEEE Workshop on Neural Networks for Signal Processing in September 1994. He chaired the Tutorial Committee for the IEEE International Conference on Neural Networks in June 1996.