Evaluation of Convolutional Neural Networks for Visual Recognition

Claus Neubauer, *Member, IEEE*

***Abstract*— Convolutional neural networks provide an efficient method to constrain the complexity of feedforward neural networks by weight sharing and restriction to local connections. This network topology has been applied in particular to image classification when sophisticated preprocessing is to be avoided and raw images are to be classified directly. In this paper two variations of convolutional networks—neocognitron and a modification of neocognitron—are compared with classifiers based on fully connected feedforward layers (i.e., multilayer perceptron, nearest neighbor classifier, auto-encoding network) with respect to their visual recognition performance. Beside the original neocognitron a modification of the neocognitron is proposed which combines neurons from perceptron with the localized network structure of neocognitron. Instead of training convolutional networks by time-consuming error backpropagation, in this work a modular procedure is applied whereby layers are trained sequentially from the input to the output layer in order to recognize features of increasing complexity. For a quantitative experimental comparison with standard classifiers two very different recognition tasks have been chosen: handwritten digit recognition and face recognition. In the first example on handwritten digit recognition the generalization of convolutional networks is compared to fully connected networks. In several experiments the influence of variations of position, size, and orientation of digits is determined and the relation between training sample size and validation error is observed. In the second example recognition of human faces is investigated under constrained and variable conditions with respect to face orientation and illumination and the limitations of convolutional networks are discussed.**

***Index Terms*— Convolutional neural networks, face recognition, handwritten digit recognition, neocognitron, neural-network applications, object recognition.**

I. Introduction

CONVOLUTIONAL neural networks with local weight sharing topology gained considerable interest both in the field of speech and image analysis [17]. Their topology is more similar to biological networks based on receptive fields and improves tolerance to local distortions. Additionally, the model complexity and the number of weights is efficiently reduced by weight sharing. This is an advantage when images with high-dimensional input vectors are to be presented directly to the network instead of explicit feature extraction and data reduction which is usually applied before classification [18], [20], [26]. Weight sharing can also be considered as an alternative to weight elimination [27] in order to reduce the number of weights [3]. Moreover, networks with local topology can more effectively be migrated to a locally connected parallel computer than fully connected feedforward networks [14].

The neocognitron [7]–[10], which can be considered as the first realization of convolutional networks, has been introduced by Fukuskima. In the neocognitron receptive fields, which were discovered in the cat's visual cortex by Hubel and Wiesel [12], [13], are used, the first extensive use of receptive fields in artificial neural networks. Fukushima applied the neocognitron primarily to handwritten digit recognition. Later variants of convolutional networks have been applied for example to large scale zip code recognition and face recognition [16], [4].

Within a convolutional architecture there are several possibilities for combining different kinds of neurons and learning rules. One method is to use McCulloch–Pitts neurons, which calculate a weighted sum plus sigmoid nonlinearity, and to train the whole network by error backpropagation [28]. This approach has been applied to zip code recognition [16]. In contrast, within the neocognitron, neurons calculate a weighted sum normalized by the incoming signal which results in a normalized convolution [9]. Furthermore, weights are trained independently layer by layer by reinforcement and a winner takes all rule. This approach has been shown to be feasible for binarized images of digits but has not been verified on large data sets. In this work a third method is proposed which combines advantages of both approaches by using McCulloch–Pitts neurons instead of more complicated neurons based on neocognitron [19]. The network is trained layer by layer similarly to the neocognitron and thus time-consuming error backpropagation is avoided. Here this approach is called modified neocognitron (NEO). Based on the previous work in this field two questions arise.

Can the neocognitron and the NEO, as examples of convolutional neural networks, be used for general recognition tasks, such as face recognition and digit recognition, on large scale databases?

How do they compare quantitatively with feedforward networks based on complete connectivity between layers without topological constraints?

The performance of the neocognitron has not previously been determined for more sophisticated problems such as face recognition, but only for character or digit recognition on a limited data sample. Thus one goal of this work is to compare the neocognitron with several fully connected networks and with the NEO. Such a comprehensive evaluation, considering the influence of varying training sample size and other key parameters, is the subject of this paper (see also [22]).

Manuscript received January 22, 1997; revised April 3, 1998.

The author is with Siemens Corporate Research, Inc., Princeton, NJ 08540 USA.

Publisher Item Identifier S 1045-9227(98)04760-2.

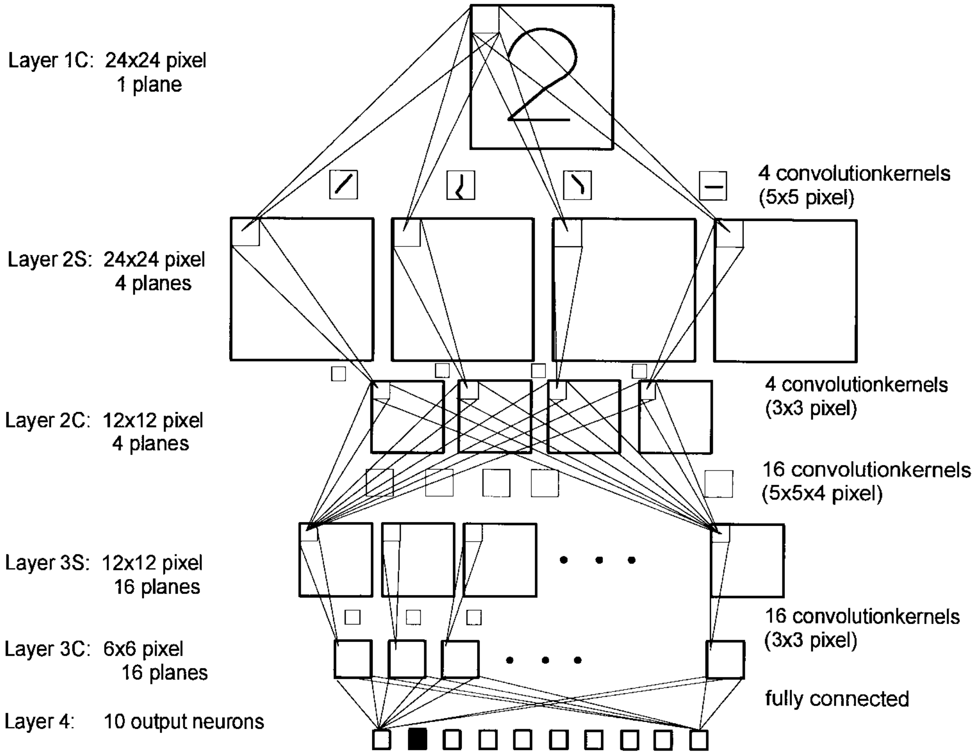

Fig. 1. The topology of the neocognitron and the NEO used for digit recognition. It consists of four layers with four convolutional planes in the first hidden layer and 16 convolutional planes in the second hidden layer. For face recognition the same number of planes is used in each layer, but due to the input resolution of 32 × 32 pixels the resolution of the layers 2S, 2C, 3S, 3C is 32 × 32, 16 × 16, 16 × 16, and 8 × 8, respectively. There are 18 output neurons according to 18 different faces.

II. Convolutional Neural Networks

In the following sections the network structure, neuron model and learning rules of the neocognitron and the NEO are described.

A. Neocognitron: Network Structure

A detailed description of the neocognitron architecture can be found in [7]. Fig. 1 shows the topology of convolutional networks used in this work. The raw image is feed into the input layer (1C) and determines the size of the input vector. Neurons perform local feature extraction and therefore each neuron is connected by a receptive field to a small area of the previous layer. A four-layer network is used. The hidden layers consist of S-sublayers and C-sublayers (see [7] for details) and each sublayer itself consists of several planes. The input layer is first mapped onto multiple planes of the 2S-sublayer. Each plane of a layer contains neurons which are extracting a particular local feature like a oriented bar or edge. The weights of the neurons in the S-sublayers are modified by training. Neurons of the same plane share the same weights in order to achieve some degree of tolerance to shift and deformation. The size of the local input window of each neuron in S-sublayers has been chosen to be 5 × 5 pixels. A mapping from one plane to the next can be considered as a convolution since all neurons of one plane have the same weight vector. From one layer to the next, the spatial resolution is reduced by two. The C-sublayers act as blurring filter (3 × 3 input window size with fixed, equally sized, positive weights) and perform subsampling by a factor two. The number of planes is increased from layer to layer in order to detect more specific features of higher complexity (curved shapes) while the spatial resolution is decreasing.

B. Neocognitron: Model of Neurons

In the neocognitron the S-neurons (together with the V-neurons) are described by (1). Is has been shown in [9] that (1) including the normalization term (V-neurons u_{vl}) approximates a convolution normalized by the length of the weight vector and the input vector. The selectivity of the S-neurons can by adjusted by hand with the parameter r. For an extensive discussion of the threshold parameter r for selectivity control during learning and testing see [11]. In contrast to [11] here the digit images are not strictly binary but the edges are smoothed due to the scanning process. It turned out by manual variation of parameter r, that the threshold for these particular gray value images has to be chosen smaller than for binary images, since the local features are not as significant as in binary images. Therefore the parameter $r = 0.5$ $(\theta = r/(1+r) = 0.33)$ in this work, while Fukushima generally used higher thresholds of $r = 1 - 4$ $(\theta = 0.5 - 0.8)$ for binary patterns. With $r = 0.5$ significant activation of only a few neurons occurs in this example, while small neuron activation is suppressed. The same value for r has been used for 2S and 3S and produced good results for the digit and face samples used in the following experiments. While the resulting planes are not strongly sensitive to small variations of r an individual adaption of parameter r for each layer to the specific recognition problem could help to slightly improve the overall performance, but has not been applied here

$$u_{sl}(n,k) = r_l(k)\phi \cdot \left(\frac{1 + \sum_{\kappa}\sum_{\nu} a_l(\nu,\kappa,k)u_{cl-1}(n+\nu,\kappa)}{1 + \frac{r_l(k)}{r_l(k)+1}\, b_l(k)u_{vl}(n)} - 1 \right)$$

$$u_{vl}(n) = \left(\sum_{\kappa}\sum_{\nu} c_l(\nu)(u_{cl-1}(n+\nu,\kappa)^2 \right)^{1/2}$$

$$\phi(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0. \end{cases} \tag{1}$$

$u_{vl}(n)$	V-neuron, layer l, position n.
$u_{cl}(n+\nu,\kappa)$	C-neuron, layer $l-1$, plane κ, position $n+\nu$.
$u_{sl}(n,k)$	S-neuron, layer l, plane k, position n.
$u_{cl-1}(n+\nu,\kappa)$	C-neuron, layer $l-1$, plane κ, postion $n+\nu$.
$a_l(\nu,\kappa,k), b_l(k)$	modifiable weights.
$c_l(\nu)$	positive fixed weights.
$r_l(k)$	selectivity.
$\phi(x)$	nonlinearity.

C. Neocognitron: Learning Rules

The neocognitron is trained layer by layer starting from the first hidden layer. After training of the first hidden layer with images containing only simple features, the training set for the next layer, containing more complex patterns, is propagated through the first layer. This procedure is repeated until the output layer is reached. Thus higher layers represent features of increasing complexity. One advantage of this approach is that the first hidden layer does not have to be retrained for each classification problem since, for typical visual recognition tasks, edges usually have to be extracted at the first level. The reinforcement learning rule proposed by Fukushima is used here both for supervised and unsupervised training of the neocognitron (2)

$$\Delta a_l(\nu,\kappa,\hat{k}) = q_l c_l(\nu) u_{cl-1}(\hat{n}+\nu,\kappa)$$

$$\Delta b_l(\hat{k}) = q_l u_{vl}(\hat{n}). \tag{2}$$

D. Modified Neocognitron: Network Structure

The NEO uses the same network topology as the neocognitron. However, in order to simplify the network, the S-neurons are modified as discussed below.

E. Modified Neocognitron: Model of Neurons

In contrast to the neocognitron, the NEO uses S-neurons based on the McCulloch–Pitts model. Thus the NEO is a combination of a neuron function based on the perceptron together with the convolutional network structure of neocognitron (3). While the original neocognitron performs a locally normalized convolution the NEO performs a convolution without normalization resulting in a simpler function, which is equivalent to a weighted sum between input vector (5 × 5 input window) and weight vector of the neuron. In the NEO a sigmoid function is used as nonlinearity. The NEO is similar to the weight sharing network of LeCun [16]. The difference between the NEO and the network used in [16] is the use of a sequential learning strategy (layer after layer) described in the next section instead of simultaneous training of all layers by error backpropagation applied by LeCun. The neurons of the C-sublayers are used in the same way as in the original neocognitron for blurring and subsampling

$$u_{sl}(n,k) = \phi\left(\sum_{\kappa}\sum_{\nu} a_l(\nu,\kappa,k)u_{cl-1}(n+\nu,\kappa) \right)$$

$$\phi(x) = 1/(1+\exp(-x)). \tag{3}$$

F. Modified Neocognitron: Learning Rules

Similar to the original neocognitron the layers of the NEO are trained independently and sequentially starting with S2-sublayer. For the NEO, the least-mean squares (LMS) rule is used for supervised learning. In (4) the LMS rule or delta rule is shown with the same notation as (2); ϵ denotes the learning rate; $o_{sl}(\hat{n},\kappa)$ is the expected output of $u_{sl}(\hat{n},\kappa)$: $o_{sl}(\hat{n},\kappa) = 1$ if the pattern is similar to the feature to be extracted by the neurons in plane κ and $o_{sl}(\hat{n},\kappa) = 0$ otherwise. This supervised algorithm is feasible for the digit recognition problem where it is intuitively clear that curved lines and endpoints are important higher level features

$$\Delta a_l(\nu,\kappa,\hat{k}) = \epsilon\; u_{cl-1}(\hat{n}+\nu,\kappa)\;(o_{sl}(\hat{n},\kappa) - u_{sl}(\hat{n},\kappa)). \tag{4}$$

For face recognition it is difficult to train intermediate layers with supervision since it is not known which higher level features are important. In this case, unsupervised algorithms for feature extraction can be applied, e.g., principle component analysis or auto-encoding learning [6], [24]. Both, principal component analysis and learning by auto-encoding, give similar results since it has been shown that the weights in the hidden layer of a three-layer auto-encoding network converge to principal components [3], [5]. In the following the steps are described, that are necessary in order to train a convolutional neural network by auto-encoding:

- Image samples with size 5 × 5 pixel (by random selection of subimages from face images) are collected and a three-layer backpropagation network (25 input neurons, four hidden neurons, and 25 output neurons) is trained in order to reproduce the input vector at the output layer.
- After training is completed, the weight vectors of the four hidden neurons are used as weight vectors for the 2S-neurons of the four feature extracting planes. All neurons of one particular plane use the same weight vector in a convolutional network.
- Images of faces are selected and processed by the 1C, 2S, and 2C sublayer in order to create a training sample for the 3S-sublayer.
- Once again random image samples with size 5 × 5 pixel are extracted and another three-layer backpropagation network with 16 hidden neurons is trained for auto-

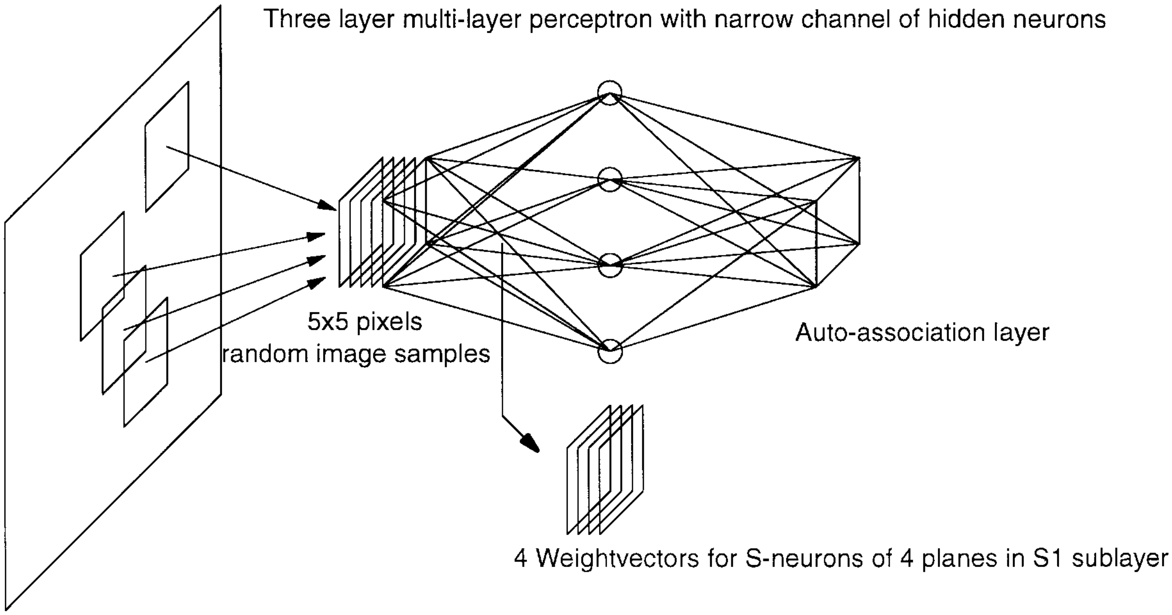
Three layer multi-layer perceptron with narrow channel of hidden neurons
5x5 pixels
random image samples
Auto-association layer
4 Weightvectors for S-neurons of 4 planes in S1 sublayer

Fig. 2. A three-layer auto-encoding network is applied in order to learn the weight vectors (local features) of 2S-sublayer in the NEO.

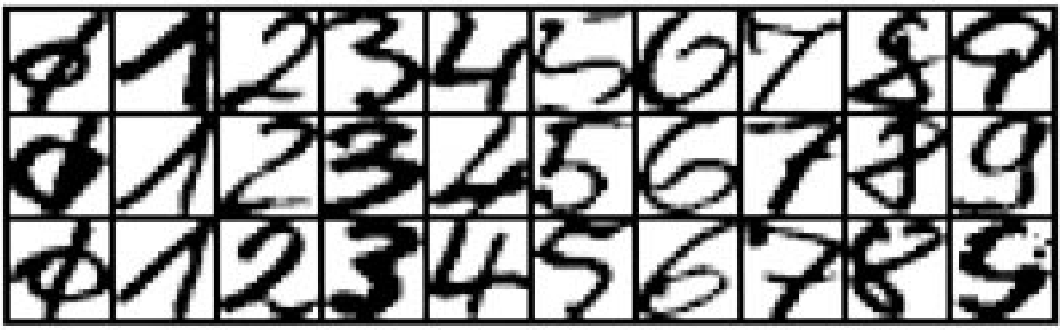

Fig. 3. Examples of the digit data set, containing original gray scale digits scanned with 8 bits per pixel. The digits used in experiment I (16 × 16 pixels) are normalized with respect to size, position, and orientation.

association in order to provide weight vectors for the 16 planes in sublayer 3S.

- After training of 3S-sublayer the training sample containing whole faces is propagated through the layers 1C to 3C. This training sample together with class labels associated with the faces is used to train the weights of the final layer (4) with supervision (LMS).

As an example training by auto-encoding of the 2S-sublayer is illustrated in Fig. 2.

III. Classification of Handwritten Digits

First, the performance of convolutional neural networks and fully connected networks is compared for recognition of handwritten digits. In the following experiments no specific feature extraction takes place but the raw images are used directly for classification. The digit data set consists of 10 000 digits (1000 per class) for learning and 10 000 digits for validation. In the original data set the digits are normalized with respect to size, position, orientation, and contrast and the resolution is 16 × 16 pixels with eight-bit gray values. Examples of the data set are shown in Fig. 3.

Two experiments are described in detail. In experiment I normalized digits are classified. Variations in the human style of writing require a strong tolerance to deformations. In experiment II (Fig. 4) additionally the digits are shifted, scaled, and rotated by affine transformations in order to measure the tolerance of the classifiers with respect to these transformations separately and in combination.

A. Experiment I: Handwritten Digits with Constant Size, Position, and Orientation

The neocognitron and the NEO are compared with three fully connected classifiers on original gray scale digits. Fully connected classifiers are represented by a two-layer perceptron, a three-layer multilayer perceptron (MLP), and a nearest neighbor classifier with the appropriate topology (Table I). In this context the nearest neighbor classifier can be regarded as a neural network where the number of hidden units is equal to the number of training samples and a winner takes all rule is applied at the output layer.

While the topology of nearest neighbor classifier and perceptron is determined by the problem, for the MLP the number

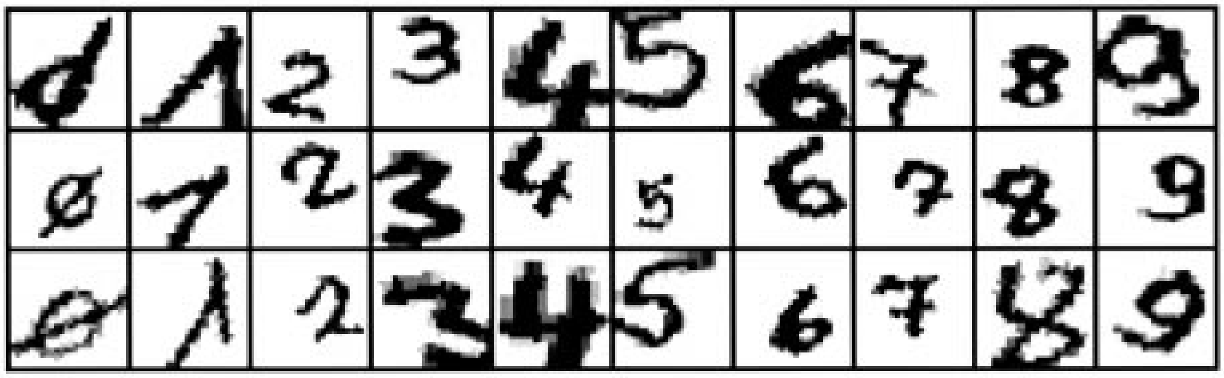

Fig. 4. Examples from the digit data set with additional variation of digit size, position, and orientation, which is used in experiment II (24 × 24 pixel). This data set is derived from the normalized data set by affine transformations.

TABLE I
Classifiers Based on Local and Global Feedforward Connections, Which Are Compared in Experiment I On Normalized Digits

Local connections		Global connections	
	planes		neurons
Modified Neocognitron	1-4-12-10	Nearest Neighbor classifier	256-x-10
Neocognitron	1-4-12-10	Two layer Perceptron	256-10
		Three layer Multi-layer Perceptron	256-50-10

of hidden neurons has to be optimized in advance by multiple training runs with different numbers of hidden neurons. Fifty neurons turned out to be a good tradeoff between classification performance and model complexity. All classifiers except the nearest neighbor classifier have to be trained iteratively, while the convergence of the validation error is observed simultaneously on an independent test set in order to avoid over-fitting. Training is stopped when the validation error stops decreasing. The training sample size is a key parameter for generalization. Therefore the learning procedure is repeated several times for each classifier with different training sample sizes. Nine training repetitions are carried out with training sample sizes ranging from ten to 1000 patterns per class. Fig. 5 shows, for the MLP, the convergence of training and corresponding validation error for different training sample sizes.

Fig. 6 summarizes the results of experiment I. The validation error is plotted as a function of training sample size for the five different classifiers. It is clearly shown that both convolutional networks generalize significantly better than the fully connected networks. The performance of the two-layer perceptron is worst since a validation error of 5.52% remains after training with 1000 samples per class. This shows that a linear separation of the ten digit classes is not sufficiently accurate on raw images. The MLP has 2.53% error. For the convolutional networks—neocognitron and NEO—only 1.23% and 1.59%, respectively, misclassifications have been observed. This indicates that the NEO has at least the same classification performance as the neocognitron despite the latter's more complicated neurons. For small training sample sizes (ten patterns per class) the nearest neighbor classifier has the worst error rate (18.93%) but with increasing training sample size it gradually improves relatively to the other algorithms and achieves 2.31% test error for 1000 learning patterns per class. This might be due to the fact that the nearest neighbor classifier approximates the optimal Bayes classifier for large sample sizes. However, we are interested primarily in good generalization for small training sample size as well as computational efficiency, so that a nearest neighbor classifier based on 1000 samples per class is not feasible for many practical applications.

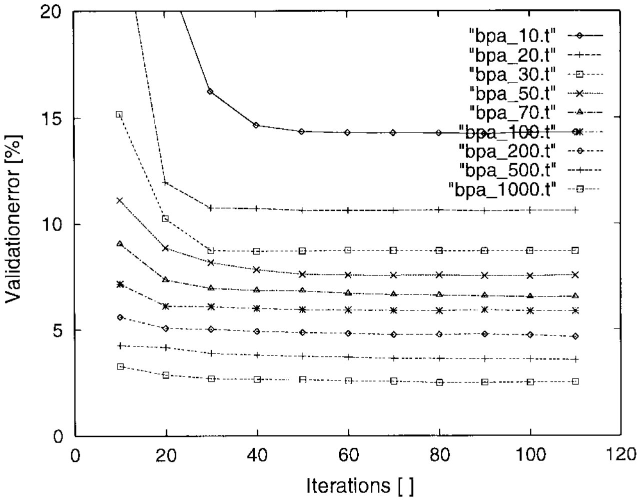

Fig. 5. Convergence of validation error of MLP for nine different training sample sizes between ten and 1000 samples per class as indicated by the numerical value in the legend.

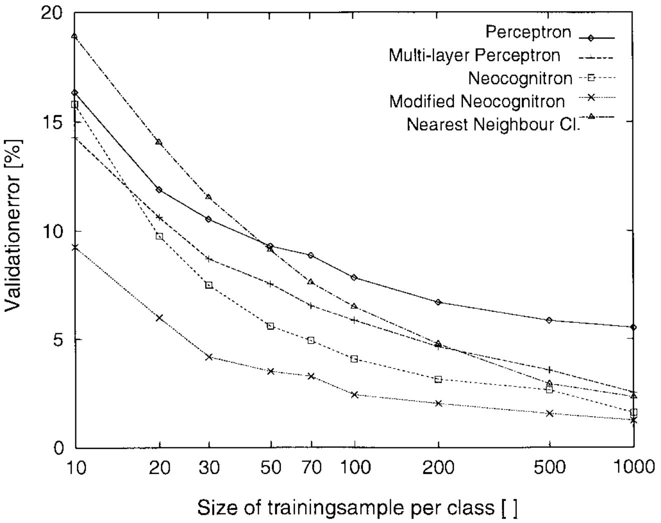

Fig. 6. Validation error as a function of training sample size for five classifiers. Classifiers based on local connectivity (neocognitron and NEO) are superior to classifiers with full feedforward connectivity in particular for small training sample size.

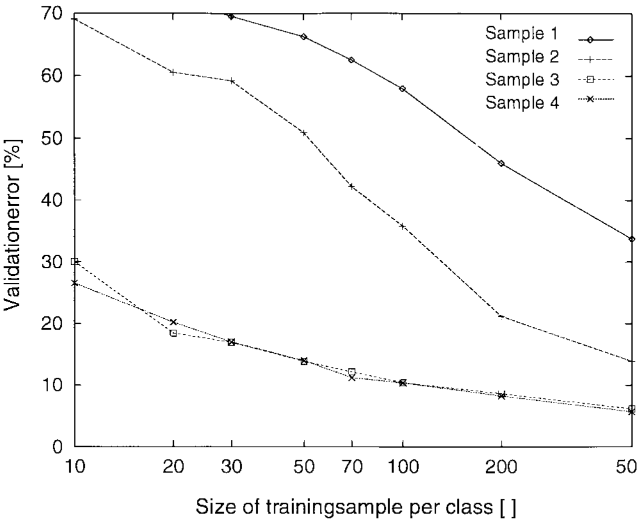

Fig. 7. Validation error of the MLP as a function of learning sample size for samples 1-4 subject to affine transformations. 1) shift, rotation, and scaling; 2) shift; 3) rotation; 4) scaling.

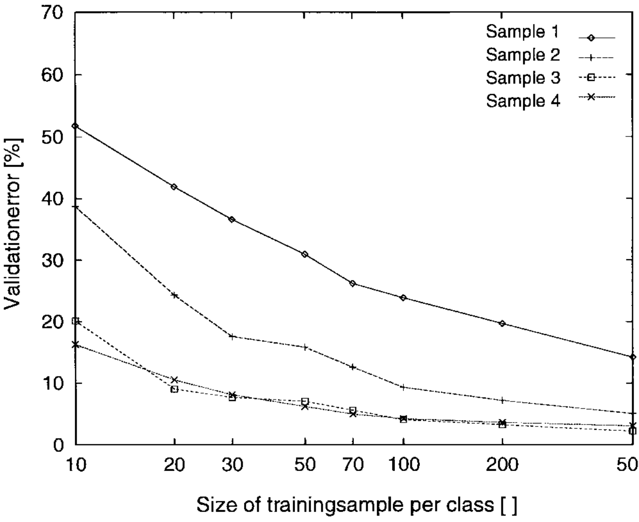

Fig. 8. Validation error of the NEO as a function of learning sample size for samples 1–4 subject to affine transformations. 1) shift, rotation, and scaling; 2) shift; 3) rotation; 4) scaling.

B. Experiment II: Digits with Variable Size, Position, and Orientation

Experiment II generalizes upon experiment I which uses normalized digits by examining the performance of convolutional networks on gray scale digits that vary with respect to position, orientation, and size within the image frame. For this experiment four data sets with 5000 patterns for training and 5000 patterns for testing are generated from the original normalized data set by random affine transformations. The spatial resolution is increased to 24 × 24 pixels. In sample 1 digits are transformed with a random combination of rotation, shift and scaling (see Fig. 4). Sample 2 contains randomly shifted patterns with shifts of maximal 50% with respect to digit size in each direction. In sample 3 rotations smaller or equal 22.5° are applied and in sample 4 digits are scaled by a factor between 0.5 and 2. This comparison focuses on the NEO and the MLP, since they performed best in their class (local and global topology) in the previous experiment. Learning curves are generated by variation of the training sample size between ten and 500 digits per class in eight steps for the four different data sets. The classification results on the four data sets are summarized in Figs. 7 and 8 for MLP and NEO, respectively.

The NEO achieves significantly better results than the MLP. For example the validation error of the NEO for sample 1 which was subject to all transformations is 14.20% compared to 33.74% for the MLP when 500 patterns per class are used for training. However, compared to the classification results in experiment I with normalized digits, a decreasing accuracy can be observed for both classifiers. This indicates that there is only limited tolerance to affine transformations, even for the NEO In Figs. 7 and 8 the influence of shift, scaling, and rotation are also shown separately, results that are important for the choice of appropriate preprocessing steps. While position variations had a high impact on the performance (MLP 13.88%, NEO 5.06%), scaling (MLP 6.16%, NEO 2.20%) and rotation (MLP 5.70%, NEO 3.08%) did not affect the classification accuracy as much. Thus it can be concluded that the convolutional architecture generalizes significantly better than the MLP but the results are still depending on proper normalization, in particular if small training sample sizes are used.

IV. Recognition of Human Faces

Human face recognition is another challenging visual classification task. For face recognition, small deviations from a three-dimensional shape have to be detected, while the recognizer has to be able to cope with a large variety of appearances due to possible illumination and pose variations. In order to simplify this task, it is necessary first to locate the face within the scene and then to normalize the face with respect to size and orientation. Therefore, in this approach a hierarchical face recognition is realized. A separate network is trained to localize a face within a scene and afterwards the subimage containing the face is analyzed by the identification network.

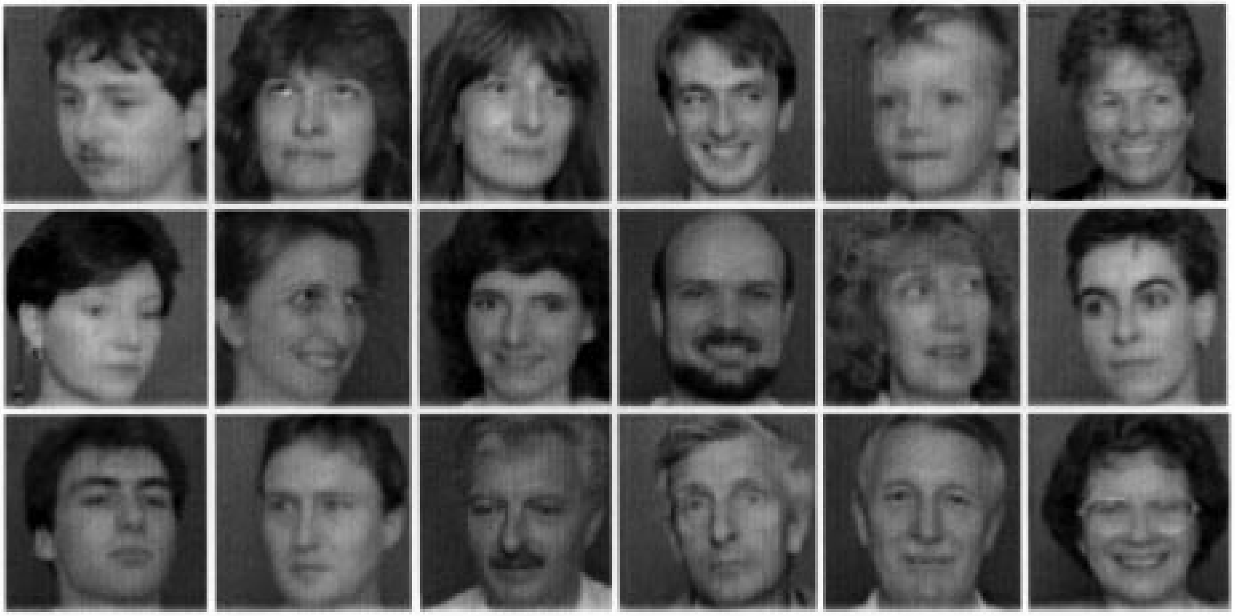

Fig. 9. Examples from the face data set with constrained illumination and homogenous background.

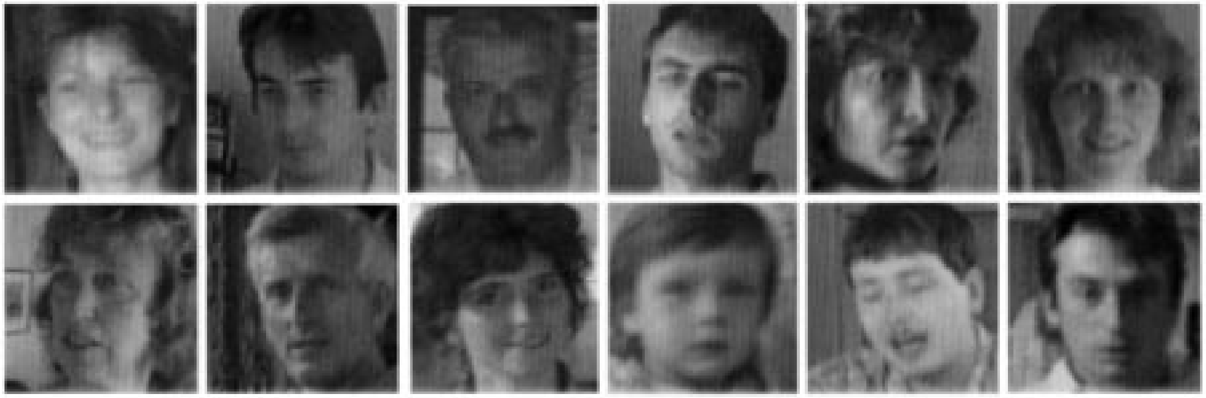

Fig. 10. Examples from the face data set with varying illumination and background.

Three experiments, carried out under different conditions, are described below. Two data sets have been gathered: one with constant illumination and homogeneous background (Fig. 9) and one with more realistic variations of illumination and background (Fig. 10).

A. Localization of Faces

For face localization a three-layer MLP with ten hidden neurons is trained to detect faces of a standard size in a window of fixed size. A resolution of 32 × 32 pixel is sufficient for this task since a face is primarily characterized by existence of eyes, nose, and mouth together with their geometrical relationship all of which can be recognized at low spatial resolution. Network training with 1000 unlabeled faces and the same number of arbitrary background images results in a sufficiently precise face detector with an accuracy summarized in Table II. For localization, a window is shifted over the whole image. Several possible sizes and orientations of the face are taken into account by a multiresolution technique and by variation of the window orientation within the scene image. These subimages are presented to the face detector so that it is possible to locate arbitrary faces in cluttered scenes if nearly frontal views are provided. From subimages with significantly high response of the localization network the background outside the central circle is deleted, the subimages are normalized with respect to brightness and contrast and they are feed into the identification network, as described in Fig. 11.

TABLE II
PERFORMANCE OF THE FACE-BACKGROUND CLASSIFIER

[%]	Training		Validation	
classified from/to	Face	Background	Face	Background
Face	95.1	1.5	86.8	2.5
Background	4.9	98.5	13.2	97.5

B. Face Identification: Constrained Training—Constrained Validation

Three experiments have been carried out for identification of faces. In the first experiment (Section B) training is based

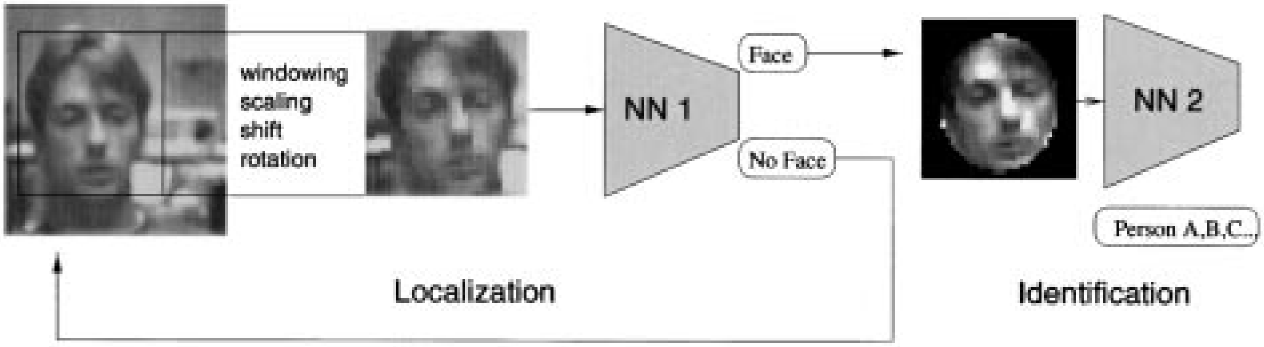

Fig. 11. Concept for localization of faces in cluttered scenes.

TABLE III
TRAINING METHODS FOR FULLY CONNECTED THREE-LAYER NETWORKS USED FOR FACE RECOGNITION

	Hidden layer	Output layer
Multi-layer Perceptron	Error backpropagation (supervised)	
Self-organizing Map + LMS	SOM (unsupervised)	LMS (supervised)
Auto-encoding network + LMS	Auto-encoding (unsupervised)	LMS (supervised)

on the constrained face data sample (Fig. 9) with constant illumination and homogenous background and validation is based on an independent constrained data set. In the second experiment (Section C) after training with constrained data a unconstrained data set (Fig. 10) is used for validation and in the final experiment (Section D) training as well as validation set are unconstrained.

For face recognition similar to the digit recognition experiments, the convolutional network (NEO) is compared with several fully connected classifiers (nearest neighbor classifier and a three-layer MLP). The fully connected network (with three layers) is trained with three different learning techniques Table III: beside error backpropagation, two alternatives have been tested, where the hidden layer is trained without supervision and the output layer is trained with supervision.

For unsupervised training of the hidden layer self-organizing feature map [1], Auto-encoding network [6] (see also Fig. 2) or principal component analysis (Eigenfaces) [25] can be applied. After the hidden layer has been trained with one of the unsupervised algorithms, the output layer is trained with supervision by least-mean square (LMS) error minimization using the face class labels.

As an example for feature extraction from faces a two dimensional self-organizing map with 10 × 10 neurons is shown in Fig. 12. After training with unlabeled faces typical prototypes evolve. The weight vectors are visualized as two-dimensional gray value images in Fig. 12. The radial distances between the prototypes and the input vector is fed into the output layer for classification. The prototypes of the 10 × 10 map represent typical faces from the training set in a ordered way. Similar faces are close together within the map. For example women are concentrated in the upper left and men in the lower right area.

In addition to feedforward classifiers, discussed here, there also exist iterative approaches such as dynamic link architecture [15], which are not considered in this paper. In [15] a flexible graph is matched between image and model and simulated annealing is used for optimization. Utilizing this concept it is possible to compensate for small deformations, but the method is quite slow due to the simulated annealing optimization necessary during classification.

In the first experiment using faces, the data set used for training is acquired under fairly constant, diffuse and frontal illumination as shown in Fig. 9. The head pose has been varied randomly within approximately 45° in each direction. One thousand eighty images from 18 persons (60 images per person) have been selected randomly from a video covering different face poses from each person. The validation set has been grabbed independently but under approximately the same illumination and with homogenous background. The classifiers achieved very different results in this comparison. In Table IV, the first column lists the type of classifier, the second column the image resolution, and the third column the results for the constrained validation set. The nearest neighbor classifier had a misclassification rate of 4.9% for images with 64 × 64 pixel resolution and 12.9% for images with 32 × 32 pixel.

The MLP correctly identified only 6.4%, because training was not successful at all. Several attempts have been made with different parameter configurations (learning rate $\epsilon = 0.01, 0, 05, 0.25$; number of hidden neurons $n = 10, 20, 50$; three different weight initializations with different fan in). Training was stopped after 200 repetitions of the whole training sample, if no further improvements took place. However, with error backpropagation the network always ended in a poor local minimum. The relatively best training error was 89.1% with 50 hidden neurons, $\epsilon = 0.25$, but the other runs resulted

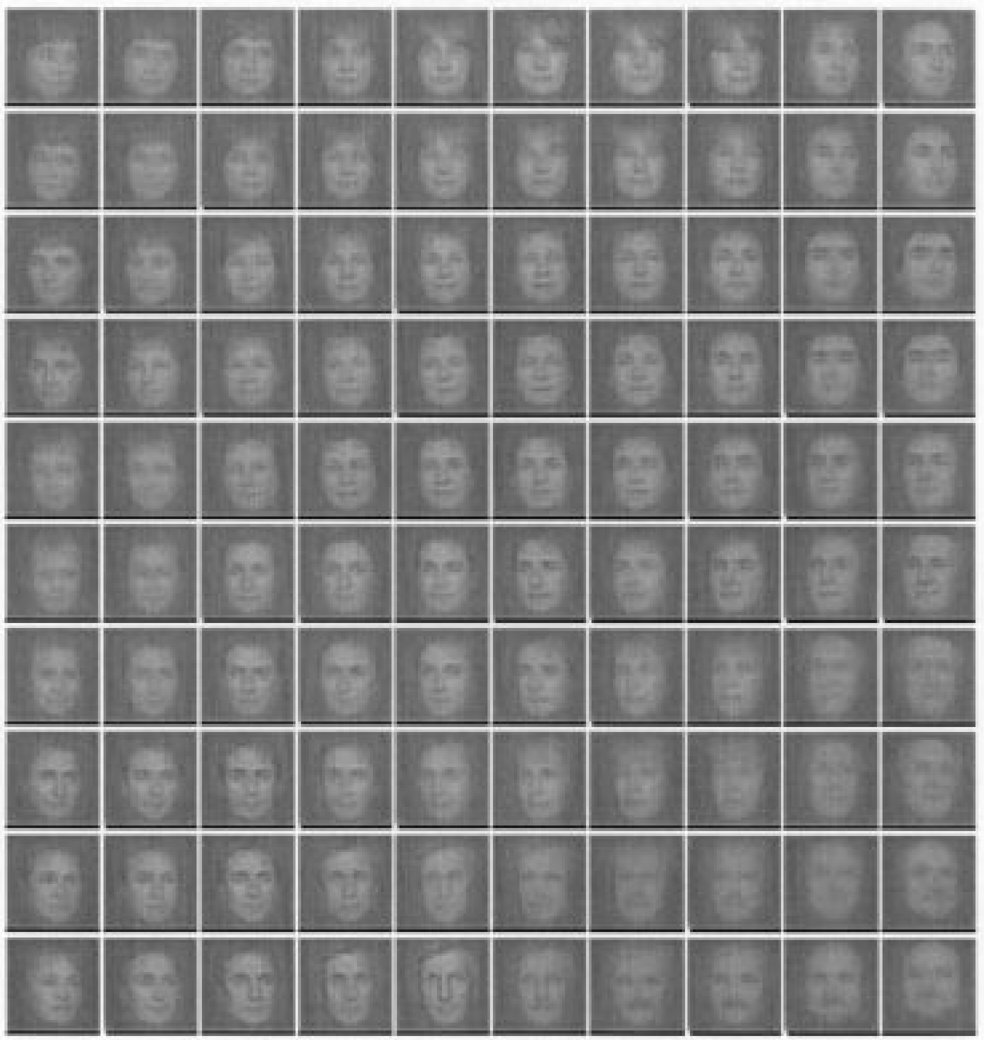

Fig. 12. Prototypes, which evolve by training a two-dimensional self-organizing map (10 × 10 neurons) with faces. Radial distances between these prototypes and input images are calculated and the resulting feature vector is fed into an LMS-network for face recognition.

in similar training errors (89–95%). Since the problems already appeared during learning, regularization such as weight decay or weight elimination for better generalization has not been applied here.

The bad performance of error backpropagation on this sample motivated other training strategies based on unsupervised learning of the hidden layer as described above. If the hidden layer was trained as a self-organizing map, the three layer network produced a 17.7% test error, while with auto-encoding learning 78.7% were misclassified. In contrast, the NEO had a 4.5% validation error on 32 × 32 pixel images, which is similar to the performance of the nearest neighbor classifier at higher input resolution (64 × 64 pixels). However the NEO requires fewer computations. In this experiment the NEO is approximately eight times faster than the nearest neighbor classifier.

C. Face Identification: Constrained Training—Unconstrained Validation

In order to determine if the classifiers, trained with the constrained data set, are capable of classifying less constrained images, a second validation set was frame-grabbed from a real life video as illustrated in Fig. 10. For such a data set all classifiers have significant problems identifying faces correctly. As shown the forth column of Table IV the misclassification rate is above 60% for all classifiers. The convolutional architecture is still relatively better compared to the other classifiers but is not able to generalize from constrained to unconstrained images.

D. Face Identification: Unconstrained Training—Unconstrained Validation

In the previous experiments the face recognition performance has been determined for training with faces under constrained conditions. If the recognition is to be improved for unconstrained images it is necessary to take these situations into account already during learning. However, it is relatively time consuming and difficult to collect various images under different pose and illumination from all persons to be recognized. Therefore, finally a restricted task is considered, where only one particular person is to be recognized and all other persons are to be rejected. Such a classifier is

TABLE IV
Validation Error for the Constrained (Section B, Fig. 9) and Unconstrained Face Sample (Section C, Fig. 10)

		constrained test sample (Section B)	unconstrained test sample (Section C)
	input resolution [pixel]	validation error [%]	validation error [%]
Nearest Neighbor Classifier	64 x 64	4.9	68.5
Nearest Neighbor Classifier	32 x 32	12.9	69.7
Multi-layer Perceptron	32 x 32	93.6	95.2
Self-organizing Map + LMS	32 x 32	17.7	71.3
Auto-encoding network + LMS	32 x 32	78.7	85.8
Modified Neocognitron	32 x 32	4.5	59.7

TABLE V
Confusion Matrix for Classification Between Person "A" and All Other Persons "Not A"

[%]	Training		Validation	
classified from/to	A	not A	A	not A
A	97.9	1.9	87.6	2.7
not A	2.1	98.1	12.4	97.3

for example necessary to verify the correspondence of a person with a personal authorization (passport, credit card, etc.). For this test 800 images of one person "A" have been grabbed randomly from a video under unconstrained conditions and the NEO has been trained to recognize this person. 1200 images containing 18 other persons have been used as examples for the rejection class "not A." The training results are shown in Table V, column 2 and 3 (training). For validation independent data sets with 300 images of "A" and 300 images of "not A" are used. Recognition rates of 87.4% and 97.3% for "A" and "not A," respectively, are achieved (Table V, column 3 and 4, validation), which is significantly better than results based on training with constrained data sets (Section C). The recognition rate for "not A" is significantly higher than for "A," because the classifier is biased toward correct classification of "not A" by using a larger training set of "not A" than of "A." This is reasonable, if security applications are considered.

V. Summary and Conclusions

In this article two types of convolutional neural networks—neocognitron and NEO—are examined. The NEO is introduced as a combination of neurons based on the perceptron with the weight sharing architecture of the neocognitron. Convolutional networks show several advantages compared to fully connected networks: they are more similar to biological neural networks, they can be easily migrated to parallel hardware and they are tolerant to small deformations of input patterns. Both networks are evaluated quantitatively based on visual recognition tasks. Handwritten digit classification and human face recognition are chosen to compare convolutional networks with fully connected classifiers.

Classification of handwritten digits showed that both neocognitron and NEO are superior to fully connected classifiers. For example after training with 1000 patterns per class (16 × 16 pixels) the NEO and the neocognitron had misclassification rates of 1.23 and 1.59%, respectively, compared to the nearest neighbor classifier, the perceptron, and the MLP with 2.31, 5.52, and 2.53% misclassification rates, respectively. The NEO has at least the same classification performance as the neocognitron even though its S-neurons are simpler. Both types of convolutional networks outperform the fully connected classifiers. If smaller learning sets are used, the performance difference is even larger. Further experiments on digit data sets which were subject to affine transformations (shift, rotation, scaling) within a 24 × 24 window confirmed the advantage of the convolutional architecture. For example, the NEO achieved a 3.08% error on digits with varying orientation while the MLP misclassified 5.70% of the same set (500 training samples per class). However, for digits with varying size, orientation, and position, the performance of both types of classifiers is significantly worse than for normalized digits. Apparently convolutional networks are only moderately tolerant to affine transformations.

In addition to the classification performance, the amount of computation required by the different classifiers also should be considered. On a serial computer, the neocognitron and the NEO require eight times less computations for the digit classification example than does the nearest neighbor classifier. The neocognitron and the NEO are about ten times slower than the MLP. However, it has to be taken into account that the convolutional networks achieved the best classification accuracy in this comparison and they can be accelerated more easily on a parallel computer with local communication than can fully connected networks.

The experiments on face recognition indicate that for reliable recognition good alignment of the faces is essential for each type of classifier. For accurate localization of faces a three-layer MLP is trained for detection of faces with fixed size. A window with varying size and orientation is moved over the whole scene. The resulting subimages are normalized with respect to brightness and contrast and evaluated by the face detection network. The face identification experiments with eighteen persons demonstrate, that the convolutional neural network (NEO) outperforms fully connected networks. The respective misclassification rates are: NEO, 4.5%; nearest neighbor classifier, 12.9%; self-organizing map, 17.7%; autoencoding network, 78.7%; and MLP, 93.6%.

The good recognition rates for nearest neighbor classifier and NEO decline significantly (Section C) when the classifiers are tested under more unconstrained conditions with respect to pose and illumination than the data sets on which they were trained. This indicates, that for classification of unconstrained face images the training sample has to cover more variations of pose, illumination, and background. The influence of varying background is less important, if the face localization and segmentation works properly. However, it can be observed in Fig. 11, that focusing on the inner circle of the face region cannot completely eliminate the background, since a face is not perfectly circular. In addition illumination changes and shading are not completely removed by contrast and brightness normalization. Further preprocessing steps such as contour extraction or high pass filtering have been avoided, since the first layer of the convolutional network already performs edge extraction. In addition to illumination and background, out of the plane rotations of faces are responsible for the declining classification performance on unconstrained face images.

The influence of out of the plane rotations and variations in illumination requires a sophisticated training set that covers the spectrum of possible face appearances as well as possible. Therefore in the last identification experiment (Section D), a larger and more general training set from real live video sequences has been gathered from one particular person. Based on the extended training set it is possible to recognize this person reliably among the other individuals under unconstrained conditions (correct recognition 87.6%, correct rejection 97.3%). However, this approach is quite impractical for many problems since a lot of instances are required from each person to be recognized.

Future attention will be focused on the representation of high level face features for improved generalization. These features have to be trained by a large amount of image data from various persons in order to cover most aspects of pose and illumination. The generation of virtual views proposed by [2] may help to further improve identification accuracy. In addition to the applications discussed here, convolutional networks will be used as classifiers in the automatic x-ray inspection of solder joints in electronic production, where three-dimensional data sets have to be evaluated. Neural networks have already been successfully applied to defect detection and classification of solder joints [21], [23] and convolutional networks combined with computer tomography will help to further improve quality control of printed circuit boards.

ACKNOWLEDGMENT

This work was performed at Fraunhofer Institute for Integrated Circuits, Erlangen, Germany. The author would like to thank Prof. Seitzer very much for his support.

REFERENCES

[1] N. M. Allinson, A. W. Ellis, B. Flude, and A. Luckman, “A connectionist model of familiar face recognition,” in *Inst. Elect. Eng. Colloquium on Machine Storage and Recognition of Faces*, 1992, pp. 5.1–5.9.

[2] D. Beymer and T. Poggio, “Image representations for visual learning,” *Science*, vol. 272, pp. 1905–1909, 1996.

[3] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[4] H. Bouattour, F. Fogelman Soulie, and E. Viennet, “Solving the human face recognition task using neural nets,” in *Artificial Neural Networks II*, I. Alexander and J. Taylor, Eds. Amsterdam, the Netherlands: North-Holland, 1992, pp. 1595–1598.

[5] H. Bourlard and Y. Kamp, “Autoassoziation by multilayerperceptrons and singular value decomposition,” *Biol. Cybern.*, vol. 59, pp. 291–294, 1988.

[6] G. W. Cottrell, “EMPATH: Face, emotion, and gender recognition using holons,” in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, vol. 3, 1991, pp. 564–571.

[7] K. Fukushima, “Neocognitron: A self-organizing neural-network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, pp. 193–202, 1980.

[8] ——, “A neural-network model for selective attention in visual pattern recognition,” *Biol. Cybern.*, vol. 55, pp. 5–15, 1986.

[9] ——, “Analysis of the process of visual pattern recognition by the neocognitron,” *Neural Networks*, vol. 2, pp. 413–421, 1989.

[10] K. Fukushima and T. Imagawa, “Recognition and segmentation of connected characters with selective attention,” *Neural Networks*, vol. 6, pp. 33–41, 1993.

[11] K. Fukushima and M. Tanigawa, “Use of different thresholds in learning and recognition,” *Neurocomputing*, vol. 11, pp. 1–17, 1996.

[12] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *J. Physiology*, vol. 160, pp. 106–154, 1962.

[13] ——, “Receptive fields and functional architecture in two nonstriate visual areas (18 und 19) of the cat,” *J. Neurophysiology*, vol. 28, pp. 229–289, 1965.

[14] T. Ito and K. Fukushima, “Realization of a neural-network model neocognitron on a hypercube parallel computer,” *Int. J. High-Speed Computing*, vol. 2, pp. 1–16, 1990.

[15] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Trans. Comput.*, vol. 42, pp. 300–311, 1993.

[16] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, pp. 541–551, 1989.

[17] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series” in *The Handbook of Brain Science and Neural Networks*, M. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258.

[18] C. Neubauer, “Fast detection and classification of defects on treated metal surfaces using a backpropagation neural network,” in *Proc. IJCNN*, Singapore, 1991, pp. 1148–1153.

[19] ——, “Shape, position, and size invariant visual pattern recognition based on principles of neocognitron and perceptron,” in *Artificial Neural Networks*, I. Alexander and J. Taylor, Eds. Amsterdam, the Netherlands: North-Holland, vol. 2, 1992, pp. 833–837.

[20] ——, “Segmentation of defects in textile fabric,” in *Proc. ICPR*, Den Haag, the Netherlands, 1992, pp. 688–691.

[21] C. Neubauer and R. Hanke, “Improving x-ray inspection of printed circuit boards by integration of neural-network classifiers,” in *Proc. IEMT*, Santa Clara, CA, 1993, pp. 14–18.

[22] C. Neubauer, “Modellierung visueller erkennungsvorgänge mit neuronalen netzen,” Ph.D. dissertation, Dept. Technische Elektronik, Univ. Erlangen-Nürnberg, Germany, 1995.

[23] ———, "Intelligent x-ray inspection for quality control of solder joints," *IEEE Trans. Comp., Packag., Manufact. Technol.-C*, vol. 20, pp. 111–120, 1997.

[24] D. Rumelhart and J. McClelland, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. Cambridge, MA: MIT-Press, 1986.

[25] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, pp. 71–86, 1991.

[26] U. Schramm, T. Wagner, S. Schmölz, K. Spinnler, F. Böbel, R. Haas, and H. Haken, "A practical comparison of synergetic computer, restricted Coulomb energy networks and multilayer perceptron," in *Proc. WCNN 93*, Portland, OR, vol. 3, pp. 657–660, 1993.

[27] A. S. Weigend, D. Rumelhart, and B. Huberman, "Generalization by weight-elimination with application to forecasting," in *Advances in Neural Information Processing*. San Mateo, CA: Morgan Kaufmann, vol. 3, 1991, pp. 875–882.

[28] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Dept. Appl. Math., Havard Univ., Cambridge, MA, 1974.

Claus Neubauer (M'97) was born in Nürnberg, Germany, in 1963. He received the diploma degree in physics from the University of Regensburg, Germany, and the Ph.D. degree in electrical engineering from the University of Erlangen, Nürnberg, in 1988 and 1995, respectively.

In 1990 he joined the Fraunhofer Institute for Integrated Circuits (IIS-A) in Erlangen, Germany. He was concerned with computer vision projects and automated X-ray inspection for quality control. In 1996 he visited the International Computer Science Institute (ICSI), Berkeley, CA. Since 1998 he has been with the Department Imaging and Visualization at Siemens Corporate Research, Inc., Princeton, NJ, where he is focusing on machine learning algorithms for industrial inspection systems. His research interests include X-ray inspection of solder joints, computed tomography, machine learning, and neural-network algorithms for visual pattern recognition.