Optical Character Recognition Using Novel Feature Extraction & Neural Network Classification Techniques

B. Gatos, D. Karras and S. Perantonis
Institute of Informatics and Telecommunications
National Research Center "Demokritos"
153 10 Agia Paraskevi, Athens, Greece

Abstract

This paper describes two novel techniques applied to the feature extraction and pattern classification stages in an OCR system for typeset characters. A technique for estimating the class discrimination ability of continuous valued features is presented leading to the formation of complex features which facilitate the classification stage. Next, a neural network classifier trained using a recently proposed powerful training algorithm, based on rigorous nonlinear programming methods, is applied to large-scale OCR problems involving typeset Greek characters and found to exhibit good generalization capabilities compared to other conventional and artificial neural network (ANN) classifiers. Combining these feature extraction and classification techniques in a unified software platform, we have designed an OCR system which achieved high recognition rates in some real world OCR experiments.

1 Introduction

Optical character recognition aims at the transformation of any written document which is created for human comprehension into an equivalent symbolic representation accessible for machine elaboration. In order to achieve optical character recognition starting with a digitized document, many individual steps must be put to use. Thus, *preprocessing* (rotation of the document to remove some skewing variation, image filtering to minimize the noise etc.), *segmentation* (separation of individual characters using horizontal and vertical vacant lines, contour following etc.), *normalization* (fitting the character to a certain grid), *feature extraction* (transformation of characters to keep only essential information) and *classification* (assigning each character to a certain class) are important stages in the recognition process, each individually affecting recognition accuracy. In order to design a successful OCR system, it is essential to combine powerful algorithms for each of these stages [1,2].

Our research team have designed and implemented a flexible OCR software platform in order to test and improve methods for all OCR stages, investigating novel techniques at each stage. In this work we concentrate on feature extraction and classification. We incorporate new powerful techniques into our platform, which improve both the separability of the extracted features and the accuracy of the classifying process. The resulting OCR system is used for the recognition of typeset Greek characters.

As regards feature extraction, preliminary features are extracted from the characters using a mask which is applied successively on the character surface. A major contribution of this paper lies in the use of an algorithm which evaluates the class discrimination ability of each preliminary feature. The obtained estimate is then used to form complex and more efficient features. The proposed technique is a continuous valued generalization of a method based on coding theory, which has been successfully applied to binary pattern classification [3,4].

As regards the classification stage, a multilayered feedforward ANN is implemented, which is trained by a recently proposed constrained learning algorithm [5] utilizing rigorous non-linear programming methods in order to make optimal use of momentum acceleration. The efficiency of ANNs as classifiers in typeset character recognition tasks is well documented [6]. Since OCR tasks involve large datasets, it is essential to devise and implement ANN classification algorithms with good scalability properties. In this context, our constrained learning algorithm – which was tested in previous related work on large-scale benchmark tasks and was found superior to other popular ANN training algorithms in terms of learning speed and convergence ability [7] – is a good candidate for training ANN classifiers applied to OCR tasks.

0-8186-5845-2/94 $3.00 © 1994 IEEE

The proposed combination of feature extraction technique and ANN classifier is applied with success to a large database of Greek typeset characters corresponding to many different fonts. Finally, options of other classical and ANN classifiers are also provided for reasons of comparison with the proposed techniques.

This paper is organized as follows: In section 2 we describe our feature extraction techniques. In section 3 we present our ANN training algorithm and discuss its relevance to OCR problems. Section 4 deals with the organization of our OCR experiments, whose results are presented in section 5. Finally, section 6 puts this work in perspective by setting goals for future research.

2 The Feature Extraction Stage

2.1 Convolutional Feature Extraction

The goal of pattern recognition is to assign input patterns $\mathbf{X}^p = (x_1^p, x_2^p, .., x_n^p)$ with $x_i^p \in \Re$, to one of a finite number of classes R, where p is the pattern indicator. One of the most important issues for pattern recognition is feature extraction [8]: Problem-dependent efficient pattern representation schemes should be chosen so as to facilitate the subsequent classification stage of the process.

The main problem of the feature extraction stage in an OCR system for typeset characters is the noise present in the rastered pattern images because of differences in the scanner and printer models used to acquire the data, as well as in the ambient illumination. Problems due to deformation or rotation are not normally encountered. In effect, our pattern representation scheme was designed so as to simultaneously achieve:

- Filtering out of the noise in the pattern images by the successive application of a low pass filtering procedure.
- Efficient image encoding, that takes into account that neighboring pixels are highly correlated [9], and forms a compressed pattern image representation of reduced dimensionality by removing redundant information. This is very important for the ANN classifiers in the next stage of the proposed system, since it is well known that improved generalization performance is dependent on the number of free parameters of the model [10,11].

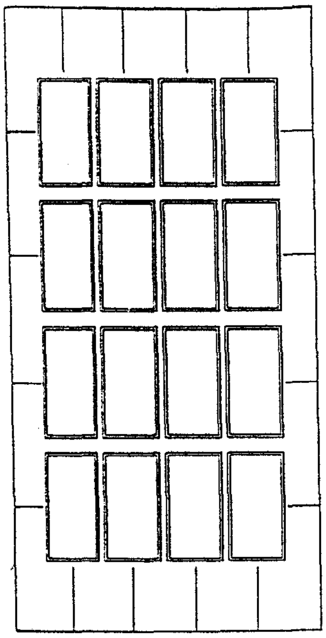
These criteria led directly to the employment of the Laplacian pyramid theory [9]. In the proposed OCR system a one level Gaussian pyramid was used with the following features:

- Two generating kernels (gaussian masks), were used (a 5-by-5 mask and a 4-by-4 mask). The pattern of weights $h(m,n)$ of the generating kernels was chosen subject to the constraints of normalization and equal contribution for the different levels [12], as well as equality for all the weights at a given level. This last constraint was selected for its simplicity instead of the separability one [9] and our experiments validated its efficiency.
- In order to achieve higher dimensionality reduction rates, we did not apply the convolution scheme required in [9]. Instead of performing the normal convolution operation as in [9,13], a different, more complex, convolution scheme of the original rastered image g_0 was employed. The resulting pattern image is derived as follows: We define $g_{l1} = h_5 \bigoplus g_0$ and $g_{l2} = h_4 \bigoplus g_0$ as the convolved images obtained by convolving g_0 (the original pattern) with the generating kernels h_5 (5-by-5) and h_4 (4-by-4) respectively. This scheme takes into account different correlations of the neighboring image pixels. The final pattern image g_l, which is the input of the classifier, is obtained by keeping the most important terms of g_{l1} and g_{l2} as shown in Fig. 1.

Using the above described procedure we obtain a training set of continuous valued convolutional features $\mathbf{X}_l^{k,p} = (x_{l,1}^{k,p}, x_{l,2}^{k,p}, .., x_{l,n}^{k,p})$ with $x_{l,i}^{k,p} \in \Re$, where $k = 1, 2, \ldots R$ is an indicator of a known class, p is a pattern indicator and l is the training set indicator. We also obtain a test set of continuous valued patterns $\mathbf{X}_t^{k,p} = (x_{t,1}^{k,p}, x_{t,2}^{k,p}, .., x_{t,n}^{k,p})$ where, k labels the unknown class to which we want to assign pattern $\mathbf{X}_t^{k,p}$ and t is the test set indicator.

2.2 Class Discrimination Ability Estimators

In the literature, all features $x_{l,i}^{k,p}$ corresponding to a pattern p are placed on the same footing, despite the fact that some of these can be more important for discriminating pattern p from patterns in other classes.



41 masks applied to the character matrix

1/112	1/112	1/112	1/112	1/112
1/112	1/28	1/28	1/28	1/112
1/112	1/28	4/7	1/28	1/112
1/112	1/28	1/28	1/28	1/112
1/112	1/112	1/112	1/112	1/112

5x5 weighted mask

1/36	1/36	1/36	1/36
1/36	1/6	1/6	1/36
1/36	1/6	1/6	1/36
1/36	1/36	1/36	1/36

4x4 convolutional weighted mask

Fig. 1 - Convolutional feature extraction.

Convolutional Features

α					ο					υ				
0.00	0.36	0.98	0.07	0.00	0.00	0.86	0.92	0.08	0.00	0.00	1.00	0.98	0.08	0.00
0.11	0.52	0.61	0.00		0.13	0.47	0.55	0.00		0.25	0.16	0.25	0.33	
0.00	0.35	0.01	0.82	0.00	0.00	0.32	0.00	0.82	0.00	0.00	0.00	0.00	0.86	0.00
0.50	0.00	**0.13**	0.16		0.33	0.00	**0.13**	0.22		0.33	0.00	**0.13**	0.77	
0.00	0.15	0.00	0.82	0.00	0.00	0.16	0.00	0.90	0.00	0.00	0.00	0.00	0.91	0.00
0.83	0.16	0.305	0.66		0.33	0.00	0.30	0.69		0.66	0.00	0.47	0.72	
0.00	**0.86**	0.01	0.82	0.00	0.00	**0.85**	0.01	0.82	0.00	0.00	**0.04**	0.05	0.32	0.00
0.16	0.19	0.27	0.00		0.03	0.19	0.25	0.00		0.11	0.22	0.27	0.00	
0.00	0.99	0.98	0.78	0.00	0.00	0.78	0.96	0.04	0.00	0.00	1.00	0.98	0.03	0.00

Discrimination Ability Estimators

α					ο					υ				
0.00	0.07	0.03	0.01	0.00	0.30	0.07	0.06	0.00	0.00	0.00	0.14	0.03	0.30	0.30
0.08	0.20	0.21	0.16		0.07	0.18	0.18	0.16		0.13	0.33	0.33	0.33	
0.00	0.44	0.01	0.02	0.00	0.00	0.42	0.00	0.02	0.00	0.00	0.83	0.00	0.04	0.00
0.17	0.00	**0.00**	0.33		0.08	0.00	**0.00**	0.30		0.08	0.00	**0.30**	0.58	
0.00	0.08	0.00	0.08	0.00	0.00	0.08	0.00	0.04	0.00	0.00	0.16	0.00	0.05	0.00
0.03	0.16	0.08	0.04		0.08	0.28	0.08	0.03		0.17	0.08	0.27	0.04	
0.00	**0.41**	0.02	0.00	0.00	0.00	**0.41**	0.02	0.00	0.00	0.00	**0.81**	0.04	0.30	0.00
0.06	0.01	0.01	0.00		0.35	0.01	0.02	0.00		0.04	0.03	0.01	0.00	
0.00	0.11	0.01	0.74	0.00	0.00	0.21	0.02	0.37	0.00	0.00	0.11	0.01	0.38	0.00

Fig. 2 - Convolutional features and corresponding CDAE for characters α, o and υ. Note the small values of the CDAE when features cannot be discriminated and the large values when features differ significantly (bold figures).

A major novelty of this paper lies in the estimation of the "class discrimination ability" of each feature. To this end, a simple algorithm is used, which is a continuous valued generalization of the one based on coding techniques and presented in [3,4]. The main idea of this new algorithm is to define a class discrimination ability estimator (CDAE) $z^k_{l,i}$ for each feature $m^k_{l,i}$ of the average prototype (derived from the training set) of each class k and assign to it a value according to the ability of $m^k_{l,i}$ to discriminate class k from every other class v, as follows:

$$z^k_{l,i} = \frac{\sum_{v \neq k} \mid m^k_{l,i} - m^v_{l,i} \mid}{R-1} \qquad (1)$$

where

$$m^k_{l,i} = \frac{\sum_{p=1}^{k_p} x^{k,p}_{l,i}}{k_p} \qquad (2)$$

is the average of feature $x^{k,p}_{l,i}$ over the k_p patterns of the training set belonging to class k.

From the above formulas, it is straightforward to see that the larger the value of a CDAE $z^k_{l,i}$, the better the ability of the corresponding feature $x^{k,p}_{l,i}$ to discriminate class k from an "average representative" of all other $R-1$ classes in the training set (Fig. 2).

Despite its simplicity and early stage of development, this technique can be very useful in the formation of more complex features as explained next. We should mention that this method has a great potential to improve, especially if the CDAE are properly defined so as to discriminate efficiently classes having similar continuous valued features.

In order to demonstrate the efficiency of our technique, we first design a weighted minimum distance classifier (WMDC) having as discriminant function weights the above derived CDAE and we compare it to the classical minimum distance classifier (MDC) [8]. The results are shown in the experimental section of this paper and prove the efficiency of our approach.

After demonstrating the classification capabilities of our method, we try to incorporate it in the formation of more complex features in the feature extraction stage of an OCR system having as classifier the ANN pattern classifier analyzed in the next section. To this end, given any pattern of the training set $\mathbf{X}^{k,p}_l = (x^{k,p}_{l,1}, x^{k,p}_{l,2}, .., x^{k,p}_{l,n})$, we form another pattern $\mathbf{Y}^{k,p}_l = (y^{k,p}_{l,1}, y^{k,p}_{l,2}, .., y^{k,p}_{l,n})$, where $y^{k,p}_{l,i} = x^{k,p}_{l,i}(z^k_{l,i})^{1/4}$ and we incorporate it into the training set along with its parental pattern $\mathbf{X}^{k,p}_l$. This novel technique aims at training the network so as to interpolate between any original detailed pattern and its transformed class "principal" feature pattern. It bears some similarity to the "signal+noise" training methods [14] and to statistical resampling techniques [15]. The results of the application of the combination of this feature extraction technique and of the ANN pattern classifier (see section 3) demonstrate the potential of our approach.

3 ANN Learning Algorithm

3.1 Relevance to OCR Tasks

ANNs have been hailed in recent years for their potential and ability to provide efficient solutions to classification, function approximation, control and optimization problems. Although it is often claimed that these networks can offer viable solutions to interesting technical problems, there is a general feeling that their full potential in real world applications remains largely unexplored [16]. It is also widely claimed that ANNs have the attributes of massive parallelism and fault tolerance, while at the same time many related papers deal only with small applications and models where these properties are absent [17].

To fill in the gap between theory and real world applications, it is imperative to study the performance of ANN learning algorithms in large scale problems and networks. In our opinion, OCR tasks are good examples of such large-scale problems: Usually, it is not only possible, but also essential to include a large dataset of different examples of each character in the training set, if acceptable recognition rates are to be achieved. Moreover, the relatively large number of categories and input features calls for networks with a large number of weights. Under these circumstances, stringent requirements are placed on learning algorithms in terms of speed, scalability properties and generalization capabilities.

In a series of papers [18,5,19,20,7,21] two of the authors have proposed that a successful way of improving these properties of ANN learning algorithms is to incorporate different forms of knowledge about learning in ANN in the form of well defined constrained optimization tasks. We have introduced a framework of basic requirements for incorporating knowledge in ANN learning algorithms [18]. This framework was used as an engine for developing different Algorithms for Learning Efficiently using Constrained Optimization techniques (ALECO). The second in the series of these algorithms (ALECO-2) was based on the incorporation of knowledge about optimal use of momentum acceleration techniques. This algorithm was tested using several different benchmark training tasks (encoder, multiplexer, counter and XOR problems) and its performance was compared to that of several different ANN training algorithms [5,20]. As regards large-scale problems (multiplexers and encoders with up to 2048 input patterns and 4360 weights), which are of interest in this work, ALECO-2 outperformed all other algorithms in terms of convergence ability, learning speed and reliability of performance.

These results illustrate the excellent capabilities of ALECO-2 as regards learning speed and scalability properties and make it an excellent candidate for training ANNs to solve large-scale problems such as the OCR tasks studied here. This paper presents our first opportunity to assess the generalization ability of ALECO-2 using extensive multifont character datasets and compare it to the ability of on-line BP, whose good performance in large-scale problems is well known [22].

3.2 Derivation of ALECO-2

Augmenting the BP algorithm with momentum is inherently heuristic in nature. The use of momentum is based on the expectation that bigger weight steps can be achieved by filtering out high frequency variations of the error surface in the weight space. ALECO-2 is based on the idea of obtaining *optimal* weight steps by optimizing, at each epoch of the learning algorithm, the euclidean distance between the current and previous epoch weights [5]. In this way, improved learning speed is achieved.

Consider a multilayered feedforward ANN with one layer of input, M layers of hidden and one layer of output units. The units in each layer receive input from all units in the previous layer. We denote the unit outputs and synaptic weights respectively by $O_{jp}^{(m)}$ and $w_{ij}^{(m)}$. The superscript (m) labels a layer within the structure of the ANN ($m = 0$ for the input layer, $m = 1, 2, \ldots, M$ for the hidden layers, $m = M + 1$ for the output layer), i and j denote units in layers $(m - 1)$ and (m) respectively and p labels the input patterns.

The training procedure in ALECO-2 solves, *for each epoch*, the following problem: First, change the cost function

$$E = \frac{1}{2}\sum_{jp} \varepsilon_{jp}, \;\; \varepsilon_{jp} = \left(T_{jp} - O_{jp}^{(M+1)}\right)^2 \qquad (3)$$

by a specified negative amount δE. After a sufficient number of epochs, the accumulated changes to the cost function should suffice to achieve the desired input-output relation. Second, simultaneously maximize the squared euclidean distance

$$\Phi = \sum_{ijm} \left(w_{ij}^{(m)} - W_{ij}^{(m)}\right)^2 \qquad (4)$$

between the weight vectors $\boldsymbol{w}$ at the present epoch and $\boldsymbol{W}$ at the immediately preceding epoch, in order to achieve optimal weight steps. This problem is solved in an elegant way by a straightforward generalization of the optimal control method introduced by Bryson and Denham [23].

ALECO-2 is an iterative procedure, whereby the synaptic weights $w_{ij}^{(m)}$ are changed by small amounts $dw_{ij}^{(m)}$ at each iteration so that the quadratic form $\sum_{ijm} dw_{ij}^{(m)} \cdot dw_{ij}^{(m)}$ takes on a prespecified value $(\delta P)^2$. Thus, at each epoch, the search for an optimum new point in the weight space is restricted to a small hypersphere centered at the point defined by the current weight vector. If δP is small enough, the changes in E and Φ induced by changes in the weights can be approximated by the first differentials dE and $d\Phi$. The problem then amounts to determining, for given values of δP and δE, the values of $dw_{ij}^{(m)}$, so that the maximum value of $d\Phi$ is attained.

Maximization of $d\Phi$ is attempted with respect to $w_{ij}^{(m)}$ and $O_{jp}^{(m)}$. In the language of non-linear programming, the synaptic weights correspond to decision variables and the unit outputs correspond to state (solution) variables. These quantities must satisfy the state equations, i.e. the constraints describing the network architecture

$$f_{jp}^{(m)}(O,w) = g\left(\sum_i w_{ij}^{(m)} O_{ip}^{(m-1)}\right) - O_{jp}^{(m)} = 0 \quad (5)$$

where g is the logistic function $g(x) = 1/(1+\exp(-x))$. Biases are treated as weights emanating from units with constant, pattern-independent activation equal to one. Apart from the state equations, the following conditions should be satisfied at each epoch of the algorithm

$$dE - \delta E = 0, \quad \sum_{ijm} dw_{ij}^{(m)} \cdot dw_{ij}^{(m)} - (\delta P)^2 = 0 \quad (6)$$

To maximize $d\Phi$, suitable Lagrange multipliers $\lambda_E^{jp(m)}$, $\lambda_\Phi^{jp(m)}$ of the $f_{jp}^{(m)}$ are introduced to take account of the architectural constraints. Two further multipliers λ_1 and λ_2 are also needed to take account of the respective terminal conditions 6. Demanding that $d\Phi$ be maximum ($d^2\Phi = 0$, $d^3\Phi < 0$), we are led to the following equations for the Lagrange multipliers

$$\lambda_E^{jp(M+1)} = O_{jp}^{(M+1)} - T_{jp},\ \lambda_E^{ip(m)} = \textstyle\sum_j \lambda_E^{jp(m+1)}$$
$$w_{ij}^{(m+1)} O_{jp}^{(m+1)} \left(1 - O_{jp}^{(m+1)}\right),\ 1 \le m \le M \quad (7)$$

$$\lambda_\Phi^{jp(m)} = 0, \quad \lambda_2 = \tfrac{1}{2}\left[\tfrac{I_{EE}(\delta P)^2 - (\delta E)^2}{I_{\Phi\Phi} I_{EE} - I_{E\Phi}^2}\right]^{-1/2},$$
$$\lambda_1 = (I_{E\Phi} - 2\lambda_2 \delta E)/I_{EE} \quad (8)$$

where

$$I_{\Phi\Phi} = \textstyle\sum_{ijm} (F_{ijm})^2, \quad I_{EE} = \sum_{ijm} (J_{ijm})^2,$$
$$I_{E\Phi} = \textstyle\sum_{ijm} J_{ijm} F_{ijm} \quad (9)$$

with

$$F_{ijm} = 2\left(w_{ij}^{(m)} - W_{ij}^{(m)}\right),$$
$$J_{ijm} = \textstyle\sum_p \lambda_E^{jp(m)} O_{jp}^{(m)} \left(1 - O_{jp}^{(m)}\right) O_{ip}^{(m-1)} \quad (10)$$

Moreover, the following updating rule is obtained for the weights:

$$dw_{ij}^{(m)} = \tfrac{1}{2\lambda_2}(F_{ijm} - \lambda_1 J_{ijm}) = \left[\tfrac{I_{EE}(\delta P)^2 - (\delta E)^2}{I_{\Phi\Phi} I_{EE} - I_{E\Phi}^2}\right]^{1/2}$$
$$\left(F_{ijm} - \tfrac{I_{E\Phi}}{I_{EE}} J_{ijm}\right) + \tfrac{J_{ijm}}{I_{EE}} \delta E \quad (11)$$

The detailed calculations are given in [5]. Note the bound $|\delta E| \le \delta P \sqrt{I_{EE}}$ set on the value of δE by equation 11. We always use a value $\delta E = -\xi \delta P \sqrt{I_{EE}}$ where ξ is a constant between 0 and 1. Thus δP and ξ are the only free parameters of the algorithm which can be tuned to obtain optimal performance. It is shown in [5] that this guarantees convergence to global or local minima of the cost function for small enough δP.

4 Experimental Study

Extensive experiments were conducted to test the efficiency of our novel feature extraction and ANN classification techniques on specific OCR tasks involving typeset Greek characters. These experiments involve different combinations of features – convolutional features and CDAE – and classifiers – MDC, WMDC, on-line BP and ALECO-2.

For the purposes of this paper, we use in all experiments the same preprocessing, segmentation and normalization regulations. At the preprocessing option we use a filter to remove pixel regions under a certain threshold (noise regions). At the segmentation option we select horizontal and vertical vacant lines segmentation. Finally, at the normalization option we use a 25x25 normalization with the character width as the width boundary and the boundary of the text line as the height boundary.

4.1 Training and Test Character Sets

For reasons of fair comparison, all experiments were conducted using the same character sets for training and recognition. Both natural and artificial sets were used. The natural sets were produced using images of plain text of more than 1500 Greek characters. The artificial sets were made from images containing 10 versions of 32 different characters (lower-case Greek letters) for a total of 320 characters. We used 6 different character fonts both in the artificial and the natural sets which are: arial, arc, times, bold arial, bold arc and bold times. We also used 2 different contrast regulations in the scanning software for the character sets which provides 2 different character thicknesses. The scanning resolution was stable for all the experiments at 300 dpi though the size of the characters varied from set to set. We chose

- the junction of arial, arc and times artificial character bases (total of 960 characters) as our training set. More specifically, the training set consisted of the following:

  arc set: We used an HP scanner at 300 dpi resolution with the default contrast regulation in the scanning software in order to binarize a text printed from an HP printer with 320 characters (32 classes with 10 prototypes each) from MS Windows arc font. We produced a 731K TIF image from which we derived the arc set.

  arial set: The same procedure as above was followed using the MS Windows arial font.

  times set: The same procedure as above was followed using the MS Windows times font.

- Various artificial and natural character sets were used for recognition (testing) purposes. In particular, 3 natural sets of arial and times fonts (total of 4647 characters) and 3 artificial sets of bold arial, bold arc and bold times fonts (total of 960 characters) were used as test sets. These recognition sets have the following specifications:

  text1 set: The scanner, printer and scanning software contrast regulation were different from those used in the training phase. We had 1554 characters in a plain text from MS Windows arial font at 300 dpi resolution in a 629K TIF file.

  text2 set: We had different scanner, printer and contrast regulation from those used in the training phase. The character set consisted of 1508 characters in a plain text from MS Windows times font at 300 dpi resolution in a 612K TIF file.

  text3 set: The scanner and printer were different from those used at the training phase, but the contrast regulation was the same as in the training phase. We had 1585 characters in a plain text from MS Windows arial font at 300 dpi resolution in a 616K file.

  barc set: The scanner, printer and scanning software contrast regulation were those used in the training phase. We had 320 characters (32 classes with 10 prototypes each) from MS Windows bold arc font at 300 dpi resolution in a 731K file.

  barial set: The same as with barc set with characters from MS Windows bold arial font.

  btimes set: The same as with barc set with characters from MS Windows bold times font.

4.2 Types of Experiments Conducted

The following architectures combining different features and classifiers were implemented:

1. A minimum distance classifier was trained using the 41 convolutional weighted mask features $x_{l,i}^{k,p}$ as inputs. Various distance metrics were used and the best recognition accuracy results, presented in Table 1, were achieved using a fourth power metric $D(k) = \left[\sum_i (x_{l,i}^{k,p} - m_{l,i}^{k})^4\right]^{1/4}$ (see section 2).

2. A weighted minimum distance classifier with weights depending on the CDAE $z_{l,i}^{k}$ was trained using the $x_{l,i}^{k,p}$ as inputs. The discriminant functions which gave the best recognition accuracy results were of the form $D(k) = \left[\sum_i (z_{l,i}^{k})^2 (x_{l,i}^{k,p} - m_{l,i}^{k})^4\right]^{1/4}$.

3. A fully connected feedforward network with two layers of weights and a 41-40-32 architecture was trained using on-line BP as the training algorithm with a learning rate of 0.4 and a momentum acceleration factor of 0.5. For each character belonging to a category k, the convolutional features $x_{l,i}^{k,p}$ were used as the network input. The desired output of all output nodes was 0, except for the k-th node, whose desired output was 1. A set of characters distinct from the training and test sets was used as a validation set: The final weights

used for testing were those for which recognition accuracy in the validation set during training had reached its maximum value.

4. A feedforward network with the same architecture, inputs and desired outputs as in case 3 was trained using ALECO-2 as learning algorithm. The parameter values $\delta P = 0.5$ and $\xi = 0.5$ were used.

5. A feedforward network with the same architecture and outputs as in cases 3 and 4 was trained using ALECO-2 (with the same learning parameters as in case 4). However, the CDAE were incorporated in the training procedure following the methodology explained in section 2. Thus, for each character in the training set, two training patterns were presented to the network as input, one consisting of the $x_{l,i}^{k,p}$ and one consisting of the quantities $y_{l,i}^{k,p} = x_{l,i}^{k,p}(z_{l,i}^{k})^{1/4}$. This led to an effective training set of 1920 patterns.

5 Results

Our results regarding the recognition accuracy of different feature extraction-classification combinations are summarized in Table 1.

	Ex#1	Ex#2	Ex#3	Ex#4	Ex#5
text1	97.94	98.13	99.16	99.09	**99.74**
text2	93.77	95.89	98.67	**99.07**	**99.07**
text3	99.37	99.56	99.56	99.75	**99.87**
barc	85.31	85.94	92.18	**95.31**	94.69
barial	91.25	92.50	**97.18**	96.88	95.62
btimes	86.88	86.31	93.75	95.31	**97.81**

Table 1 - Classification accuracy results for the characters in our 6 different test sets using various feature-classifier combinations. Each column corresponds to an experiment described in section 4.2.

From these results the following conclusions can be drawn:

- The use of CDAE helps improve recognition accuracy results. This is evident in the results obtained using the WMDC with weights depending on the CDAE. Note that in five out of six test sets, the recognition accuracy was improved. This supports the conclusion that CDAE can be successfully applied not only to binary patterns (as in [3,4], but also in the case of continuously valued features.

- ALECO-2 emerges as a powerful neural network training algorithm for large-scale OCR tasks. In our experiments, the excellent speed and scalability properties of ALECO-2 were confirmed: Convergence of the training procedure for experiment no. 5 in a network with 2992 weights and a task with 1920 training patterns was achieved in just 150 epochs (using Fahlman's 0.4-0.6 criterion [24]). Moreover, our results show attractive generalization ability properties: Compared with on-line BP, ALECO-2 achieved better recognition rates in 4 out of 6 test sets, including substantial improvements in the barc and btimes test sets; in the remaining two test sets, its recognition accuracy was marginally inferior to that of on-line BP. The good generalization ability of ALECO-2 can probably be attributed to the fact that the cost function is changed monotonically and gradually [5], without the abrupt jumps sometimes involved in learning algorithms which incorporate heuristics in their formulation (including on-line BP). Note that in the same spirit of constrained learning, it is possible to augment ALECO-2 with weight elimination techniques [10] which will hopefully further improve its generalization ability without adverse effect on its learning speed.

- Finally, the combination of using CDAE in conjunction with the neural classifier trained by ALECO-2 gave the best results from all other feature-classifier combinations in 4 out of 6 test sets, including the three continuous texts, where excellent recognition accuracies (consistently over 99%) were recorded.

6 Prospects

The promising results obtained by incorporating novel techniques in the feature extraction and classification stages of an OCR system, encourage us to continue this research with the following aims: First, to improve the generalization capability of ALECO-2 by employing judiciously chosen functional conditions, so as to control searching in the weight space in an attempt to focus on the best generalization performance areas. Second, to improve the CDAE estimation algorithm so as to discriminate between classes with similar features, by focusing on their principal differences. Finally, to efficiently combine these novel techniques as well as other ones – pertaining not only to feature extraction and classification, but also to all other OCR stages – in order to design a very high recognition rate OCR system for real world problems.

References

[1] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

[2] M. Bokser, "Omnidocument technologies," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1066–1078, 1992.

[3] N. Gaitanis, G. Kapogianopoulos, and D. A. Karras, "Pattern classification using a generalized hamming distance metric," in *Proceedings of WCNN'93*, (Portland, Oregon, US), pp. IV260–IV263, 1993.

[4] N. Gaitanis, G. Kapogianopoulos, and D. A. Karras, "Minimum distance pattern classifiers based on a new distance metric," 1993. Accepted for presentation at *ICANN'93*.

[5] S. J. Perantonis and D. A. Karras, "An efficient constrained learning algorithm with momentum acceleration," 1993. Submitted to *Neural Networks*.

[6] A. Khotanzad and J. H. Lu, "Distortion invariant character recognition by a multilayer perceptron and backpropagation learning," in *Proceedings of the IEEE First International Conference on Neural Networks*, (San Diego), pp. I625–I632, 1988.

[7] D. A. Karras and S. J. Perantonis, "Comparison of learning algorithms for feedforward networks in large scale networks and problems," in *Proceedings of IJCNN'93*, (Nagoya, Japan), pp. 532–535, 1993.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[9] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[10] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman, "Generalization by weight elimination with application to forecasting," in *Adv. in Neural Inf. Proc. Sys.*, pp. 875–882, 1991.

[11] J. Moody, "Note on generalization, regularization, and architecture selection in nonlinear learning systems," in *Neural Networks for Signal Processing*, (B. H. Juang, S. Y. Kung, and C. A. Kamm, eds.), Piscataway, NJ: IEEE press, 1991.

[12] P. J. Burt, "Fast filter transforms for image processing," *Computer Graphics and Image Processing*, vol. 16, pp. 20–51, 1981.

[13] D. A. Karras, S. J. Varoufakis, and G. Carayannis, "A neural network for character recognition using a preprocessing Gaussian pyramid encoding block," *Neural Network World*, pp. 347–352, 1992.

[14] L. Holmström and P. Koistinen, "Using additive noise in back propagation training," *IEEE Trans. on Neural Networks*, vol. 3, no. 1, pp. 24–38, 1992.

[15] T. Hastie and R. Tibshirani, *Generalized Additive Models*. Chapman and Hall, 1990.

[16] K. A. Marko, "Real world issues in installing neural networks," in *Proceedings of the IJCNN'92*, (Baltimore), pp. IV301–IV303, 1992.

[17] L. A. Feldkamp, G. V. Puskorius, L. I. Davis, and F. Yuan, "Strategies and issues in applications of neural networks," in *Proceedings of the IJCNN'92*, (Baltimore), pp. IV304–IV309, 1992.

[18] D. A. Karras and S. Perantonis, "Efficient constrained training algorithms for feedforward networks,". *IEEE Trans. on Neural Networks* under review, 1993.

[19] D. A. Karras, S. J. Perantonis, and S. J. Varoufakis, "Constrained learning: a new approach to pattern classification using feedforward networks," in *Proceedings of WCNN'93*, (Portland, Oregon, USA), pp. IV235–IV238, 1993.

[20] S. J. Perantonis and D. A. Karras, "A fast constrained learning algorithm based on the construction of suitable internal representations," in *Proceedings of IJCNN'93*, (Nagoya, Japan), pp. 536–539, 1993.

[21] S. J. Varoufakis, S. J. Perantonis, and D. A. Karras, "A family of efficient learning algorithms for feedforward networks based on constrained optimization techniques," in *Proceedings of NEURONET'93*, (Praga, Czech Republic), 1993.

[22] F. Fogelman Soulie, "Neural network architectures and algorithms: a perspective," in *Artificial Neural Networks*, (T. Kohonen, K. Makisara, O. Simula, and J. Kangas, eds.), pp. 605–615, Elsevier (North Holland), 1991.

[23] A. E. Bryson and W. F. Denham, "A steepest-ascent method for solving optimum programming problems," *Journal of Applied Mechanics*, vol. 29, pp. 247–257, 1962.

[24] S. E. Fahlman, "Faster learning variations on back propagation: an empirical study," in *Proceedings of the 1988 Connectionist Models Summer School*, (San Mateo), pp. 38–51, Morgan Kaufmann, 1988.