Artificial Neural Networks for Document Analysis and Recognition

Simone Marinai[+], Marco Gori[*], Giovanni Soda[+]

∗ Dipartimento di Ingegneria dell'Informazione, Università di Siena (Italy)

+ Dipartimento di Sistemi e Informatica, Università di Firenze (Italy)

Abstract

Artificial neural networks have been extensively applied to document analysis and recognition. Most efforts have been devoted to the recognition of isolated handwritten and printed characters with widely recognized successful results. However, many other document processing tasks like pre-processing, layout analysis, character segmentation, word recognition, and signature verification have been effectively faced with very promising results. This paper surveys most significant problems in the area of off-line document image processing where connectionist-based approaches have been applied. Similarities and differences between approaches belonging to different categories are discussed. A particular emphasis is given on the crucial role of the prior knowledge for the conception of both appropriate architectures and learning algorithms. Finally, the paper provides a critical analysis on the reviewed approaches and depicts most promising research guidelines in the field. In particular, a second generation of connectionist-based models are foreseen which are based on appropriate graphical representations of the learning environment.

Index Terms: Character segmentation, document image analysis and recognition, layout analysis, neural networks, pre-processing, recursive neural networks, word recognition.

1 Introduction

Todays' computers equipped with cameras or optical scanners can read documents and provide their faithful electronic reproduction. In spite of these technological achievements, however, stacks of documents still flood desks of most offices. Whereas documents can be read and accurately stored, the processing required for extracting information is still only in its infancy. Unfortunately, either the presence of noise or the hardly predictable document structure make it very hard to extract information automatically. In the last decade, the decreasing cost of document acquisition, storage and processing, and the renewal of interest in Artificial Neural Networks (ANNs) have given rise to many novel solutions to different tasks of document processing. The topic of Document Image Analysis and Recognition (DIAR) is thoroughly dealt with in books and survey papers (see e.g. [1, 2, 3]). In spite of the emphasis on tasks like OCR and word recognition, the application of ANNs to other important DIAR tasks has not received much attention yet. An extensive bibliography on applications of artificial neural networks to document processing tasks[1] shows that most papers deal with OCR-related tasks (65%) and with word recognition (15%). To the best of our knowledge, important complementary topics have not been the subject of surveys like for OCR and word recognition [4, 5]. The purpose of this paper is to fill this gap by providing an introduction to most significant problems of DIAR where connectionist-based approaches have demonstrated their effectiveness. To narrow the scope of the paper, we focused attention on applications dealing with document images only, thus leaving out the large literature related to the on-line processing of cursive words and signatures.

To face document analysis and recognition tasks one needs to select adequate neural

[1] The bibliography can be found at: (http://www.dsi.unifi.it/ANNxDAR).

architectures and learning schemes, and to conceive appropriate representations of the data to be processed. Most of the applications to DIAR rely on traditional schemes based on multilayer perceptrons (MLPs, e.g. [6]) and do not propose novel methodologies. The original contributions concern the way MLPs are applied to the specific task. There are a number of straightforward applications to tasks like filtering and noise removal, character, word, and graphical item recognition. The massive experimentation carried out in the last few years demonstrates that, unfortunately, in spite of very interesting and promising results, critical issues of learnability and computational capabilities often arise when dealing with real-world problems. The pattern representation plays also a very crucial role for the effectiveness of the proposed solution. In many tasks, the input to be pre-processed has typically a flat representation based on a vector of features. For instance, zoning (based on grouping together features in each region of a grid superimposed to the character) is frequently used in OCR for feeding a neural network with a sub-sampled image of variable-size characters. Whereas a flat representation is appropriate for many tasks, structural representations seem to be very appropriate for all tasks which involve the whole document or any parts which exhibit relevant structure. The recent developments in the field of learning in structured domains (see e.g. [7, 8]) offer new unexplored and promising research directions, some of which are reviewed in the following.

The paper is organized into three main parts. The first part (Sections 2-4) is devoted to the analysis of neural methods related to pre-processing and segmentation of document images. The second part covers the “reading” of written documents, and includes OCR, graphical item recognition, word recognition (Section 5), and signature verification (Section 6). In the last part, we briefly discuss some aspects of pattern representation with emphasis on graph-based descriptions, and we propose some research directions (Section 7).

2 Pre-processing

Typical pre-processing operations in DIAR include binarization, noise reduction, skew detection, and character thinning (see e.g. [1], chapter 2). Most significant results on the use of ANNs in pre-processing are discussed in this section and summarized in Table 1. Neural networks are frequently used for image pre-processing by learning the appropriate filters from examples. The MLP can be used by feeding it with the pixels of a fixed size moving window (see e.g. [9]). For instance, document image binarization is performed in [10] by feeding an MLP with features computed in a 5 by 5 moving window. Text image restoration is designed to clean broken and touching characters and is aimed at improving the quality of document images before layout analysis and OCR. Typical approaches include morphological filtering, filling operations (e.g. the kFill algorithm [1], page 13), Kalman filtering, and line following methods [11]. All these methods are computationally expensive, are tuned to specific kind of noise, and cannot be easily applied to other noise sources. Obviously, a single MLP-based filter can hardly be used for cleaning a wide range of noisy documents. An adaptive approach to text image restoration using MLPs is proposed in [12]. MLPs are used as filters with a square input window to model the inverse of the distortion process. This allows one to adapt the filter by re-training the MLP on each separate page to be processed. Broken and touching characters can be unintentionally generated when removing lines overlapping text in checks, forms, and music notation. The main problem consists of reconstructing the broken symbols after line removal. In music drawing processing, Martin and Bellisant [13] propose a line removal method based on an MLP that suggests whether a pixel of a staff line also belongs to an overlapping symbol. Skewed pages can affect document image segmentation as well as the reading of textual parts. Some de-skewing methods which use ANNs have been suggested for dealing with both handwritten pages [14] and isolated

Task	**Neural architecture**	**Input to neural network**	**Output of neural network**	**Refs.**
Binarization	MLP	Features in 5x5 moving window	Foreground/Background pixel	[10]
Image restoration	MLP	Pixels in a moving window	Restored value of current pixel	[12]
Removal of touching lines	MLP with receptive fields	Histogram of chord lengths	Membership of current pixel	[13]
Page de-skew	MLP	Features computed from the page	Skew angle	[14]
Symbol de-skew	MLP	Moments computed from the symbol	Skew angle	[15]
Symbol thinning	SOM	Normalized bitmap of the symbol	Thinned symbol	[16]

Table 1: Neural approaches to the pre-processing of document images.

symbols [15]. In both cases a set of appropriate features describing the input is obtained and, subsequently, an MLP with one output is trained to reproduce the mapping from these features to the estimated angle. In the domain of thinning algorithms, Ahmed [16] proposes a clustering-based skeletonization algorithm (CBSA) implemented by Self Organizing Maps (SOM, e.g. [17]). CBSA is articulated into two main steps: a set of clusters corresponding to adjacent pixels are located and the skeleton is constructed by connecting the cluster centers. The first step is implemented in [16] by taking the clustering capabilities of SOM during network training into account. Other thinning algorithms are based on topology-adaptive SOMs [18, 19].

Discussion

The main property of ANNs which is useful for pre-processing applications is their ability to learn from examples complex nonlinear input-output relationships. The learning of these maps makes it possible a rapid adaptation to different kind of noise, without needing the handling of different cases arising in real-world applications. When compared to other learning algorithms, a remarkable advantage of ANNs is that they demonstrate a relevant degree of immunity to erroneous patterns that can be unintentionally produced when generating training patterns (e.g. [20], page 85). In pre-processing applications, ANNs are mainly used for regression, which is quite common in filtering. Sometimes classification approaches are also considered for binarization and for assigning a foreground pixel to an overlapping line. A complementary situation occurs in layout analysis (Section 3), where ANNs are mostly used as classifiers. Similarly to other filtering approaches, the choice of an appropriate input window size is a challenging problem. To partially overcome this problem, the use of multiresolution approaches can be considered. It is important to point out that, as yet, there is no clear evidence of the effectiveness of ANN in this task, since the reported tests are made on small data sets. Benchmarking of the results is required to emphasize benefits and limits of ANN-based approaches. Some methods have been recently proposed for ANN-based image pre-processing which take into account edge information (e.g. [21, 22]). These methods seem to be promising also in the domain of DIAR. With the exception of binarization methods, the approaches here described deal with black and white (B/W) images. However, the extension to gray or color images (in a way similar to layout analysis) is straightforward.

3 Layout Analysis

Document layout analysis is performed for segmenting the document into homogeneous regions (*physical layout analysis*), and for a subsequent labeling (*logical layout analysis*). Segmentation algorithms in image processing are usually classified into three main categories: pixel classification, region-based segmentation, and edge-based segmentation. Document segmentation approaches pertain to the first two categories. Pixel classification methods are related to those proposed for image segmentation [23]. Most common approaches in DIAR (e.g. [1]) belong to the family of region-based segmentation, comprising bottom-up (e.g. the Run Length Smoothing Algorithm, RLSA, and the Document Spectrum), and top-down methods.

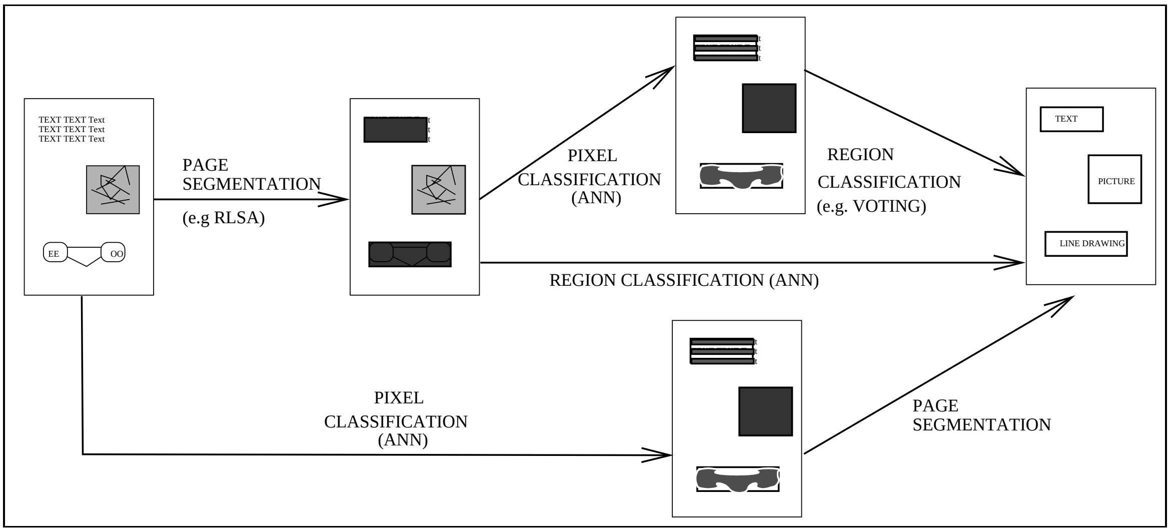

Figure 1: Neural classifiers in layout analysis. Different approaches described in Section 3 follow different paths in the picture. Pixel classification – page segmentation [24]. Page segmentation – region classification [25, 26]. Page segmentation – pixel classification – region classification [27].

In layout analysis, the classification capabilities of ANNs have been exploited at three levels: *pixel classification*, *region classification*, and *page classification*. Fig. 1 and Table 2 summarize some neural approaches to layout analysis described in this section.

3.1 Pixel classification

Pixel classification was initially applied to the binarization of document images (Section 2), and later on extended to deal with additional classes (e.g. text, graphics, and line drawing). This class of methods can be applied to different kinds of images (B/W, gray level, or color) by considering the appropriate pixel information. Most pixel-based segmentation methods rely on the assumption that different types of regions have different textures.

Etemad *et al.* [28] use MLPs in conjunction with a multi-scale wavelet packet representation for document segmentation. The features are the local moments of wavelet packet components computed in local windows at different resolutions. For each pixel various class estimations at different resolutions can be obtained by various networks. The pixel class is computed by combining the MLP outputs that encode the class membership with a one-hot coding. In this coding one output unit is assigned to each class, and all the outputs are set to 0, with the exception of the one corresponding to the correct class, which is set to 1. When the text size is fixed in several pages, the texture can be analyzed by considering a single resolution, and by feeding one MLP with the pixels in a small window around the pixel to be labeled. An example of this approach is given by Jain and Zhong [24], using an MLP to discriminate between three classes (see Table 2). Only a subset of the pixels of the input window are used as inputs so as to reduce the number of connections and improve generalization capabilities. Node pruning is performed to remove hidden units that do not significantly contribute to classification. The pixel classification methods summarized in Table 2 deal with grey level or color images. The processing of color images is performed by feeding the ANN with the color values extracted from three moving windows, one for each RGB color band [29, 30].

3.2 Region classification

Region classification was initially performed by using global features computed from each region as inputs to linear classifiers designed with user-defined parameters [36, 37]. Similarly, when using neural networks for region classification, some features are computed from each region and used as inputs to neural classifiers. Local and global features can be considered.

Local features are computed for each pixel in the region. In [25] each region, extracted by an RLSA-based algorithm, is identified as *text/non-text* by a SOM classifier with local features. The features are the occurrences in the region of some masks used to extract textural patterns, and the spatial relationships between these masks. When the computation of local features is expensive, a few random blocks within each region can be analyzed. For instance,

Task	**Classification level**	**Neural architecture**	**Input to neural network**	**Output of neural network**	**Refs.**
Page segmentation	Pixel	MLP	Pixels in a window around current pixel	Background, halftone, text & line drawings	[24]
Text location in color images	Pixel	Parallel combination of 3 MLPs: one for each color band	Pixels in a window around current pixel	Text, non-text	[30]
Multi-resolution segmentation	Pixel	MLP	Moments of wavelets computed in a window around current pixel	Image, text, graphics	[28]
Identification of text blocks - Global features	Region	MLP	Region size, number of black pixels, properties of connected components	Text, non-text	[26]
Identification of text blocks - Local features	Region	SOM	Occurrences of pre-defined 3x3 masks in the region	Text, non-text	[25]
Region labeling - Voting of labels of few random windows	Region	MLP	Gradient vector and luminance computed in a window	Background, printed & handwritten characters, photo, image	[27]
Color clustering for text extraction	Region	SOM for color clustering	Pixel color (R,G,B)	Pixel color cluster	[31]
Logical labeling of text blocks	Region	Recurrent neural network	Block size, number of lines, etc., and label of previous block	Date, Address, etc.	[32]
Logical labeling of text lines	Region	Recurrent neural network	Features of connected components	Postal code or not	[33]
Form classification	Page	MLP	Line crossings	Form identifier	[34]
Page classification in digital libraries	Page	MLP	MXY tree encoding	Page class (title, index, regular, etc.)	[35]

Table 2: Neural approaches to layout analysis.

in [27] the histograms of gradient vector directions and luminance levels are computed for each block that is classified by an MLP into five classes (see Table 2). Finally, the region class is determined by a majority voting over the selected blocks. Le, Thoma and Wechsler [26] use global features for region classification in black and white images, and compare four neural networks with the same set of features. Similarly to non-connectionist methods [36, 37], the features are computed using the region width and height, the number of black pixels, and the length of black runs. MLP with sigmoidal neurons, Radial Basis Function (RBF) network, Probabilistic Neural Network, and SOM are compared. The authors claim that the RBF networks yield a better classification accuracy.

The logical classification of textual regions has been recently addressed in [32] with a recurrent neural network used to exploit contextual information. The text blocks are extracted from the top-left to the bottom-right corners and are organized as a "temporal sequence".

For each step in the sequence, the input contains quantitative information on the current block (dimensions, position, number of lines, and number of words) as well as information from the output of the previous block. The recurrent neuro-fuzzy system called RFasArt (Recurrent Fuzzy Adaptive System ART based) is adopted which uses the same general structure of the Fuzzy-ARTMAP [38]. Experimental results are reported for business letters and scientific papers. Examples of logical classes for business letters are *date, address*, and *sender*, while for scientific papers logical classes can be *title, authors*, and *abstract.* A similar problem is the logical labeling of handwritten text lines in postal envelopes. In this domain, De Waard [33] proposes the use of recurrent neural networks for identifying sequences of connected components that correspond to the zip code.

3.3 Page classification

Page classification has been addressed with different approaches. Earlier applications concerned forms and business letters. In the last few years the classification of journal and book pages has received a significant attention (e.g. [39]).

Taylor *et al.* [34] propose an MLP-based system for the classification of forms described by line crossings. Considering the intersections between horizontal and vertical lines nine crossings can be defined. Forms can be distinguished considering the position of line crossings in the page. Zoning-based encoding is considered with a 3 by 3 grid, obtaining nine equally-sized non-overlapping regions. The number occurrences of each kind of crossing is considered for each region, and a feature vector containing 81 values is computed.

Page layout is generally of a hierarchical nature, and can be represented by graphs [40] or trees [41]. Neural networks can deal with structural representations using two main approaches: either by coding the representation into a fixed-size feature vector, or by using recursive neural networks. An example of the first approach has been recently proposed in [35]. Here document images are represented by means of a modified XY (MXY) tree where the cuts are made along horizontal and vertical lines, in addition to classical cuts along spaces considered in XY trees [41]. MXY trees are encoded into a fixed-size representation by considering the occurrences of tree-patterns in the tree corresponding to each page.

Discussion

Neural networks are used in layout analysis mainly for pattern classification, in contrast to pre-processing tasks, where they are used as non-linear trainable filters. Pixel classification approaches are in a certain sense on the borderline between filtering and classification, because as with filtering methods, there is a moving window, but here the output is a discrete decision among few classes. The most apparent advantage of neural networks acting as classifiers, which is common to other trainable classifiers, is their ability to learn the decision function from examples. This is an advantage when compared with approaches where the classification algorithm is hand-tuned by the user. As already pointed out, an additional benefit of ANNs over other learning approaches is that ANN learning methods are quite robust to noise in the training data caused, for instance, by erroneously labeled patterns (e.g. [20], page 85). When dealing with pixel classification problems, wrongly labeled pixels can be caused by frontier pixels that could be assigned to more than one class, since the window centered on these pixels covers different regions. In this case the best approach is the rejection of frontier pixels, and other approaches are to be preferred to MLPs to provide a reliable rejection (Section 7).

The comparison of pixel classification with connected component-based approaches reveals that the former is more time consuming, since all the pixels in the page must be processed [42, 30]. However, pixel classification can provide a better segmentation in some domains (e.g.

Web pages and video documents) where the location of connected components in color images is difficult [30]. Perspective uses of ANNs in layout analysis are related to graph-based representations of the page layout. With few exceptions analyzed in this section, ANNs have not been used with these graphical representations yet. In the last few years ANN-based layout techniques have been extended to color document images (e.g. [29, 31, 30]). These methods can be used in new applications of DIAR related to Web image processing, and to text location in natural images or video frames.

4 Character segmentation

Character segmentation seeks to decompose a sequence of characters into individual symbols. Segmentation strategies can be divided into three main categories [3]. *Dissection methods* partition the input image into sub-images having "character-like" properties. *Recognition-based methods* rely on the integration of segmentation and recognition. *Holistic approaches* avoid segmentation by recognizing entire words as units. In this section we analyze dissection methods employing ANNs.

4.1 Identification of touching characters

In dissection-based segmentation, touching characters are generally detected either on the basis of their size or by a low recognition confidence of a classifier trained to recognize isolated characters. ANNs with different architectures and experimental settings have been proposed as a valuable alternative. In [43] an MLP with two output neurons is trained to distinguish isolated characters from pairs of touching characters. The network input is a normalized image corresponding to one or two characters (Figure 2 (a)). A training set with more than 17,000 pairs of touching and non-touching characters was artificially generated starting from scanned samples of more than 30 common fonts. Each touching pair was generated by the

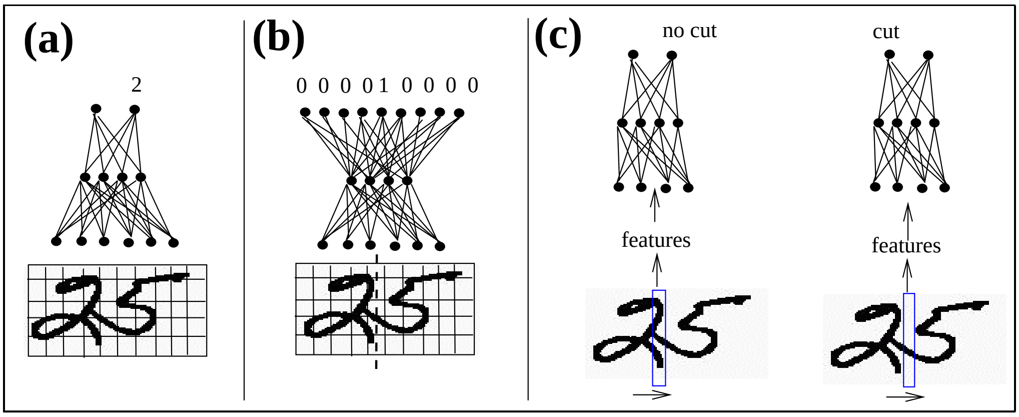

Figure 2: Neural approaches to character segmentation. In (a) and (b) an MLP is fed with a sub-sampled image. In (a) the MLP is used to establish the number of characters. In (b) the MLP suggests the cutting point. In (c) a set of features are extracted at each horizontal position in the image, and one MLP suggests cutting points.

simple concatenation of two characters. This approach to training set generation has some obvious advantages over a more immediate strategy of looking for touching and non-touching pairs in "real" documents. However, a major concern is the generalization capabilities of the network, that is the ability to detect touching pairs found in scanned documents and not artificially generated. The results reported in [43] are encouraging, since all the touching pairs not detected using traditional approaches (based on aspect-ratio and classifier rejection) were correctly detected. The previous approach can be extended to the more general problem of estimating the length (number of characters) of connected strings. For instance Lu, Chi and Siu [44] train an MLP to classify sub-images on the basis of the expected number of connected digits. The MLP is fed with features extracted from the sub-image, and has four outputs, corresponding to four string lengths (one to four digits). The features are related to the horizontal foreground-background and background-foreground transitions in the image, as well as to feature points extracted from the skeleton image. The dataset considered in the experiments was built starting from a NIST database. Since the number of connected strings with three and four characters is very small (only 48 connected 4-digit strings are available), some additional 3-digit and 4-digit strings are produced by artificially merging existing samples.

4.2 Location of cutting points

Neural networks can be used for locating cutting points as well. Eastwood *et al.* [45] propose using an MLP for cursive word segmentation. Similarly to neural filters (Section 2), an MLP is "moved" across the input image (with a horizontal displacement) resulting in a labeling of the corresponding horizontal position (Figure 2 (c)). Ten simple features (based on the vertical projection profile, the histogram of line crossings, the position of holes and the upper contour of the word) are computed for each x position in the word. At each position the features are fed to an MLP with one output node which is trained to recognize segmentation points based on local information. When dealing with handwritten strings the segmentation points can be located by analyzing the primitives (horizontal strokes) instead of the raw image. This approach is considered in [46] where an MLP is used in order to identify strokes that can correspond to cutting points. If the sub-image to be segmented is constrained to contain at most two characters, then a sub-sampling of the whole sub-image can be considered as an MLP input (Figure 2 (b)). This is the approach considered in [47]. Zoning is applied to these sub-images, which are linearly scaled to fit into a 30 by 60 frame. A feature vector of 72 elements is computed from the scaled image by counting the number of black pixels in small, non-overlapping windows. One-hot coding is used on the 60 outputs of the MLP to describe the position of the cutting point in the scaled image.

Discussion

In dissection-based segmentation a simple sub-sampling can be considered as input to

Task	**Number of characters**	**Neural architecture**	**Input to neural network**	**Output of neural network**	**Refs.**
Length estimation	1-2	MLP	Normalized sub-image	Number of characters	[43]
Length estimation	1-4	MLP	Features computed from the whole image	Number of characters	[44]
Neural dissection	2	MLP	Normalized sub-image	Cutting position	[47]
Neural dissection	Variable	TDNN	Features computed at each x position	Current x is a cutting point	[45]
Neural dissection	Variable	MLP	Stroke features	Cutting stroke	[46]

Table 3: Neural approaches for segmentation of touching characters.

a neural classifier when at most two characters are expected to be found in a word image. In presence of more characters different approaches can be pursued, such as computing global features, or using a sliding window approach (Table 3). One critical aspect in neural dissection methods is the collection of training patterns, since touching characters are not very frequent in average quality documents. The training set collection is complex, consequently many methods automatically generate synthetic touching pairs of characters (see also Section 7). In spite of the difficulties for the collection of training samples, the use of ANNs for the segmentation of touching characters has some advantages over more traditional approaches. The first advantage is the ability to adapt to different levels of character overlapping. The low execution time during the classification (e.g. [43] reports a throughput of 570 patterns per second with an accuracy of 99.8 %) is another benefit when compared, for instance, to contour and stroke analysis methods ([3], page 696). Similarly to layout analysis the use of RBF classifiers can improve the reliability of neural dissection methods. Most neural segmentation methods are limited to linear dissections, where the separating line is a vertical cut. For this reason traditional contour-based methods currently outperform ANN-based approaches, since they are able to take into account more complex splitting paths that are required with handwritten and cursive text. Graph-based representations have been applied also to this task (e.g. [48]) and can be used in conjunction with recursive neural networks.

5 OCR and word recognition

As highlighted in the introduction, word recognition and OCR have been the subjects of several survey papers. For instance, various connectionist models dealing with handwritten words have been compared by LeCun *et al.* [4]. Other authors compared neural and non-neural classifiers, and evaluated the influence of the training set size on the generalization capabilities of MLPs (e.g. [4, 49, 5]). We analyze some recent approaches for the recognition of words, characters and graphical items that are closely related to other document processing tasks. More details can be found in [9].

5.1 OCR and graphical item recognition

We analyze three crucial aspects for OCR and graphical item recognition: the relationship between feature extraction and learning algorithms, the encoding of structural features, and the adoption of modular classifiers.

Relevant shapes in printed and handwritten characters can be located by feature extraction (e.g. using the Hough transform [50]), or by forcing their recognition by the ANN, as it is exploited in Neocognitron training [51]. Similarly, features related to edges can be extracted by computing the gradient of the image. In this case, a valuable option is to insert the gradient information into the learning procedure. Gori *et al.* [52] propose a modified learning scheme for the use of autoassociators (discussed in this section among modular classifiers) to classify graphical items in presence of spot noise. An appropriate weighted norm, which is based on the gradient of the gray level, is used instead of the Euclidean norm to measure the input-output accuracy of the neural network. A modified learning algorithm (Edge-Backpropagation) is derived from the classical Backpropagation by considering the weighted error function.

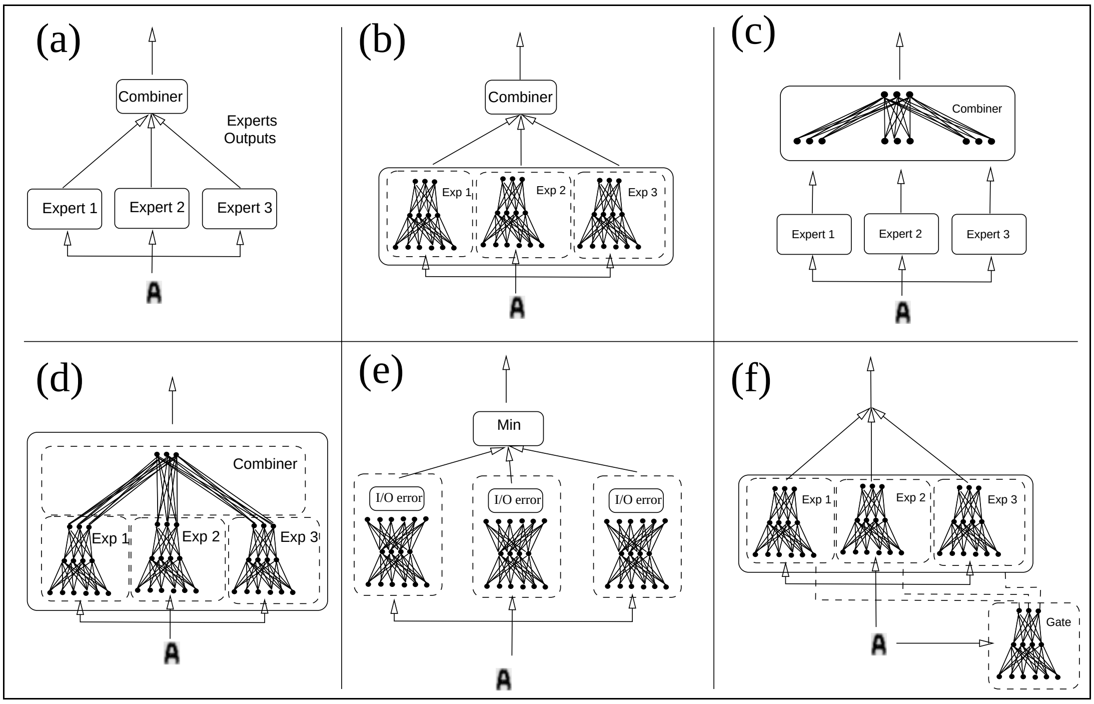

Figure 3: The use of neural networks in modular parallel classifiers for OCR. (a) represents the general architecture containing a set of experts and a combiner. (b-d) show the use of neural classifiers for the individual experts, the combiner or both modules. The autoassociator-based classifier is represented in (e). A gating network, activating different modules, is shown in (f).

Structural features describe the symbols in terms of sub-parts and relationships between these sub-parts. Amin *et al.* [53] describe the character with a graph whose nodes correspond to sub-patterns extracted from the skeleton, and edges correspond to mutual positions between sub-patterns. The graph encoding is based on the assignment of some pre-defined slots of the feature vector to each node and edge of the graph. Another method (e.g. [54]) consists in computing the occurrences of all combinations of node and edge attribute values. Other approaches rely on recursive neural networks [7] which can process graphs instead of simple sequences like recurrent networks. For instance Diligenti *et al.* [55] use recursive neural networks to recognize logos described with contour-trees [56] (see Section 7).

With reference to Fig. 3, in modular parallel classifiers ANNs can be used for implementing either the individual experts or the combiner. Connectionist-based experts for digit recognition are described in [57] (Fig. 3 (b)). A neural network can be used for implementing the combiner (Fig. 3 (c)): Lee and Srihari [58] propose a network where the input neurons (outputs of individual classifiers) are directly connected with output neurons. A parallel classifier can be designed relying only on neural networks (Fig. 3 (d)). An example of this architecture for the recognition of unconstrained handwritten digits is proposed by Mui *et al.* [59]. The network contains ten small independent sub-nets, each of which is responsible for a particular class. Independent networks dedicated to separate classes are considered also in autoassociator-based classifiers (Fig. 3 (e)). An autoassociator is an MLP with the same number of input and output units, and fewer neurons in the hidden layer. When fed with samples of the corresponding class, the autoassociator is trained to map the identity function. A classifier can be built by feeding several autoassociators in parallel (one for each class), and considering a decision module which interprets the distances between the reconstructed outputs (for each network) and the presented example. The lower is the distance, the higher is the similarity between the input sample and the corresponding autoassociator class. When the number of classes is high, characters can be split, for example, into upper-case, lower-case, digits, and special symbols; a gating network can activate the appropriate classifier (Fig. 3 (f)). In [60] this architecture is contemplated for building a two-stage OCR system with a gating network selecting the specialized networks used for more accurate classification. Likewise, in [61], a gating network pre-sorts Hangul syllables into six global types.

In serial combinations, the selection of subsequent classifiers can either be made on-line, or off-line. In on-line selection the classifier to be activated depends on the input pattern. The outputs of an MLP-based classifier trained with one-hot coding can be used for selecting the most appropriate verification modules in a serial combination. Two examples of this approach are proposed by Takahashi and Griffin [62], and by Francesconi *et al.* [63]. In off-line selection the architecture is defined during the learning of the overall classifier. Structure adaptation methods automatically adjust the network structure to the uneven distribution of classes in the pattern space. These methods are appropriate when dealing with many classes. Clustering algorithms are frequently used for grouping together most similar classes. Some methods analyze the confusion table of the first classifier in order to find most confused classes (e.g. [64]). Alternatively, clustering algorithms can be applied to the characters belonging to the learning set (e.g. [65]).

5.2 Word recognition

The segmentation-based approach to word recognition can hardly be considered when the location of segmentation points is difficult, as in cursive handwriting. One alternative is to use holistic word recognition (Fig. 4 (a)), where the words are recognized as single units. This approach is effective when the words belong to a small lexicon, for instance when reading bank checks [66]. Keyword spotting is another application where holistic word recognition can be appropriate [67]. When the problem at hand requires a larger lexicon, then segmentation-based reading is appropriate, nevertheless segmentation requires some feedback from recognition. Integrated segmentation and recognition (ISR) techniques are related to the contextual development of segmentation and recognition, with three main approaches: Heuristic Over Segmentation (HOS), Time-Delay Neural Networks (TDNN), and Space Displacement Neural Networks (SDNN) (see e.g. [9]).

In HOS a segmentation algorithm is applied to a word image to locate a large number of candidate cutting points. Subsequently, a recognizer is employed to score the alternative

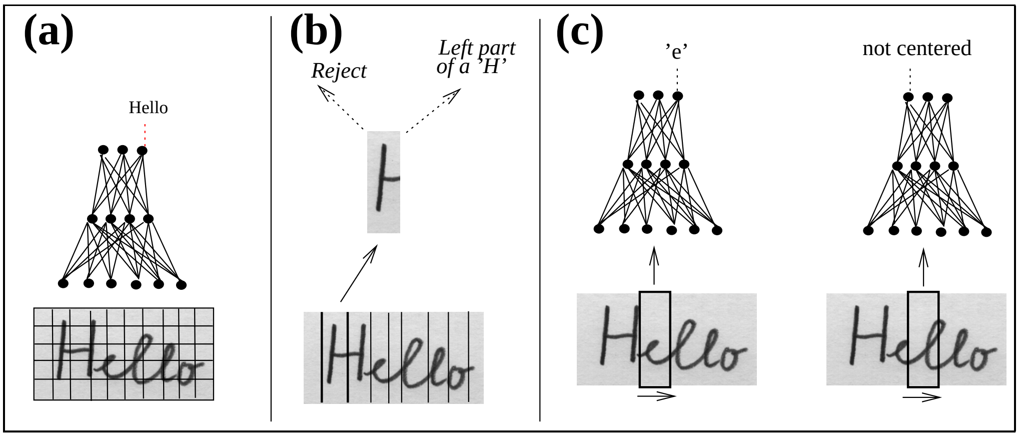

Figure 4: Some approaches to word recognition. (a) Holistic word recognition. (b) Two possible interpretations of broken characters in HOS. (c) "Non centered" output in TDNN.

segmentations generated and to find the best character sequence. Dynamic programming approaches and Hidden Markov Model (HMM)-based methods are frequently taken into account for optimizing the recognition in relation to a given lexicon [68]. When using ANNs in HOS two training strategies can be adopted for dealing with broken characters due to segmentation errors (Fig. 4 (b)). In the first approach (e.g. [69]) we assume that the broken characters still provide information, and appropriate sub-classes are introduced for each class. In this way a broken symbol can be classified as a cut symbol of the correct class. The second approach assumes that the correct segmentation points are definitely among the proposed ones. In this case the character classifier should be trained to be resistant to non-characters and a broken character should be rejected as noise by the classifier, to look for another combination of cutting points that will produce the correct character image. Generative classifiers (e.g. Gaussian density model or autoassociators) are inherently resistant to non-characters. Discriminative neural classifiers can be trained to be resistent to non-characters with outlier samples [70]. Modeling between-character shape compatibility can also overcome the non-character problem [71].

Time-Delay Neural Networks are used to deal with temporal sequences, and are implemented with an MLP that is "moved" across the sequence, providing a label for each position. TDNNs have been used for on-line word recognition: in this case the meaning of "time" is quite straightforward. In off-line word recognition the horizontal axis in the window containing the word is considered as a temporal scale. One variation of TDNN has been proposed by Martin [72] where a fixed-size window scans a word horizontally feeding a TDNN (Fig. 4 (c)). One output neuron is active when the window covers portions of two digits (the "non-centered" output). The other output neurons describe the digit membership with a one-hot encoding when the window is centered over a digit.

Discussion

Word recognition approaches are tightly connected with other tasks described in this survey. Holistic word recognition is related to methods used for isolated character recognition, and zoning is frequently considered as feature extraction. A significant difference concerns the number of classes that must be managed by the classifiers. In OCR the classes are pre-defined and known in advance, whereas a limited lexicon can be taken into account for word recognition only in some applications. The basic idea of TDNNs, which are based on a sliding window feeding an MLP, is very similar to the sliding window approach that is used for character segmentation (Section 4). The main difference is the purpose of the classification performed by the "moving" neural network. In character segmentation the network is used for identifying cutting points, whereas the TDNN provides a character classification in addition to the cutting information. In OCR and word recognition, the use of connectionist architectures is a mature field well investigated, and widely applied also to commercial systems. These DIAR tasks are examples of successful applications of ANNs to Pattern Recognition problems, and there is clear experimental evidence that well designed

ANN-based character and word recognizer can outperform other approaches. For instance, in [4] a convolutional network is compared with k-NN and Support Vector Machine classifiers showing better performance for the former one.

6 Signature verification

Signature verification is a very important topic in document processing systems (e.g. in check reading applications), that has been the subject of massive investigation (see e.g. [73]). Many connectionist approaches have considered the problem of recognizing random forgeries (affixed without knowing the name of the signer) as a preliminary step to a more complete verification system. The training set can be built up simply by collecting signatures of different writers, without requiring actual forgeries. Random forgeries can be recognized with an MLP having two outputs (one for genuine signatures, class ω_1, and one for random forgeries, class ω_2 e.g. [74]). Plamondon and Lorette [75] as well as Murshed *et al.* [76], point out the limits of a random selection of training samples from patterns of the ω_2 class. To solve this problem in [76] the verifier is trained using only the genuine signatures of one writer. Basically, the signature is divided into m equal-size regions, each assigned to a Fuzzy ARTMAP network which acts like an expert examining one region. A combination of SOM-based classifiers and MLPs is proposed in [77] in order to select patterns belonging to class ω_2. Initially two SOMs are considered to group together the most similar signatures corresponding to neurons of the SOM output layer. Subsequently, an MLP classifier is used to verify a given signature. It is worthwhile noting that the training samples of class ω_2 correspond to signatures that are clustered together by the SOM. The related problem of writer identification has been considered in [78] with an MLP fed with features computed from a single text line.

Discussion

Two related problems affect the design of signature verification systems: the collection of training samples and the choice of classifiers able to discriminate between ω_1 and ω_2 classes. In particular, we emphasize the limits of MLP-based classifiers when used for classification purposes [79]. Other architectures, like autoassociator-based classifiers and RBF networks, are more appropriate for dealing with problems of verification with strategies that can be extended from other tasks (e.g. [80] apply RBF for outlier rejection in word recognition).

7 Future research directions

The first observation that can be drawn from this survey is that simple applications of MLPs acting as classifiers are generally inadequate for solving complex PR tasks that emerge in DIAR. On the other hand, by using state of the art approaches it is possible to reach significant results with a low programming effort mainly relying on the learning from example capabilities of ANNs (e.g. [81]).

Instead of using simple models, appropriate architectures must be considered. In this paper we pointed out several times that, as discussed in [79], MLPs are very good tools for performing a discriminative classification between patterns of well defined classes, but are not adequate for applications requiring a reliable rejection. Other architectures, like autoassociator-based classifiers and RBF networks, are more suitable. Bianchini *et al.* [82] provide an in-dept analysis of the behavior of autoassociators from a theoretical point of view. An experimental analysis of the use of RBF for speech verification is provided in [83], whereas recent experiments on performance of MLP and RBF networks for outlier rejection in word recognition are reported in [80]. Hybrid classifiers based on the combination of generative and discriminative learning may provide both high classification accuracy and resistance to outliers [84, 85].

Other architectures that deserve consideration in various DIAR tasks are the Polynomial classifier, that was shown to outperform most other neural classifiers in character recognition (e.g. [86]), and the SOM that has been recently applied also to character encoding for performing text retrieval from document images [87].

Some typical examples of modular trainable architectures are receptive fields and convolutional neural networks that provide invariance under translations and have been proven to significantly increase OCR performance [4]. Modular classifiers, that have been explored mainly in OCR (Section 5.1), are other examples of trainable architectures. The extension of the receptive field approach to strings of characters has lead to the SDNN approach [9], where convolutional neural networks are replicated over the input field. Another popular approach for sequence recognition is the integration of HMM and MLP (e.g. [68, 88]). The gradient descent algorithm, that is the basis of Backpropagation, has been recently extended to graph transformer networks, which are based on a composition of standard modules [4]. These networks have been applied on document recognition systems composed by several modules that can be trained globally for word recognition by using SDNN. This nice approach is a promising direction for future DIAR applications.

Classic Backpropagation learning schemes have been recently extended to the case in which the patterns are represented by Directed Ordered Acyclic Graphs (DOAG) [7]. The basic idea of the computational scheme is depicted with an example in Fig. 5, where the contour-tree algorithm [56] is used to represent each pattern by a tree (a node is associated to each contour by a recursive scheme). The computation scheme consists of unfolding the neural architecture through the structures. More general graphs can easily express complex patterns, but the corresponding extension of classic neural network architectures and learning algorithms is

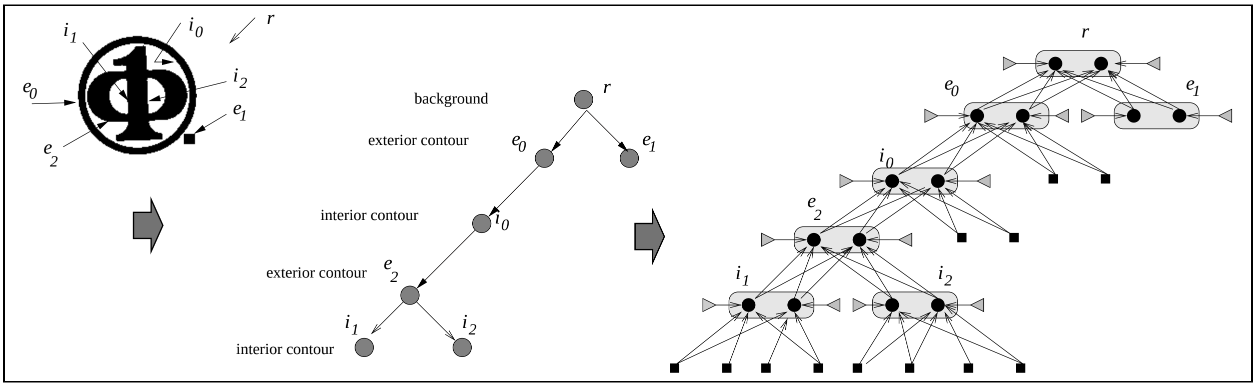

Figure 5: Graphical pattern representation and the corresponding connectionist processing, which inherits the input graphical structure. In this example, two neurons are associated with each node.

shown to be less effective. This extended view of neural networks operating on graphs gives rise to a new wave of connectionist-based techniques for pattern recognition, which is somehow in between traditional decision-theoretic and structural approaches. In this paper, we point out that this new approach, and its applications to tasks of pattern classification and image retrieval offer one of the most promising and viable research directions in the field. A general discussion on this new computational scheme can be found in [89]. It is worth mentioning that, so far, the application to pattern recognition has been essentially limited to non-stationary processing of DOAGs. Recent analyses on graphs different from DOAGs as well as models which allows one to incorporate non-stationarity, have not been applied so far to pattern recognition. The development of these issues is likely to be of crucial importance for further developments of adaptive graphical pattern recognition.

Recursive neural networks are closely related to Bayesian networks. In [7] a unified framework is proposed for the analysis of the neural and the statistical approaches. The full understanding of their connections is very important for application purposes and still requires an in-depth investigation (see also a recent paper by Caelli *et al.* [90]).

New neural approaches to DIAR can be extended from those developed in image processing or other pattern recognition applications. In the last few years, Support Vector Machine (SVM) classifiers gained a large interest. SVMs are at the borderline between ANNs and other statistical approaches, and are increasingly used for character recognition tasks (e.g. [91, 92]). Several methods have been proposed for image pre-processing, including image restoration [21, 93, 94], and image compression [95, 96]. New methods developed for image segmentation can be extended to document images as well [97, 98, 99, 100]). Another important issue in DIAR applications is the uneven distribution of training samples among various classes. One approach is based on the automatic generation of artificial patterns (see Section 4). Other solutions rely on general training strategies designed to deal with this problem (e.g. [101, 102]).

8 Conclusions

Artificial neural networks have been massively used for nearly all the tasks in document image analysis and recognition. Most connectionist approaches rely on the use of simple MLPs, and the relationships between different uses of ANNs in different tasks have been only partially considered. In fact, many lessons learned in some tasks should be properly considered in other domains. Several pointers to these similarities have been reported in this paper: strict relationships exist between pre-processing and layout analysis, between OCR and holistic word recognition, and between character segmentation and word recognition by TDNNs. Two other issues have been stressed in the paper. First, the need for selecting appropriate architectures in order to deal with patterns to be rejected. The confidence values which are the outputs of MLP are frequently taken into account for rejecting dubious patterns. Two alternatives for handling outliers are RBF, and autoassociators. Second, the emergence of new architectural approaches designed for processing structural representations. Several tasks, such as page classification, require to deal with graphical structures (e.g. trees and graphs). For these cases recursive neural networks represent some of the most promising methods to be taken into consideration in future applications.

Acknowledgments. We thank all the members of the DANTE research group for their support to most of the research we have been carrying out in the field. We are also very grateful to the participants of the tutorial [2] "Artificial Neural Networks for Document Analysis and Recognition" (ICDAR 2001 - Seattle, September 2001, and ICPR 2002 - Québec, August 2002) for their questions and constructive comments.

References

[1] L. O'Gorman and R. Kasturi, *Document Image Analysis.* Los Alamitos, California: IEEE Computer Society Press, 1995.

[2] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE TPAMI*, vol. 22, no. 1, pp. 38–62, 2000.

[3] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE TPAMI*, vol. 18, no. 7, pp. 690–706, 1996.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] M. D. Garris, C. L. Wilson, and J. L. Blue, "Neural network-based systems for handprint OCR applications," *IEEE Trans. Image Processing*, vol. 7, no. 8, pp. 1097–1112, 1998.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing* (D. E. Rumelhart and J. L. McClelland, eds.), vol. 1, ch. 8, pp. 318–362, Cambridge: MIT Press, 1986.

[2] The slides of the tutorial can be downloaded at `http://www.dsi.unifi.it/ANNxDAR`.

[7] P. Frasconi, M. Gori, and A. Sperduti, “A general framework for adaptive processing of data structures,” *IEEE TNN*, vol. 9, no. 5, pp. 768–786, 1998.

[8] P. Frasconi, M. Gori, and A. Sperduti, “Guest editors’ introduction: special section on connectionist models for learning in structured domains,” *IEEE Trans. Knowledge and Data Engineering*, vol. 13, no. 2, pp. 145–147, 2001.

[9] M. Gori, S. Marinai, and G. Soda, “Artificial neural networks for document analysis and recognition,” tech. rep., N. 1/2003 - DSI - University of Florence, “http://www.dsi.unifi.it/ANNxDAR/”, 2003.

[10] Z. Chi and K.W.Wong, “A two-stage binarization approach for document images,” in *Proc. Int’l Symp. Intelligent Multimedia, Video and Speech Processing*, pp. 275–278, 2001.

[11] A. P. Whichello and H. Yan, “Linking broken character borders with variable sized masks to improve recognition,” *Pattern Recognition*, vol. 29, no. 8, pp. 1429–1435, 1996.

[12] P. Stubberud, J. Kanai, and V. Kalluri, “Adaptive image restoration of text images that contain touching or broken characters,” in *ICDAR 95*, pp. 778–781, 1995.

[13] P. Martin and C. Bellisant, “Low-level analysis of music drawing images,” in *ICDAR 91*, pp. 417–425, 1991.

[14] N. Rondel and G. Burel, “Cooperation of multilayer perceptrons for the estimation of skew angle in text document images,” in *ICDAR 95*, pp. 1141–1144, 1995.

[15] R. Palaniappan, P. Raveendran, and S. Omatu, “New invariant moments for non-uniformly scaled images,” *PAA*, vol. 3, no. 2, pp. 78–87, 2000.

[16] P. Ahmed, “A neural network based dedicated thinning method,” *PRL*, vol. 16, no. 6, pp. 585 – 590, 1995.

[17] T. Kohonen, “The self-organizing map,” *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[18] R. Singh, V. Cherkassky, and N. Papanikolopoulos, “Self-organizing maps for the skeletonization of sparse shapes,” *IEEE TNN*, vol. 11, no. 1, pp. 241–248, 2000.

[19] A. Datta, S. K. Parui, and B. B. Chaudhuri, “Skeletonization by a topology-adaptive self-organizing neural network,” *Pattern Recognition*, vol. 34, pp. 617–629, 2001.

[20] T. Mitchell, *Machine Learning.* McGraw Hill, 1997.

[21] K. Suzuki, I. Horiba, and N. Sugie, “Neural edge enhancer for supervised edge enhancement from noisy images,” *IEEE TPAMI*, vol. 25, no. 12, pp. 1582–1596, 2003.

[22] D. Wang, A. Talevski, and T. Dillon, “Edge-preserving nonlinear image restoration using adaptive components-based radial basis function neural networks,” in *Int’l Joint Conference on Neural Networks*, vol. 2, pp. 1243–1248, 2003.

[23] K. Chen, D. Wang, and X. Liu, “Weight adaptation and oscillatory correlation for image segmentation,” *IEEE TNN*, vol. 11, no. 5, pp. 1106–1123, 2000.

[24] A. K. Jain and Y. Zhong, “Page segmentation using texture analysis,” *Pattern Recognition*, vol. 29, no. 5, pp. 743–770, 1996.

[25] C. Strouthopoulos and N. Papamarkos, “Text identification for document image analysis using a neural network,” *Image and Vision Computing*, vol. 16, no. 12/13, pp. 879–896, 1998.

[26] D. X. Le, G. R. Thoma, and H. Wechsler, “Classification of binary document images into textual or nontextual data blocks using neural network models,” *MVA*, vol. 8, no. 5, pp. 289–504, 1995.

[27] S. Imade, S. Tatsuta, and T. Wada, “Segmentation and classification for mixed text/image documents using neural networks,” in *ICDAR 93*, pp. 930–934, 1993.

[28] K. Etemad, D. S. Doermann, and R. Chellappa, “Multiscale segmentation of unstructured document pages using soft decision integration,” *IEEE TPAMI*, vol. 19, no. 1, pp. 92–96, 1997.

[29] H. Yan and J. Wu, “Character and line extraction from color map images using a multi-layer neural network,” *PRL*, vol. 15, pp. 97–103, April 1994.

[30] K. Jung, “Neural network-based text location in color images,” *PRL*, vol. 22, pp. 1503–1515, 2001.

[31] C. Strouthopoulos, N. Papamarkos, and A. Atsalakis, "Text extraction in complex color documents," *Pattern Recognition*, vol. 35, pp. 1743–1758, 2002.

[32] G. Sainz Palmero and Y. Dimitriadis, "Structured document labelling and rule extraction using new recurrent fuzzy-neural systems," in *ICDAR 99*, pp. 181–184, 1999.

[33] W. P. de Waard, "Neural techniques and postal code detection," *PRL*, vol. 15, no. 2, pp. 199 – 206, 1994.

[34] S. L. Taylor, R. Fritzson, and J. A. Pastor, "Extraction of data from preprinted forms," *MVA*, vol. 5, no. 5, pp. 211–222, 1992.

[35] F. Cesarini, M. Lastri, S. Marinai, and G. Soda, "Encoding of modified X-Y trees for document classification," in *ICDAR 01*, pp. 1131–1136, 2001.

[36] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM Journal of Research and Development*, vol. 26, no. 6, pp. 647–656, 1982.

[37] F. Y. Shih and S. S. Chen, "Adaptive document block segmentation and classification," *IEEE Trans. SMC*, vol. 26, no. 5, pp. 797–802, 1996.

[38] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: a neural network architecture for incremental learning of analog multidimensional maps," *IEEE TNN*, vol. 3, no. 5, pp. 698–713, 1992.

[39] J. Hu, R. Kashi, and G. Wilfong, "Document image layout comparison and classification," in *ICDAR 99*, pp. 285–288, 1999.

[40] J. Yuan, Y. Y. Tang, and C. Y. Suen, "Four directional adjacency graphs (FDAG) and their application in locating fields in forms," in *ICDAR 95*, pp. 752–755, 1995.

[41] G. Nagy and S. Seth, "Hierarchical representation of optically scanned documents," in *ICPR 84*, pp. 347–349, 1984.

[42] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE TPAMI*, vol. 20, no. 3, pp. 294–308, 1998.

[43] J. Wang and J. Jean, "Segmentation of merged characters by neural networks and shortest path," *Pattern Recognition*, vol. 27, no. 5, pp. 649–658, 1994.

[44] Z. K. Lu, Z. Chi, and W. C. Siu, "Length estimation of digits strings using a neural network with structure based features," *SPIE/IS&T Journal of Electronic Imaging*, vol. 7, pp. 79–85, January 1998.

[45] B. Eastwood, A. Jennings, and A. Harvey, "A low level feature based neural network segmenter for fully cursive handwritten words," in *ICDAR 97*, p. 523, 1997.

[46] D. You and G. Kim, "An approach for locating segmentation points of handwritten digit strings using a neural network," in *ICDAR 03*, pp. 142–146, 2003.

[47] J. H. Bae, K. Jung, J. Kim, and H. Kim, "Segmentation of touching characters using an MLP," *PRL*, vol. 19, no. 8, pp. 701–709, 1998.

[48] S. W. Lee, D.-J. Lee, and H.-S. Park, "A new methodology for gray-scale character segmentation and recognition," *IEEE TPAMI*, vol. 18, no. 10, pp. 1045–1050, 1996.

[49] D. J. Burr, "Experiments on neural net recognition of spoken and written text," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1162–1168, 1988.

[50] W. Utschick, P. Nachbar, C. Knobloch, and J. A. Nossek, "The evaluation of feature extraction criteria applied to neural network classifiers," in *ICDAR 95*, pp. 315–318, 1995.

[51] K. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by the Neocognitron," *IEEE TNN*, vol. 2, no. 3, pp. 355–365, 1991.

[52] M. Gori, M. Maggini, S. Marinai, J. Sheng, and G. Soda, "Edge-backpropagation for noisy logo recognition," *Pattern Recognition*, vol. 36, no. 1, pp. 103–110, 2003.

[53] A. Amin, H. Al-sadoun, and S. Fischer, "Hand-printed arabic character recognition system using an artificial network," *Pattern Recognition*, vol. 29, no. 4, pp. 663–675, 1996.

[54] L. Cordella, C. De Stefano, and M. Vento, "A neural network classifier for OCR using structural descriptions," *MVA*, vol. 8, no. 5, pp. 336–342, 1995.

[55] M. Diligenti, M. Gori, M. Maggini, and E. Martinelli, "Adaptive graphical pattern recognition for the classification of company logos," *Pattern Recognition*, vol. 34, pp. 2049–2061, 2001.

[56] G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni, “Trademark shapes description by string-matching techniques,” *Pattern Recognition*, vol. 27, no. 8, pp. 1005–1018, 1994.

[57] N. W. Strathy and C. Y. Suen, “A new system for reading handwritten zip codes,” in *ICDAR 95*, pp. 74–77, 1995.

[58] D.-S. Lee and S. N. Srihari, “Dynamic classifier combination using neural network,” in *Proc. SPIE - Doc. Rec. II*, pp. 26–37, 1995.

[59] L. Mui, A. Agarwal, A. Gupta, and P. S.-P. Wang, “An adaptive modular neural network with application to unconstrained character recognition,” *IJPRAI*, vol. 8, no. 5, pp. 1189–1204, 1994.

[60] J. Mao, K. Mohiuddin, and T. Fujisaki, “A two-stage multi-network OCR system with a soft pre-classifier and a network selector,” in *ICDAR 95*, pp. 78–81, 1995.

[61] S.-B. Cho and J. H. Kim, “Recognition of large-set printed Hangul (Korean script) by two-stage backpropagation neural classifier,” *Pattern Recognition*, vol. 25, no. 11, pp. 1353–1360, 1992.

[62] H. Takahashi and T. D. Griffin, “Recognition enhancement by linear tournament verification,” in *ICDAR 93*, pp. 585–588, 1993.

[63] E. Francesconi, M. Gori, S. Marinai, and G. Soda, “A serial combination of connectionist-based classifiers for OCR,” *Int’l J. Doc. Anal. Rec.*, vol. 3, no. 3, pp. 160–168, 2001.

[64] R. Y.-M. Teo and R. Shingal, “A hybrid classifier for recognizing handwritten numerals,” in *ICDAR 97*, pp. 283–287, 1997.

[65] H. Su, W. Wang, X. Li, and S. Xia, “Hierarchical neural network for recognizing hand-written characters in engineering drawings,” in *ICDAR 95*, pp. 46–49, 1995.

[66] J. H. Kim, K. K. Kim, and C. Y. Suen, “An HMM-MLP hybrid model for cursive script recognition,” *PAA*, vol. 3, no. 4, pp. 314–324, 2000.

[67] F. Cesarini, M. Gori, S. Marinai, and G. Soda, “INFORMys: A flexible invoice-like form reader system,” *IEEE TPAMI*, vol. 20, no. 7, pp. 730–745, 1998.

[68] T. Steinherz, E. Rivlin, and N. Intrator, “Offline cursive script word recognition - a survey,” *Int’l J. Doc. Anal. Rec.*, vol. 2, no. 2/3, pp. 90–110, 1999.

[69] P. D. Gader, M. Mohamed, and J.-H. Chiang, “Comparison of crisp and fuzzy character neural networks in handwritten word recognition,” *IEEE Trans. Fuzzy Systems*, vol. 3, no. 3, pp. 357–363, 1995.

[70] J. Bromley and J. S. Denker, “Improving rejection performance on handwritten digits by training with rubbish,” *Neural Computation*, vol. 5, no. 3, pp. 367–370, 1993.

[71] P. Gader, M. Mohamed, and H. Chiang, “Handwritten word recognition with character and inter-character neural networks,” *IEEE Trans. SMC*, pp. 158–164, 1997.

[72] G. Martin, “Centered-object integrated segmentation and recognition of overlapping handprinted characters,” *Neural Computation*, vol. 5, no. 3, pp. 419–429, 1993.

[73] F. Leclerc and R. Plamondon, “Automatic signature verification: the state of the art,” *IJPRAI*, vol. 8, no. 3, pp. 643–660, 1994.

[74] J. P. Drouhard, R. Sabourin, and M. Godbout, “A neural network approach to off-line signature verification using directional PDF,” *Pattern Recognition*, vol. 29, no. 3, pp. 415–424, 1996.

[75] R. Plamondon and G. Lorette, “Automatic signature verification and writer identification: the state of the art,” *Pattern Recognition*, vol. 22, no. 2, pp. 107–131, 1989.

[76] N. A. Murshed, F. Bortolozzi, and R. Sabourin, “Off-line signature verification, without a priori knowledge of class ω_2. A new approach,” in *ICDAR 95*, pp. 191–195, 1995.

[77] H. Cardot, M. Revenu, B. Victorri, and M.-J. Revillet, “A static signature verification system based on a cooperating neural networks architecture,” *IJPRAI*, vol. 8, no. 3, pp. 679–692, 1994.

[78] U.-V. Marti, R. Messerli, and H. Bunke, “Writer identification using text line based features,” in *ICDAR 01*, pp. 101–105, 2001.

[79] M. Gori and F. Scarselli, “Are multilayer perceptrons adequate for pattern recognition and verification?,” *IEEE TPAMI*, vol. 20, no. 10, pp. 1121–1132, 1998.

[80] J. Liu and P. Gader, "Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition," *Pattern Recognition*, vol. 35, pp. 2061–2071, 2002.

[81] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR 03*, pp. 958–963, 2003.

[82] M. Bianchini, P. Frasconi, and M. Gori, "Learning in multilayered networks used as autoassociators," *IEEE TNN*, vol. 6, no. 2, pp. 512–515, 1995.

[83] P. Frasconi, M. Gori, and G. Soda, "Links between LVQ and Backpropagation," *PRL*, vol. 18, no. 4, pp. 303–310, 1997.

[84] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum, "Classification with hybrid generative/discriminative models," in *Advances in Neural Information Processing Systems 16* (S. Thrun, L. Saul, and B. Schölkopf, eds.), Cambridge, MA: MIT Press, 2004.

[85] C.-L. Liu, H. Sako, and H. Fujisawa, "Discriminative learning quadratic discriminant function for handwriting recognition," *IEEE TNN*, vol. 15, no. 2, pp. 430–444, 2004.

[86] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, pp. 2271–2285, 2003.

[87] S. Marinai, E. Marino, and G. Soda, "Indexing and retrieval of words in old documents," in *ICDAR 03*, pp. 223–227, 2003.

[88] E. Trentin and M. Gori, "Robust combination of neural networks and hidden markov models for speech recognition," *IEEE TNN*, vol. 14, no. 6, pp. 1519–1531, 2003.

[89] P. Frasconi, M. Gori, A. Kuechler, and A. Sperduti, "From sequences to data structures: Theory and applications," in *A field guide to dynamical recurrent networks* (J. Kolen and S. Kremer, eds.), New York: IEEE Press, 2001. Chapter 22.

[90] T. Caelli, W. F. Bischof, and M. Ferraro, "A comparison of neural and graphical models for syntactic and structural pattern recognition," in *Proc. 1st IAPR-TC3 Workshop ANNPR* (M. Gori and S. Marinai, eds.), pp. 1–7, ISBN 88-7957-221-0, 2003.

[91] J.-X. Dong, C. Suen, and A. Krzyzak, "High accuracy handwritten chinese character recognition using support vector machine," in *Proc. 1st IAPR-TC3 Workshop ANNPR* (M. Gori and S. Marinai, eds.), pp. 39–45, ISBN 88-7957-221-0, 2003.

[92] L. Zeyu, S. Tang, and H. Wang, "Fast recognition of handwritten digits using pairwise coupling support vector machine," in *Int'l Joint Conference on Neural Networks*, vol. 1, pp. 878–883, 2002.

[93] H.-S. Wong and L. Guan, "A neural learning approach for adaptive image restoration using a fuzzy model-based network architecture," *IEEE TNN*, vol. 12, no. 3, pp. 516–531, 2001.

[94] S. Lu, Z. Wang, and J. Shen, "Neuro-fuzzy synergism to the intelligent system for edge detection and enhancement," *Pattern Recognition*, vol. 36, pp. 2395–2409, 2003.

[95] D.-C. Park and Y.-J. Woo, "Weighted centroid neural network for edge preserving image compression," *IEEE TNN*, vol. 12, no. 5, pp. 1134–1146, 2001.

[96] D. Feiden and R. Tetzlaff, "Binary image coding using cellular neural networks," in *Int'l Joint Conference on Neural Networks*, pp. 1149–1152, 2003.

[97] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, "Support vector machines for texture classification," *IEEE TPAMI*, vol. 24, no. 11, pp. 1542–1550, 2002.

[98] P. Clark and M. Mirmehdi, "Combining statistical measures to find image text regions," in *ICPR 00*, pp. 450–453, 2000.

[99] E. Lopez-Rubio, J. Munoz-Perez, and J. Gomez-Ruiz, "A robust two-stage system for image segmentation," in *ICPR 00*, pp. 606–609, 2000.

[100] X. Feng, A. K. I. Williams, and S. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE TPAMI*, vol. 24, no. 4, pp. 467–483, 2002.

[101] G. Karystinos and D. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE TNN*, vol. 11, no. 5, pp. 1050–1057, 2000.

[102] J. Ghosn and Y. Bengio, "Bias learning, knowledge sharing," *IEEE TNN*, vol. 14, no. 4, pp. 748–765, 2003.