IJDAR
DOI 10.1007/s10032-011-0151-y

ORIGINAL PAPER

Labelling logical structures of document images using a dynamic perceptive neural network

Yves Rangoni · Abdel Belaïd · Szilárd Vajda

Received: 30 March 2010 / Revised: 22 January 2011 / Accepted: 7 February 2011

Abstract This paper proposes a new method for labelling the logical structures of document images. The system starts with digitised images of paper documents, performs a physical layout analysis, runs an OCR and finally exploits the OCR's outputs to find the meaning of each block of text (i.e. assigns labels like "Title", "Author", etc.). The method is an extension of our previous work where a classifier, the perceptive neural network, has been developed to be an analogy of the human perception. We introduce in this connectionist model a temporal dimension by the use of a time-delay neural network with local representation. During the recognition stage, the system performs several recognition cycles and corrections, while keeping track and reusing the previous outputs. This dynamic classifier allows then a better handling of noise and segmentation errors. The experiments have been carried out on two datasets: the public MARG containing more than 1,500 front pages of scientific papers with four zones of interest and another one composed of documents from the Siggraph 2003 conference, where 21 logical structures have been identified. The error rate on MARG is less than 2.5% and 7.3% on the Siggraph dataset.

Keywords Document image analysis and recognition · Layout analysis · Logical labelling · Perceptive neural network · Time-delay neural network

Y. Rangoni (✉) · A. Belaïd
Nancy 2 University, LORIA, Vandœuvre-Lès-Nancy, France
e-mail: rangoni.yves@googlemail.com

A. Belaïd
e-mail: abelaid@loria.fr

S. Vajda
Computer Science Department, TU Dortmund, Dortmund, Germany
e-mail: szilard.vajda@udo.edu

1 Introduction

The role of a document image analysis and recognition (DIAR) system is to provide an electronic and editable version of a paper document. For example, a user wants to find quickly some interesting documents inside a corpus, based on some keywords. He could use a plain-text search while an Optical Character Recognition (OCR) would be able to extract the text from the pages. However, generally it is a waste of time for the user and also the content providers, digital libraries or publishers (e.g. CiteSeer, European Library, Library of Congress, Springer, Elsevier, etc.). The raw results of an OCR appear insufficient when the user needs to focus on some structural metadata such as specific titles, list of authors, paragraphs, tables. Indeed, both users and content providers prefer working on a specific part of the document ("the advanced search") to focus their search on more meaningful zones like "Title" or "Author". For an advanced query, the amount of computations is reduced and it should return less but more interesting documents for the reader [9,29]. On top of that, if the document is fully zoned (i.e. each zone has a logical label), it can be easily transformed, reformatted or reorganised; process which is really important for all viewer devices, especially for handheld devices [19]. Thus, the automatic labelling of documents is highly desirable [2]. Several large scale digitisation initiatives such as the Million Book project, the efforts of the Open Content Alliance, or the digitisation works of Google [13,15], want to make the logical structures available in their systems to provide richer browsing and searching experience for the public.

Getting the logical information is a task that can be done easily by a human, but actually it is still an open problem for a computer. More precisely, it is still widely done manually for documents where scanned or paper versions are

Published online: 03 March 2011

only available. Indeed, the logical structure extraction is a challenging problem due to the inherent complexity of the documents. Starting at the pixel level, the gap between physical observations and logical interpretations is huge. The goal is to find the best labelling function which maps a logical label to each block of the physical page layout. To deal with this issue, two types of approaches have been considered in the literature: model-driven and data-driven [36].

The model-driven approach is the most frequent one. It needs a representation of the knowledge, a model, to interpret the input data. The model contains all the information necessary to transform a physical structure into a logical one. Usually, these models are made up either by rules, or trees, or grammars. Syntactical analysis is often employed to perform the labelling [36]. Rule-based systems such as [22,24,30] are fast and human-understandable but are poorly flexible and cannot really handle difficult cases and varying layouts. To avoid writing huge lists of empirical rules, knowledge databases or thesaurus can be considered as proposed for example by [14,25,50]. However, it requires on the other hand the reading and understanding of the text. This implies other issues.

Grammars are also very popular solutions [10,21,26]. The physical and logical layouts are described as a sentence of tokens while syntactic parsers do the labelling. They produce deterministic models which are improper for processing noisy documents. Stochastic extensions [52] or extended formalisms [12] have been proposed in order to overcome this issue.

Although model-driven approaches seem to transcribe the structure hierarchy, some drawbacks still remain. They are not designed to handle complex and noisy document structures. Moreover, a lot of parameters need to be tuned. The model building requires an expert and cannot deal with all the possible interpretations of a document.

The data-driven approaches make use of raw physical data to analyse the document and no knowledge or static rules are given. The underlying idea is to let the system find the labelling function by itself and stop relying on rules or heuristics of an expert. Few contributions using classical-machine-learning tools like Neural Networks (NNs) can be found in the literature [45,46]. There are chiefly extensions of model-driven systems where a training stage is introduced. In that case, grammars are still popular [8,18]. NNs are used in other steps of DIAR and are appreciated for their robustness faced to noisy data. Indeed, NNs are largely exploited in image preprocessing and physical layout analysis as related in the survey of [39]. However, NNs are deprecated because of the workload of the training process (the ground truth documents are expensive to produce and the learning stage can be slow). Furthermore, they are not really designed to work on structured patterns and do not integrate domain-specific knowledge.

In this paper, we introduce an extension of our previous work [44] where we have proposed to combine capabilities of model-driven and data-driven approaches, respectively. It is based on a hybrid method using a special kind of NN with local representation, the so-called Percepto by [11] or Transparent Neural Network by other authors [34].

The architecture is with a local representation. Our model incorporates the two structures, both physical and logical, as concepts in the neurons. A training stage allows learning the relationships between the two structures from samples. The recognition is not only a classic forward propagation over the NN, but it performs many perceptive cycles as well. A perceptive cycle consists in forwarding the physical features, getting the logical output, and if an ambiguity occurs, correcting the input vector according to what it has been seen during the recognition. Considering that the system will deal with erroneous input features, it can refine the recognition progressively thanks to the input correction. We called it PNN for Perceptive Neural Network. The new contribution consists in implanting into the previous PNN the time dimension. As the network is working with different and corrected input features at each cycle, we introduce the usage of a Time-Delay Neural Network (TDNN) which takes into account the results of the previous perceptive cycles at times $T-n$ while making a decision at time T. The extended system, called Dynamic Perceptive Neural Network (DPNN), as opposed to PNN which is static, is as fast as its predecessor (PNN), thanks to a better behaviour during the recognition step.

The paper is organised as follows: the next section describes the PNN's running and analyses its drawbacks. Section 3 introduces the Dynamic PNN extension, based on a TDNN. Finally, Sect. 4 reports experimental results and discussion on the MARG dataset and a new dataset we created. This contains scientific papers as well but with more logical labels to be identified.

2 Perceptive neural network

This section describes our previous work before introducing the extension in Sect. 3. By outlining the most important points from it, we hope it will prevent the reader to refer to several other papers.

The initial Perceptive Neural Network itself borrowed some ideas from the Percepto system introduced by Côté [11]. Percepto is a perceptual model designed for handwritten word recognition, based on the [40] reading mode. The main idea is to integrate knowledge as interpretable concept associated to each neuron. Contrary to a classical Multi-Layer Perceptron (MLP) [28], all the neurons are transparent, which means each output is known. There are no hidden layers at all. The network has a "local representation" as opposed to a distributed representation like a MLP. The recognition is

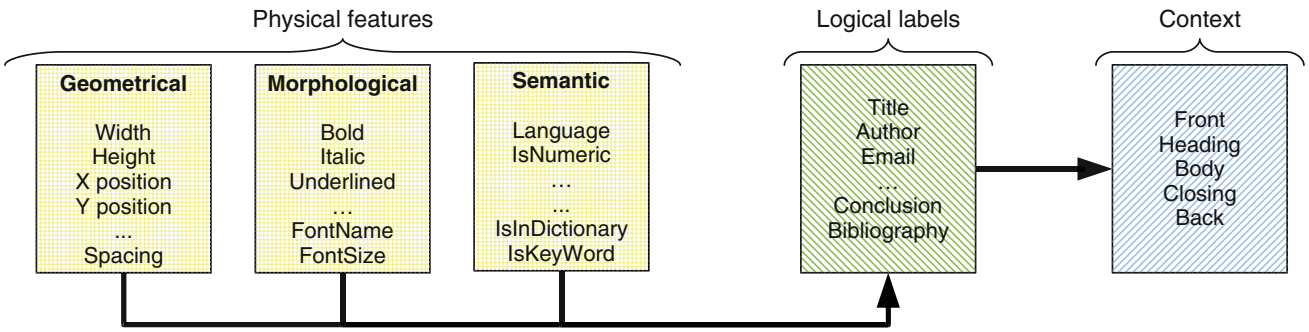


Fig. 1 A PNN instance with 3 layers for scientific articles

performed through several bottom-up and top-down processes (perceptive cycles) just like human vision operates.

The Percepto architecture is organised in layers of neurons. The activation function works with saturation. It accumulates only positive activation, and there is no competition between the neurons. The activation function A is given by (1), where θ_i is a decreasing constant, r_i is the activation threshold, $E_i(t)$ the neighbourhood contribution, M and m are respectively superior and lower activation bounds, α_{ij} and β_{ij} are the positive and negative stimulations from j to i, $a_j(t)$ is the activation of the node j.

$$\begin{aligned} A_i(t+\delta t) &= A_i(t) - \theta_i(A_i(t) - r_i) + E_i(t) \\ E_i(t) &= \begin{cases} n_i(t)(M - A_i(t)) & \text{if } n_i(t) > 0 \\ n_i(t)(A_i(t) - m) & \text{if } n_i(t) < 0 \end{cases} \\ n_i(t) &= \sum_j (\alpha_{ij} - \beta_{ij}) a_j(t) \end{aligned} \tag{1}$$

The main concepts of the Percepto system were retained for our PNN. In [5,43,44], it has been shown how this model for handwriting recognition can be adapted to logical document zoning. We have shown its utility for the logical structure recognition. The input data corresponds to text blocks (bounding boxes). The extracted features are related to the text block descriptions (e.g. style, font, etc.), while the output designates the logical labels.

As reported in [41] concerning logical structure recognition, a perceptive and knowledge guided solution seems to be more appropriate to cope with the gap between physical observation and logical interpretation. That is the reason why a regular MLP is not adapted: knowledge is difficult to integrate and it behaves as a "black-box", resulting in a complex understanding of its behaviour [17]. The PNN has "organised" neurons and each of them corresponds to an interpretable concept and it is linked to an element of the logical structure. Excluding the first layer containing the physical inputs, the following layers unfold the logical layout by introducing fine concepts in the first layers and general/coarse concepts in the following layers. The Fig. 1 presents an instance of the PNN used for logical labelling of scientific articles. Input features (geometrical, morphological and semantic) make up the first layer, the second layer contains logical labels while the third layer includes coarse concepts called context. The context is provided by the expert. It is during the construction of this context that the "knowledge" mentioned before is brought. For our application, we decided to follow the logical splitting of the text. As proposed by the Text Encoding Initiative[1] (TEI), we created five independent concepts (front, heading, body, closing, back) clustering all the logical labels.

In the former system [11] the connections between adjacent layers were bidirectional and only stimulating (just positive activation values were forwarded). Moreover, the connection weights were determined manually according to some prior knowledge. This choice sounded inappropriate in the case of logical structure recognition as the relationships between the layers are not straightforward. Thus, we have chosen to perform a training phase for the PNN, similar to MLP, in order to compute all the weights w according to the input patterns x_p. As opposed to Percepto, the PNN is fully connected and the connections can be inhibitive. The training is done by a gradient descent algorithm. The error $E_p(w)$ between the desired output d_q and the computed output $o_{l,q}$ is minimised for each input pattern p (2) where L stands for the last layer.

$$\begin{aligned} E_p(w) &= \sum_{q=1}^{N_L} \left(o_{L,q}(x_p) - d_q(x_p)\right)^2 \\ o_{l,j} &= f\left(\sum_{i=0}^{N_{l-1}} w_{l,j,i} o_{l-1,i}\right) \end{aligned} \tag{2}$$

The weight between the unit i in layer l and j in layer $l+1$ is modified according to (3) [28].

$$\Delta w_{l,i,j} = \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial o_{l,j}} f'\left(\sum_{m=0}^{N_{l-1}} w_{l,j,m} o_{l-1,m}\right) o_{l-1,i} \tag{3}$$

where f is an activation function (e.g. the sigmoid).

[1] http://www.tei-c.org/index.xml.

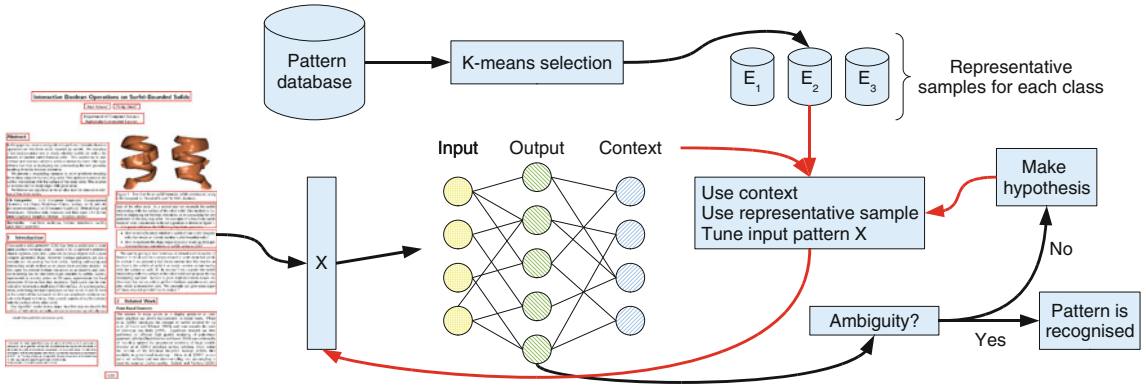


Fig. 2 Perceptive cycles and input correction. As soon as a propagation is done, the output vector is analysed to decide whether the pattern is well recognised or needs a correction

All the neurons carry interpretable concepts, so the desired output is known for all of them. The partial term is given by (4) and the PNN can be trained as a cascade of mono-layer Perceptrons (no hidden layers included).

$$\forall l, \frac{\partial E_p(w)}{\partial o_{l,j}} = o_{l,j}(x_p) - d_j(x_p) \tag{4}$$

During the recognition step, the PNN is employed such as a MLP, but after each propagation, the outputs are analysed. It means that if the output vector is close to a canonical basis vector (5) and (6), the pattern is considered as classified. Otherwise, the following layers (the context) are taken into account to inject additional information. We defined two rejection criteria: $M(O)$ checks if the vector has at least one component with a high value (close to 1) and $\Gamma(O)$ checks if the greatest component has a value largely higher than all the other components. These rules decide when a block is accepted or some corrections are needed.

$$M(O) = \|O\|_\infty > \varepsilon \quad \text{with } 0 \ll \varepsilon < 1 \tag{5}$$

$$\Gamma(O) = \frac{n\left(\left(\sum O_i\right)^2 - \sum O_i^2\right)}{(n-1)\left(\sum O_i\right)^2} < \eta \quad \text{with } 0 < \eta \ll 1 \tag{6}$$

The final layers contain global information and coarse concepts, but they are more robust and easier to find. They can be used to generate hypothesis on the pattern. This context information manages the correction of the input features. Once a label is supposed to be the good one, the input vector is corrected according to this hypothesis. Actually, during the training step, several representative samples, or prototypes, for each logical label are extracted from the training set. The correction consists in modifying the current input to make it closer to a representative sample. The correction is mainly focused on the block bounding boxes: merging blocks together or splitting them into smaller sub-blocks. The rules are straightforward. If a block contains several lines and the hypothesis indicates that the bounding box for this kind of label is smaller, the current box is split into a different number of lines in such a way that its new dimensions better fit a correct prototype (Fig. 3).

More precisely, when extracting the representative samples, a k-means algorithm is used to store the width, the height and the number of contained lines. Three possible candidates are determined for each class. One of the most likely representative samples for the "Author" label could be the vector (20; 0.2; 1), which means that the candidate is 20% of the page width, 0.2% of the height and contains 1 line. If an ambiguous block comes to the PNN with the hypothesis of being an "author" and with the vector (18; 1; 3), we look if it can be divided and if so, we try the best split. Here, it could be split after the first line which produces the vectors (18; 0.3; 1) and (18; 0.7; 2). The corresponding bounding boxes have better chance now to be labelled as "Author" and "Affiliation".

Merging blocks occurs rarely, because the behaviour of the OCR [1] is mainly under-segmentation. It could be supposed that it uses a fast top-down approach which stops dividing the blocks, maybe a little bit too early when they are considered sufficiently homogeneous. In the rare case when two consecutive blocks are ambiguous, we merge them. At the end, it amounts to correct the physical layout analysis where errors occur quite often [23,54]. Perceptive cycles are completed in a loop (Fig. 2) until no ambiguity persists (i.e. (5) and (6) are fulfilled).

The perceptive cycles (propagation-correction) allow a bottom-up and top-down resolution and refine the recognition. However, if too many cycles must be performed, the process could be time consuming because it implies many physical extractions. In order to face this problem, the solution could be to prune some unnecessary extractions. Instead of feeding the network with the whole amount of features at each cycle, they are introduced progressively, in groups, during the recognition and only if the pattern is considered

to be too ambiguous. The groups are the result of a clustering of the full set of features (e.g. geometrical, morphological and semantic). This simulates at the same time a global and local vision of the recognition. It is global when using the context information and hypothesis generation; and local when extracting or correcting specialised features. The input data clustering does not improve the recognition quality, but for the same recognition rates the speed-up is considerable.

Technically speaking, as the number of features is not constant, we create as many PNNs as clusters. In the implementation, the number of clusters is set to three, so three PNNs are used. If the groups of variables can be defined arbitrary as suggested in Fig. 1, the input feature space can also be divided automatically according to several criteria. We proposed in [44] a fast filter-based selection [7] to construct subsets of variables. It ranks the subsets by predictive power and in each of them the variables are the most independent as possible. It avoids redundant variables, and allows feeding the PNN with relevant information.

3 Dynamic perceptive neural network

This section describes the contribution we propose in this work. First we show why the PNN is perfectible and secondly we present how to extend the PNN in a dynamic model, which provides a better behaviour during the recognition stage.

3.1 Static recognition with dynamic data

As mentioned in Sect. 2, a PNN is based on a static model but is improved to manage knowledge and input data correction. During the recognition, several forwards are performed with possibly corrected inputs. As the segmentation is often corrected, it snowballs most of the features in the input vector. However, the network is trained only once, with a fixed training database. Indeed, the features until now called x_i are different for each cycle. The x_i should be named $x_i(t)$ where t is the number of the current perceptive cycle. Unfortunately, the weights $w^l_{i,j}$ are always constant scalars and are not adapted to $x_i(t)$ because $x_i(0) \neq \cdots \neq x_i(n)$.

The contribution of this paper is to demonstrate how to integrate the data correction occurring between each perceptive cycle within the training step. The goal is to modify the training schema in such a way as the recognition step has a more appropriate behaviour.

3.2 Considering temporal dimension with a time-delay neural network

The Time-Delay Neural Network (TDNN) can handle this kind of unsettled data. Contrary to a MLP, where the output $o_{l,j}$ depends on the outputs of the preceding layer, the output of a neuron $o(t)$ at time t in a TDNN corresponds to a weighted sum of the past delayed values (7).

$$o(t) = \sum_{n=0}^{T} w(n)x(t-n) \tag{7}$$

There are two main methods to train such a network. The first one removes all time delays by unfolding the network into an equivalent static structure [32], but this method is not used in practice (time consuming). The second solution is an extension of the back-propagation algorithm called temporal back-propagation. The idea is to consider the weights as constants when computing an approximation of the gradient (8) of the cost function Err (9).

$$Err = \sum_{n=0}^{T} e(n)^2 \tag{8}$$

$$\frac{\partial Err}{\partial W^l_{ij}} = \sum_{n=0}^{T} \frac{\partial Err}{\partial o^{l+1}_j(n)} \cdot \frac{\partial o^{l+1}_j(n)}{W^l_{ij}} \tag{9}$$

where $W^l_{ij} = [w^l_{ij}(0), w^l_{ij}(1), \ldots, w^l_{ij}(D)]^T$
With a complete derivation of the terms [53], the weights are updated according to (10).

$$W^l_{ij}(n+1) = W_{ij}(n) - \mu\delta^{l+1}_j(n) \cdot X^l_i(n) \tag{10}$$

with $X^l_i(n) = [x^l_i(n), x^l_i(n-1), \ldots, x^l_i(n-T)]$ and $\delta^l_j(k)$ defined by (10).

$$\delta^l_j(k) = \begin{cases} -2e_j(t) \cdot f'\left(o^L_j(t)\right) & l = L \\ f'\left(o^l_j(t)\right) \cdot \sum_k \delta^{l+1}_k(t) \cdot W^{l+1}_{jk} & l < L \end{cases} \tag{11}$$

If W and X are considered as scalars, the algorithm is similar to the back-propagation for static networks. The Algorithm 1 summarises the main steps of the training.

```
repeat
  Choose a random sample
  for all the layers do
    for all the neurons in the layer do
      Compute δ^l_j(k)
      for all incoming connections of the neuron do
        Compute ΔW^l_ij(k) = μ δ^l_j(k) o_j(k)
      end for
    end for
  end for
  for all the weights do
    W^l_ij(k) ← W^l_ij(k) + ΔW^l_ij(k)
  end for
until the training converges
```

Algorithm 1: DPNN backpropagation algorithm

This dynamic training is slower than the static back-propagation one, even if $T = 1$. This is due to the

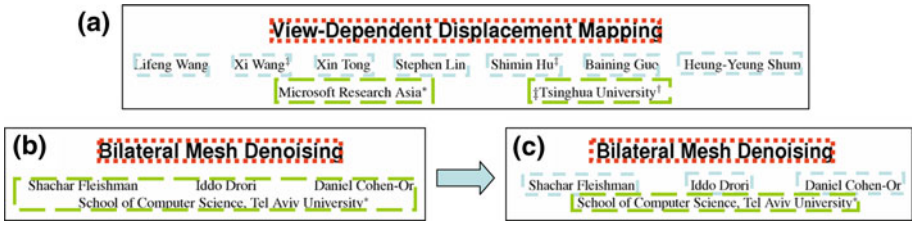


Fig. 3 Image **a**: The segmentation is correct. Image **b**: The authors are in the same *big bounding box*, it is split into *smaller boxes* as in (**c**)

presence of vectorial data instead of scalars. The algorithm also suffers from the same drawbacks as those of the classic back-propagation one [31], especially convergence problems, which are amplified by the new T parameter [42,51]. These are some of the possible reasons why TDNN are not so common, and why other methods such as HMM are preferred when a problem requires the time dimension [33].

Fortunately, in the case of logical structure extraction, the T factor depends on the number of perceptive cycles. Some experiments have shown that T can be small for the Static PNN (SPNN). On top of that, as the weights are more suitable now to the current input X_i, the number of cycles can be lower compared to the static version. Moreover, each X_i is given by the same OCR and the variation between the data ($\|X_i(t) - X_i(t-1)\| < \epsilon$) is rather small. We have chosen to set $T = 2$ for the Dynamic PNN, which means 3 perceptive cycles. It is a reasonable trade-off between accuracy and time consumption.

3.3 Training and recognition with the DPNN

3.3.1 Training

Unlike the Static PNN, the training database for a Dynamic PNN has to be modified to integrate the time dimension. The raw (uncorrected) outputs of a commercial OCR [1] are collected to have the feature vectors at time $t = 0$. Then, a manual correction is performed on the segmentation: the erroneous bounding boxes are tuned to match with logical elements. It is sometimes necessary to merge or split some bounding boxes (Fig. 3). The OCR is run another time on the new segmentation and the outputs are kept to feed data at time $t = 1$. These data are also modified manually for the second time and provide the network inputs at time $t = 2$.

These manual corrections could be tedious, but not impossible to handle. The system needs some ground truth documents as a first bootstrapping and the user has to correct and label manually several documents (assuming there is no other ready-to-use material to start a training). Then, it can continue the ground truth generating process by using a first trained DPNN, with a high rejection rate and tries to use it for a new semi-automatic labelling of another set of document in order to make the training/testing dataset growing. This manual effort is not more difficult than any other data-driven methods, and it is a very profitable contribution to large and homogeneous datasets.

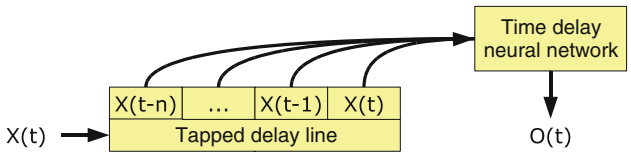


Fig. 4 Tapped delay line [20]

3.3.2 Recognition

The recognition step is similar to the static version one. The input data are extracted first, then passed in the DPNN with $t = 0$. The outputs are analysed and if the answer is ambiguous (5) and (6), the input features are corrected by resizing the bounding box as it was already done in the static case. Another perceptive cycle follows with a new OCR extraction but on the DPNN with $t = 1$, where the information of $t = 0$ is taken into account. A last cycle is finally executed, using the current extraction results and the previous ones at times $t = 1$ and 0.

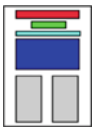
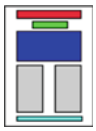
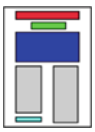
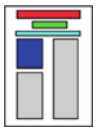
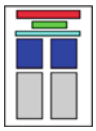
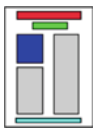
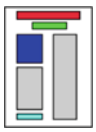
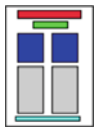
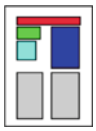
Additional perceptive cycles can be considered and two solutions can be considered. Planning more cycles during the training stage, even if it means to stop the correction of the input data after $t > 2$. It is also possible and more appropriate to keep the same network, but shifting the data with $t \leftarrow t-1$ as the "tapped delay line" [20] of the TDNN (Fig. 4). That would amount to reduce the memory capability by remembering only the latest events and discard the first extractions which were maybe too erroneous.

4 Experimentations

4.1 The MARG dataset

There are just a few available datasets and few contributions reporting results involving these data. We applied the DPNN on the publicly available dataset MARG (Ground Truth Data for Biomedical Journals) of [16] which is one the most used ones, even though it is not the best test scenario for our DPNN. MARG contains binarized, skew corrected images. They are title pages of medical journal

Table 1 MARG dataset: visual definition of the nine layout types

Type	A	B	C	D	E	F	G	H	Other
- Title - Author - Affiliation - Abstract - Text									
Distrib.	12.6	15.3	14.4	14.0	27.8	1.2	3.8	1.3	9.5

Distribution of the page layouts in percentage [38]

Table 2 Logical labels used for the Siggraph dataset. Input features used by the DPNN

Logical labels		Physical features			
		Geometrical (bounding box level)	Morphological (line level)		Semantic (word level)
Title	Simple Paragraph	isText	Bold	Scaling	IsNumeric
Author	List item	isImage	Italic	Spacing	KeyWords
Email	Enumerate item	isTable	Underlined	Alignment	%KnownWords
Affiliation	Float (e.g. picture)	isOther	Strikethrough	Left Indent	%Punctuation
Abstract	Conclusion	X position	Upper Case	Right Indent	Bullet
Keywords	Bibliography item	Y position	Small Capitals	First Indent	Enum
CR Categories	Algorithm	Width	Subscript	Num Lines	Language
Introduction	Copyright	Height	Superscript	Boxed	Baseline
Section	Acknowledgments	UpSpace	Font Name	Red/Green/Blue	
SubSection	Page number	BottomSpace	Font Size		
SubSubSection		LeftSpace			
		RightSpace			
		NumPage			

articles and the corresponding ground truths are given. There are 1, 553 images divided into 8 classes (named with letters A to H) depending on their layout type, plus an additional class "others", where all the unusual article layouts belong. The images are logically labelled with four labels: "Title", "Authors", "Affiliation" and "Abstract". The frequencies of each class and a visual aspect of the layouts are given in Table 1.

We made 10 rounds of cross-validations by splitting the database every time in two equal parts between training and validation. The results are averaged over the rounds to reduce variability. The 10×2 sets were chosen with a controlled random seed, so that the experimentation is reproducible. The initial segmentation and all the physical features (Table 2) were obtained with the [1] OCR.

Our features are the direct transcriptions of those described in the complete schema [47]. What we called *Geometrical features* corresponds to the "BlockType" of the schema. We added the Up/Bottom/left/Right space values which are the widths of the white margins between a block and the other blocks around it. *Morphological features* are the combination of "Formatting type" and "ParagraphType" of the schema. As some features are only given at the word or character level (e.g. boldness), we computed the global ratio at the block level by an elementary cross-multiplication rule. The *Semantic features* are mostly taken in the "CharParamsType" section of the schema. We added in it: "Bullet" and "Enum" to describe if some bullet or enumeration symbols are found in the text. The *Keywords* feature describes if a predefined word is found in the text. The keyword set is {'abstract', 'introduction', 'keyword', 'table', 'figure', 'conclusion', 'references', 'appendix'}. Except this last feature, all the others can be extracted by most of the commercial OCR (e.g. FineReader, Omnipage, ReadIris) and are also listed in the ALTO (Analyzed Layout and Text Object) XML Schema [3].

The Fig. 5 shows the box-and-whisker diagrams of the recognition rates obtained for each type of layout. In addition to sample min/max, lower/upper quartile and median, we added the mean with a blue round dot.

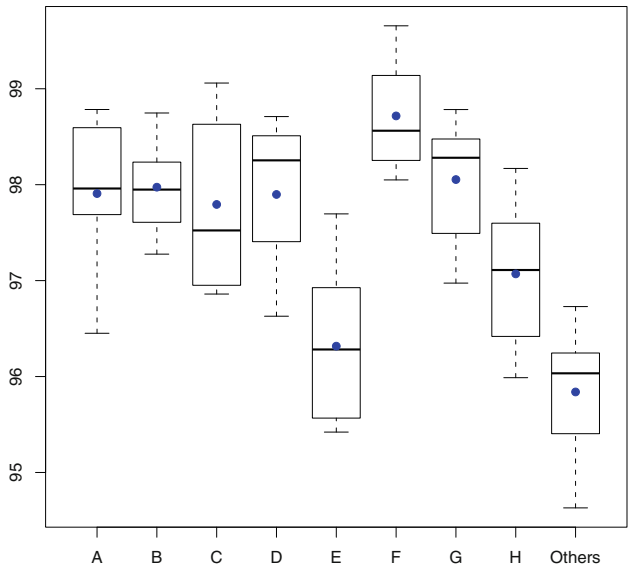


Fig. 5 Results for the MARG dataset. Layout types are in *x*-axis and recognition rates in *y*-axis

It can be shown that whatever the layout type is, the results are always good. The easiest layout types were 'B', 'F' and 'G', while the hardest was the "others" type. Compared to what can be found in the literature, the results are satisfactory [36], although the DPNN was designed to treat more logical labels (more than a dozen) and few layout classes. The MARG dataset is somewhat the opposite. There are few labels and several layout classes.

A lot of methods exist in the literature, but most of them have been tested on confidential documents or on small datasets. Furthermore, the wide range of input features, labels, training/test sets, etc. makes the comparisons almost impossible. Recently, [6] published a work on example-based logical labelling of document title page images. Firstly, they reported the few other representative works where the MARG dataset was used: [24] with a rule-based system obtained 96.7%, [35] with also a rule-based system obtained 86% (due to a bad recognition of the affiliations) and [37] with a hidden semi-Markov model obtained 91%. [6] pointed the fact that the MARG dataset was not considered in its totality and that only subsets of it were used. Secondly, they introduced a lightweight method with a similarity measure for layout combining structural layout and textual similarity. The features they employed are fast to extract and the matching step is also fast, flexible and efficient. They obtained on the full MARG dataset accuracy rates from 94.8 up to 99.6% with an average of 97.8%.

Our rates are ranging from 94.6 to 99.7%, with a global average of 97.5%. At first glance, [6] can obtain up to 99.6%, but we are also acting in different conditions. Indeed, to reach 99.6% on all the 1,553 images, they used a leaving-image-out cross validation, which means that the labelling of one document is determined knowing the labelling of all the 1,552 others documents. Analogously, the leaving-journal-out gives 98.9% by testing documents of the same journal (a dozen in average) knowing all the other documents. Finally, the leaving-type-out gives 94.8%, meaning that the tests where done on one type of layout knowing all the other types which roughly means that 90% of the full dataset is used for training (exact proportions can be deduced from Table 1).

For the MARG dataset, we did not use the context layer because 4 labels are too few to have a helpful context. The first cycle gives roughly 80%, while the second one is around 95% and 97.5% for the last one. Most of the errors are due to the bad recognition of the affiliations, which are most of the time labelled as authors. However, we do not have poor result on the label "Affiliation" like in the work of [35]. The perceptive cycles are able to correct most of the miss-segmentations when a big block includes both lines of "Authors" and "Affiliations". The dynamic version of the PNN is better than the static one. Only the class 'E' was requiring a lot of correction. This class 'E' was also difficult because some documents were really different one from one another. There were also some pages with missing labels (e.g. layout122/14230829, layout122/14234603,…), complex layouts of affiliations and abstracts (e.g. layout122/15111832), and also, according to us, some errors in the ground truth (e.g. layout122/18447737 or layout122/18628210: Abstract is the full body of the document (Fig. 6), sometimes the keyword lines are included in the abstract for layout122/18445612, sometimes excluded for layout122/18447737, missing title in layout 122/18583868, etc.)

4.2 The siggraph dataset

The DPNN was designed to deal with several labels in order to propose a complete document analysis, meaning that all the blocks must have a label. The MARG dataset is somewhat different: with few labels and several classes of documents. In our best knowledge, there is no publicly available dataset containing a full physical and logical description of the documents. For this purpose, we have created a more detailed dataset. It is composed of 74 papers from the ACM Siggraph 2003 conference [48]. The documents are scientific articles in Portable Document Format (PDF) having a considerable amount of logical structure elements.

The PDFs were printed at 600 dpi with a laser printed HP Color LaserJet 4650 PCL[2] and then digitised with a digital copier-printer-scanner device Gestetner DSm745[3] at 600 dpi in black and white (Fig. 7). In these 74 documents, 21

[2] http://h10010.www1.hp.com.

[3] http://www.gestetnerusa.com.

State-of-the-Art Review

Vascular Thrombohemorrhagic Disorders: Hereditary and Acquired

Rodger Bick, M.D., Ph.D., F.A.C.P.

Dallas Thrombosis Hemostasis Clinical Center, Dallas, Texas, USA

Disorders of the vasculature are common, but often unappreciated causes of bruising, bleeding, and vasculitis/small vessel thrombosis. Petechiae and purpura and other dermal findings, including dermal vascular thrombosis, livido reticularis, and variants such as cutus marmorata, pallor, cyanosis, and dusky cyanosis, are hallmark findings of vascular disorders, depending upon whether vascular leakage, occlusion, or both are occurring. Most typically, patients with vascular disorders usually complain of mild to moderate mucosal membrane bleeding, often manifesting as bilateral epistaxis, gastrointestinal bleeding (often occult), intrapulmonary bleeding, or genitourinary bleeding. Patients also may present with a history of easy and spontaneous bruising or gingival bleeding with tooth brushing. Many normal individuals experience occasional gingival bleeding with tooth brushing; however, if occurring almost daily, or more than two to three times a week, a vascular or platelet defect should be considered. Another clinical clue to the presence of a vascular disorder is the finding of dependent petechiae and purpura primarily found on the extremities and usually absent from the torso. This is a characteristic of vascular bleeding, whereas platelet defects are typically associated with symmetrical petechiae and purpura found on the extremities and torso. If microvascular disorders are manifested by occlusion, likewise, the findings are usually dependent or distal. Common clinical findings of vascular disorders are summarized in Table 1 (1–3). When suspecting a vascular disorder, one must first rule out coagulation protein defects (prothrombin time and activated partial thromboplastin time) and thrombocytopenia. If these are normal, in the appropriate setting, the differential diagnosis is, therefore, a vascular defect versus a platelet function defect. If the distinction cannot be made clinically, the most reliable method of diagnosing a vascular defect is to document normal platelet function. In the past, prolongation of the standardized template bleeding time (TBT) has been used to document the presence of vascular or platelet dysfunction; however, this test is very unreliable for this particular use. Thus, when suspecting vascular dysfunction, the appropriate ways to rule out platelet dysfunction are platelet aggregation or Lumi-aggregation studies or use of the newer PFA-100 or Thrombostat 4000 platelet function analyzers (4–6). Once platelet dysfunction is ruled out, vascular dysfunction is likely.

Address correspondence and reprint requests to Dr. Rodger Bick, Dallas Thrombosis Hemostasis Clinical Center, 10455 North Central Expressway, Suite 109, PMB 320, Dallas, TX 75231.

CLINICAL VASCULAR DISORDERS

Vascular disorders are best categorized as hereditary, acquired, and drug induced. These are summarized in Table 2. The hereditary vascular disorders generally are the hereditary collagen vascular diseases and most are rare clinical oddities. The one exception to rarity is Osler–Weber–Rendu disease (hereditary hemorrhagic telangiectasia [HHT]), which is common. Alternatively, the acquired vascular disorders are very common and all clinicians should be familiar with them. The importance of becoming familiar with acquired vascular disorders is several-fold: when a patient presents with dependent petechiae and purpura, and easy or spontaneous bruising or the other clinical manifestations of vascular dysfunction previously discussed, the patient should be evaluated for vascular disorders. Also, if an individual has one of the acquired disorders known to be associated with vascular defects and undergoes surgery or sustains trauma, it should be assumed the patient has a systemic vascular defect that might lead to clinically significant thrombohemorrhagic problems.

Vascular disorders may present in bizarre and varied ways. The reasons for this are the many complex determinants seen in individual paticnts, which may alter clinical findings. These determinants, which account for varied clinical presentations, are summarized in Table 3. There are many potential host responses to a vascular disorder or defect. For example, there may be simply an antigenic response, or may only be activation of the coagulation system, activation of fibrinolysis, kinin activation, activation of only complement, or any combination

178

Fig. 6 Scan of a page from the MARG dataset

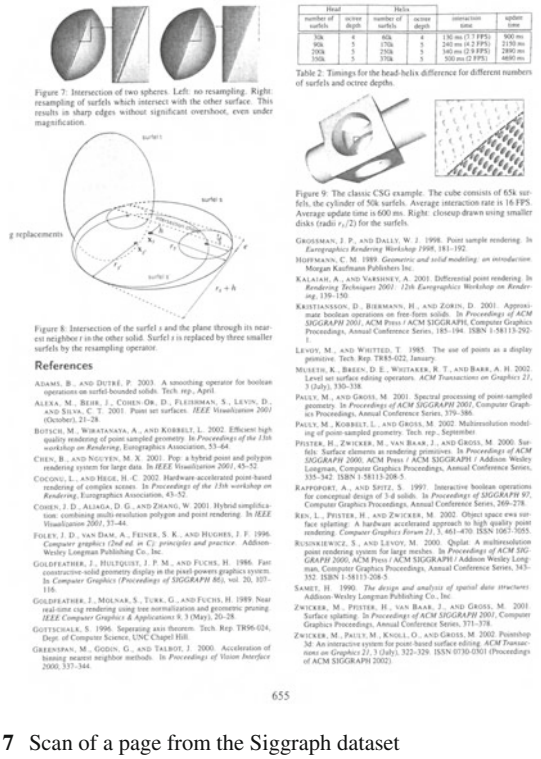


Figure 7: Intersection of two spheres. Left: no resampling. Right: resampling of surfels which intersect with the other surface. This results in sharp edges without significant overshoot, even under magnification.

Figure 8: Intersection of the surfel s and the plane through its nearest neighbor t in the other solid. Surfel s is replaced by three smaller surfels by the resampling operator.

References

Table 2: Timings for the head-helix difference for different numbers of surfels and octree depths.

Figure 9: The classic CSG example. The cube consists of 65k surfels, the cylinder of 50k surfels. Average interaction rate is 16 FPS. Average update time is 600 ms. Right: closeup drawn using smaller disks (radii $r_s/2$) for the surfels.

655

Fig. 7 Scan of a page from the Siggraph dataset

logical structures have been identified (Table 2) for a total of more than 2,000 patterns. The input feature information is still coming from the commercial OCR [1]. The training stage uses 44 documents and the remaining 30 are considered for testing. The results of the Dynamic PNN are shown in the last columns of the Table 3. We also reported the results obtained with a Static PNN with details for each cycle and a regular MLP.

After three cycles, the Static PNN can already outperform almost 10 points of recognition rate than a simple MLP (from 81.7 to 90.2%), but the extraction time is multiplied by a factor of 1.85. It has been shown, that three perceptive cycles are a good trade-off between quality and speed. A possible fourth cycle does not bring highly better recognition and slows down the recognition (not in Table 3: 91.7% for recognition and 2.1 speed factor for time extraction). Here, the Dynamic PNN is better than the SPNN with 3 and also with 4 cycles. It can reach 92.7% with a 1.8 time factor. Although the training of the DPNN is up to 10 times slower than for the SPNN, the propagation of the inputs inside the network during the recognition is neglectable compared to OCR extractions. Results show that the 3-cycle DPPN is more accurate and faster than a 4-cycle SPNN. For the Siggraph dataset, the DPNN clearly outperforms the static version.

The Table 4 gives a closer look at the recognition rates for each logical label. The DPNN never returns worse results than the SPNN. The fact of using results from preceding

Table 3 Logical classification rates for a regular MLP, a Static PNN and a Dynamic PNN after 3 perceptive cycles

		Static-PNN			DPNN
	MLP	C_1	C_2	C_3	C_3
All	81.7	45.2	78.9	90.2	92.7
Best class	98.9	66.7	85.3	100.0	100.0
Worst class	0.0	0.0	0.0	28.9	66.7
Relative time. MLP as reference	1.0	0.7	1.45	1.85	1.8

cycles is favourable for the DPNN. On the other hand, the DPNN cannot return tremendous results. It fails also on difficult cases. For instance, the "Acknowledgment" and "Conclusion" are still not very well recognised. With the exception of segmentation problems, to which we can find a solution with more appropriate algorithms [4]; we still have problems with these types of labels: the bounding boxes and physical features are correct, but we think that more informative features are missing to deal with this kind of logical labels.

The results we obtained are difficult to compare with other reported results in literature, for the reasons we described before. We did not find publications using as many labels as we did. For example, if we refer to the work of [26], which deals with a similar dataset of document images, they obtained a recognition rate of 94.4% for 9 labels (Title, Author, Page Number, Abstract, KeyWords, Copyright, Sec-

Table 4 Recognition rates for all the labels for a standard MLP, a Static PNN, and a Dynamic PNN

Labels	#	MLP (%)	SPNN (%)	DPNN (%)
Title	15	93.3	100.0	100.0
Author	44	88.6	90.9	93.2
E-mail	5	0.0	80.0	100.0
Locality/Affiliation	21	47.6	66.7	85.7
Abstract	15	93.3	100.0	100.0
Keywords	14	92.8	92.9	92.9
CR Categories	9	88.9	100.0	100.0
Introduction	73	80.8	80.8	80.8
Paragraph	440	96.1	95.7	97.3
Section	92	97.8	97.8	97.8
SubSection	62	98.3	98.4	98.4
SubSubSection	17	76.4	76.5	82.4
List	69	97.1	98.6	98.6
Enumeration	44	95.4	97.7	97.7
Float	105	91.4	99.1	99.1
Conclusion	38	28.9	28.9	28.9
Bibliography	187	98.9	98.9	100.0
Algorithm	86	95.3	97.7	98.8
Copyright	9	88.8	100.0	100.0
Page Number	30	96.6	93.3	96.7
Acknowledgement	10	70.0	60.0	70.0

tion, Algorithm, Float). With the same set of logical labels, our DPNN reaches 96.3%.

5 Conclusions

We presented an extension of the Static Perceptive Neural Network to a dynamic version, called Dynamic Perceptive Neural Network, using a Time-Delay Neural Network. The new network architecture is designed to recognise the logical structures in document images with several cycles of recognition-correction.

The method uses all the concepts of the static version: introduction of knowledge in each neuron, network topology in hierarchy from fine (the labels to recognise) to coarse (the global context), analysis of the outputs and correction of the inputs (perceptive cycles), input feature clustering for speeding up the recognition.

We introduced the time dimension in the network allowing to take into account the inputs of the preceding cycles during the recognition step. The static version has been extended through the usage of a TDNN. As a consequence, the proposed DPNN is more flexible and more adapted to handle the variability of the input features between each cycle. The correction of these features is also much more efficiently supervised thanks to the weight adaptations provided by the TDNN architecture.

The previous SPNN was already giving encouraging results and the DPNN improves the existing recognition rates. The DPNN has been tested on the MARG dataset and has been compared with other methods in the literature. Although it was not specifically designed to deal with such a database, it is able to outperform most of the previous methods, especially those using rule-based systems and gives similar results to the state-of-the-art on this dataset. As there are no publicly available datasets with many different labels to be recognised, we tested the model on a set of documents where 21 logical structures have been considered. The experiments show that the DPNN outperforms the SPNN and gives roughly 93% of recognition rate versus 90% for the same processing time.

In a future work, we plan to integrate a partial or totally recurrent network in order to provide an architecture with a temporal and space recurrence [27,49]. A further step will be to identify and test more informative input features for dealing with highly difficult cases where it seems that "reading" and "understanding" the content of a block of text is necessary to find the right label for it.

References

1. ABBYY FineReader Engine: http://www.abbyy.com/ocr_sdk/ (2003)
2. Alam, H., Hartono, R., Kumar, A., Rahman, A.F.R., Tarnikova, Y., Wilcox, C.: Assuming accurate layout information for web documents is available, what now? Int. Workshop Document Layout Interpret. Appl. **1**(3), 27–30 (2003)
3. Analyzed Layout and Text Object: http://www.loc.gov/standards/alto/ (2010)
4. Antonacopoulos, A., Pletschacher, S., Bridson, D., Papadopoulos, C.: ICDAR2009 page segmentation competition. Int. Conf. Document Anal. Recognit. **1**(10), 1370–1374 (2009)
5. Belaïd, A., Rangoni, Y.: Structure extraction in printed documents using neural approaches. Mach. Learn. Document Anal. Recognit. Ser. Stud. Computat. Intell. **90**, 21–43 (2008)
6. van Beusekom, J., Keysers, D., Shafait, F., Breuel, T.M.: Example-based logical labeling of document title page images. Int. Conf. Document Anal. Recognit. **1**(9), 919–923 (2007)
7. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. Artif. Intell. **97**(1–2), 245–271 (1997)
8. Brugger, R., Bapst, F., Ingold, R.: A DTD extension for document structure recognition. Int. Conf. Electron. Publ. **1375**(7), 343–354 (1998)
9. Candela, L., Castelli, D., Pagano, P.: A reference architecture for digital library systems: principles and applications. LNCS Digit. Libr. Res. Dev., Springer, Berlin **4877**(1), 22–35 (2007)
10. Conway, A.: Page grammars and page parsing. A syntactic approach to document layout recognition. Int. Conf. Document Anal. Recognit. **1**(2), 761–764 (1993)
11. Côté, M., Lecolinet, E., Cheriet, M., Suen, C.: Automatic reading of cursive scripts using a reading model and perceptual concepts. Int. J. Document Anal. Recognit. **1**(1), 3–17 (1998)

12. Coüasnon, B.: DMOS, a generic document recognition method: Application to table structure analysis in a general and in a specific way. Int. J. Document Anal. Recognit. **8**(2), 111–122 (2006)
13. Coyle, K.: Mass digitization of books. J. Acad. Librariansh. **32**(6), 641–645 (2006)
14. Dengel, A.R., Klein, B.: Smartfix: a requirements-driven system for document analysis and understanding. Int. Conf. Document Anal. Recognit. **2423**(5), 77–88 (2002)
15. Doucet, A., Kazai, G.: ICDAR 2009 book structure extraction competition. Int. Conf. Document Anal. Recognit. **1**(10), 1408–1412 (2009)
16. Ford, G., Thoma, G.: Ground truth data for document image analysis. Symp. Document Image Underst. Technol. **1**(5), 199–205 (2003)
17. Hruschka, H.: Interpretation Aids for Multilayer Perceptron Neural Nets. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin (2005)
18. Hurst, M.: Layout and language: an efficient algorithm for detecting text blocks based on spatial and linguistic evidence. SPIE, Document Recognit. Retr. **4307**(8), 56–67 (2001)
19. Hurst, N., Li, W., Marriott, K.: Review of automatic document formatting. Symp. Document Eng. **1**(9), 99–108 (2009)
20. Hush, D., Horne, G.: Progress in supervised neural networks: what's new since Lippmann? IEEE Signal Process. Mag. **10**(1), 8–38 (1993)
21. Ingold, R., Armangil, D.: A top-down document analysis method for logical structure recognition. Int. Conf. Document Anal. Recognit. **1**(1), 41–49 (1991)
22. Ishitani, Y.: Logical structure analysis of document images based on emergent computation. Int. Conf. Document Anal. Recognit. **1**(5), 189–192 (1999)
23. Kanai, J., Rice, S.V., Nartker, T.A., Nagy, G.: Automated evaluation of OCR zoning. IEEE Trans. Pattern Anal. Mach. Intell. **1**(17), 86–90 (1995)
24. Kim, J., Le, D.X., Thoma, G.R.: Automated labeling in document images. SPIE, Document Recognit. Retr. VIII **4307**(1), 111–122 (2001)
25. Kreich, J., Luhn, A., Maderlechner, G.: An experimental environment for model based document analysis. Int. Conf. Document Anal. Recognit. **1**(1), 50–58 (1991)
26. Krishnamoorthy, M., Nagy, G., Seth, S., Viswanathan, M.: Syntactic segmentation and labeling of digitized pages from technical journals. IEEE Trans. Pattern Anal. Mach. Intell. **7**(15), 737–747 (1993)
27. Küchler, A., Goller, C.: Inductive learning in symbolic domains using structure-driven recurrent neural networks. German Conference on Artificial Intelligence: Advances in Artificial Intelligence **1137**(20), 183–197 (1996)
28. Le Cun, Y., Bottou, L., Orr, G., Muller, K.: Efficient backprop. Neural Netw. Tricks Trade **1524**, 9–50 (1998)
29. Lervik, J., Brygfjeld, S.: Search engine technology applied in digital libraries. ERCIM News **1**(66), 18–19 (2006)
30. Lin, C., Niwa, Y., Narita, S.: Logical structure analysis of book document images using contents information. Int. Conf. Document Anal. Recognit. **2**, 1048–1054 (1997)
31. Lodwich, A., Rangoni, Y., Breuel, T.: Evaluation of robustness and performance of early stopping rules with multi layer perceptrons. Int. Joint Conf. Neural Netw. **1**(19), 1877–1884 (2009)
32. Logar, A.M., Corwin, E.M., Oldham, W.J.B.: A comparison of recurrent neural network learning algorithms. IEEE Trans. Neural Netw. **2**, 1129–1134 (1993)
33. Schenkel, M.I., Guyon, D.H.: On-line cursive script recognition using time delay neural networks and hidden markov models. Int. Conf. Acoustics Speech Signal Process. **2**, 637–640 (1994)
34. Maddouri, S.S., Amiri, H., Belad, A., Choisy, C.: Combination of local and global vision modelling for arabic handwritten words recognition. Int. Workshop Frontiers Handwrit. Recognit. **1**(8), 128–135 (2002)
35. Mao, S., Kim, J.W., Thoma, G.R.: Style-independent document labeling: design and performance evaluation. SPIE, Document Recognit. Retr. XI **5296**(1), 14–22 (2003)
36. Mao, S., Rosenfeld, A., Kanungo, T.: Document structure analysis algorithms: a literature survey. SPIE, Electron. Imaging **50**(10), 197–207 (2003)
37. Mao, S., Thoma, G.R.: Bayesian learning of 2D document layout models for automated preservation metadata extraction. Int. Conf. Vis. Imaging Image Process. **1**(4), 329–334 (2004)
38. MARG: Medical Records Groundtruth: http://marg.nlm.nih.gov (2003)
39. Marinai, S., Gori, M., Soda, G.: Artificial neural networks for document analysis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **27**(1), 23–35 (2005)
40. McClelland, J., Rumelhart, D.: An interactive activation model of context effects in letter perception. Psychol. Rev. **88**(1), 375–407 (1981)
41. Nagy, G.: Twenty years of document image analysis in PAMI. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 38–62 (2000)
42. Pearlmutter, B.A.: Gradient calculations for dynamic recurrent neural networks:a survey. IEEE Trans. Neural Netw. **6**(5), 1212–1228 (1995)
43. Rangoni, Y., Belaïd, A.: Data categorization for a context return applied to logical document structure recognition. Int. Conf. Document Anal. Recognit. **1**(8), 297–301 (2005)
44. Rangoni, Y., Belaïd, A.: Document logical structure analysis based on perceptive cycles. Conf. Document Anal. Syst. **1**(7), 117–128 (2006)
45. Sainz Palmero, G.I., Cano Izquierdo, J.M., Dimitriadis, Y.A., Lopez Coronado, J.: A new neuro-fuzzy system for logical labeling of documents. Int. Conf. Pattern Recognit. **18**(4), 431–435 (1996)
46. Sainz Palmero, G.I., Dimitriadis, Y.A.: Structured document labeling and rule extraction using new recurrent fuzzy-neural systems. Int. Conf. Document Anal. Recognit. **1**(5), 181–184 (1999)
47. Schema for representing OCR results exported from FineReader 6.0: http://www.abbyy.com/FineReader_xml/FineReader6-schema-v1.xml (2002)
48. Siggraph: http://www.siggraph.org/s2003/ (2003)
49. Sperduti, A., Starita, A.: Supervised neural networks for the classification of structures. IEEE Trans. Neural Netw. **8**(3), 714–735 (1997)
50. Summers, K.: Near-wordless document structure classification. Int. Conf. Document Anal. Recognit. **1**(3), 462–465 (1995)
51. Szilas, N., Cadoz, C.: Adaptive networks for physical modeling. Neurocomputing **20**(1-3), 209–225 (1998)
52. Tateisi, Y., Itoh, N.: Using stochastic syntactic analysis for extracting a logical structure from a document image. Int. Conf. Pattern Recognit. **12**(2), 391–394 (1994)
53. Wan, E.: Time series prediction by using a connectionist network with internal delay lines. In: Weigend A.S., Gershenfeld N.A. (eds.) Time Series Prediction. Forecasting the Future and Understanding the Past, SFI Studies in the Science of Complexity, vol. 17, pp. 195–217. Addison-Wesley, CA (1994)
54. Yanikoglu, B.A., Vincent, L.: Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation. Pattern Recognit **31**(9), 1191–1204 (1998)