DAN: An Automatic Segmentation and Classification Engine for Paper Documents

L. Cinque, S. Levialdi, A. Malizia, and F. De Rosa

Dept. of Information Science,
Universita' "La Sapienza" di Roma Via Salaria 113, 00198 Roma, Italy
{cinque,levialdi,malizia,derosa}@dsi.uniroma1.it

Abstract. The paper documents recognition is fundamental for office automation becoming every day a more powerful tool in those fields where information is still on paper. Document recognition follows from data acquisition, from both journals, and entire books in order to transform them in digital objects. We present a new system DAN (Document Analysis on Network) for Document recognition that follows the Open Source methodologies, XML description for documents segmentation and classification, which turns to be beneficial in terms of classification precision, and general-purpose availability.

1 Introduction

The document analysis field is related to the semi-automatic management of paper documents. Such automation systems have been used in several fields: typically cataloguing and storing of documents, blueprints, faxing servers, character recognition software. Many different approaches have been used and standards are still lacking. Typical problems of document analysis systems are: layout segmentation, syntactic parsing, but also the selective extraction of information such as document types and semantic contents [1,2]. Most document processing packages are designed either for document recognition (i.e., indexing and archiving of document images) or for data acquisition (i.e., extracting data from filled forms). Document recognition is essentially the process of converting paper documents into digital images and indexing such data. Images are stored as data files (typically as TIFF files) and together with indexes are stored in a content management system. Most forms oriented products available on the market today instead require the user to redesign his forms in order to achieve acceptable recognition rates. In our case we can show that our model will work both with document images and forms. A central question is how to evaluate the effectiveness of such complex document analysis systems involving rather distinct components. In addition, we are interested in comparing different document categorization algorithms by their effectiveness [3,4,5]. A lot of different approaches have been proposed in the Document Analysis field. The existing works could be classified into three main classes: top-down systems, bottom-up systems, and mixed systems.

D. Lopresti, J. Hu, and R. Kashi (Eds.): DAS 2002, LNCS 2423, pp. 491–502, 2002.

Top-down methods start from the original image, and, using cut operations obtain regions, which have to be catalogued [6,7,8]. In order to be efficient, this method needs a priori information, which leads to a non-general approach. Bottom-up systems start from small parts of the document image, and then merge regions having similar characteristics [9,10]. Mixed systems instead optimize the use of both bottom-up and top-down methods in order to avoid their limititations [11,12]. Our approach, as shown in the following paragraphs, falls in the mixed systems class.

2 System Architecture Overview

Our system uses a split and merge technique similar to the approach that has been obtained by Nagy's X-Y cut algorithm [13], but instead of working top-down, we use the recognized horizontal and vertical lines to cut the image into small regions, merging with a quad-tree technique and image processing algorithms. We have built a system based onto these different phases, in order to perform the document classification in a modular and efficient way. The DAN (Document Analisys on Network) system is based onto the Segmentation and Classification engine, which takes a document as input, and performs an automatic classification, which is then presented to the end user with an interface called DANEditor. The DANEditor allows users to evaluate the automatic classification performed by the system and edit the textual annotation for indexing. The user can also edit the recognized regions by the classification engine and adjust their values and sizes. The output of these phase is an XML file which could be imported in a server with an xml DB for indexing and querying. The Figure (1) shows the DANEditor module, with it's tools for annotating and edit the automatic recognized region sizes and properties.

3 Segmentation and Classification Engine

The DAN system architecture includes four main components: the preprocessor (1), the split module (2), the merge module (3) and the classification module (4) as shown in Figure (2). The preprocessing algorithm (1) component is performed in order to enhance the quality of input data, removing portions of the image, which could be considered as noise. Moreover in this phase the original image of the document is loaded into main memory so obtain better computation performances. The Split module (2) takes input from the preprocessing phase and applies a particular quad-tree technique in order to split the document into small blocks. Then the Split module passes its result to the Merge module (3), which applies pre-classification criterion merging similar regions into big regions. We use local operators with variable thresholds in order to compute the pre-classification phase. Finally the engine of our system is in the Classification module (4) which executes the classification procedure according to the classification logic. In fact, the "brain" of our system is this classification engine, which

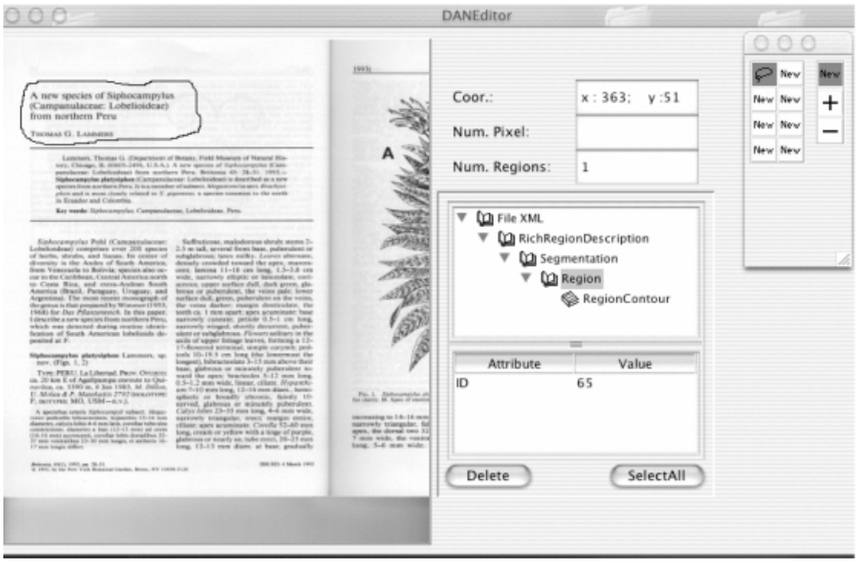

Fig. 1. Annotation and Editor tool diagram.

outputs segmented regions and their attributes, like type and size, in an XML file.

4 Split and Merge Phases

In this section we will explain, the details of the documents segmentation phases. We will show the techniques used for preprocessing, splitting and then merging the regions, with pre-classification information. It is important to understand that all of these phases are optimized for using them in a general environment without "a priori" knowledge of the documents format or size.

4.1 Preprocessing

In a production environment, the preprocessing stage of the recognition process may be very important in order to enhance the quality of the input data. The preprocessing phase performs two steps: loads the scanned image in main memory and computes the gray-level histogram extracting the three parameters discussed below. This approach is due to the fact that we want to reduce the amount of computations in the preprocessing phase obtaining a quick response method. Starting from a 256 gray levels document we compute the RI1 parameter, as the maximum value in the first half of the gray-level histogram. This value is not the absolute maximum of the histogram function, but considering intervals in a window of $\varepsilon = 5$ gray levels length. Then for the RI2 parameter we compute it in the same way but on the other half of the histogram, thus considering the

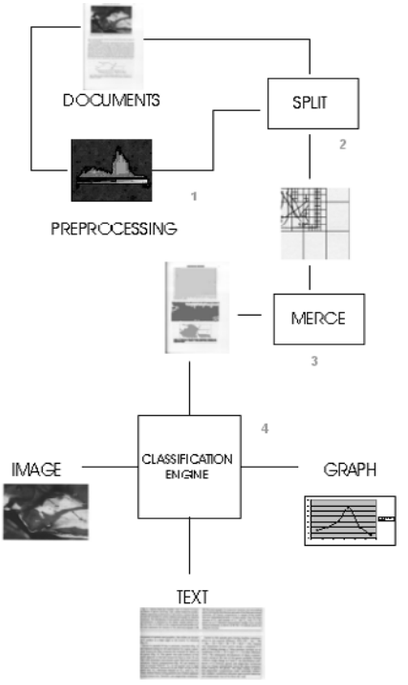

Fig. 2. Segmentation and Classification Engine architecture diagram.

whitening colors. Finally the last parameter RI3, which represents the point of separation between background and text, is the minimum between the RI1 and RI2 parameter. The use of these parameters will help in the split and merge phase in order to let our method adapt to different documents.

4.2 Split

The split procedure is based on two functions: the mean and the variance of scanned documents. Using the mean function we set a window (ranging along x and y coords) of pixels and compute the value:

$$f_m(i,j) = \left[\frac{\sum_i \sum_j f(i,j)}{(x_w - x_i + 1)(y_w - y_i + 1)} \right], \tag{1}$$

with $f(i,j)$ representing the gray-level values of the image pixels and (x_i, y_i), (x_w, y_w) the pixels between the chosen window in which we apply the mean operator.

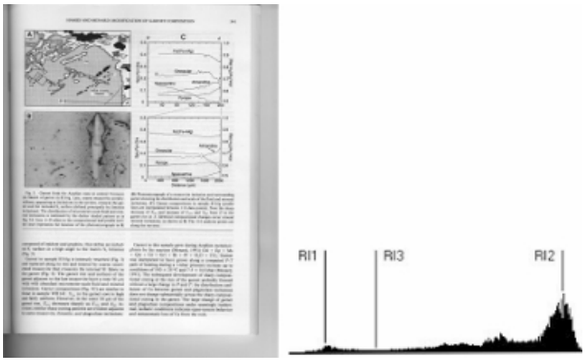

Fig. 3. Original document and its gray-levels histogram, we can see also the three parameters $RI1$, $RI2$, $RI3$.

Calling that mean value $f_m(i,j)$, we can compute the variance operator for every pixels of the document image, as described in the above function:

$$f_v(i,j) = \left[\sqrt{\frac{\sum_i \sum_j (f(i,j) - f_m(i,j))^2}{(x_w - x_i + 1)(y_w - y_i + 1)}} \right] \tag{2}$$

During the split phase, the whole image of the document is split according to the extracted vertical and horizontal lines as well as the boundaries of recognized images. This results in many small zones (block sizes are within a range depending on the size of the image). We have decided to use an already known technique but modeling it with our specific parameters. This technique is known as *quad-tree*[14]. We use the quad-tree decomposition to perform spatial segmentation by assigning a condition by which nodes are split, according with the Ojala and Pietikainen idea [15]. We also added a post-processing routine for adjoining similar spatially adjacent nodes with different parents. A final block grouping stage can be added to merge all similar blocks to obtain arbitrarily shaped regions.

In order to choose if going on with splitting or stop the method we use the mean and variance operators. We simulate a split step in four regions, and using the mean and variance of the regions we find if these values are in a range; more deeply we have a variable threshold range (depending on the pre-processing phase) in order to distinguish low variance regions (thus with less information); moreover ranges are adaptive depending on the region area, in fact ranges are related to the inverse of region areas thus enhancing precision with wider regions. Finally we define the result of the split procedure so that:

$$\cup_i r_i = I \quad and \quad \cap_i r_i = \emptyset \tag{3}$$

We apply labels to regions in order to pre-classify them. These pre-classification is useful to pass information to the merge phase. We define three

class of regions: *text-graph*, *image*, *background*. In case of low variance and low mean values we label the region as *background*, instead if we have high variance and low mean we label the region as *text-graph*; else the label will be *image*. Let we define the classify function as:

$$f(r_i) = \{Text - Graph, Image, Background\}. \tag{4}$$

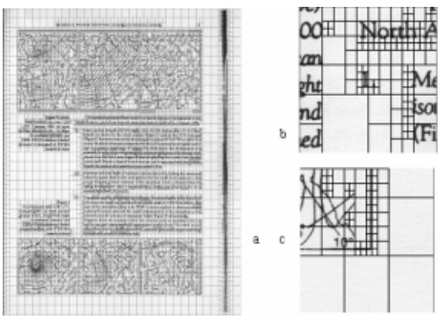

Fig. 4. Original document after the split phase (a), and two highlighted particulars (b) and (c)

4.3 Merge

The split operation results in a heavily over-segmented image. The goal of the merge operation is to join neighboring zones to form bigger rectangular zones. The first phase of merging consists of connecting neighbors regions with the same pre-classification value. Let's define the merge procedure above descript in terms of a boolean function regarding the r_i and r_j generic regions:

$$R = \{r_i | r_i \in I, \exists k \in P | \forall i, f(r_i) = k, \forall (i,j) C(i,j)\} \tag{5}$$

$$C(i,j) = \{(f(r_i) = f(r_j)) \wedge adjacent(i,j)\} \tag{6}$$

$$P = \{Text - Graph, Image, Background\} \tag{7}$$

Using only pre-classification we don't have all the information we need, but with this approach we follow one of the targets of our method, the computing performance efficiency. The second step of the merging phase is the Union phase. The Union procedure will be used to enhance the pre-classification results. First of all, the regions, which are in the external edges of the document, are removed, then all other regions will be consider for the further phase. Now let's introduce the Macro-Regions, as those regions with a spanned area greater than a threshold, which is related to the document region sizing. All the adjacent Macro-Regions with the same pre-classification values will be merged thus obtaining our segmentation.

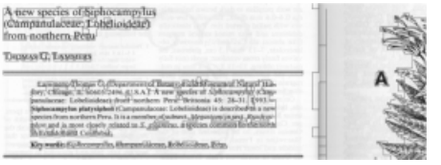

Fig. 5. Original document after the first phase of the merge and its highlighted particular.

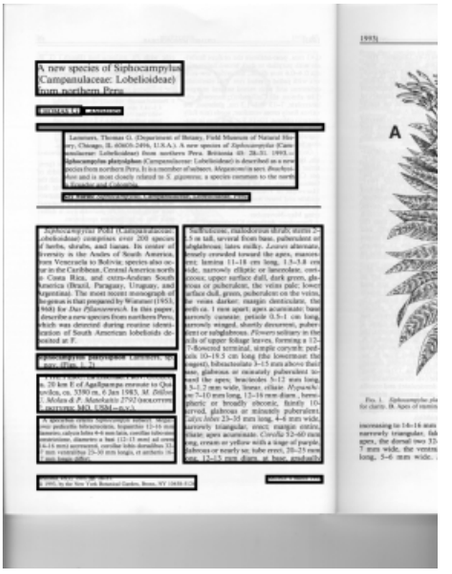

Fig. 6. Original document after the second phase of the Merge, the "Union" phase.

5 Region Classification

Now, regions of interest has been segmented, so we have to catalogue them . In order to catalogue regions we introduce global operators, which are universally used in Image Processing. These global operators will work on the entire segmented areas. In order to interpret in a precise manner all these regions, more operators than in the split and merge phase, are required.

5.1 Global Operators

The selected global operators are: the mean, which is applied to the region with the median, the threshold that computes the values of the darkness pixels in the region, the variance which is an average square difference, the gray-levels which consists of the grays, blacks and white pixels, edges that are related to color

variations in the document and ratio which is the ratio between the region and the enclosing rectangle.

- $mean(M_k) = \frac{\sum_l \sum_i \sum_j f(i,j)}{\sum_l \sum_i \sum_j 1}$, with $l \in M_k$, with M_k is a Macro-Region
- $median(M_k) = min_m m | \sum_{k=0}^{m} f_k(R) \leq \sum_l \sum_i \sum_j \frac{1}{2} f_k(R) = \#\{(i,j)|f(i,j) = k\}$
- $threshold(M_k) = \frac{\sum_l \sum_i \sum_{j|f(i,j)>k} 255}{\sum_l \sum_i \sum_j 1}$, with $l \in R$
- $variance(M_k) = \sqrt{\frac{\sum_l \sum_i \sum_j (f(i,j) - mean(M_k))^2}{\sum_l \sum_i \sum_j 1}}$, with $l \in R$
- $gray - levels(i,j) = \{0, 127, 255\}$, 0 (black) if $f(i,j) < (2 * RI1 + RI3)/3$, 255 (white) if $f(i,j) < (2 * RI2 + RI3)/3$, 127 (gray) else
- edges $g(i,j) = \{0, 1, g(i,j-1)\}, 0$ if $f(i,j) < (2 * R21 + RI3)/3$, 1 if $f(i,j) < (2 * RI1 + RI3)/3$, $g(i,j-1)$ else
- $edge(M_k) = \frac{\#\{(i,j)|g(i,j) \neq g(i,j-1)\} * 256}{\#\{(i,j)\}}$
- $ratio(M_k) = \frac{\sum_l \sum_i \sum_j 1}{(Rx_f - Rx_i + 1)(Ry_f - Ry_i + 1)}$, with $l \in R$

5.2 Classification Procedure

We will introduce the $F(M) = \{Text, Graph, Image\}$ as the classification function, where M is a Macro-Region, which will be valorized by the following procedure:

1. if the mean and the variance are higher than RI3 (one of the pre-processing parameters) M could be *Text* or *Graph* and $F(M) = \emptyset$
2. if (white $> 40\%$ and gray $< 25\%$ of pixel $\in M$ and the ratio is less than 60% of threshold(M)) then $F(M) = Text$
3. else if (gray $> 25\%$ or mean is higher then RI3) then $F(M) = Image$
4. else $F(M) = Graph$
5. if $F(M)$ is $\emptyset$ then M is removed (could be background or border).

The DAN system produces an XML (eXtensible Markup Language) description of the regions recognized with the above procedure, which is useful for creating a standard base query system. In fact the user can define a query module based on standard HTML+XML source code using the system output. Moreover users can build their own central repository and acquire different documents in different states or cities and then using Internet to upload the XML region descriptions to a central repository. The users can interact with an Internet Browser in order to produce queries onto the central repository, thus obtaining interesting documents. An example of the XML file produced and the DocRegionDescription DTD (Document Type Definition) used are shown in the following code examples.

Example of an XML file produced by DAN

```
<?xml version="1.0"?>
<!-- XML example -->
<!DOCTYPE DRD SYSTEM "DRD.dtd">
<DRD  FName="2.tif" W="640" H="480">
<Segmentation Type="Auto" Algo="Cinque-Levialdi-Malizia-DeRosa">
<Region ID="1" type="TEXT">
 <NumPixels Num="144"/>
 <Size x1="52" x2="75" y1="5" y2="8"/>
 <Barycenter x="463" y="17"/>
</Region>
</Segmentation>
</DRD>
```

Example of the DocRegionDescription DTD used by DAN

```
<?xml version="1.0" encoding="us-ascii"?>
<!-- DTD "DRD" version 0.93 -->
<!ELEMENT DRD (Segmentation+)>
<!ATTLIST DRD
        Version   CDATA   #IMPLIED
        FileName   CDATA   #REQUIRED
        Height   CDATA   #REQUIRED
        Width   CDATA   #REQUIRED>
<!ELEMENT Segmentation (Region+)>
<!ATTLIST Segmentation
        Type   (Auto | Manual)   #IMPLIED
        Param1   CDATA   #IMPLIED
        Param2   CDATA   #IMPLIED
        Param3   CDATA   #IMPLIED
        Algo   CDATA   #IMPLIED>
<!ELEMENT Region ((NumPixels | Barycenter | Size | Shape| Label)+)>
<!ATTLIST Region
        ID   CDATA   #REQUIRED
        Type  CDATA   #REQUIRED>
<!ELEMENT NumPixels EMPTY>
<!ATTLIST NumPixels
        Num   CDATA   #REQUIRED>
<!ELEMENT Barycenter EMPTY>
<!ATTLIST Barycenter
        x   CDATA   #REQUIRED
        y   CDATA   #REQUIRED>
<!ELEMENT Size EMPTY>
<!ATTLIST Size
        x1   CDATA   #REQUIRED
        y1   CDATA   #REQUIRED
        x2   CDATA   #REQUIRED
        y2   CDATA   #REQUIRED>
```

```
<!ELEMENT Shape EMPTY>
<!ATTLIST Shape
        Shape   CDATA   #REQUIRED>
<!ELEMENT Label EMPTY>
<!ATTLIST Label
        Label   CDATA   #REQUIRED>
```

5.3 Noise Reduction

We have also improved our system with noise reduction of original document caused by scanning, borders, extra pages and transparencies. Scanning is a very important issue, and we have considered images scanned at 200-400 dpi range. Using the RI_i (described in preprocessing phase) we avoid the problem of having scanned images darker or lighter than the original paper; in fact the problem of acquisition could affect splitting parameters causing errors. Borders are often obtained from photocopies or central part of two pages magazines or books. In order to avoid errors of this kind, like black borders, we use a post-process routine. This routine uses Macro-Regions which are a small number thus having good computation performances. The routine finds a central region and tries to enlarge that region joining it to Macro-Regions with at least one edge into the central region. Extra pages are caused acquiring not only the original page but also part of another page in a two pages book or magazine. Different techniques exist in order to correct this problem; we choose to eliminate all the regions that after the merge phase are adjacent to external borders. Transparencies are originated by thin paper or very old ink absorbed by the paper. We have avoided this problem, which can cause region detection errors, by the use of the RI_i parameters, in fact these parameters help our method to understand where information are in a manner which is independent from gray-levels absolute values thus avoiding errors from transparencies.

6 Experimental Results

We have tested our system over the UW-II database that is the second in series of document image databases produced by the Intelligent Systems Laboratory, at the University of Washington, Seattle, Washington, USA. UW-II is designed and constructed by a team of undergraduate and graduate students, led by Dr. Ihsin Phillips and Dr. Robert Haralick at the Intelligent Systems Laboratory. The database contains 624 English journal document pages (43 complete articles) and 63 MEMO pages. All pages are scanned pages.

Each document in the database has been taken from scientific journals and contains text, graphs and images. The average sizes of documents are 2500 x 3000 pixels and about 1 MB of storage required for each document. We can see in table 1, a sample of 8 documents taken from the database. The fields respectively indicate D (Document ID), R (Regions number), C R (Correctly Recognized), N C R (Not Correctly Recognized), T B D (To Be Defined, this

are regions for which our system hasn't enough information to decide if they're text, graph or image). This sample even with this low number of documents is an important example of the results obtained over the entire database because of the different qualities of the selected documents.

Table 1. A sample of 8 eigth docs extracted from the database.

D	R	C R	N C R	T B D
1	14	14	0	0
2	8	8	0	0
3	10	9	1	0
4	11	8	3	0
5	5	5	0	0
6	10	8	2	0
7	9	7	2	0
8	8	6	1	1
TOTAL	75	65	9	1

We have tested our method over the entire database (600 images) obtaining an 84% of correctly recognized regions, 14% of not correctly recognized and 2% to be defined. The 84% of correctly recognized regions could be sub dived into a 59% of entirely recognized and 25% of partially recognized, which means a single text regions was interpreted as two text regions (this usually happens in titles where there are many spaces).

Table 2. Values over the entire database.

D	R	C R	N C R	T B D
600	5625	4725	788	112

7 Conclusion and Further Works

This paper presents our approach for segmenting and classifying documents. In this approach we have worked on the reduction of the split operations using a preprocessing routines. In future this work could be improved in sub-classification, such as the title and subtitle detection inside text block, table recognition, and the use of some OCR technique in the text regions. Moreover the system could be improved adapting parameters to specific fields of applications where documents are of a certain type, thus we could obtain a better percentage of correctly recognized regions. The goal is general document recognition system that assists the user in recognizing documents also using segmentation algorithm [16] for classifying image regions, and some shape algorithm [17] in order to model documents with convex regions, which are more precise representation than the rectangular ones. Finally, our DAN system may be consider as an automatic documents

recognition system with region segmentation and classification, with the chance of semi-automatic or interactive action by the user; thus this system should be very suitable for automatic recognition of image database and batch acquisition of multiple documents types and formats.

References

1. F. Bapst, R. Brugger, and R. Ingold. Towards an Interactive Document Recognition System. *Internal working paper* 95– 09, IIUF-Université de Fribourg, March 1995.
2. B. Gatos, S. L. Mantzaris, K. V. Chandrios, A. Tsigris, and S. J. Perantonis. Integrated algorithms for newspaper page decomposition and article tracking. In *ICDAR'99: Fifth International Conference on Document Analysis and Recogntion*, pages 559–562, Bangalore, India, Sept. 1999.
3. D. Lewis. Representation and Learning in Information Retrieval. *PhD thesis*, Department of Computer Science, University of Massachusetts, 1992.
4. Y. Yang and J. Pedersen. A Comparative Study on Feature Selection in Text Categorization. *Machine Learning. Proceedings of the 14th International Conference (ICML 97)*, pages 412-420, Nashville, TE, USA, July 6–12 1997.
5. M. Junker and R. Hoch. An Experimental Evaluation of OCR Text Representations for Learning Document Classi- fiers.*International Journal on Document Analysis and Recognition*, 1(2):116–122, June 1998.
6. G. Nagy, S. Seth, S. Stoddard. Document analysis with an expert system.*Pattern Recognition*, Vol. 19 N.1, pp 149–159, 1986.
7. K. C. Fan, L. S. Wang, Y. K. Yang. Page segmentation and identification for intelligent signal processing. *Signal Processing*, Vol 45 N.2, pp 329–346, 1995.
8. T. Pavlidis, J. Zhou. Page segmentation and classification. Graphical Models and Image Processing. *CVGIP*, Vol 54, pp 484–496, November 1992.
9. L. Fletcher, R. Katsuri. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol 10, pp 910–918, 1998.
10. J. Litcher, F. Hones. Layout extraction of mixed mode document. *IEEE Transaction on Machine Vision and Application*,Vol 6, pp 477–486, 1994.
11. L. O'Gorman, R. Katsuri. Document image analysis. *IEEE Computer Society.* Press Los Alamos, California, pp 161–181, 1995.
12. F. Esposito, D. Malerba, G. Semeraro, E. Annese, G. Scafuro. An experimental page layout recognition system for office document automatic classification: an integrated approach for inductive generalization. *Proceedings of 10th ICPR*, pp 557–562.
13. G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, July 1992
14. M. Span, R. Wilson. A quad-tree approach to image segmentation which combines statistical and spatial information. *Pattern Recognition*, Vol.18:257–269,1985.
15. T. Ojala, M. Pietikainen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, Vol.32:477–486,1999.
16. L. Cinque, F. Lecca, S. Levialdi, S. Tanimoto – "Retrieval of images using Rich Region Descriptions". *Proceeding of the International Conference of Pattern Recognition*, Brisbane, Australia, 1998, Volume I, pp. 899–109.
17. L. Cinque, S. Levialdi, A. Malizia, K.A. Olsen. "A Multidimensional Image Browser". *Journal of Visual Language and Computing*,Vol. 9, 1998.