2014 11th IAPR International Workshop on Document Analysis Systems

Ground-truth and Performance Evaluation for Page Layout Analysis of Born-digital Documents

Xin Tao, Zhi Tang, Canhui Xu and Liangcai Gao
Institute of Computer Science and Technology
Peking University
Beijing, China
Email: {jolly.tao, tangzhi, gaoliangcai}@pku.edu.cn, ccxu09@yeah.net

***Abstract*—In this paper, a new dataset is proposed for page layout analysis of born-digital documents. By extracting uniformly the document contents, an XML based data format is designed in terms of raw data and structure data. Utilizing a self-developed ground-truthing tool, a public dataset is constructed from diverse styles of document resources. With consideration of physical segmentation and logical labeling, automatic performance evaluation methods are adjusted to cope with different scenarios. The applications of the proposed dataset have shown that it is suitable for evaluating various layout analysis tasks.**

***Keywords*—dataset, born-digital document, ground-truthing, performance evaluation**

I. Introduction

Page layout analysis is a prerequisite step in the pipeline of document understanding. A complete layout analysis process usually involves phases of physical segmentation and logical labeling. Its performance has significant impact on succeeding structure analysis processes at higher semantic levels. For the purpose of comparing the considerable layout analysis methods proposed over past two decades and those to be reported, research on ground-truth construction [1]–[5] and evaluation metrics [6]–[8] has attracted continuous interests, since no single algorithm is uniformly optimal due to document layout versatility [9].

While majority of current layout analysis research has been applied to image based document pages, there exists increasing attention on born-digital fixed-layout documents, typically PDF (Portable Document Format) documents. During last decade, pioneering research groups have devoted to layout analysis of legacy PDF files. DIVA group proposed reverse engineering tools [10], [11] to analyze the embedded resources of PDF files, generate physical structures, and then rebuild the logical structure. Déjean and Meunier [12] reconstruct the logical hierarchy of PDF documents through extraction of internal objects and recognition of table of contents. Marinai's research dedicated to table of contents detection during the conversion of PDF books to reflowable XHMTL based format [13]. Tang focuses on converting fixed-layout documents to fluid CEBX documents [14].

It is known that there exists no standard benchmarks or evaluation sets for PDF based layout analysis. Existing datasets available for page layout analysis are built on document images. The public UW-III dataset uses DAFS as data format and contains image based physical segmentation and logical classification ground-truth [15]. The PRImA dataset [1] has been used for the ICDAR page segmentation competition. Generally, these datasets use rectangle or polygon for region representations, which do not reflect the PDF format. Though pages from PDF documents can degenerate to images ignoring the content streams, the attributes of primitive objects are additionally beneficial for layout analysis [10]. Therefore, a dataset conforming to PDF format is highly desirable, especially for research on repurposing of legacy fixed-layout documents, like extraction and conversion.

This paper presents a dataset constructed from PDF documents for in-depth evaluation of layout analysis. The dataset adopts an identifier-based representation to describe structure objects. A graphical user interface tool is developed to assist ground-truthing. Performance metrics are also adapted to this representation for physical, logical and overall evaluations. The paper is organized as follow. Section II introduces the format of the dataset. Section III describes the dataset construction workflow. Evaluation schemes for analysis tasks are formulated in section IV. Finally, section V concludes the paper.

II. Data Format Design

Compared with document images, a PDF document page contains a content stream that fully specifies its appearance and formatting. The content stream consists of any combination of primitive objects (text, images and graphics) necessary to display. With an eligible parser, accurate description of each visible content can be obtained from the page. In addition to pure pixels, these content attributes give richer information and thus can be profitable for layout analysis task. Ideally, the physical structure of a page can be regarded as a composition hierarchy built upon atomic components, and logical labels are assigned to physical segments to reflect their functional roles.

For a given page, no matter how the upper structures differ, its building units remain unchanged. In our dataset, The primitives are stored apart from structure information. An advantage of such separation is that structures from either ground-truth or analysis algorithms may share the same set of primitives. Unique identifiers are attached to both primitives and structure objects. Instead of outline based representation, segments at higher levels are precisely defined using the identifiers of their components. This representation allows ground-truthing tool to use easier shapes (e.g., bounding boxes as in our case) to display the content objects, because possible overlapping regions are disambiguated through identifiers. The identifers also offer a chance of more accurate evaluation metrics.

978-1-4799-3243-6/14 $31.00 © 2014 IEEE
DOI 10.1109/DAS.2014.37

Though it is possible to record structure information within the same documents using features like "Tagged PDF" of the PDF format, such solution will make the ground-truth less transparent and difficult to understand. On the other hand, direct investigation of the PDF documents at primitive level requires a parser, which is a burden for researchers. Hence, the data is better represented by an intermediate format preserving the attributes of PDF primitive contents and easy to import. In our dataset, PDF primitives and layout structures are represented in XML formats. In this way, the data is more comprehensible, and can be parsed using any off-the-shelf XML tools.

A. Raw Data

A page from fixed-layout document is perceivable in two ways: visual appearance and primitive content attributes. These types of data are "raw" since they are extracted from original documents prior to execution of any layout analysis algorithm. The two aspects of page information are stored separately. Appearance of a whole page is exported as a raster image with resolution of 300 DPI. This image not only provides intuitive impression which aids ground-truthing, but also serves as source of pixel based feature engineering during analysis.

Primitive contents derived from PDF files are described in XML format. There are three types of primitives parsed from PDF documents: text, image and graphic, following the original PDF format respectively. The primitives are regarded as "frozen" in the context of evaluation, which means that their identifiers and attributes are immutable though the structures at higher levels may vary. XML of raw primitives organizes the data as follows:

- Text. Text content information contains character codes and its attributes like font family, font size and text location in terms of bounding box coordinates.
- Image. Bounding boxes of image primitives are available. Their pixel values are obtainable through referring to the raw image of the whole page.
- Graphic. Vector graphics are described as path operations used for drawing straight lines, rectangles and cubic curves, etc.

Each primitive has its unique identifier. All the coordinates are expressed in multiples of 1/72 inch. These primitives are distill by applying a commercial PDF parser engine. Fig 1(a) illustrated the raw XML description of three kinds of primitives.

B. Structure Data

Despite of the precise descriptions of primitives, PDF generally has no structure information at higher level. Physical and logical structures are supported by PDF format, but they are not a requirement in PDF manufacture. These structures are missing in most PDF documents, which are exactly the targets of tasks like layout analysis and document understanding.

Page structure data in our dataset currently serve for two purposes: physical segmentation and logical labeling. Physical segmentation is to divide the page contents into a hierarchy, where segments at higher level are compositions

```
<?xml version="1.0" encoding="UTF-8"?>
<raw:page
  xmlns:raw="http://www.founderrd.com/marmot/schema/1.1/raw"
  pageNum="2">
  <box x="0.000" y="0.000" w="512.000" h="768.000"/>
  <contents>
    <chars>
      <char id="p2t40c0" char="物" textState="1">
        <box x="395.108" y="747.988" w="8.037" h="8.037"/>
      </char>
      <char id="p2t41c0" char="理" textState="1">
        <box x="404.235" y="748.437" w="8.071" h="7.174"/>
      </char>
    </chars>
    <images>
      <image id="p2i363">
        <box x="19.541" y="200.899" w="226.771" h="258.695"/>
      </image>
    </images>
    <paths>
      <path id="p2p1557">
        <operations>
          <m>17.700 31.500</m>
          <l>499.800 31.500</l>
        </operations>
        <box x="17.700" y="31.500" w="482.100" h="0.300"/>
      </path>
    </paths>
  </contents>
</raw:page>
```

(a) Sample of raw data

```
<?xml version="1.0" encoding="UTF-8"?>
<physical:page
  xmlns:physical="http://www.founderrd.com/marmot/schema/1.1/physical"
  pageNum="2">
  <contents>
    <fragments>
      <fragment id="p2f37" children="p2t40c0 p2t41c0" logical="body"/>
      <fragment id="p2f38" children="p2i363" logical="figure"/>
      <fragment id="p2f45" children="p2p1557" logical="footer"/>
    </fragments>
    <blocks>
      <block id="p2b1" children="p2f37 p2f38"/>
    </blocks>
  </contents>
</physical:page>
```

(b) Sample of structure data

Fig. 1. Examples of XML description of page raw data and structure data

of components at lower level. All physical segments can be decomposed down to primitives. In this work, the relationship of containment is established through taking advantage of identifiers. Physical segmentation data is currently presented in two levels:

- Fragment. Fragment aggregates homogeneous primitives with proximity and belonging to the same type of content stream. For example, textual fragment are usually text lines grouping characters attached closely along horizontal direction. Each fragment has its own unique identifier. The identifiers of aggregated primitives are also recorded in a fragment.
- Block. Block is a cluster of fragments. For example, textual block is an aggregation of text line fragments. Similarly, each block has its unique identifier and identifiers of its children fragments'.

Since the compositions are recorded using identifiers, regions of physical segments are not explicitly given. The attributes of physical segments, like bounding box and text, can be propagated from its children in a bottom-up manner. Logical label is

enclosed in physical segment as an attribute, so as to indicate its semantic role. The logical label set commonly includes body, title, list, figure, figure caption, table, table caption, equation, header, footer, page number, marginal note and foot note, etc. Fig 1(b) gives an example of XML descriptions of both hierarchical physical fragment, block and logical labels.

III. Dataset Construction

A new dataset is constructed from a wide range of document sources so as to represent various layouts and styles. And a graphical user interface tool is developed to efficiently generate ground-truth which is then stored in the data format introduced above with XML representation. The dataset is publicly available at http://www.icst.pku.edu.cn/cpdp/data/marmot_data.htm

A. Document Selection

Our dataset contains pages from publicly available electronic documents, in both Chinese and English. Styles of these documents range from technical journals, magazines to books, with layouts of single or multiple columns. Among these pages, 2000 pages are partially labeled for task of table detection and 400 pages for mathematical formula detection. Our recent contribution to this dataset with increasing size is to fully label 300 pages selected from over 40 born digital documents, most of which belong to books related to social science, business, medicine and technology. The fragments and blocks in each page are completely segmented and labeled with the aim of assessing a complete PDF document understanding system, which is invaluable to reflow the fixed layout documents.

B. Ground-truthing

The ground-truthing of the dataset involves physical segmentation and logical labeling. We also developed a wxpython-based annotator to accelerate ground-truthing. The annotator renders the whole page, offering a better visual perception. The raw primitive and physical segments are represented with their bounding boxes in distinct color. Their logical roles, if available, are indicated with text labels. Figure 2 shows the visualization of a manually labeled example in the scales of primitive, fragment and block. Bounding boxes, identifiers, children elements and logical labels are recorded using the data format designed in section II-B.

In our experience, the most time-consuming part of ground-truthing is selecting and grouping of text primitives in initial steps. The massive amount and relatively tiny sizes of text can result in numerous burdensome manual actions. To reduce this effort, a heuristic algorithm is used to automatically group horizontally aligned text primitives. The algorithm considers a pair of text primitives as connected if they overlap along y axis and are close along x axis. The horizontal adjacency between two primitives is measured by the minimal distance between vertical edges of their bounding boxes. Following this heuristic, text primitives are divided into groups by their connectivity. Each group of primitives form a new fragment. The heuristic algorithm works properly in most of the situations and sometimes needs minor corrections. On average, this preprocessing step reduces ground-truthing time from 8 minutes to 3 minutes per page.

A typical ground-truthing workflow is summarized as follows:

1) Using an eligible parser, export a PDF page as an image and an XML file containing its raw primitive in format described in II-A.
2) Group the primitives into fragments; group the fragments into blocks.
3) Label the logical roles of physical segments using a context menu.
4) Save the results of segmentation and labeling in format described in II-B.

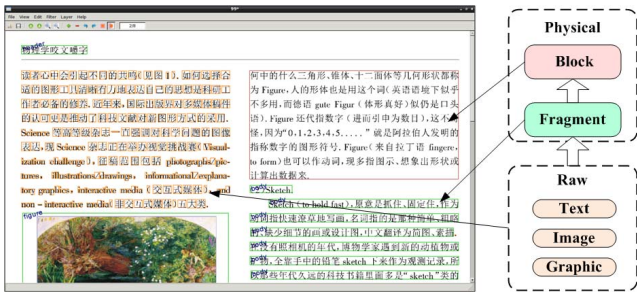

Fig. 2. An illustration of ground-truthing tool. The contents are displayed with their bounding boxes in distinct colors. The logical labels are demonstrated using rotated text in blue.

Since the primitives are raw content objects (text, images and graphics) rather than pixels, the ground-truthing is less arduous than image based annotation. On this constructed dataset, layout analysis methods can be evaluated, and the performance can be examined in detail, as is expounded in following section.

IV. Performance Evaluation

A. Physical Segmentation Performance Evaluation

Physical segmentation aims to divide the basic page primitives such as text, images and graphics into regional groups according to geometric relevance measurements. Grouping can be carried out on different granularities, producing a hierarchical segmentation where segments at higher level are compositions of those at lower level. For example, one block level division can include several fragment level divisions. Set $R = \{r_1, \ldots, r_k\}$ as basic raw page primitives, $G = \{g_1, \ldots, g_m\}$ as ground-truth, $S = \{s_1, \ldots, s_n\}$ as segmentation results, where $g_i = \{r_{i_1}, \ldots, r_{i_{size(i)}}\}$ and $s_j = \{r_{j_1}, \ldots, r_{j_{size(j)}}\}$.

Performance of physical segmentation is measured by the matching degree of S and G. One possible evaluation method is the maximum-weight bipartite graph matching. To construct a bipartite B, the vertex set is defined as $V = G \cup S$, where each vertex is a segment from either G or S. The edges are established as $E = \{(g_i, s_j) | g_i \cap s_j \neq \varnothing\}$. Weights are then assigned to the edges based on how much the segments overlap with each other. A maximum-weight bipartite matching M in B is a set of pairwise non-adjacent edges and the sum of the weights of the edges is maximal. The Hungarian algorithm [16] is one of the methods to find M. The performance of

physical segmentation S with regard to ground-truth G can then be normalized as

$$BGM(S,G) = \frac{\sum_{(g_i,s_j)\in M} w(g_i \cap s_j)}{\sum_{g\in G} w(g)} \quad (1)$$

The weight function w is defined using area or number of primitives.

B. Logical Labeling Performance Evaluation

The target of logical labeling is to assign each physical segment a semantic role, like body, title, figure, etc.. Let $\boldsymbol{o} = \{o_1, \ldots, o_n\}$ be the segments to be labeled, $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ the observations over $\boldsymbol{o}$. The logical labeling is a function $f(\boldsymbol{x}) = \hat{\boldsymbol{y}}$ that maps the observations to labels. $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ the ground-truth labels, $\hat{\boldsymbol{y}} = \{\hat{y}_1, \ldots, \hat{y}_n\}$ the results of labeling algorithm, where $\hat{y}_i \in L$. The metrics with regard to label $l \in L$ are conventionally calculated as:

$$\text{TruePositive} : tp(l) = \{i | y_i = l \wedge \hat{y}_i = l\} \quad (2)$$

$$\text{FalsePositive} : fp(l) = \{i | y_i \neq l \wedge \hat{y}_i = l\} \quad (3)$$

$$\text{FalseNegative} : fn(l) = \{i | y_i = l \wedge \hat{y}_i \neq l\} \quad (4)$$

$$precision(l) = \frac{|tp|}{|tp| + |fp|} \quad (5)$$

$$recall(l) = \frac{|tp|}{|tp| + |fn|} \quad (6)$$

$$f(l) = \frac{2 \cdot precision(l) \cdot recall(l)}{precision(l) + recall(l)} \quad (7)$$

All these metrics are defined upon each logical label. Micro- and macro-average can be used to obtain performance on all labels.

C. Overall Evaluation

In practical document analysis tasks, usually both physical and logical analyses are performed on the documents. Naturally we want to know the performance of the whole process. In such situation, it is inadequate to simply evaluate in terms of physical segmentation or logical labeling. Evaluation giving consideration to both sides is more scientifically reasonable.

Each segment g_i in G or s_j in S is a set of primitive identifiers. By tracking identifiers of the their primitives, g_i and s_j can be decomposed as $g_i = TP(g_i) \cup FN(g_i)$ or $s_j = TP(s_j) \cup FP(s_j)$, where

- $TP(g_i) = \{g_i \cap s_j | s_j \in S\}$
- $FN(g_i) = \{g_i - \bigcup_{s_j \in S}(g_i \cap s_j)\}$
- $TP(s_j) = \{g_i \cap s_j | g_i \in G\}$
- $FP(s_j) = \{s_j - \bigcup_{g_i \in G}(s_j \cap g_i)\}$

The correspondences C between S and G are found based on their overlaps (see Figure 3 for an instance). With ground-truth in gray rectangle and analysis result in dashed rectangle, the correspondence types are depicted in Table I. For simplicity, only one-to-one (match, over detection, under detection), one-to-many (split, merger) and one-to-null (miss, false alarm) types are considered.

TABLE I. CORRESPONDENCE TYPES BETWEEN GROUND-TRUTH AND ANALYSIS RESULTS

type	*definition*	*illustration*
match	$TP(s_j) = \{s_j\}$ or $TP(g_i) = \{g_i\}$	
miss	$TP(g_i) = \varnothing$	
false alarm	$TP(s_j) = \varnothing$	
split	$\lvert TP(g_i)\rvert \geqslant 2$	
merger	$\lvert TP(s_j)\rvert \geqslant 2$	
over detection	$\lvert TP(s_j)\rvert = 1 \wedge FP(s_j) \neq \varnothing$	
under detection	$\lvert TP(g_i)\rvert = 1 \wedge FN(g_i) \neq \varnothing$	

For $c \in C$, let $v(c)$ denote the weight of true positive part, $u(c)$ the weight of c. For correspondence types of match, miss, under detection and split, there is only one segment g_c from G in c. Hence we define $v(c) = area(TP(g_c))$ and $u(c) = area(g_c)$, where $area(x)$ is defined as the sum of areas of primitives belonging to x. In the cases of false alarm, over detection and merger, we can find s_c from S in c and define $v(c)$ and $u(c)$ similarly.

Furthermore, the significance of errors in correspondence depends on application scenarios [6]. We introduce a function $p(c) \in [0, 1]$ to penalize the correspondence with regard to the logical labels. For example, a split in table can result in loss of comprehensibility in mobile reading, making it more intolerable than a merger between body paragraphs. So we assign 0.5 and 0.9 to $p(c)$ in these cases, respectively.

The overall performance involving both segmentation and labeling is normalized as:

$$score = \frac{\sum_{c\in C} p(c)v(c)}{\sum_{c\in C} u(c)} \quad (8)$$

D. Case Study

The proposed performance evaluation method has been proved effective by application cases including table detection and formula detection. The dataset is not limited to evaluation of specific application oriented tasks. It is adequate for evaluation of detection and recognition of multiple logical labels.

- Table detection. Table detection method via visual separators and geometric content layout information for PDF documents in [17] applied the performance system in section 4 to evaluate miss tables, fake tables, acceptable tables when compared with other representative table detection algorithms: Pdf2table and TableSeer. The performance metrics integrating penalty scores and content-based quantitative calculation. The comparison of table detection algorithms have shown the reliability of dataset and effectiveness of performance evaluation.

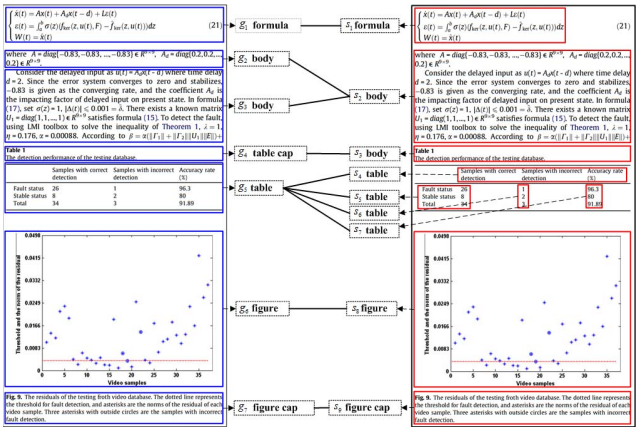

Fig. 3. Correspondences between ground-truth and analysis result of a sample page. On the left is the ground-truth with segments in blue. On the right is the analysis result with segments in red. The solid lines denote the correspondences between them. Logical labels are given in black boxes in the middle outside the pages. In this example, the correspondences are $(\{g_1\}, \{s_1\})$, $(\{g_2, g_3\}, \{s_2\})$, $(\{g_4\}, \{s_3\})$, $(\{g_5\}, \{s_4, s_5, s_6, s_7\})$, $(\{g_6\}, \{s_8\})$ and $(\{g_7\}, \{s_9\})$. There is an under detection between g_1 and s_1; s_2 is a merger of g_2 and g_3; s_3 is wrongly labeled compared with g_4; g_5 is split into 4 pieces in the analysis result; s_8 and s_9 are correct in both segmentation and labeling.

- Formula identification. Mathematical formula identification using ruled based, SVM-based and hybrid methods in [18] are evaluated. Eight types of error for each method are weighted, and an overall performance score is computed based on the significance of different types of identification results.
- Logical label recognition. In this case, the physical segmentation is fixed and the observations of segments are used as input for analysis algorithms. Using the fully labeled subset of our dataset, evaluations can be carried out as described in IV-B. The logical label set currently includes body text, title, list, figure, figure caption, table, table caption, equation, header, footer, page number, marginal note and footnote. This label set can be representative for the understanding of most book pages. As for the evaluation of detection of specific semantic class, the label set is simplified by treating labels that are not targets as "others".

V. Conclusion

With the aim of layout analysis for born-digital documents, this work proposes a new public dataset constructed from PDF documents. An XML based ground-truth data format is designed to uniformly describe the PDF primitives so as to avoid its inherent intricacy, simplifying the input of page layout analysis algorithms. The practical dataset selects diverse styles of born-digital documents representing layout varieties. All the data are produced with assistance of a self-developed GUI ground-truthing tool. The evaluation methods adjusted to this dataset can be customized to meet various assessing needs of specific application cases for physical segmentation and logical labeling. For future work, our emphasis will be put on enlarging the present dataset and evaluating more structure analysis tasks.

Acknowledgment

This work is supported by National Natural Science Foundation of China (No.61202232).

References

[1] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A realistic dataset for performance evaluation of document layout analysis," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 296–300.

[2] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Aletheia-an advanced document layout and text ground-truthing system for production environments," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 48–52.

[3] B. A. Yanikoglu and L. Vincent, "Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation," *Pattern Recognition*, vol. 31, no. 9, pp. 1191–1204, 1998.

[4] C. Ha Lee and T. Kanungo, "The architecture of trueviz: A groundtruth/metadata editing and visualizing toolkit," *Pattern recognition*, vol. 36, no. 3, pp. 811–825, 2003.

[5] E. Saund, J. Lin, and P. Sarkar, "Pixlabeler: User interface for pixel-level labeling of elements in document images," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 646–650.

[6] A. Antonacopoulos and D. Bridson, "Performance analysis framework for layout analysis methods," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 1258–1262.

[7] F. Shafait, D. Keysers, and T. M. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 941–954, 2008.

[8] W. Seo, M. Agrawal, and D. Doermann, "Performance evaluation tools for zone segmentation and classification (pets)," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 503–506.

[9] G. Paaß and I. Konya, "Machine learning for document structure recognition," in *Modeling, Learning, and Processing of Text Technological Data Structures*. Springer, 2012, pp. 221–247.

[10] K. Hadjar, M. Rigamonti, D. Lalanne, and R. Ingold, "Xed: a new tool for extracting hidden structures from electronic documents," in *Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on*. IEEE, 2004, pp. 212–224.

[11] J.-L. Bloechle, C. Pugin, and R. Ingold, "Dolores: An interactive and class-free approach for document logical restructuring." in *Document Analysis Systems*, 2008, pp. 644–652.

[12] H. Déjean and J.-L. Meunier, "A system for converting pdf documents into structured xml format," in *Document Analysis Systems VII*. Springer, 2006, pp. 129–140.

[13] S. Marinai, E. Marino, and G. Soda, "Conversion of pdf books in epub format," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 478–482.

[14] R. Qiu, Z. Tang, L. Gao, and Y. Yu, "A novel xml-based document format with printing quality for web publishing," *Imaging and Printing in a Web*, vol. 2, p. 2, 2010.

[15] I. Guyon, R. M. Haralick, J. J. Hull, and I. T. Phillips, "Data sets for ocr and document image understanding research," *Handbook of character recognition and document image analysis*, pp. 779–799, 1997.

[16] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[17] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual seperators and tabular structures," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 779–783.

[18] X. Lin, L. Gao, Z. Tang, X. Lin, and X. Hu, "Performance evaluation of mathematical formula identification," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 287–291.