

An open approach towards the benchmarking of table structure recognition systems

Asif Shahab, Faisal Shafait, Thomas Kieninger and Andreas Dengel
German Research Centre for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
{shahab,shafait,kieni,dengel}@dfki.uni-kl.de

ABSTRACT

Table spotting and structural analysis are just a small fraction of tasks relevant when speaking of table analysis. Today, quite a large number of different approaches facing these tasks have been described in literature or are available as part of commercial OCR systems that claim to deal with tables on the scanned documents and to treat them accordingly.

However, the problem of detecting tables is not yet solved at all. Different approaches have different strengths and weak points. Some fail in certain situations or layouts where others perform better. How shall one know, which approach or system is the best for his specific job? The answer to this question raises the demand for an objective comparison of different approaches which address the same task of spotting tables and recognizing their structure.

This paper describes our approach towards establishing a complete and publicly available, hence open environment for the benchmarking of table spotting and structural analysis. We provide free access to the ground truthing tool and evaluation mechanism described in this paper, describe the ideas behind and we also provide ground truth for the 547 documents of the UNLV and UW-3 datasets that contain tables.

In addition, we applied the quality measures to the results that were generated by the T-Recs system which we developed some years ago and which we started to further advance since a few months.

Categories and Subject Descriptors

I.7.5 [Document and Text Processing]: Document Capture—*Document Analysis*

Keywords

Table structure recognition, Benchmarking, Performance eval-

uation, Performance measures, Ground truth preparation, Image segmentation

1. INTRODUCTION

Tables in documents contain important data that is arranged in a matrix-like fashion for easier visual perception. Each row and column in the table has a specific meaning and table cells that belong to one row or column share certain attributes that are optionally indicated at the row or column heading. When a document containing a table is to be converted into an editable electronic representation by means of an optical character recognition (OCR) system, it is desired to extract the information in the table with the same level of detail as it was laid out in the original document. This is usually done in two steps. First, a table detection algorithm spots the tables in the document image marking their boundaries. Then, a table recognition module analyzes the row/column structure of the table and segments the table into individual cells while identifying the relation of each cell to rows and columns. This step is also called table structure extraction.

Several algorithms for table detection and table structure extraction have been reported in literature. A good overview of the state-of-the-art in table recognition is given in [25, 7]. The very early approaches focused on the detection of table delineation, significantly large column spacing or known column headings and have thus been applicable only to a fraction of table models [18]. An early model free approach was described by Kieninger et al. [15, 16]. Their approach performed a bottom up clustering of blocks and potential column candidates based on overlapping words segments in adjacent lines. These blocks are then designated as candidate table elements if they satisfy certain criterion. The candidate table regions are then decomposed into rows and columns based on the alignment and overlap of individual cells.

Wang et al. [24] uses a statistical learning approach for the table recognition problem. The table lines are identified from the set of text-lines using word spacing. Then vertically adjacent lines with large gaps and horizontally adjacent words are grouped together to make table entity candidates. Finally, a statistical based learning algorithm is used to refine the table candidates and reduce false alarms. Hu et al. [11] presented a system for table recognition from scanned documents or from their plain text representation. Their system assumes a single-column input page that can

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAS '10, June 9-11, 2010, Boston, MA, USA
Copyright 2010 ACM 978-1-60558-773-8/10/06 ...\$10.00

be easily segmented into individual text-lines (for instance by horizontal projection). The table detection problem is then posed as an optimization problem where start and end text-lines belonging to a table are identified by optimizing some quality function.

More recent work in table recognition is reported by Gatos et al. [8]. Their approach focused on locating tables that have both horizontal and vertical rulings and find their intersection points. Then, table reconstruction is achieved by drawing the corresponding horizontal and vertical lines that connect all line intersection pairs.

One major limitation of the literature on table recognition is a lack of common datasets in experimental evaluation as well as common benchmarking mechanisms and measures. Unlike for document analysis technologies that apply to non-tabular text (zoning, logical labeling, text categorization, content extraction etc.), there is little or no efforts towards a comparative evaluation of table recognizers. Hu et al. [13] used a dataset consisting of 26 Wall Street Journal article in text format and 25 email messages. Kieninger et al. [16] used a proprietary dataset of invoices of a purchasing department. Gatos et al [8] evaluated their algorithm on a test corpus consisting of scanned forms, news papers, magazines, scientific papers, tickets, bank cheques etc. Public in-availability of these datasets proved to be a major hindrance towards comparative evaluation of these algorithms.

However, publicly available datasets containing tabular elements such as UW-3[21] and UNLV[1] exists. But, the ground truths only provide table bounding boxes making them only suitable for table spotting tasks. Wang et al. [24] proposed a technique for automatic generation of ground truths for table documents using UW-3 dataset as the seed and some control parameters but the prepared dataset is no longer publicly available and hence not usable for evaluation tasks.

This situation is worsened by the use of custom-defined performance measures. Hu et al. [13] used *graph probing* for evaluating the results of table recognition. The output of their table analysis method is a graph model of the table. In parallel, a ground truth graph model is created manually. Three classes of questions are asked to both graphs and the percentage of agreement is measured. The first class aims at evaluating the quality of the segmentation task (how many columns does the table have?); the second evaluates table functional model (how many attributes contain "Open"?); and the third the structural model (mimicking database-type queries). Performance measures from the information retrieval domain such as *Recall*, *Precision* [19] and *combined F-measure* have also been used by several authors for evaluating the performance of their table recognition algorithm [17, 20]. Silva et al. [6] proposed *Completeness(C)* and *Purity(P)* as the new evaluation metrics for table evaluation task.

These two issues make direct comparison of table recognition algorithms particularly hard. In this paper we address this problem by

1. Proposing the application of established performance

measures used in the image segmentation community [10, 4, 14, 23] for evaluating the performance of table recognition algorithms.

2. Presenting a tool for preparing ground-truth of tables in a naive and intuitive way.
3. Generating ground-truth of table structure (cells, row, columns, table, row-spanning cells, column-spanning cells) for the publicly available UNLV [1, 22] and UW-3 [21] dataset.
4. Making the source code for the ground-truthing tool and the prepared ground-truth of the UNLV and UW-3 dataset freely available to the community.
5. Benchmarking a reference implementation of the T-Recs table recognition system [15] to provide a basis for comparative evaluation.

2. IMAGE BASED SEGMENTATION EVALUATION

An adequate and complete representation of table ground truth is very important for systematic evaluation of any table recognition algorithm. Table detection and decomposition of a table into rows, columns, and cells has strong analogies to hierarchical image segmentation. At the first level, tables are segmented from the rest of the document image. Then, the segmented tables are further decomposed into rows and columns that overlap with each other. These overlapping partitions are further decomposed into disjoint sets in the form of table cells. Therefore, if the hierarchical structure of tables can be represented in an image format, established methods [14, 10, 23] for evaluating image segmentation performance could be applied to table recognition as well.

Encoding table information directly in the image format allows easy visualization of the table ground truth data at different levels of abstraction such as cells, rows, columns as shown in Figure 2(a) etc. It also enables the evaluation algorithm to directly show errors made by the analysis algorithm on the image for visual comparison. Since PNG (Portable Network Graphics) is used as the image format, no proprietary tools are needed to interpret the image information in contrast to the early efforts such as DAFS [9] which requires proprietary software for visualization. All information will be encoded in the foreground(black) pixels belonging to a table which makes the ground truth free from errors induced by OCR software (word boxes extraction, text extraction) which are commonly used to assist in ground truth generation [17]. It will also be possible to prepare ground truths for skewed documents and warped camera captured documents which would otherwise be impossible with the traditional bounding box approach of encoding ground truth information.

In this section, we first present the representation of tables in a color-coded image format and then describe the application of general image segmentation evaluation measure to table recognition.

2.1 Encoding Table Structure Information in Images

The key problem when representing table structure information in an image format is multiple associations of each pixel. For example, a particular foreground pixel in a table region belongs to a particular cell, row, column, and table. Hence we need a segmentation representation that can encode this multiple levels of hierarchical information. This problem is similar to the document image segmentation problem in general, where each foreground pixel belongs to a particular word, text-line, and text-column. In that domain, the concept of encoding these multiple levels of information into a single image has been introduced in [23]. The main idea is to use different color channels to represent different levels of information. This idea works in the domain of document analysis since we are typically dealing with binary images.

Embedding the segmentation information into the color channels of a binary document image has several advantages. First, the representation is not limited to any particular types of region shapes (rectangular, isothetic polygons, ...). Second, the segmentation information can be easily exchanged and distributed using standard loss-less compressed image formats like PNG. Hence one does not need any specialized software to view the segmentation result. Last but not least, the segmentation information is pixel-accurate. Due to these advantages, the OCRopus [5] open source OCR system uses a color-based representation as a basis for exchanging segmentation information between different components.

In this paper, we propose a color-coding scheme for representing table segmentation. The main idea is to use the red color channel for representing the index of the table, i.e. the table number in the current document image. The green channel represents the row number within the table, and the blue channel represents the column number. In this way, each cell in the table gets a unique color since it has a unique row, column, and table number. If we want to access the table level information, we can easily obtain that by masking the green and blue channels from the image. Similarly, row level information can be accessed by getting the brightness-values in the green channel and the column level by accessing the blue channel.

This assignment of colors leaves an important question unanswered. How to encode cells that span multiple columns or rows? We propose to use 16-bit color channels for this purpose. The higher order byte represents the start index, and the lower order byte represents the end index. For instance, in order to represent a cell that spans column numbers three to five, the higher order byte of the blue channel will have the value three, and the lower order byte will have the value five. Similarly, for representing multi-row cells, this information goes into the green color channel. All cells that lie within a single row and column have the same start and end index in both green and blue channels.

2.2 Color-Based Segmentation Evaluation

Given two segmentations in color-coded representation, the goal of color-based segmentation evaluation is to identify the differences between the two segmentation, specifically highlighting different classes of segmentation errors. The

color assigned to a particular segment is not important. The only requirement is that each segment gets a unique color.

Consider a ground-truth segmentation G and a segmentation S performed by an algorithm. The correspondence between the segmentations G and S can be obtained by examining color values of corresponding pixels (pixels at the same location) in each image. For this purpose, the colors in each image are enumerated using a Hash table. The key of an entry in the Hash table is the index of a particular color in the image, and the value of that entry is the RGB color value associated with that index. For table images, we reserve the key 0 for the background color (0xffffffff) and the key 1 for the foreground pixels that do not belong to any table. Hence if there are three tables in an image, and we are only interested in table-level segmentation, each table will have a unique color that is uniform throughout the table region. In that case, the table regions will be assigned indices (Hash keys) 2, 3, 4 respectively. In this manner, segments in both images G and S are enumerated.

After enumerating colors in both images, a correspondence matrix is built by counting the number of pixels locations that have the same color index in both segmentations. Let n and m be the number of colors, i.e. the number of entries in the hash table, in the ground-truth segmentation G and the obtained segmentation S respectively. Then we initialize an $n \times m$ matrix with all zero entries. The entry (i, j) of the matrix is incremented by one if at a particular pixel location, the color index of the ground-truth pixel is i and the index of the obtained segmented image pixel is j . Hence, in a single pass through the images G and S simultaneously, we obtain a correspondence matrix between the ground-truth image and the obtained segmentation.

It is interesting to note here that the sum of a particular row gives us the total number of foreground pixels in the corresponding segment of the ground-truth segmentation G . The value in each cell of that row tells us how many pixels in that segment overlap with a certain segment in the obtained segmentation S . For a perfect segmentation, each segment in G should correspond to exactly one segment in S . Hence, in case of a perfect recognition result for each row and column there should be only one non-zero entry. However, in practice we might still have some pixels mismatching between the G and S . Therefore we define a threshold T for the ratio of overlapping pixels to define a significant overlap. An overlap of a ground-truth segment G_i with a segment S_j is called significant if the area of the overlapping region $|G_i \cap S_j|$ divided by the area of the ground-truth region $|G_i|$ is larger than T . Similarly, an overlap of an obtained segment S_j with a ground-truth segment G_i is called significant if $|G_i \cap S_j| / |S_j| > T$. Based on the overlap between segments in G and S , the following performance measures are defined:

- **Correct Detections:** These are the number of segments in G that have a one-to-one correspondence with a detected segment, i.e. they have a large overlap ($|G_i \cap S_j| / |G_i| > 1 - T$) with one of the segments S_j and the segment S_j does not have a significant overlap with any other ground-truth segment ($|G_k \cap S_j| / |S_j| < T; \forall k \neq i$).

- **Partial Detections:** These are the number of segments in G that have a one-to-one correspondence with a detected segment, however the amount of overlap is not large enough ($T < |G_i \cap S_j|/|G_i| < 1 - T$) to be classified as a correct detection.
- **Over-Segmentation:** These are the number of ground truth segments that have a major overlap ($T < |G_i \cap S_j|/|G_i| < 1 - T$) with more than one detected segments. This indicates that different parts of the ground truth segment were detected as separate segments.
- **Under-Segmentations:** These are the number of detected segments that have a major overlap ($T < |G_i \cap S_j|/|S_j| < 1 - T$) with more than one ground truth segment. This indicates that more than one segment (possibly adjacent) were merged by the detection algorithm and were reported as a single segment.
- **Missed Segments:** These are the number of segments in G that do not have a major overlap with any of the detected segments ($|G_i \cap S_j|/|G_i| < T; \forall j$). These segments are regarded as missed by the detection algorithm.
- **False Positive Detections:** These are the number of detected segments that do not have a major overlap with any of the ground-truth segments ($|G_i \cap S_j|/|S_j| < T; \forall i$). These segments are regarded as false positive detections since the system mistook some non-interesting region as a segment.

3. TABLE GROUND TRUTH GENERATION

We have developed an easy to use ground truth preparation tool, called "T-Truth" which allows quick and accurate labeling of tables in a document. A table consists of a row-column structure with cells containing data at the intersection of these rows and columns. However, some cells can span rows, columns or both. Our tool allows capturing of all this information in simplified user interactions using mouse alone.

T-Truth tool as shown in Figure 1, consist of an image canvas where an operator can load a document image. The image canvas supports variety of image formats (png, tif, pgm, etc). It also allows zooming in and out of the image. Once the image file is loaded the user can start with the following labeling tasks:

Mark Table Zone: The user draws a bounding box around a table. This is performed by clicking on the starting position (top-left) of the table, dragging mouse and dropping at the end position (bottom-right). The tool paints a rectangle which follows the mouse, thus providing visual feedback to the user. The user can mark multiple tables on a page using the same interaction. The rectangular region drawn with the mouse will automatically be cropped to cut off regions without black pixels. The table co-ordinates are translated and stored in terms of image co-ordinates internally by the canvas. The table region is shown by its bounding box in Figure 1(a).

Mark Row/Column Separators: Once the table zones are marked in the document image, the user can click inside

the bounding box of one of the table zones which highlights the table indicating that the current table is selected for marking rows or columns. The user can then use left mouse click to mark row separators and right mouse click to mark column separators. Each of these clicks draws a horizontal(blue) and a vertical(red) line marking row and column separations inside the table zone as shown in Figure 1(a). This labeling task can be performed easily by zooming in to the image and then clicking in the area which effectively separates rows and columns. The user doesn't have to worry about row/column spanning cells at the moment which will be dealt later. Further, the operator need not be very precise to draw these separators even if they cut few pixels of the neighboring cells because this problem will be handled by the evaluation framework . The tool internally stores the image points for these row and column separators. It also builds an internal table structure which stores the bounding box for each of the cells.

The tool also provides undo/redo operations to support the above labeling task. After the above labeling tasks the user sees a table grid which marks cells inside a table. However, some of the cells (so called *spanning cells*) are erroneously divided into multiple cells by the row-column separators. The user can then proceed with removing these errors with the following labeling procedure:

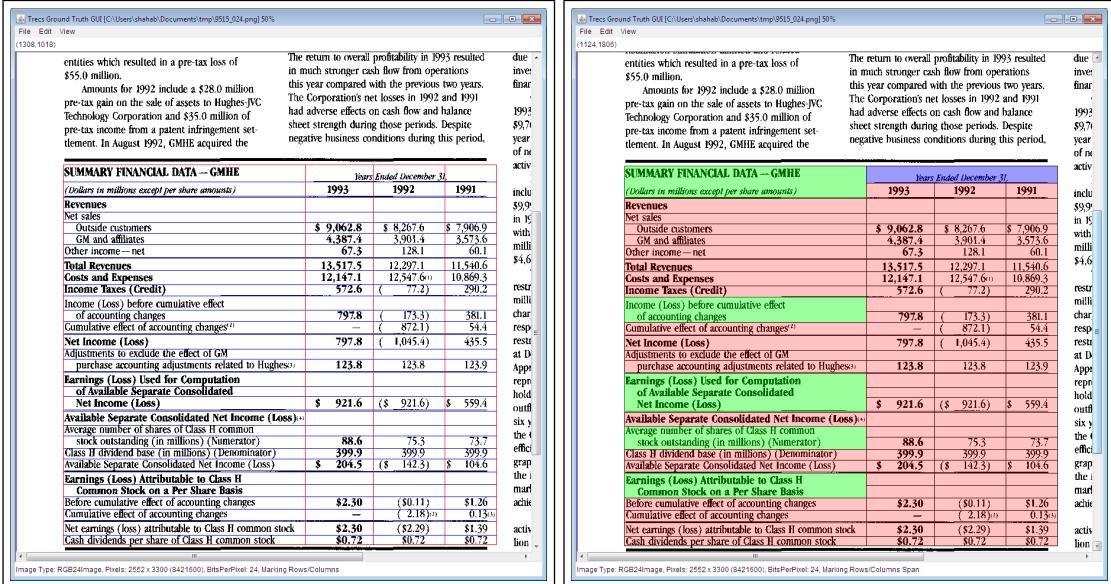
Mark Row/Column Span: The user first clicks inside one of the tables thus selecting it to define row/column span. For row spanning cells, the user drags the mouse down with the left button pressed from the start cell to the end cell. After this drag and drop the cells are merged together into one cell and are painted in distinguishing color and the row-column separators are updated to give direct visual feedback to the user as shown in Figure 1(b). Similarly, for column spanning cells the user drags the mouse to the right from the start cell to the end cell by pressing the right button. This merges the cells together defining a column-spanning cell as shown in Figure 1(b). For, cells spanning rows and columns, the user can define either of the row or column span first and then define the other effectively combining all the cells into a row-column-spanning cell. The user can define these spanning cells anywhere in the table.

After completion of above labeling tasks, the user can save the ground truth for the given image. The tool stores ground truth in the XML format following a predefined naming convention for a unique association with the original document image.

The T-Truth tool also provides a preview-edit mode for ground truth validation. In this mode, the tool can be initialized with the folders containing document images and their ground truths and the operator can browse through the dataset using next-previous menu items or shortcuts. The table ground truths are read from the file automatically and shown on the image for manual validation. In case of errors the operator can quickly edit the ground truth using the labeling operations defined above.

3.1 Preparation of ground truth images

In order to feed our color encoded segmentation approach, we need to transform the XML-encoded ground truth data



(a) Mark Table and Row/Column

(b) Mark Row/Column Span

Figure 1: T-Truth Tool

acquired with T-Truth into according image representations. We therefore use a 16-bit RGB encoding to re-color the foreground (black) pixels within each cell bounding box based on its table-, row- and column indices as described in section 2.1. However, if in the document image there are physical delineation for rows and/or columns then they will be erroneously colored with multiple colors for each of the cells it intersects. We take care of this problem by first extracting the word bounding boxes for the given document image using an OCR software[2] and then re-coloring only those pixels inside cells which comprise words. This step also ensures that noise elements in the document image are not colored and thus do not interfere with the evaluation framework. Alternatively, a line detection algorithm can be used to identify horizontal and vertical lines and avoid the recoloring of their pixels.

4. EXPERIMENTS AND RESULTS

For our experimental evaluation, we used the following publicly available datasets:

1. **UNLV dataset[1]:** The dataset contain 2889 pages of scanned document images from variety of sources (Magazines, News papers, Business Letter, Annual Report etc). The scanned images are provided at 200 and 300 DPI resolution in bitonal, grey and fax format. The ground truths with manually marked zones with zone types are provided in text format. Closer examination of the dataset reveal that there are no marked table zones in the fax images. The grey images are all present in bitonal images. Therefore we concentrated on bitonal documents with resolution of 300 dpi for the preparation of ground truths. We first separated those images for which table zones have been marked in the ground truth. There are around 427 such images with at least one marked table zone.

2. **UW-3 dataset[21]:** The database consists of 1600 skew corrected English document images with manually edited ground-truth of entity bounding boxes. These bounding boxes enclose page frame, text and non-text zones, textlines, and words. The type of each zone (text, math, table, half-tone, ...) is also marked. There are around 120 document images containing at least one marked table zone. We chose these document images for our evaluation.

We trained a user to operate the T-Truth tool and asked him to prepare the ground truth for the target images from above dataset. The ground truth for each image is stored in an XML file. The ground truths were manually validated by another expert using the preview edit mode of the T-Truth tool and improper ground truths were corrected. These iterations were made several times to ensure the accuracy of the ground truth. We observed that most of the mistakes are made in labelling column spanning cells where column boundaries coincide with the word boundaries. Problems can also occur where alternate interpretations of a table structure exist as described by Nagy et al[12] and domain knowledge is required to accurately label the table structure.

We then prepared the ground truth images as described in section 3.1. These images are stored in PNG format and are used directly by our evaluation framework for comparison. The "T-Truth tool" and the ground truths for UW-3 and UNLV document pages containing tables can be found at [3].

4.1 T-Recs Evaluation Results

T-Recs is a table spotting and structure recognition system that was developed and first published more than 10 years ago. Details on principles of T-Recs can be found in [15, 16]. It has since then undergone one major and several mi-

Table 1: Results of evaluating a T-Recs system on 120 binary 300-dpi scanned UW-3 dataset pages containing table zones.

	Table(%)	Row(%)	Column(%)	Cell(%)	Row Span(%)	Col Span(%)
Correct Detections	10.71	41.86	32.95	62.92	0	32.35
Partial Detections	24.82	20.84	20.37	0.35	2.38	19.61
Over-Segmented	4.38	9.15	8.85	6.06	0	11.76
Under-Segmented	12.41	7.68	6.05	5.29	0	3.92
Missed	10.219	13.82	13.50	11.95	95.24	28.43
False Positive Detections(scaled) ¹	0.065	6.92	2.80	28.14	0.13	0.77

Table 2: Results of evaluating a T-Recs system on 427 binary 300-dpi scanned UNLV dataset pages containing table zones.

	Table(%)	Row(%)	Column(%)	Cell(%)	Row Span(%)	Col Span(%)
Correct Detections	34.86	54.98	40.51	74.28	0.84	15.24
Partial Detections	20	12.45	18.57	1.43	2.52	23.17
Over-Segmented	3.30	6.27	13.50	8.55	0	7.93
Under-Segmented	8.62	7.70	5.11	3.29	0	4.88
Missed	13.76	10.69	13.50	7.76	96.63	42.37
False Positive Detections(scaled) ¹	0.04	0.12	0.88	8.84	0.05	0.43

nor evolution steps. Its relevance to practice has recently been boosted by industry partners who became aware of the potentials of this model free approach in contrast to their own custom made approaches that are tailored to a well defined narrow table model but fail if just small variation to that model appear. We have thus decided to reactivate research in that area and to continue developing T-Recs.

Although being a model free approach, T-Recs is controlled by a set of numerical parameters, typically thresholds that are considered during the various processing steps to control the treatment of layout elements. Slight variations to the parameters might result in worse or better results depending on the layout that is given.

Hence, numerous applications of T-Recs in different document domains (e.g. bank account statements, business documents like invoices, multicolumn research papers) and specific layouts always required a balancing of system parameters to yield best results. That was so far always made by hand.

With the described benchmarking at hand we are now planning to extend T-Recs to a self adapting system that can tune itself towards a given set of training-documents for which ground truth data is defined.

Whereas we primarily aimed at an open and universally applicable benchmarking environment for the evaluation of table spotting and structural recognition systems, we also targeted the improvement of the T-Recs system.

Consequently, we have chosen T-Recs to produce sample analysis results for the UNLV and UW-3 datasets. T-Recs system provides recognition results in form of table bounding box and its cells associated to particular rows and/or

columns. It is also able to determine row/column spanning cells. We wrote a small export routine to export the results of T-Recs in the same XML format we used for generating ground truths. The T-Recs analysis was then performed on the target images from these two datasets and the results are stored in the matching XML files. These results are then transformed to the document images for direct comparison with the ground truth images generated earlier. The comparison results are shown in the Table 1 and 2.

Table 3: Results of evaluating a T-Recs system on 1-column, 2-column and 3-column 300-dpi scanned UNLV dataset pages containing table zones.

	Table Detection		
	1-Col(%)	2-Col(%)	3-Col(%)
Correct	50.40	5.67	5.71
Partial	22.44	10.68	34.28
Over-Segmented	5.15	1.42	0
Under-Segmented	4.61	19.15	11.42
Missed	9.75	17.02	5.71
False Positive ¹	0.06	0.03	0.04

These results were obtained by evaluating the base T-Recs system without any optimization of parameters for the given datasets. Since T-Recs is a bottom up approach, the evaluation results clearly shows the strength of the technique at identifying cells(74.28% for the UNLV and 62.92% for the UW-3 correct detections) belonging to a particular table.

¹False Positives for tables are scaled to per document page, for rows, columns, and cells it is scaled to per detected table region and for row and column spanning cells it is scaled to per detected rows and columns

However, there is room for improvements and optimization of parameters for row-, column- and table detection. The low percentage of table detections for UNLV and UW-3 datasets is also due to the fact that T-Recs was designed for single column layout documents whereas these dataset contains several document with multiple columns. This can be seen from the evaluation results in Table 3 which are obtained by performing T-Recs analysis on 1-column, 2-column and 3-column documents from the UNLV dataset separately.

The strength of this evaluation scheme is that it allows evaluation at each level of abstraction (Table, Rows, Columns, Row-spans etc) independently, thus is suited to evaluating variety of approaches ranging from top-down to bottom-up. Furthermore, the evaluation scheme can directly show different error classes on the document image as shown in Figure 2(b). This gives these evaluation measures a meaningful visualization which highlights the shortcomings of the given table recognition system and thus opens the possibility for improvements. Additionally, the breadth of measures provided for comparison allows developers to focus on tuning the set of measures specific to their domain of application. e.g. Table/Figure segmentation application can focus on optimizing the correctness and false positive measure for table detection where as an indexing application might be interested in good percentages of correct cell detections.

5. CONCLUSION

This paper presented an open framework for comparative evaluation and benchmarking of table recognition systems. We presented an image-based representation of a given table structure and applied established image segmentation evaluation techniques for the benchmarking of table recognition algorithms. We developed an open-source tool for ground-truthing table regions and prepared ground-truth for tables in the UW3 and UNLV datasets. The prepared ground-truth is also made publicly available. The T-Recs table recognition system was chosen as a reference to be evaluated on the UW3 and UNLV datasets. We hope that this framework will provide solid foundations for comparing different table recognition algorithms on common grounds.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project PaREn (01 IW 07001).

6. REFERENCES

- [1] <http://www.isri.unlv.edu/ISRI/OCRtk>.
- [2] <http://www.nuance.com/imaging/products/omnipage.asp>.
- [3] <http://www.dfki.uni-kl.de/~shahab/t-truth>.
- [4] T. M. Breuel. Representations and metrics for off-line handwriting segmentation. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 428–433, Ontario, Canada, Aug. 2002.
- [5] T. M. Breuel. The OCropus open source OCR system. In *Proc. SPIE Document Recognition and Retrieval XV*, pages 0F1–0F15, San Jose, CA, USA, Jan. 2008.
- [6] A. Costa e Silva. New metrics for evaluating performance in document analysis tasks - application to the table case. In *Proc. ICDAR'07*, pages 481–485, Washington, DC, USA, 2007. IEEE Computer Society.
- [7] D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy. Table-processing paradigms: a research survey. *IJDAR'06*, 8(2):66–86, 2006.
- [8] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis. Automatic table detection in document images. In *Proc. Int. Conf. on Advances in Pattern Recognition*, pages 612–621, Path, UK, Aug. 2005.
- [9] I. Guyon, R. M. Haralick, J. J. Hull, and I. T. Phillips. Data sets for OCR and document image understanding research. In H. Bunke and P. Wang, editors, *Handbook of character recognition and document image analysis*, pages 779–799. World Scientific, Singapore, 1997.
- [10] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.
- [11] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *Proc. SPIE Document Recognition and Retrieval VII*, pages 291–302, San Jose, CA, USA, Jan. 2000.
- [12] J. Hu, R. Kashi, D. Lopresti, G. Wilfong, and G. Nagy. Why table ground-truthing is hard. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 129–133, Seattle, WA, USA, Sep. 2001.
- [13] J. Hu, R. S. Kashi, D. Lopresti, and G. Wilfong. Evaluating the performance of table processing algorithms. *IJDAR'02*, 4(3):140–153, 2002.
- [14] X. Jiang, C. Marti, C. Irniger, and H. Bunke. Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing*, 2006(1):Article ID 35909, 10 pages, 2006.
- [15] T. Kieninger and A. Dengel. A paper-to-HTML table converting system. In *Proc. DAS'98*, pages 356–365, Nagano, Japan, Nov. 1998.
- [16] T. Kieninger and A. Dengel. Applying the T-RECS table recognition system to the business letter domain. In *Proc. ICDAR'01*, pages 518–522, Seattle, WA, USA, Sep. 2001.
- [17] T. Kieninger and A. Dengel. An approach towards benchmarking of table structure recognition results. In *Proc. 8th ICDAR'05*, pages 1232–1236, Seoul, Korea, Aug. 2005.
- [18] B. Klein, S. Gökkus, T. Kieninger, and A. Dengel. Three approaches to "industrial" table spotting. In *ICDAR'01*, volume 0, CA, USA, 2001.
- [19] D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In *Proc. of the 18th annual Intl. ACM SIGIR conf. on Research and development in information retrieval*, pages 246–254, New York, NY, USA, 1995. ACM.
- [20] S. Mandal, S. Chowdhury, A. Das, and B. Chanda. A simple and effective table detection system from document images. *IJDAR'06*, 8(2-3):172–182, 2006.
- [21] I. T. Phillips. User's reference manual for the UW English/Technical Document Image Database III. Technical report, Seattle University, Washington, 1996.
- [22] S. V. Rice, F. R. Jenkins, and T. A. Nartker. The fourth annual test of OCR accuracy. Technical report, Information Science Research Institute, University of Nevada, Las Vegas, 1995.
- [23] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(6):941–954, 2008.
- [24] Y. Wang, R. Haralick, and I. T. Phillips. Automatic table ground truth generation and a background analysis based table structure extraction method. In *Proc. ICDAR'01*, pages 528–532, Seattle, WA, USA, Sep. 2001.
- [25] R. Zanibbi, D. Blostein, and J. R. Cordy. A survey of table recognition: Models, observations, transformations, and inferences. *IJDAR'04*, 7(1):1–16, 2004.

GMHE's business mix in 1993, 1992, and 1991 was as follows:

GMHE BUSINESS MIX			
Automotive	1993	1992	1991
Telecommunications & Space	33%	32%	32%
Defense	16	16	16
Commercial	45	45	46
Diverse mix indicates GMHE growth potential.	6	7	6

GMHE also provides direct support to GM by supplying components and technologies for GM vehicles.

In July 1993, GMHE sold its 30% ownership interest in the Japan Communications Satellite Corporation, resulting in a \$6.0 million pre-tax gain. In December 1993, GMHE sold Hughes Rediffusion Satellite Limited and related entities which resulted in a pre-tax loss of \$5.0 million.

损益表 for 1992 include a \$26.0 million pre-tax income on the sale of assets to Hughes JVC Technology Corporation and \$35.0 million of pre-tax income from a patent infringement settlement. In August 1992, GMHE acquired the

SUMMARY FINANCIAL DATA — GMHE
(in millions except per share amounts)

	Years Ended December 31:		
	1993	1992	1991
Revenues			
Net sales	\$ 9,062.8	\$ 8,207.6	\$ 7,006.6
GMH affiliates	4,387.4	3,901.4	3,575.6
Other income — net	67.3	128.1	60.1
Total Revenues	13,517.5	12,297.1	11,540.6
Costs and Expenses			
Income Tax (Credit)	12,147.1	12,547.6	10,869.3
Income Tax (Credit) (continued)	(77.3)	(27.3)	(29.0)
Income (Loss) before cumulative effect of accounting changes	797.8	(175.3)	381.5
Cumulative effect of accounting changes ⁽¹⁾	—	(872.1)	54.4
Net Income (Loss)	797.8	(1,045.4)	435.5
Adjustments to exclude the effect of GM purchase accounting adjustments related to Hughes ⁽²⁾	125.8	125.8	125.8
Net Income (Loss)	\$ 921.6	(\$ 921.6)	\$ 559.5
Net Income (Loss) of Non-Controlling Interest in Separate Consolidated Entity			
Available Separate Consolidated Net Income (Loss)			
Average Number of Class II Common Stock Outstanding (in millions) (Denominator)	88.6	75.3	73.7
Class II Dividend Rate (in millions) (Denominator)	399.9	399.9	399.9
Available Separate Consolidated Net Income (Loss)	\$ 204.5	(342.5)	\$ 106.6
Earnings (Loss) Attributable to GMHE II Common Stock on a Per Share Basis			
Before cumulative effect of accounting change	\$2.30	(0.11)	\$1.38
Cumulative effect of accounting changes ⁽¹⁾	—	(2.89)	0.13
Net earnings (loss) attributable to Class II common stock	\$2.30	(3.29)	\$1.39
Cash dividends per share of Class II common stock	\$0.72	\$0.72	\$0.72

(1) include one-time \$1,215 million (after-tax) FWD 4 million or EUR 8.7 million of Class II common stock restructuring charges primarily allocated to GMHE's facilities in Germany and related employees at Hughes.
(2) Effective January 1, 1992, GMHE adopted SFAS No. 106, "Employer's Accounting for Postretirement Benefits Other Than Pensions." Hughes changed its revenue recognition policy for certain commercial businesses from the accrual method to the cash method. This change increased GMHE's revenue and decreased operating profit. Hughes Electronics' change to include in inventory general purchase parts were generally charged directly to expense and were all SFAS No. 106, "Employer's Accounting for Postretirement Benefits Other Than Pensions."
(3) includes an adjustment for the change in GMHE's accounting for GMHE's acquisition of Hughes.
(4) Available Separate Consolidated Net Income (Loss) is determined quarterly.

Tables

(a) Ground Truth Image with cells painted at different levels of abstractions..

Table ground truth (Model Image)

GMHE BUSINESS MIX			
	1993	1992	1991
Automotive	33%	32%	32%
Telecommunications & Space	16	16	16
Defense	45	45	46
Commercial	6	7	6

Diverse mix indicates GMHE growth potential.

GMHE BUSINESS MIX			
	1993	1992	1991
Automotive	33%	32%	32%
Telecommunications & Space	16	16	16
Defense	45	45	46
Commercial	6	7	6

Cells

GMHE BUSINESS MIX			
	1993	1992	1991
Automotive	33%	32%	32%
Telecommunications & Space	16	16	16
Defense	45	45	46
Commercial	6	7	6

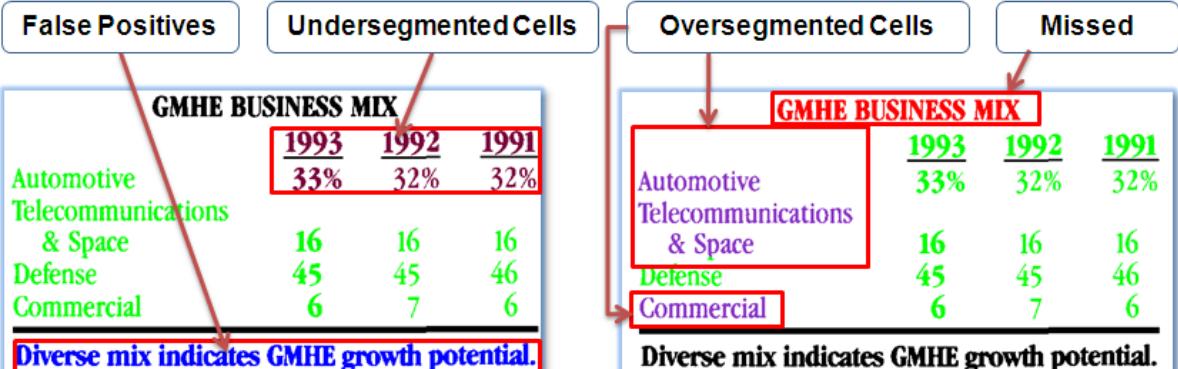
Columns

Table reognition result(Result Image)

GMHE BUSINESS MIX			
	1993	1992	1991
Automotive	33%	32%	32%
Telecommunications & Space	16	16	16
Defense	45	45	46
Commercial	6	7	6

Diverse mix indicates GMHE growth potential.

Segmenation Errors shown on the Image



(b) Segmentation Errors shown on images generated by the Evaluation Framework

Figure 2: Sample Visualization of the Ground Truth data (a), and comparison of analysis result with Ground Truth (b)