



Robust Detection of Tables in Documents Using Scores from Table Cell Cores

Md. Ajjij¹ · Sanjoy Pratihar² · Diptendu Sinha Roy¹ · Thomas Hanne³

Received: 17 August 2021 / Accepted: 18 January 2022 / Published online: 12 February 2022
© The Author(s) 2022

Abstract

Table detection is an essential step in many document analysis systems. Tabular data are a pivotal form of information representation that can organize data in a conventional structure for comfortable and quick information retrieval and comparison. Detection of table structures in PDF files or images is a challenging task because of the variability of table layouts, and sometimes the tabular structures' similarities with non-tabular elements like charts, plots, etc. In this work, we have presented a table detection method using a geometric analysis of the table cell cores that represents the table cell texts. The proposed method works by analyzing the text gap information, and hence it can detect the table cell cores, irrespective of the presence of the table boundary lines and cell-separating rule-lines. Experimentations have been done on various document images of complex structures from well-known datasets. The detection accuracies obtained by us corroborate the usefulness of the proposed method.

Keywords Table recognition · Document analysis · Information retrieval

Introduction

One of the significant challenges of document layout analysis is table understanding in the document image. Tables are broadly present in a prodigious variety of documents such as official documents, bills, scientific articles, reports, or archival documents among others; and, hence, techniques for table analysis are instrumental to automatically extract important information kept in a tabular form from numerous sources [1]. Tables facilitate readers to easily compare, analyze and understand facts present in documents [2]. So, table detection is an essential task as the accurate table detection will enhance document analysis addressing important information extraction. Due to the diversity of table styles, table detection and extraction is a popular and challenging

task. There is no such general algorithm that can detect the presence of the tables in the document irrespective of the styles of the tables.

A conventional optical character recognition (OCR) system consists of three significant steps, i.e., layout analysis, character recognition and text string generation using a language modeling tool [3]. Since layout analysis is the first step in such a process, all subsequent stages rely on layout analysis to work correctly. One of the significant difficulties faced by layout analysis is detecting table regions. Tables are made of horizontal and vertical lines or by introducing uniform spaces to differentiate the cells within it. The variety of styles makes it difficult to provide a generic algorithm for table detection [2]. Our main contribution in this paper is writing a generalized algorithm for table detection followed by information extraction. The rest of the paper is organized as follows: we discuss related work in “[Related Works and Motivation](#)”. In “[Our Proposed Method](#)”, we present our proposed approach and further details on checking components are in “[Score Computation](#)”. Results of the method are shown in “[Results and Discussions](#)” and we conclude with “[Conclusions](#)”.

✉ Thomas Hanne
thomas.hanne@fhnw.ch

¹ Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong, India

² Department of Computer Science and Engineering, Indian Institute of Information Technology, Kalyani, India

³ FHNW University of Applied Sciences and Arts Northwestern Switzerland School of Business, Riggbachstrasse 16, 4600 Olten, Switzerland

Related Works and Motivation

Several methods have been proposed for table detection and are available in the literature. There are approaches which use purely geometric features extracted from the ruled lines, pixel distributions, white gaps and finally those features help detecting tables using machine learning. Our approach is based on a geometric analysis of the table cell centers. In general, these methods can be divided into two main categories, text analysis based and ruling line based.

Anh et al. [4] proposed a hybrid approach for the detection of table structures, irrespective of the style, a ruling line table or a non-ruling line table. Experimental results are shown by them for the ICDAR-2013 table competition dataset. Jahan et al. [5] proposed a method where local thresholds for word gaps and the line-heights have been used to locate and extract all categories of tables. The system shows a 75% overall detection rate which was not very promising. Bansal et al. [6] presented a learning-based framework which identifies tables from scanned document images. They proposed a scheme for analyzing and labeling different document elements, their contexts, and finally to define and understand the table boundaries from the context informations. Kasar et al. [7] presented a method which works by identifying the column and row line separators. The horizontal and vertical-aligned lines are extracted first using run-length thresholds and then those aligned lines are used for feature generation and subsequent classification in to tables and non-tables.

Many recent works are available for the proposed problem, which use neural networks or deep learning models. For example, Forczmański et al. [8] presented an object detection approach using a Convolutional Neural Network. They focused on automatic segmentation of elements from documents. The elements considered by them were stamps, logos, text blocks, tables, and signatures. The authors have collected various documents from internet and created their own dataset. The method presented by them works in two stages. In the first stage, a rough classification of the detected regions of interest is done, and then in the second stage, verification of found elements are done. They experimented on public datasets and obtained a table detection accuracy of 97.79%. In another recent work, the authors have used a Convolutional Neural Network with 28 layers for the detection of tables [9]. In a study by Shah Rukh Qasim et al. [10], a graph model is used for the structural analysis problem of documents. Table recognition from scanned document images is the main point of interest. The proposed architecture is a combination of convolutional neural network (CNN) and graph network, where CNN helps in visual feature extraction and the graph network deals with the problem structure. For experimentation, the

authors used UNLV [11] and ICDAR 2013 [12] datasets. The absence of the ruling lines, it was reported as 94.7%. Kavasidis et al. [13] solved the automated table or chart detection task by a combination of deep convolutional neural networks, graphical models and saliency concepts are presented in this article. Localization of tables and charts in documents was carried out using the saliency-based fully-convolutional neural network followed by a fully-connected conditional random field (CRF). Performance was tested on the ICDAR 2013 dataset and they observed a precision of 97.5%, and a recall of 98.1%. Arif et al. [14] suggested a novel data-driven approach for table detection from document images using foreground and background features. The observations the authors were that the tables normally contain more numeric data, they focused on differentiating the numerical and other text data. They obtained a precision of 86.33% and a recall of 93.21% when applied on the UNLV dataset. Schreiber et al. [15] presented a system for table detection using deep learning which works by analyzing the cell positions after detection of rows and columns present in the tables. The accuracy for table detection and structure recognition by their method was 91.44% when applied on ICDAR 2013 dataset. Li et al. [16] proposed a convolutional neural networks based method which applies some loose heuristic rules to extract meta-information from the PDF documents and used those meta-informations for table detection purposes. The crucial limitation of the method is that it only works for PDF documents.

Our Contributions

- From the current works available, we can see that the approaches are based on the table lines' geometry or gap between contents only. The method in [17] is made only for ruled tables. A document may have ruled, non-ruled tables, and partially ruled tables, and our proposed method aims to work for all of them. In our proposed method, we are not relying on any horizontal or vertical lines for detecting tables.
- A hybrid method to detect both ruled and non-ruled tables has been proposed in [4]. However, this method is very complicated and time consuming. It categorizes the tables as ruled and non-ruled and processes them differently. We do not classify tables into ruled and non-ruled tables, and neither do we classify text and non-text elements in the documents. Hence our method is more simple and yet useful.
- The method in [18] relies on graphic lines, which sometimes leads to false detections of tables when there is a line in a paragraph with sparsely populated text. Our proposed method of score computation for the recognition solves this problem to some extent.

- Mandal et al. [19] have proposed a method based on the fact that the gaps between the fields have to be larger than the gap between the words in text lines. Though, this may not always be true as tables can be densely populated. Our proposed method also uses gaps between elements in a page, but relies on a more accurate assumption that tables can be recognized seeing the well-structured point set representing the table cell cores. Cores are understood as the text blocks' centers in the tabular cells and represented as a set of points.

Our Proposed Method

Our objective is to find tables present in a document image using simple methods. We start with a gray-scale image of the page. It is assumed that the image is already skew corrected. The image is then binarized using adaptive thresholding. Then the average character height is estimated. The next step's goal is to separate the page into regions, each containing a single component, such as a paragraph, image, table, figure, etc. It can be shown that the gap between components is significantly larger than the gap between text lines. Next, we examine the elements inside each component and try to

group them into rows and columns. These elements' relative positions are further examined to categorize the components into two categories, tables and non-tables. These steps are described in more detail below. The proposed methodology is shown in Fig. 1.

Pre-processing

We start with a grayscale image of the document. If the image is skewed, it must be skew corrected for this method to work. We assume that the given image is already skew corrected. The input image is binarized using the method proposed by Sauvola et al. [20]. The method uses adaptive thresholding and can produce good quality binarized images even for input images that have a change of illumination or noise issues. A sample output image is shown in Fig. 2. We have slightly modified the method to extract an estimate of character height in the document image. In the final step of the binarization method, the connected black elements (say a text character) are plotted on the resulting final image (which was initially taken to be white). While plotting these elements, we keep track of the height of each of them. We find the mean and median of these heights and estimate the

Fig. 1 Flow diagram of the proposed method

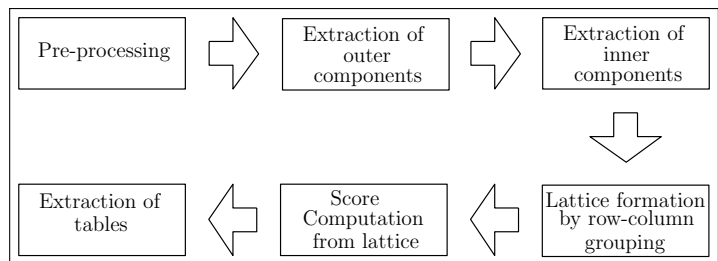


Fig. 2 Binarization

and FL_{5mm} , have been considered as per Figs. 16 and 17. The global sensitivity index is found to be given in Table 11, in which SD1 was found the most sensitive (44.54%) followed by SD2 with 37.7% and lowest variation at all in SD3.

Elementary and interaction global sensitivity analysis (Elementary and interaction global sensitivity analysis)	SD1	SD2	SD3
Mean	49.97	39.4	21.71
Std	21.5	47.6	21.71
Std	22.47	39.4	0
Global importance (%)	44.54	37.7	0
Global Threshold (%)	33.4	32.4	4.7

4) Sensitivity analysis to ambient temperature during Resonance tests. Experimentally, the temperature variation in the different rubber design was measured using a laser-thermocouple. For the data collected, the measurements were performed more than three times for the same scenario. Fig. 19 and Table 12 give the different temperature measurements at the initial and final state (after a couple of cycles) in the rubber of SD1, SD2 and SD3. In Table 12, the predicted final temperatures from (23) are given along with the ambient temperature.

SD	Initial Temp. (°C)	Temperature (°C)	Temperature (°C)
SD1	21.8	21.8	21.8
SD2	21	21.4	21.2
SD3	21	21.2	21.4

5) Sensitivity analysis to ambient temperature during Resonance tests. Experimentally, the temperature variation in the different rubber design was measured using a laser-thermocouple. For the data collected, the measurements were performed more than three times for the same scenario. Fig. 19 and Table 12 give the different temperature measurements at the initial and final state (after a couple of cycles) in the rubber of SD1, SD2 and SD3. In Table 12, the predicted final temperatures from (23) are given along with the ambient temperature.

SD	Initial Temp. (°C)	Temperature (°C)	Temperature (°C)
SD1	21.8	21.8	21.8
SD2	21	21.4	21.2
SD3	21	21.2	21.4

The final temperature was calculated using (23) using the measured values from the experimental measurements and the average initial temperature value, the average energy absorbed E_a , a frequency of about 0.3 Hz, the equivalent thickness of the rubber, and the standard thermal conductivity of the rubber, which is about 0.17 W/mK. Consequently, a discrepancy was observed between the measured and the predicted final temperature in the rubber.

This may be due to several factors such as:

- The assumption taken in the estimation that the rubber thickness is a linear shape. Finite element analysis is required for better results.
- The measurements with the infrared camera were taken at different locations with no easy access and using a rubber consistency of about 0.05 (which varies between 0.02–0.07). There is lack of an accurate consistency value from the supplier of the spacer-damper.
- Accumulation of different uncertainties from different measurements.

Table 13

AMBIENT, INITIAL, ANGLE, TEMPERATURE AND RESONANCE RESPONSE MEASUREMENTS FOR DIFFERENT RUBBER TYPES UNDER THE FINAL RESONANCE MEASUREMENTS ON RESONANCE TESTS

Temperature	Temperature	Temperature
SD	Initial Temp. (°C)	Temperature (°C)

6) Sensitivity analysis to ambient temperature during Resonance tests. Experimentally, the temperature variation in the different rubber design was measured using a laser-thermocouple. For the data collected, the measurements were performed more than three times for the same scenario. Fig. 19 and Table 12 give the different temperature measurements at the initial and final state (after a couple of cycles) in the rubber of SD1, SD2 and SD3. In Table 12, the predicted final temperatures from (23) are given along with the ambient temperature.

SD	Initial Temp. (°C)	Temperature (°C)	Temperature (°C)
SD1	21.8	21.8	21.8
SD2	21	21.4	21.2
SD3	21	21.2	21.4

VI. CONCLUSIONS

This paper illustrates that the selection criteria for a spacer-damper should include the sensitivity response to frequency and force-displacement. Sensitivity could be a significant parameter not only in the spacer-damper optimization, but also in the improvement of the existing design. The sensitivity evaluation shows that there is a significant frequency effect on the damping features of the rubber-like housing for different temperatures.

Comparison of the vibration performance of available spacer-dampers in South Africa shows that their capacity in absorbing energy resides in the design of the rubber housing and the arm length. The latter has to be designed in a way that its impedance reduces the frequency received from the wind to less than 1 Hz at the building rubber in order to enable effectively absorption of energy. However, this present study has examined only the damping as an inertia implication performance and did not consider the inertia implication of the rigid frame. The damping model was studied for one arm and rigid frame of the spacer-damper as if it was independent from the rest of the arm.

Further studies, which should take the inertia of the spacer-damper into account in vibration performance, should include, together with study of the resonance problem to allow for active control, using, for instance, the transfer matrix approach [26] to implement the single dynamic model of spacer-dampers and frame in the spacer-damper by taking into account also the synchronization of all arm inputs into the spacer-damper conductor system.

and FL_{5mm} , have been considered as per Figs. 16 and 17. The global sensitivity index is found to be given in Table 11, in which SD1 was found the most sensitive (44.54%) followed by SD2 with 37.7% and lowest variation at all in SD3.

Elementary and interaction global sensitivity analysis (Elementary and interaction global sensitivity analysis)	SD1	SD2	SD3
Mean	49.97	39.4	21.71
Std	21.5	47.6	21.71
Std	22.47	39.4	0
Global importance (%)	44.54	37.7	0
Global Threshold (%)	33.4	32.4	4.7

4) Sensitivity analysis to ambient temperature during Resonance tests. Experimentally, the temperature variation in the different rubber design was measured using a laser-thermocouple. For the data collected, the measurements were performed more than three times for the same scenario. Fig. 19 and Table 12 give the different temperature measurements at the initial and final state (after a couple of cycles) in the rubber of SD1, SD2 and SD3. In Table 12, the predicted final temperatures from (23) are given along with the ambient temperature.

SD	Initial Temp. (°C)	Temperature (°C)	Temperature (°C)
SD1	21.8	21.8	21.8
SD2	21	21.4	21.2
SD3	21	21.2	21.4

5) Sensitivity analysis to ambient temperature during Resonance tests. Experimentally, the temperature variation in the different rubber design was measured using a laser-thermocouple. For the data collected, the measurements were performed more than three times for the same scenario. Fig. 19 and Table 12 give the different temperature measurements at the initial and final state (after a couple of cycles) in the rubber of SD1, SD2 and SD3. In Table 12, the predicted final temperatures from (23) are given along with the ambient temperature.

SD	Initial Temp. (°C)	Temperature (°C)	Temperature (°C)
SD1	21.8	21.8	21.8
SD2	21	21.4	21.2
SD3	21	21.2	21.4

Fig. 19 Temperature measured in SD1, SD2 and SD3 corresponding respectively to A, B, and C, with index 1 and 2 shows the temperature and index 3 and 4 the temperature during the resonance at a fixed frequency of 0.3 Hz.

Fig. 18 Temperature measured in SD1, SD2 and SD3 corresponding respectively to A, B, and C, with index 1 and 2 shows the temperature and index 3 and 4 the temperature during the resonance at a fixed frequency of 0.3 Hz.

model using Karl Pearson’s formula of computing *mode*, as shown below in Eq. (1). We take this *mode* to be the estimated character height *h*.

$$\text{mode} = 3 \times \text{median} - 2 \times \text{mean} \tag{1}$$

Component Extraction

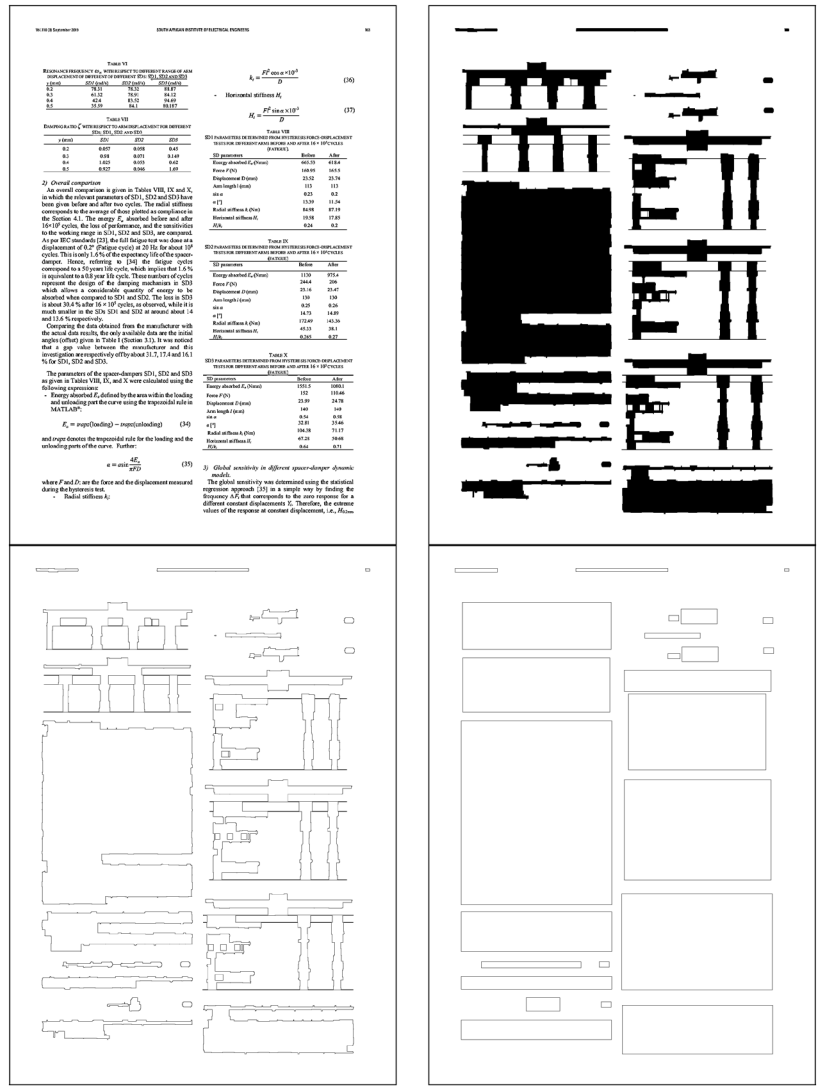
A document consists of a variety of *components* or regions such as text blocks, paragraphs, images, tables, figures, etc. It is helpful to separate them before further processing. Document structure and layout analysis can be used to decompose these components from a document image. Various such techniques exist and are mentioned in [21]. We use a simple smoothing-based technique.

Component Bounding Box Detection

To detect the bounding boxes, we start by smearing the foreground pixels, like coalescing nearby black pixels and forming blobs. We use the run-length smoothing algorithm (RLSA) [22] for this. The RLSA can be used for block segmentation and text discrimination. The algorithm converts white pixels in the input image to black if the number of adjacent white pixels is less than or equal to some predefined limit *l*. We set this limit *l* to be some multiple of the estimated character height *h*. RLSA is applied both horizontally and vertically with respective parameter values *l_h* and *l_v*, respectively (horizontal or vertical run-lengths).

We then traverse the edge of the blobs and find the four extreme points of each blob, namely *x_{min}*, *y_{min}*, *x_{max}* and *y_{max}*.

Fig. 3 Outer bounding box detection steps



These four points are enough to define a bounding box (see Fig. 5). We call these bounding boxes outer bounding boxes as they represent the outer boundary of each component. The steps are shown in Fig. 3.

Inner Elements Detection

To detect the elements inside a component boundary, we use a similar approach to detect components on a page. We start with a copy of the binarized image combining nearby black pixels and forming blobs but this time only horizontally. That is, RLSA is applied only in the horizontal direction. In this way, elements in separate lines do not coalesce into a single blob. We then find the extreme points of each blob and store them as an array of bounding boxes. We call these boxes as inner bounding boxes because these are obtained from the elements

inside each component. We found that rough removal of the long vertical or horizontal lines (table boundaries or separator lines) before applying RLSA in this step gives better results. Occasionally, the cell contents in a ruled table are too close to the table boundary lines. The steps are shown in Fig. 4.

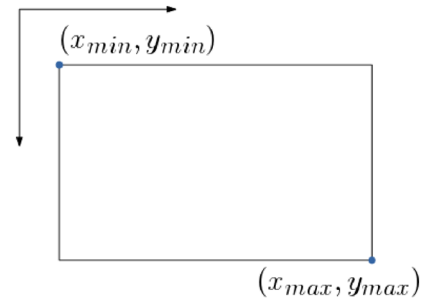


Fig. 5 Bounding box

Fig. 4 Inner bounding box detection steps

The figure contains several tables and diagrams. The top-left table (Table VI) lists parameters for different paper types. The middle-left table (Table VII) lists parameters for different paper types. The top-right table (Table VIII) lists parameters for different paper types. The middle-right table (Table IX) lists parameters for different paper types. The bottom-left diagram shows a document page with bounding boxes overlaid on text and tables. The bottom-right diagram shows a document page with bounding boxes overlaid on text and tables.

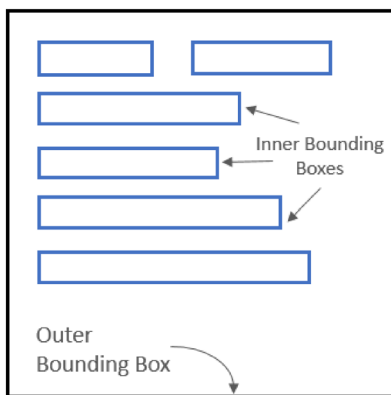


Fig. 6 Component representation

Combining Inner and Outer Bounding Boxes

We now have a list of outer bounding boxes for each component and a list of inner bounding boxes (Fig. 5). These are now combined into a list of components. Each component has an outer bounding box which contains the smaller inner bounding boxes as shown in Fig. 6. These inner components are processed individually for all outer boxes one by one.

Component Representation

We represent each component with the following attributes as shown in the example in Fig. 6.

- Outer bounding box: a rectangular box which contains all inner components.
- An array of inner bounding boxes: an array of bounding boxes of all the elements inside the Outer bounding box. These can be text lines for paragraphs, a cell for a table, or some arbitrary region from graphics.

Table Detection

Once the components have been extracted, we need to identify the ones that could be tables. We will examine the relative position of all the inner elements to see if their structure is close to that of the table cells in any way. However, to do that, we need to find a more straightforward way to represent the inner elements. So, we attempt to group all the inner elements into rows and columns by testing if their X or Y axes projections overlap. Then, based on their overlapping areas, each inner element that could be successfully grouped into rows and columns is allocated one single 2D point. We call this point the overlapping center. Now that a single point can represent each element and by comparing their relative positions and further examination of their layout structure would be more comfortable (Fig. 7).



Fig. 7 Row, column grouping through projection on X and Y axes

Row-Column Grouping

In this step, each component’s inner bounding boxes are grouped into rows and columns and marked accordingly. We ignore any inner bounding box with a width more than 75% of the width of the outer bounding box as cells are generally not of this big size. It may be a header, but we only focus on the cells.

Two inner bounding boxes *A* and *B* can be said to be in the same row if their projections on the Y axis overlap. This can be checked easily with the following formula given in Eq. (2).

$$F_y(A, B) = \begin{cases} \text{True} & (A \cdot y_{\min} \geq B \cdot y_{\min} \text{ and } A \cdot y_{\min} \leq B \cdot y_{\max}) \\ & \text{OR} \\ & (B \cdot y_{\min} \geq A \cdot y_{\min} \text{ and } B \cdot y_{\min} \leq A \cdot y_{\max}) \\ \text{False} & \text{otherwise} \end{cases} \tag{2}$$

The result $F_y(A, B)$ indicates whether the two boxes *A* and *B* have a Y-axis projection intersection or not. Similarly, two inner bounding boxes can be said to be in the same column if their X axis projections overlap each other. The previous formula can be tweaked a little to use in this case, as shown in Eq. (3).

$$F_x(A, B) = \begin{cases} \text{True} & (A \cdot x_{\min} \geq B \cdot x_{\min} \text{ and } A \cdot x_{\min} \leq B \cdot x_{\max}) \\ & \text{OR} \\ & (B \cdot x_{\min} \geq A \cdot x_{\min} \text{ and } B \cdot x_{\min} \leq A \cdot x_{\max}) \\ \text{False} & \text{otherwise} \end{cases} \tag{3}$$

The result $F_x(A, B)$ indicates whether the two boxes *A* and *B* have an X-axis projection intersection or not.

Representing Cells by Single Points

Once the groups are formed, for each element, we check the minimum area that is overlapping with all the elements in the same row group and the same column group. Then we assign the center of this overlapped area to the element as shown in Fig. 8. To describe this more formally, let us suppose that the inner elements $I_1, I_2, I_3, \dots, I_k$ are there in the same column, and the elements $J_1, J_2, J_3, \dots, J_m$ are coming from the same row. Then the coordinates of the center *p* of

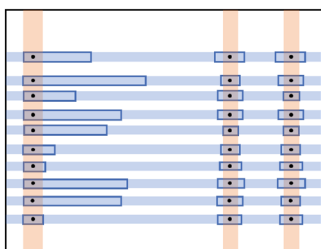


Fig. 8 Overlapped region centers

the overlapping region is defined as shown in Eqs. (4) and (5)

The collection of all p values from all overlapping regions is referred to as the core C . Hence, C represents the well-structured point set representing the table cell cores which are the text blocks' centers within the tabular cells.

$$p_x = x_1 + (x_2 - x_1)/2 \tag{4}$$

where, $x_1 = \text{Max}(I_1 \cdot x_{\text{min}}, I_2 \cdot x_{\text{min}}, I_3 \cdot x_{\text{min}}, \dots, I_k \cdot x_{\text{min}})$,
 $x_2 = \text{Min}(I_1 \cdot x_{\text{max}}, I_2 \cdot x_{\text{max}}, I_3 \cdot x_{\text{max}}, \dots, I_k \cdot x_{\text{max}})$
 and

$$p_y = y_1 + (y_2 - y_1)/2 \tag{5}$$

where,

$$y_1 = \text{Max}(J_1 \cdot y_{\text{min}}, J_2 \cdot y_{\text{min}}, J_3 \cdot y_{\text{min}}, \dots, J_m \cdot y_{\text{min}}),$$

$$y_2 = \text{Min}(J_1 \cdot y_{\text{max}}, J_2 \cdot y_{\text{max}}, J_3 \cdot y_{\text{max}}, \dots, J_m \cdot y_{\text{max}}).$$

Score Computation

Now, we have a set of points, each representing an inner element of the outer components. In tables, these points would represent cells and hence would be arranged as the core structure representing the table as a whole. The cells of a table are generally group-wise uniformly spaced. This helps us in identifying a table even in the absence of ruling lines. We define a score based examination method from the relative distances of these points to identify if a given component is to be considered as a table or not.

Vertical and Horizontal Relations and Distances

For two given points p and q in C , we say, pR_vq (p is vertically related to q) if $|p \cdot y - q \cdot y| \leq \delta$ where δ is some distance threshold. Similarly, we define pR_hq if $|p \cdot x - q \cdot x| \leq \delta$. Next, we do the following using the point set to calculate the tabular structure score:

1. For every point p find a point q , if any, such that pR_hq holds, $p \cdot x > q \cdot x$, and q is closest to p .

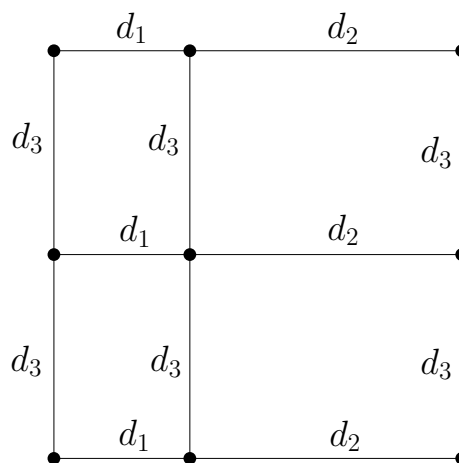


Fig. 9 Example of core point set C

2. For every point p find a point q , if any, such that pR_hq holds, $p \cdot x < q \cdot x$, and q is closest to p .
3. For every point p find a point q , if any, such that pR_vq holds, $p \cdot y > q \cdot y$, and q is closest to p .
4. For every point p find a point q , if any, such that pR_vq holds, $p \cdot y < q \cdot y$, and q is closest to p .

Here, q is closest to p in terms of distances and these distances are stored. The patterns of this distances will reveal the nature of the component. Some distance values will repeat (or close enough) many times in case of tables. We define our score for recognition of tables using the frequencies of these distances as shown in Eq. 6.

$$\text{Score} = \frac{\sum n_{d_i} \times r_{d_i}}{\max\{|pR_hq| + |pR_vq|, |C|\}} \tag{6}$$

Here, n_{d_i} denotes the frequency of the distance d_i , r_{d_i} denotes the number of points involved with distance d_i , $|pR_hq|$ and $|pR_vq|$ denotes, respectively, the number of horizontal and vertical relations, and $|C|$ denotes the size of core C (in terms of number of points). For example, with respect to the following point set shown in Fig. 9, we have, $r_{d_1} = 6$, $n_{d_1} = 3$, $r_{d_2} = 6$, $n_{d_2} = 3$, $r_{d_3} = 9$, $n_{d_3} = 6$, $|pR_hq| = 6$, $|pR_vq| = 6$, and $|C| = 9$. Therefore, for this example we have, $\text{Score} = 7.5$.

Results and Discussions

Our program was tested on a computer with Intel Core i3 – 6098P Processor which has a base frequency of 3.60 GHz. Our method was tested on 80 input documents images taken from various scholarly articles. the document pages contain various types of tables along with other types of graphics elements like plots, equations, images etc.

Fig. 10 Scores of the components detected as tables are 11.33333 and 9.71429

164

SOUTH AFRICAN INSTITUTE OF ELECTRICAL ENGINEERS

Vol.110 (3) September 2019

and H_0 5mm, have been considered as per Figs. 16 and 17. The global sensitivity (relative based) is given in Table 11, in which SD1 was found the most sensitive (44.54 %) followed by SD2 with 37.7 % and then no variation at all in SD3.

11.33333 TABLE XI
GLOBAL AND THEORETICAL GLOBAL SENSITIVITY ANALYSIS TECHNIQUE (GSAT) IN PERCENTAGE (%) OF SD1, SD2 AND SD3

Sensitivity	SD1	SD2	SD3
A1	97	97	97
A2	97	97	97
A3	97	97	97
A4	97	97	97
CA1 experiments	97	97	97
CA1 Theoretical	97	97	97

1.00000 analysis to ambient temperature during hysteresis tests.

Experimentally, the temperature variation in the different rubber design was measured using a laser-thermo-camera. For the data collected, the measurements were performed more than three times for the same scenario. Fig. 19 and Table 12 give the different temperature measurements at the initial and final states (after a couple of cycles) in the rubber of SD1, SD2 and SD3. In Table 12, the predicted final temperatures from (23) are given along with the ambient temperature.



Fig. 19. Temperature measured in SD1, SD2 and SD3 corresponding respectively to A1, B1 and C1 (with index 1 and 2 denotes the temperature state before and after measurement) during the hysteresis testing at fixed frequency of 0.35 Hz.

The final temperature was calculated using (23) using the numerical values from the experimental measurements and the average initial temperature value, the average energy absorbed E_a , a frequency of about 0.35 Hz, the equivalent thickness of the rubber, and the standard thermal conductivity of the rubber K , which is about 240 W/mm²K. Consequently, a discrepancy was observed between the measured and the predicted final temperature in the rubber.

1.50000 due to several factors such as:

- The assumption taken in the estimation that the rubber thickness is a plane shape; finite element analysis is required for better results.
- The measurements with the infrared camera were taken at different locations with no easy access and using a rubber emissivity of about 0.95 (which varies between 0.85-0.97). There is lack of an accurate emissivity value from the supplier of the spacer-damper;
- Accumulation of different uncertainties from different measurements.

9.71429 TABLE XII
INITIAL AND FINAL TEMPERATURE MEASURED AND PREDICTED DURING THE HYSTERESIS TESTING AT FREQUENCY EQUAL TO 0.35 HZ. THE FINAL PREDICTED TEMPERATURE WAS DETERMINED USING (23).

SD	Temperature ambient °C	Temperature measured °C	Temperature predicted °C
SD1	3.18182	23.6	23.6
SD2	3.17647	22.5	22.5
SD3	2.92308	23.6	23.6

VI. CONCLUSIONS

1.33333 illustrates that the selection criteria for a spacer-damper should include the sensitivity response to frequency and force-displacement sensitivity could be a significant parameter not only in the spacer-damper optimisation, but also in the improvement of its existing design. The sensitivity evaluation shows that there is a significant frequency effect on the damping features of the rubberlike bushing for different temperatures.

Comparison of the vibration performances of available spacer-dampers in South Africa shows that their capacity in absorbing energy resides in the design of the rubber bushing and the arm length. The latter has to be designed in a way that its impedance reduces the frequency received from the wind to less than 1 Hz at the bushing rubber in order to enable effectively absorption of energy. However, this present study has examined only the damping and the stiffness factor performance and did not consider the inertia implication of the rigid frame. The damping model was studied for one arm and rigid frame of the spacer-damper as if it was independent from the rest of the arms.

Further studies, which should take the inertia of the spacer-damper into account in vibration performance, should follow, together with study of the inverse problem to allow for active control, using, for instance, the transfer matrix approach [36] to implement the single dynamic model of representation arm and frame in the spacer-damper by taking into account also the synchronization of all arm inputs into the spacer-damper-conductor system.

Fig. 11 Score of the component detected as table is 13.4761

Date	Score
1 March st	13.47619
30 April st	
30 May st	
30 June st	
30 July st	
30 August st	
30 September st	
2 October st	

Table 2.1: Correspondence

2.1 The prediction on the number of confirmed cases in Europe

1.00000 compute the trend of R_0 , or the basic production number changes over time. After we take the daily number of confirmed cases in Europe at the end of each day running from 1 March to 2 May, numbered as day 1 to day 64, [11][6], we apply historical data with the formula $R_0(t) = \left(\frac{t_1 - t_0}{t_0}\right)$ to derive the composite R_0 , that is, we take the ratio of current day increase over the previous day increase. Next, we need to compute the proportion of population currently vulnerable, it starts with unity of 1 at day 1 and decreases toward 0 as greater proportion of population have been diagnosed who eventually either recovers with immunity or deceased. We use the formula:

1.42857

$$P_{ortion}(t) = 1 - \frac{\sum_{n=0}^t I_{europe}(n)}{P_{europe}} \tag{2.1}$$

Where: $P_{europe} = 63746974$, that is the population of Europe excluding Russia, and $I_{europe}(t)$ is the number of diagnosed in Europe at any given day. The total number of diagnosed up to day t is therefore $\sum_{n=0}^t I_{europe}(n)$. We divide $R_0(t)$ by $P_{ortion}(t)$ to derive the raw basic production number of the given day.

1.50000

$$R_{aw}(t) = \frac{R_0(t)}{P_{ortion}(t)} \tag{2.2}$$

The raw $R_{aw}(t)$ is guaranteed to become somewhat larger because a lower proportion of the entire population becomes vulnerable overtime, helping reducing the composite $R_0(t)$ number. Next, its moving average over a window size of 7 days is applied so that the fluctuation of the infection cycle with 7 day incubation is reduced. A simple exponential decay fit in the form if ax^b is applied to the data points as an approximation. After removing extreme outlier at day

Figure 14 shows the required CPU times for detection of tables for some sample documents of different size.

The computed scores from the core points for some documents are shown in Figs. 10, 11, 12, 13. Based on experiment observations, we have classified components as tables when the score exceeds 5.00 (Fig. 14).

Results are shown for various types of document images in Figs. 15, 16 and 17. Here, Figs. 15 and 16 show pages containing ruled tables, whereas Fig. 17 shows document pages with tables where cell contents are not separated by ruled lines.

Evaluation Metrics

To evaluate the classification accuracy, four metrics have been used in our work: Precision, Recall, F1 score, and Accuracy. Respective definitions are shown in Eq. (7) where TP, FP, FN, and TN represent true positives, false positives, and false negatives, and true negatives, respectively. Here, TP represents the count of tables correctly predicted as a table. The figure FP shows the number of non-tables (plots, graphs, graphics) predicted as a table. FN represents the count of tables not detected as tables, and TN denotes the count of non-tables predicted as a non-table.

$$\text{Accuracy} = (TP + TN)/(TP + FP + FN + TN)$$

$$\text{Precision} = TP/(TP + FP)$$

$$\text{Recall} = TP/(TP + FN)$$

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (7)$$

Another metric we used for evaluation, Intersection over Union (IoU), which is used widely in the object detection benchmarks [11]. It measures the overlap between predicted and ground truth tables' covering rectangles or polygons. The value of IoU lies in the range of [0, 1]. The higher value of IoU designates maximum match in the ground truth and predicted tables.

Our Results

Initially, we tested our method on our own dataset. Our dataset contains 80 document images with 99 tables in total. We obtained $FP = 8$, $FN = 6$, and $TP = 93$ thereby giving precision = 0.921, recall = 0.939, and F_1 score equal to 0.93.

We have also tested our method on datasets like ICDAR-2013, Marmot, TableBank, and ICDAR-2019. ICDAR-2013 [12] is one of the most popular datasets for table detection and structure recognition. This dataset is created by documents obtained from web pages. This dataset was made for a competition concentrated on detecting figures, tables, and mathematical equations from document images. The dataset is composed of PDF files which we converted to images to be used within our research work. The dataset contains 59 PDF files, a total of 117 tables. To give the algorithms ample possibility to find false positives, approximately two pages before and after the table included as excerpts. A comparative discussion on accuracy figures with respect to other methods applied on ICDAR-2013 is shown in Table 1, which clearly shows that the proposed method outperforms the other methods. Our proposed method detects well irrespective of the presence of the table boundary rule lines, which is the major contribution of our work.

The Marmot [23] dataset comprises English and Chinese documents. The dataset consists of 2000 images, where a ratio of almost 1:1 is present between the positive to negative samples. The pages show a great variety in language type, page layout, and table styles. Over 1500 conference and journal papers pages were taken into the dataset, covering various fields, spanning from the year 1970 to the latest 2011 publications. The e-Book pages are primarily in one-column layout, while the English pages are mixed with both one-column and two-column layouts. Our method was tested on English pages, and it achieved the precision of 0.960, recall of 0.984, and F1 score equal to 0.972.

The TableBank dataset [24] consists of 417, 234 high quality labeled tables as well as their original documents in a variety of domains. Our method achieved the precision of

Table 1 Results of proposed method on ICDAR-2013 dataset in comparison with the state-of-the-art

Method	Approach	Accuracy on ICDAR-2013
Casado-García et al. [1]	Fine-tuning from a closer domain over deep learning algorithms	Average F_1 Score = 0.74 using the RetinaNet algorithm (best)
Kavasidis et al. [13]	Combination of deep convolutional neural networks, graphical models and saliency concepts	Precision and recall are 97.5% and 98.1%, respectively
Schreiber et al. [15]	DeepDeSRT: deep learning for detection and structure recognition of tables in document images	Accuracy for table detection and structure recognition is 91.44%
Anh et al. [4]	Hybrid approach: algorithm uses an alternative bottom-up and top-down approach to identify the table regions in document image	Un-ruled: 97.63%, ruled: 99.05%
Proposed method	Core points extraction and score finding	Un-ruled: 99.02%, ruled: 99.14%

Fig. 12 Score of the component detected as table is 8.6666

It is observed that the peak in confirmed cases is left skewed due to a delayed peak from the rest of the world.

This allows a prediction of the world wide number of diagnoses at specific dates in the future.

	Daily number of confirmed cases	Cumulative number of confirmed cases
June 1st	104,62	6,33,054
July 1st	78,90	9,19,222
August 1st	43,41	11,07,025
September 1st	22,46	12,05,790
October 1st	12,55	12,57,132

Table 2.6: Worldwide daily confirmed cases predictions

When taking the limit, the final number of diagnoses in the world is 13,325,574.

$$\lim_{x \rightarrow \infty} \exp(F(x)) = 133,5574 \quad (2.8)$$

Compared with our earliest rough assessment on the global cases, this result falls within the lower and upper bound; however, the trend of change has a much higher resolution due to a better comprehension of the underlying data.

And based on the final number of diagnoses in Europe and the United States. From this, the proportion of each region out of all confirmed cases can be observed.

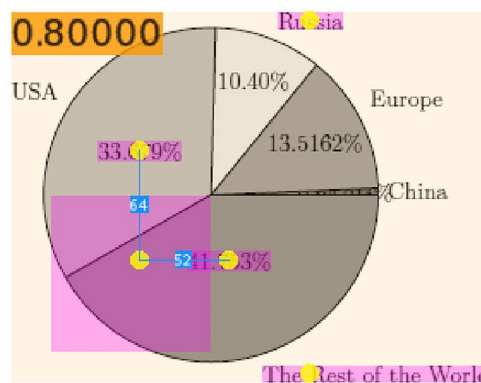


Figure 2.24: The proportion of each region out of all confirmed cases

2.5.2 Lower Bound based on $R_{cumulative}(t)$ (updating soon)

Much like our earlier prediction based on R_0 , the plateau lasts by end of May and starts to drop. Unlike earlier scenario, due to smaller number of expected cases from US at 2.1 million,

Fig. 13 Scores of the components detected as tables are 13.5714 and 12.2857

$$\int_0^d \frac{d}{dx} D_{death}(x) dx \tag{3.6}$$

13.57143	200	Predicted	242	Actual
Mar 21st	200	1670	242	1239
Apr 5th	200	5221	242	4934
Apr 10th	200	7217	242	7915
Apr 15th	200	9279	242	8934
Apr 18th	200	10496	242	10178

Table 3.1: cross check

It implies that the average mortality rate in Europe is very high, at about 11.2%. It seems no surprise since a correlation exists between high mortality rates and an overwhelmed Europe’s health care system from early on, a higher proportion of the elderly, no special therapeutic treatment, and no large-scale inoculation of the Bacillus Calmette-Guérin vaccine. [1][5]
 After deriving the death curve, we quickly obtain the amplitude of the recovery curve at $A=0.888$. While the value of W set to 0.58 best fits the existing recovery data.

$$R_{recovery}(x) = 0.888 \times \exp(F_{europe}(0.58x)) \tag{3.7}$$

we also double checked the cumulative number of recoveries by integration:

$$\int_0^d \frac{d}{dx} R_{recovery}(x) dx \tag{3.8}$$

12.28571	200	Predicted	240	Actual
Mar 21st	200	2632	240	3152
Apr 5th	200	12638	240	12633
Apr 10th	200	18696	240	20690
Apr 15th	200	26052	240	26242
Apr 18th	200	31118	240	30818

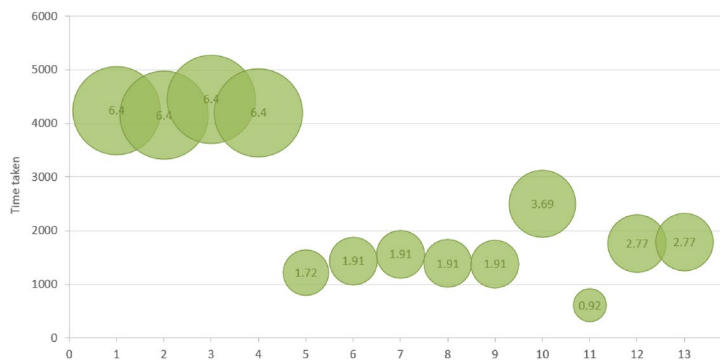
Table 3.2: cross check

Diagnosis, recovery, and death curves in Europe:

0.9813, recall of 0.9482, and F1 Score 0.9645, respectively, on a subset of this dataset. We tested only on Part 1 of this dataset. The Part 1 of this dataset consists of 1379 document pages.

The ICDAR-19 represents a recent time table detection dataset introduced in the table detection competition at ICDAR 2019 [25]. Our approach achieved the precision

Fig. 14 Time taken for table localization in millisecond (the sizes of the circles represent the size of the images)



40 Chapter 2. Intelligent Agents

2.3 THE NATURE OF ENVIRONMENTS

TASK ENVIRONMENT Now that we have a definition of rationality, we are almost ready to think about building rational agents. First, however, we must think about **task environments**, which are essentially the “problems” to which rational agents are the “solutions.” We begin by showing how to specify a task environment, illustrating the process with a number of examples. We then show that task environments come in a variety of flavors. The flavor of the task environment directly affects the appropriate design for the agent program.

2.3.1 Specifying the task environment

In our discussion of the rationality of the simple vacuum-cleaner agent, we had to specify the performance measure, the environment, and the agent’s actuators and sensors. We group all these under the heading of the **task environment**. For the acronymically minded, we call this the **PEAS** (Performance, Environment, Actuators, Sensors) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

PEAS The vacuum world was a simple example; let us consider a more complex problem: an automated taxi driver. We should point out, before the reader becomes alarmed, that a fully automated taxi is currently somewhat beyond the capabilities of existing technology. (page 28 describes an existing driving robot.) The full driving task is extremely *open-ended*. There is no limit to the novel combinations of circumstances that can arise—another reason we chose it as a focus for discussion. Figure 2.4 summarizes the PEAS description for the taxi’s task environment. We discuss each element in more detail in the following paragraphs.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figure 2.4 PEAS description of the task environment for an automated taxi.

First, what is the **performance measure** to which we would like our automated driver to aspire? Desirable qualities include getting to the correct destination; minimizing fuel consumption and wear and tear; minimizing the trip time or cost; minimizing violations of traffic laws and disturbances to other drivers; maximizing safety and passenger comfort; maximizing profits. Obviously, some of these goals conflict, so tradeoffs will be required.

Next, what is the driving **environment** that the taxi will face? Any taxi driver must deal with a variety of roads, ranging from rural lanes and urban alleys to 12-lane freeways. The roads contain other traffic, pedestrians, stray animals, road works, police cars, puddles,

40 Chapter 2. Intelligent Agents

2.3 THE NATURE OF ENVIRONMENTS

TASK ENVIRONMENT Now that we have a definition of rationality, we are almost ready to think about building rational agents. First, however, we must think about **task environments**, which are essentially the “problems” to which rational agents are the “solutions.” We begin by showing how to specify a task environment, illustrating the process with a number of examples. We then show that task environments come in a variety of flavors. The flavor of the task environment directly affects the appropriate design for the agent program.

2.3.1 Specifying the task environment

In our discussion of the rationality of the simple vacuum-cleaner agent, we had to specify the performance measure, the environment, and the agent’s actuators and sensors. We group all these under the heading of the **task environment**. For the acronymically minded, we call this the **PEAS** (Performance, Environment, Actuators, Sensors) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

The vacuum world was a simple example; let us consider a more complex problem: an automated taxi driver. We should point out, before the reader becomes alarmed, that a fully automated taxi is currently somewhat beyond the capabilities of existing technology. (page 28 describes an existing driving robot.) The full driving task is extremely *open-ended*. There is no limit to the novel combinations of circumstances that can arise—another reason we chose it as a focus for discussion. Figure 2.4 summarizes the PEAS description for the taxi’s task environment. We discuss each element in more detail in the following paragraphs.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figure 2.4 PEAS description of the task environment for an automated taxi.

First, what is the **performance measure** to which we would like our automated driver to aspire? Desirable qualities include getting to the correct destination; minimizing fuel consumption and wear and tear; minimizing the trip time or cost; minimizing violations of traffic laws and disturbances to other drivers; maximizing safety and passenger comfort; maximizing profits. Obviously, some of these goals conflict, so tradeoffs will be required.

Next, what is the driving **environment** that the taxi will face? Any taxi driver must deal with a variety of roads, ranging from rural lanes and urban alleys to 12-lane freeways. The roads contain other traffic, pedestrians, stray animals, road works, police cars, puddles,

Fig. 15 CPU time taken = 6.41 s, image size = 6.4 megabytes

of 0.9777, recall of 0.9518, and F1 Score 0.9645 on this dataset. Sample results using the proposed method on some sample document pages from various datasets are shown in Fig. 18.

For the proposed method, we obtained the IoU values as 0.90, 0.65 and 0.60 with respect to our dataset, the Marmot dataset, and the TableBank (Part 1), respectively (Tables 2, 3, 4).

Challenges

Sample document pages where our proposed method failed are shown in Fig. 19 from the Marmot dataset. In the image shown in Fig. 19a (Marmot-123), horizontal spacing is not there in the first column, and this fact lowers the score. In our experimentation, the horizontal (l_h) and vertical (l_v)

Chapter 2. Intelligent Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 2.5 Examples of agent types and their PEAS descriptions.

we list the dimensions, then we analyze several task environments to illustrate the ideas. The definitions here are informal; later chapters provide more precise statements and examples of each kind of environment.

Fully observable vs. partially observable: If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. A task environment is effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action; relevance, in turn, depends on the performance measure. Fully observable environments are convenient because the agent need not maintain any internal state to keep track of the world. An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data—for example, a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares, and an automated taxi cannot see what other drivers are thinking. If the agent has no sensors at all then the environment is **unobservable**. One might think that in such cases the agent's plight is hopeless, but, as we discuss in Chapter 4, the agent's goals may still be achievable, sometimes with certainty.

Single agent vs. multiagent: The distinction between single-agent and multiagent en-

FULLY OBSERVABLE
PARTIALLY OBSERVABLE
UNOBSERVABLE
SINGLE AGENT
MULTIAGENT

Chapter 2. Intelligent Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 2.5 Examples of agent types and their PEAS descriptions.

we list the dimensions, then we analyze several task environments to illustrate the ideas. The definitions here are informal; later chapters provide more precise statements and examples of each kind of environment.

Fully observable vs. partially observable: If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. A task environment is effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action; relevance, in turn, depends on the performance measure. Fully observable environments are convenient because the agent need not maintain any internal state to keep track of the world. An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data—for example, a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares, and an automated taxi cannot see what other drivers are thinking. If the agent has no sensors at all then the environment is **unobservable**. One might think that in such cases the agent's plight is hopeless, but, as we discuss in Chapter 4, the agent's goals may still be achievable, sometimes with certainty.

Single agent vs. multiagent: The distinction between single-agent and multiagent en-

FULLY OBSERVABLE
PARTIALLY OBSERVABLE
UNOBSERVABLE
SINGLE AGENT
MULTIAGENT

Fig. 16 CPU time taken = 8.05 s, image size = 6.4 megabytes

run-length parameters are taken as $1.5h$ and $1.5h$ to detect the outer and inner boxes. Here, h is the average height of characters on the page. In the case of the image shown in Fig. 19b (Marmot-190), the presence of the vertical separator line and its proximity with the text words are the reasons for failure.

For the sample shown in Fig. 20 (Tablebank-007), we see that the distributions of texts within the table cells are not uniform. So, a proper selection of horizontal and vertical smoothing parameters detects the table correctly. We see that a run-length of value $2h$ for horizontal smoothing (l_h) and of h for vertical smoothing (l_v) detects the table correctly.

Optimized selection of the run-length parameter values can be done in terms of the average height of characters. The inter-line, inter-paragraph, or the table cell gaps are decided in terms of average height of characters.

Conclusions

This paper presents a novel method for the detection of tables from document images. Table detection in documents is, for instance, necessary to convert tables in a document image into an editable format. Once the tables are

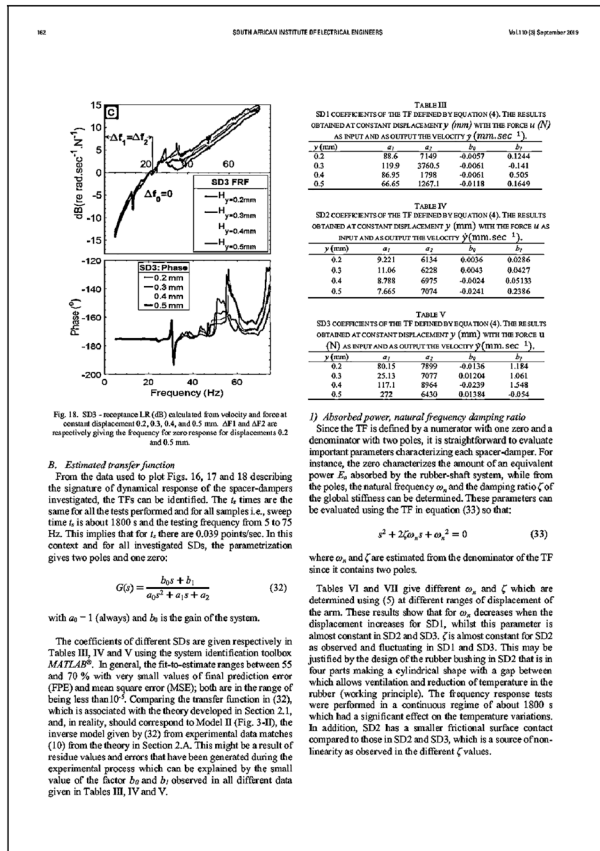


Fig. 17. SD3 - resonance LR (dB) calculated from velocity and force at constant displacement 0.2, 0.3, 0.4, and 0.5 mm. ΔF_1 and ΔF_2 are respectively giving the Frequency for zero response for displacements 0.2 and 0.5 mm.

B. Estimated transfer function
From the data used to plot Figs. 16, 17 and 18 describing the signature of dynamical response of the spacer-dampers investigated, the TFs can be identified. The t_s times are the same for all the tests performed and for all samples i.e., sweep time t_s is about 1800 s and the testing frequency from 5 to 75 Hz. This implies that for t_s there are 0.039 points/sec. In this context and for all investigated SDs, the parametrization gives two poles and one zero:

$$G(s) = \frac{b_2s + b_1}{a_2s^2 + a_1s + a_0} \quad (32)$$

with $a_0 = 1$ (always) and b_2 is the gain of the system.

The coefficients of different SDs are given respectively in Tables III, IV and V using the system identification toolbox *MATLAB*[®]. In general, the fit-to-estimate ranges between 55 and 70 % with very small values of final prediction error (FPE) and mean square error (MSE); both are in the range of being less than 10^{-3} . Comparing the transfer function in (32), which is associated with the theory developed in Section 2.1, and, in reality, should correspond to Model II (Fig. 3-4I), the inverse model given by (32) from experimental data matches (10) from the theory in Section 2.A. This might be a result of residue values and errors that have been generated during the experimental process which can be explained by the small value of the factor b_2 and b_1 observed in all different data given in Tables III, IV and V.

1) Absorbed power, natural frequency damping ratio
Since the TF is defined by a numerator with one zero and a denominator with two poles, it is straightforward to evaluate important parameters characterizing each spacer-damper. For instance, the zero characterizes the amount of an equivalent power P_z absorbed by the rubber-shaft system, while from the poles, the natural frequency ω_n and the damping ratio ζ of the global stiffness can be determined. These parameters can be evaluated using the TF in equation (33) so that:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad (33)$$

where ω_n and ζ are estimated from the denominator of the TF since it contains two poles.

Tables VI and VII give different ω_n and ζ which are determined using (5) at different ranges of displacement of the arm. These results show that for ω_n , decreases when the displacement increases for SD1, whilst this parameter is almost constant in SD2 and SD3. ζ is almost constant for SD2 as observed and fluctuating in SD1 and SD3. This may be justified by the design of the rubber bushing in SD2 that is in four parts making a cylindrical shape with a gap between which allows ventilation and reduction of temperature in the rubber (working principle). The frequency response tests were performed in a continuous regime of about 1800 s which had a significant effect on the temperature variations. In addition, SD2 has a smaller frictional surface contact compared to those in SD1 and SD3, which is a source of non-linearity as observed in the different ζ values.

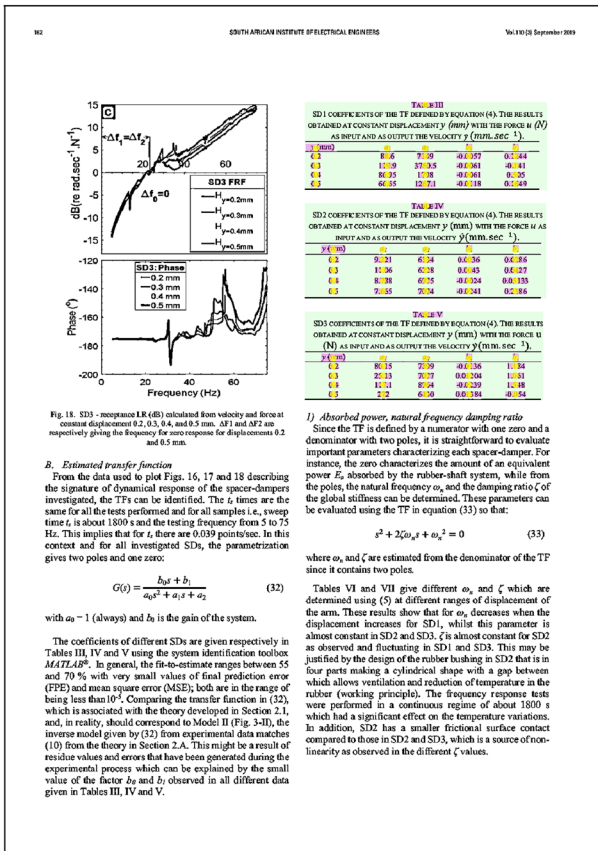


Fig. 18. SD3 - resonance LR (dB) calculated from velocity and force at constant displacement 0.2, 0.3, 0.4, and 0.5 mm. ΔF_1 and ΔF_2 are respectively giving the Frequency for zero response for displacements 0.2 and 0.5 mm.

B. Estimated transfer function
From the data used to plot Figs. 16, 17 and 18 describing the signature of dynamical response of the spacer-dampers investigated, the TFs can be identified. The t_s times are the same for all the tests performed and for all samples i.e., sweep time t_s is about 1800 s and the testing frequency from 5 to 75 Hz. This implies that for t_s there are 0.039 points/sec. In this context and for all investigated SDs, the parametrization gives two poles and one zero:

$$G(s) = \frac{b_2s + b_1}{a_2s^2 + a_1s + a_0} \quad (32)$$

with $a_0 = 1$ (always) and b_2 is the gain of the system.

The coefficients of different SDs are given respectively in Tables III, IV and V using the system identification toolbox *MATLAB*[®]. In general, the fit-to-estimate ranges between 55 and 70 % with very small values of final prediction error (FPE) and mean square error (MSE); both are in the range of being less than 10^{-3} . Comparing the transfer function in (32), which is associated with the theory developed in Section 2.1, and, in reality, should correspond to Model II (Fig. 3-4I), the inverse model given by (32) from experimental data matches (10) from the theory in Section 2.A. This might be a result of residue values and errors that have been generated during the experimental process which can be explained by the small value of the factor b_2 and b_1 observed in all different data given in Tables III, IV and V.

Fig. 17 CPU time taken = 4.96 s, image size = 1.90 megabytes

detected, the table cells can be localized, and finally, using OCR, we can extract cell contents. While it is straightforward to detect ruled tables and is often done by identifying the horizontal and vertical lines of the table borders, it is more challenging to detect unruled or partially ruled tables. We presented a method that can recognize tables irrespective of whether it is ruled, partially ruled or unruled. We do not look for any lines or boundaries; instead, we rely on the fact that cells of a table are arranged in well

structured way. We have shown that the structure can be represented as a set of core points if the cell contents are replaced by representative points. In this work, we have shown the score computation method only for the detection table structures. Further, the work can be extended to design the score formulae for other types of graphics elements like plots, graphs, equations, etc. Further automated selection of run-length parameters for smoothing will be worthy to explore for fine-tuning.

Fig. 18 Results using the proposed method on some sample document pages from various datasets; **a, b** samples from TableBank dataset, **c, d**: samples from Marmot dataset, **e, f**: samples ICDAR 2019 dataset

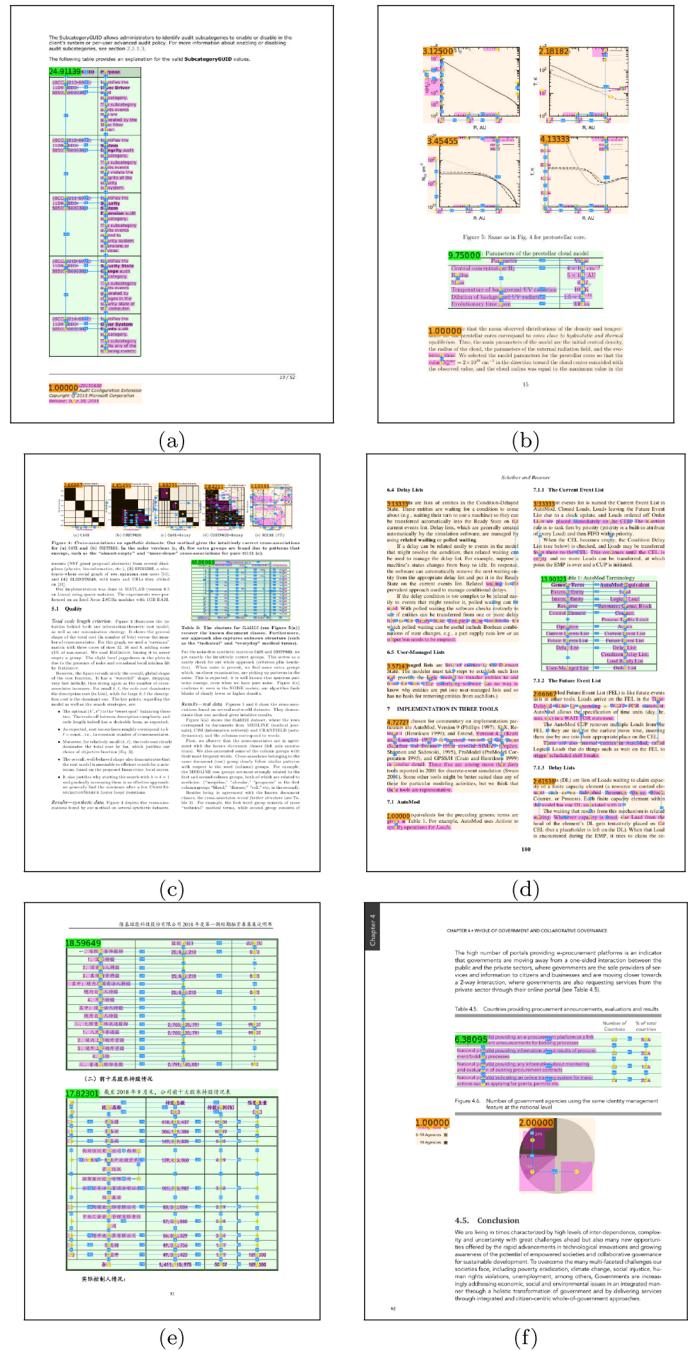


Table 2 Results of proposed method on Marmot dataset in comparison with the state-of-the-art

Article	Precision	Recall	F1 Score	Method
Nazir et al. [26]	0.962	0.961	0.956	ResNeXt-101 backbone and hybrid task cascade (HTC)
Hashmi et al. [27]	0.952	0.965	0.958	Recursive feature pyramids (RFP) and switchable atrous convolutions (SAC), cascade mask R-CNN
garwal et al. [28]	0.975	0.930	0.952	CDeC-Net, MaskR-CNN
He et al. [29]	0.753	0.70	–	Multi-scale, multi-task fully convolutional neural network (FCN)
Proposed	0.960	0.984	0.972	Geometric analysis of the text block cells and scores from the distributions of the cell cores

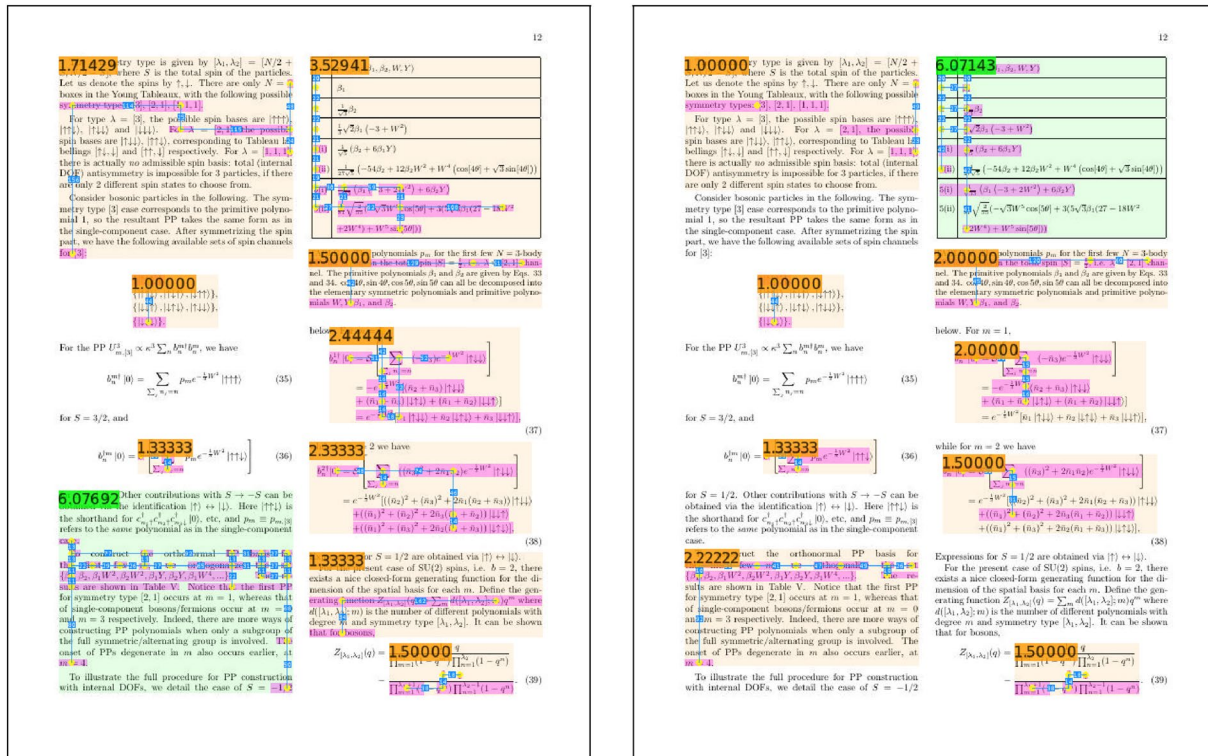


Fig. 20 Sample document page from the TableBank dataset where the method fails using both the l_h and l_v values equal to $1.5h$ (left); proper detection using the l_h and l_v values as $2h$ and h , respectively

Acknowledgement We would like to acknowledge IIIT Kalyani undergraduate students Sayantan Das, Somenath Maji, and Somnath Pal for their help on this paper.

Funding Open access funding provided by FHNW University of Applied Sciences and Arts Northwestern Switzerland.

Data Availability The used datasets are available at the locations specified in the references [11, 12, 23–25]. Our software is available at Zenodo at <https://zenodo.org/record/5824970>.

Declarations

Conflict of Interest The authors declare that there is no conflict of interest related to this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Casado-García Á, Domínguez C, Heras J, Mata E, Pascual V. The benefits of close-domain fine-tuning for table detection in document images. In: International workshop on document analysis systems. Cham: Springer; 2020. p. 199–215.
2. Gilani A, Qasim SR, Malik I, Shafait F. Table detection using deep learning. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR), vol. 1, IEEE; 2017, p. 771–6.
3. Shafait F, Smith R. Table detection in heterogeneous documents. In: Proceedings of the 9th IAPR international workshop on document analysis systems, IEEE; 2010, p. 65–72.
4. Anh TT, In-Seop N, Soo-Hyung K. A hybrid method for table detection from document image. In: 2015 3rd IAPR Asian conference on pattern recognition (ACPR), IEE; 2015, p. 131–5.
5. Jahan MA, Ragel RG. Locating tables in scanned documents for reconstructing and republishing. In: 7th international conference on information and automation for sustainability, IEEE; 2014, p. 1–6.
6. Bansal A, Harit G, Roy SD. Table extraction from document images using fixed point model. In: Proceedings of the 2014 Indian conference on computer vision graphics and image processing, ACM; 2014, p. 1–8.
7. Kasar T, Barlas P, Adam S, Chatelain C, Paquet T. Learning to detect tables in scanned document images using line information. In: 2013 12th international conference on document analysis and recognition, IEEE; 2013, p. 1185–1189.

8. Forczmański P, Smoliński A, Nowosielski A, Małecki K. Segmentation of scanned documents using deep-learning approach. In: International conference on computer recognition systems. Springer, Cham; 2019, p. 141–152.
9. Redmon J, Farhadi A. Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE; 2017, p. 7263–7271.
10. Qasim SR, Mahmood H, Shafait F. Rethinking table recognition using graph neural networks; 2019. p. 142–7.
11. Please update the reference with complete details.
12. Göbel M, Hassan T, Oro E, Orsi G. Icdar 2013 table competition. In: 2013 12th international conference on document analysis and recognition, IEEE; 2013, p. 1449–53.
13. Kavasidis I, Pino C, Palazzo S, Rundo F, Giordano D, Messina P, Spampinato C. A saliency-based convolutional neural network for table and chart detection in digitized documents. In: International conference on image analysis and processing. Springer, Cham; 2019, p. 292–302.
14. Arif S, Shafait F. Table detection in document images using foreground and background features. In: Digital image computing: techniques and applications (DICTA). IEEE; 2018, p. 1–8.
15. Schreiber S, Agne S, Wolf I, Dengel A, Ahmed S. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR), vol. 1, IEEE; 2017, p. 1162–7.
16. Hao L, Gao L, Yi X, Tang Z. A table detection method for pdf documents based on convolutional neural networks. In: 12th IAPR workshop on document analysis systems (DAS). IEEE; 2016, p. 287–92.
17. Gatos B, Danatsas D, Pratikakis I, Perantonis SJ. Automatic table detection in document images. In: International conference on pattern recognition and image analysis. Springer; 2005, p. 609–18.
18. Ramel J-Y, Crucianu M, Vincent N, Faure C. Detection, extraction and representation of tables. In: Seventh international conference on document analysis and recognition. Proceedings., IEEE; 2003, p. 374–8.
19. Mandal S, Chowdhury S, Das AK, Chanda B. A simple and effective table detection system from document images. *Int J Doc Anal Recogn.* 2006;8(2):172–82.
20. Hadjadj Z, Meziane A, Cherfa Y, Cheriet M, Setitra I. Isauvola: improved sauvola's algorithm for document image binarization. In: International conference on image analysis and recognition. Springer, Cham; 2016, p. 737–45.
21. Nambodiri AM, Jain AK. Document structure and layout analysis. London: Springer; 2007. p. 29–48.
22. Wong KY, Casey RG, Wahl FM. Document analysis system. *IBM J Res Dev.* 1982;26(6):647–56.
23. Fang J, Tao X, Tang Z, Qiu R, Liu Y. Dataset, ground-truth and performance metrics for table detection evaluation. In: 10th IAPR international workshop on document analysis systems. IEEE. 2012; p. 445–9.
24. Li M, Cui L, Huang S, Wei F, Zhou M, Li Z. Tablebank: table benchmark for image-based table detection and recognition. In: Proceedings of the 12th language resources and evaluation conference; 2020, p. 1918–25.
25. Gao L, Huang Y, Déjean H, Meunier J-L, Yan Q, Fang Y, Kleber F, Lang E. Icdar 2019 competition on table detection and recognition (CTDAR). In: 2019 international conference on document analysis and recognition (ICDAR), IEEE; 2019, p. 1510–5.
26. Nazir D, Hashmi KA, Pagani A, Liwicki M, Stricker D, Afzal MZ. Hybridtabnet: towards better table detection in scanned document images. *Appl Sci.* 2021;11(18):8396.
27. Hashmi KA, Pagani A, Liwicki M, Stricker D, Afzal MZ. Castabdetectors: cascade network for table detection in document images with recursive feature pyramid and switchable atrous convolution. *J Imaging.* 2021;7(10):214.
28. Agarwal M, Mondal A, Jawahar C. Cdec-net: composite deformable cascade network for table detection in document images. In: 2020 25th international conference on pattern recognition (ICPR), IEEE; 2021, p. 9491–8.
29. He D, Cohen S, Price B, Kifer D, Giles CL. Multi-scale multi-task fcn for semantic page segmentation and table detection. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR), vol 1, IEEE; 2017, p. 254–61.
30. Prasad D, Gadpal A, Kapadni K, Visave M, Sultanpure K. Cascadetabnet: an approach for end to end table detection and structure recognition from image-based documents. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops; 2020, p. 572–3.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.