Abderrahmane Kefali
Soumia Drabsia
Toufik Sari
Mohammed Chaoui
Chokri Ferkous

# EXTRACTION OF SCORES AND AVERAGE FROM ALGERIAN HIGH SCHOOL DEGREE TRANSCRIPTS

**Abstract** *A system for extracting scores and the average from Algerian high school degree transcripts is proposed. The system extracts the scores and average based on the localization of tables gathering this information; it consists of several stages. After preprocessing, the system locates the tables using ruling-line information as well as other text information. Therefore, the adopted localization approach can work even in the absence of certain ruling lines or the erasure and discontinuity of the lines. After this, the localized tables are segmented into columns and the columns into information cells. Finally, cell labeling is done based on prior knowledge of the table structure, allowing us to identify the scores and the average. Experiments have been conducted on a local dataset in order to evaluate the performances of our system and compare it to three public systems at three levels; the obtained results show the effectiveness of our system.*

**Keywords** localization of areas of interest, table localization and recognition, table understanding, document analysis and recognition, digital archiving, physical and logical structure

**Citation** Computer Science 21(1) 2020: 59–96

## 1. Introduction

The development of information and communication technologies has profoundly changed the working methods by greatly facilitating and accelerating the production, sharing, and storage of digital information. In parallel, the recognition of electronic writing has paved the way for "electronic administration," the dematerialization of business processes, and the production of digital originals. Indeed, after several years of trial and error, the dematerialization is today part of the daily life of the contemporary citizen. It becomes generalized for all areas of business life, administrative authorities, and even those of individuals. Like any new habit, the dematerialization involves important consequences. One of them is that it will now safely keep huge volumes of dematerialized information. This preservation will sometimes be done for a long time. Once the documents are dematerialized, it is also necessary to set up an electronic document management solution so that they can be stored and modified if necessary. This is why digital/electronic archiving has become a real issue for information system managers.

However, electronic archiving and digitization are sometimes confused. Although digitization allows us to reproduce the original document with sufficient quality as an electronic document for long-term preservation and communication, electronic archiving is more general. In addition to the storage, saving, and electronic management of documents, electronic archiving may be defined as ő*all actions aiming to identify, collect, classify, and conserve information for future reference on a suitable and secure support, for the time necessary to satisfy legal obligations or information needs*ś [37]. In fact, electronic archiving may be in the form of office documents, digitized files, data exchanged via remote-procedures, databases, etc. For these, electronic archiving is not just a simple digitization of company documents - it also includes a whole set of processing leading to the storing, protection, understanding, and easy consultation of these documents.

In most cases, the documents to be archived contain several pieces of information, not only text or the written characters but also the type and size of the used font and the writing color in addition to other additional information describing the organization and structuring of the different elements of the document. Without this additional information provided by the structure of a document, the correct reading or localization of the document would be impossible. Therefore, the understanding of a document requires the recognition of its structure in addition to its textual content; then, any information in this document can be located correctly.

Indeed, the digitization and dematerialization of archives is a current trend of Algerian universities. This digitization allows them to preserve the records of students, employees, etc. in electronic form. However, this must be accompanied by methods and techniques facilitating their automatic analysis and search. The present work is part of this process. In this paper, we propose a system for the automatic extraction of scores and average from Algerian high school degree transcripts (we use the acronym "AHST" in the remainder of the paper). An AHST is one of the most important

documents in a student's record, and it is stored in a university's archive. It should be noted that, in Algeria (and French-speaking countries in general), secondary education is completed by passing a final exam called an "exam of Baccalauréat". Successful candidates in the exam obtain a diploma of "Baccalauréat" (which is equivalent to a "high school degree" in Anglo-Saxon countries) in addition to a transcript of a high school degree indicating the scores of the student in each course of the exam in addition to his average in the exam. This transcript is necessary for the registration of students at Algerian universities.

The proposed system analyzes the structure of the AHSTs in order to extract the scores and average of the student. This information is gathered in two tables in the AHSTs: the *scores table* and *average table*. Therefore, the extraction of this information must be preceded by the localization of the two previous tables. Thus, the long-term goal of our work is the development of an electronic archiving system of AHSTs that integrates several functionalities, acquisition, compression, preprocessing, analysis, recognition, retrieval, etc. Without a doubt, this system will facilitate the work of agents in the education offices and archive services of universities.

The remainder of this paper is organized as follows. First, we present some previous works of table detection in document images. Then, we describe the characteristics of the documents that are the subject of the study; namely, AHSTs. After that, we show the architecture of our system while detailing its different stages. Finally, we present the obtained results before concluding.

## 2. Previous works

According to [39], table-detection methods can be divided into two main classes: non-text-analysis methods (using ruling lines) and text-analysis methods. Non-text-analysis methods were the first methods of table detection proposed in the literature. They detect the location of a table by analyzing the ruling lines of the table, and they usually require a phase of preprocessing (like skew correction). The disadvantage of these methods is that they are only effective for some individual tables that are comprised of full horizontal and vertical ruling lines [39]. Text-analysis methods, on the other hand, do not require the presence of any guideline, and they use text information to analyze and recognize the table regions. Although these methods can identify a large number of tables in a document image, they are only effective on single-column or non-complex documents [39].

In this section, we present some of the most important works and methods on table detection while grouping them into non-text-analysis and text-analysis methods.

### 2.1. Non-text-analysis methods

One of the earliest works on identifying tabular regions in document images is that of Watanabe and his coauthors. Thus, Watanabe et al. [43–45] aim for a complete description of the various types of information necessary to interpret a ruled scanned

table. In [43], a method was proposed to recognize the layout structure of table-form documents analytically. This method identifies the line segments directly and interprets the layout structure with the extracted segments. In this method, the mutual relationships among line segments are very important for the structure analysis because the line segments specify the document layout structure. In addition, the individual blocks partitioned by the line segments define the particular domains of meaningful items. Reference [44] presented a method for recognizing tables guided by a generic model of the treated table. The model is described by a graph of cells governing all of the arrays of the same class. Such a technique is not robust for handling broken rulings. The method proposed in [45] aims to recognize the layout structures of multi-kinds of table-form document images. For this purpose, the authors introduced a classification tree to manage the relationships among the different classes of layout structures. The proposed recognition system has two modes: layout knowledge acquisition and layout structure recognition. In the layout knowledge-acquisition mode, table-form document images are distinguished according to this classification tree; then, those structure description trees that specify the logical structures of the table-form documents are generated automatically. In the layout structure-recognition mode, individual item fields in the table-form document images are extracted and classified successfully by searching the classification tree and interpreting the structure description tree.

Laurentini and Viada [20] propose a method for detecting tables where the text and lines are horizontal or vertical. The arrangement of detected lines is compared with that of the text blocks in the same area. Furthermore, the algorithm attempts to add the missing horizontal and vertical lines using the horizontal and vertical projection profiles in order to fully understand the table structure.

In [10], Green and Krishnamoorthy discuss their system of model-based analyses of printed tables. The proposed system uses a top-down approach of analysis by a hierarchical characterization of the physical cells. The goal of the system is to extract and associate parts of a table's image into related segments. For example, it can locate the columns and rows as well as the column and table headings of a table's image. Horizontal lines, vertical lines, horizontal space, and vertical space are used as features to extract the table region. Elementary cell characterization is performed to label individual cells; these labels are matched to a table model such that the relational information can be extracted.

Reference [28] proposes a bottom-up method for recognizing tables within a document. This method is based on the paradigm of graph-rewriting. First, the document image is transformed into a layout graph, whose nodes and edges represent the document entities and their interrelations, respectively. This graph is subsequently rewritten using a set of rules that are designed based on a priori document knowledge and general formatting conventions. The resulting graph provides a logical view of the document content; it can be parsed to provide general format analysis information.

Gatos et al. [6] propose a technique for automatic table detection in document images based on detecting horizontal and vertical ruling lines and progressively identifying all possible types of line intersections. Thus, the proposed technique is comprised of three distinct steps: *(i) image pre-processing* – this mainly involves binarization and image enhancement, skew correction, and marginal noise removal; *(ii) horizontal and vertical line detection* – using a novel method of detection. The latter is mainly based on horizontal and vertical black run processing as well as on image/text area estimation in order to exclude line segments that belong to these areas; *(iii) table detection* – which, in turn, involves two distinct steps: detection of line intersections and table detection (reconstruction).

In [34], the authors propose a practical algorithm for table detection that works with high accuracy on documents with varying layouts (company reports, newspaper articles, magazine pages,...). An open-source implementation of the algorithm is provided as part of the Tesseract OCR engine. The table regions are determined using certain heuristic rules based on analysis of the column layout of the page and column partitions. However, it requires the presence of large text regions (paragraphs) so that the column layouts can be reliably estimated. Evaluation of the algorithm on the document images from a publicly available dataset shows competitive performance as compared to the table-detection module of a commercial OCR system.

In [30], Santosh presents a document information content-extraction technique via graph mining and claims that this technique is well-suited for table processing (i.e., extracting repeated patterns from a table). Real-world users first provide a set of key text fields from the document image that they think are important. These fields are used to initialize a graph where the nodes are labeled with the field names in addition to other features such as size, type, and the number of words; the edges are attributed with relative positioning between them. Such an attributed relational graph (ARG) is then used to mine similar graphs from the document images, which are used to update the initial graph iteratively each time we extract them to produce a graph model. Graph models, therefore, are employed in the absence of users.

Reference [33] presents a method for locating tables and their cells in camera-captured document images. In order to deal with this problem in the presence of geometric and photometric distortions, the authors develop new junction-detection and labeling methods. After the junction detection, the method encodes the connectivity information between the junctions into 12 labels and designs a cost function that reflect the pairwise relationships as well as any local observations. The cost function is minimized via the belief propagation algorithm; this can locate tables and their cells from the inferred labels. Also, in order to handle multiple tables on a single page, the authors propose a table area-detection method based on the well-known recursive X-Y cut. However, they modify the method so that they can also deal with the curved seams caused by geometric distortions.

To increase the efficiency of the non-text-analysis approach, a number of improvements have been proposed such as those of [1,13,17,38]. In [13], the authors propose

a technique to deal with broken rulings, but limited to a small gap. The method, called Box Driven Reasoning (BDR), allows one to robustly analyze the structure of table form documents that include touching characters and broken lines. BDR deals with regions directly, in contrast with other previous methods. Cesarini et al. [1] propose an approach that requires at least two parallel table-structure lines. The documents are described by means of a hierarchical representation that is based on an MXY tree. The presence of a table is hypothesized by searching parallel lines in the MXY tree of the document. The method of Kasar et al. [17] is a learning approach that allows one to detect table regions in document images by identifying the column and row line separators as well as their properties. In [38], Tran et al. propose a method to identify the table region from document images that requires the existence of table-structure lines or table-structure boundaries. The proposed method proceeds as follows. The method starts by recognizing the regions of interest (ROIs) as table candidates. In each ROI, the text components are located and text blocks extracted. After this, it checks all text blocks to determine whether they are arranged horizontally or vertically and compare the height of each text block with the average height. Finally, the ROI is regarded as a table if the text blocks satisfy a series of rules.

## 2.2. Text-analysis methods

When some ruling lines are missing, non-text-analysis methods becomes insufficient; in this case, table detection must be based on other physical elements, and we recourse to text-analysis methods.

Hu et al. [14] describe a technique for detecting tables based on computing an optimal partitioning of a document into some number of tables. A dynamic programming algorithm is given to solve the resulting optimization problem. This high-level framework is independent of any particular table-quality measure and independent of the document medium. Moreover, it does not rely on the presence of ruling lines and has the desirable property that an identical high-level approach can be applied to tables expressed as ASCII text (or any other symbolic format) and those in image form. The authors report on some preliminary experiments using this method to detect tables in both the ASCII text and scanned images, yielding promising results.

In [40], a statistical learning approach is used for the table-recognition problem. After preprocessing, the approach uses word spacing to identify table lines from the set of text-lines. Then, vertically adjacent lines with large gaps and horizontally adjacent words are grouped together to create table entity candidates. Finally, a statistical-based learning algorithm is used to refine the table candidates and reduce false alarms [35].

Pinto et al. [27] describe an approach for locating and extracting tables based on conditional random fields (CRFs) and compares them with hidden Markov models (HMMs). Table extraction using CRFs starts by labeling each line of a document with a tag that describes that line's function relative to the tables. Twelve labels are established, and they are designed by examining a large number of tables in web

documents. After this, a set of features (including white space features, text features, and separator features) are extracted. The final step is the training of the two versions of the CRFs.

In [29], the authors are concerned with the extraction of tables from exchange format representations of very diverse composite documents. They put forward a flexible representation scheme for complex tables based on a clear distinction between the physical layout of a table and its logical structure. Relying on this scheme, they develop a new method for detecting and extracting tables by an analysis of the graphic lines. To deal with tables that lack all or most of the graphic marks, one must focus on the regularities of the text elements alone; a multi-level analysis of the layout of text components on a page is thus completed. A general graph representation of the relative positions of the blocks of text is exploited.

In [24], the authors report a new simple approach for detecting any tables present in document pages. The algorithm relies on the observation that tables have distinct columns, so the gaps between the fields are substantially larger than the inter-word gaps in normal text lines. According to the authors, this deceptively simple observation has led to the design of a simple but powerful table-detection system with low computational cost. Moreover, the mathematical foundation of the approach is also established, including the formation of a regular expression for ease of implementation. Reference [17] announces that this method works only for the Manhattan layout and may fail with complex documents. All lines are removed as a pre-processing step; this can result in inaccurate detections for partially-filled tables.

In [2], a method to detect tables in scanned handwritten documents subject to challenging artifacts and noise is proposed. The text components (machine-print, handwriting) are first separated from the rest of the page using an SVM classifier. Then, the table regions are determined based on a correlation-based approach measuring the coherence between adjacent text lines that may be part of the same table. The resulting page-decomposition problem is solved using dynamic programming. Like other text-based approaches, the detected regions can still have a great number of discrepancies when compared with the ground-truth (even for correct detections) [17].

Harit and Bansal [12] present a new approach for detecting tabular structures present in document images and in low-resolution video images that uses both of the analyses of text and non-text. The proposed technique for table detection is based on identifying the unique table start pattern and table trailer pattern. However, the major contribution is the characterization of the table header and trailer patterns using a set of layout attributes and to formulate the rules that can govern the grouping of adjacent patches. The perceptual attributes used for characterizing the patterns are as follows: the presence of ruling lines; the thickness of the white space separators, the thickness, color, and proximity of the ruling lines; the background color in the divisions formed between the vertical separators; and the characteristics of the text blobs, such as their alignment, font size, and font color. According to the authors, the

proposed approach is tested on a set of document images; it demonstrates improved detection for different types of table layouts (with or without ruling lines).

A query-based approach to selectively extract the tabular information and recognize the table structure from scanned documents is described in [18]. Unlike conventional table-processing paradigms, the authors adopt a client-driven approach where clients provide a query pattern by specifying a set of key fields in the document image. The query pattern is first transformed into an attributed relational graph where each node is described with the features and the edges with the spatial relationships between the nodes. After this, the approach uses a fast graph-matching technique to retrieve other similar graphs from the document image. Furthermore, it collectively analyzes the extracted graphs in order to deduce the overall tabular structure.

In [39], the authors propose a novel method for detecting table regions by using a new shape, which is called Random Rotation Bounding Box. This shape is used for the illustration and description of the table regions. The proposed system consists of the following fundamental steps to detect table zones: binarization; classification of the text and non-text elements in the document image; segmentation of the text elements and classification of the non-text elements into several types; detection of the ruling-line tables and identification of the non-ruling-line tables; and finally, refinement of the regions and labeling. The authors claim that their approach can detect most kinds of tables with a high precision (even when they are skewed).

In [9], Gilani et al. present a deep learning-based method for table detection in document images. The proposed method consists of two major modules: image transformation and table detection. Image transformation is applied in order to separate the content regions in the document, while the table-detection module uses Faster R-CNN as a basic element of a deep network. Faster R-CNN is highly dependent on a combined network that is composed of Region Proposal Networks (RPN) and Fast R-CNN. The authors claim that the proposed method works with high precision on document images with varying layouts, which include documents, research papers, and magazines.

Huynh-Van et al. [16] present a hybrid method for detecting table zones in document images. This method consists of three fundamental steps: classification of the regions, detection of the tables that constitute intersecting horizontal and vertical lines, and identification of the tables made up of only parallel lines.

## 2.3. Survey papers

Finally, a number of surveys on table processing (localization, recognition, understanding, representation, etc.) have appeared over the past several years. We cite, for example, the overviews described in [3–5, 11, 15, 22, 36, 46]. Readers are referred to these papers for more comprehensive information on this topic.

## 3. Characteristics of Algerian high-school degree transcripts

After the physical analysis of the AHSTs of our test corpus distributed over different years (from 1990 to 2017), we noticed that the format of the transcripts changes almost every year; however, the data remains the same {Frame, Heading, Student ID, Student Information, Year, Branch of Study, Scores Table, ... }. Thus, five models of AHSTs exist from 1990 to 2017 (as illustrated in Figure 1).

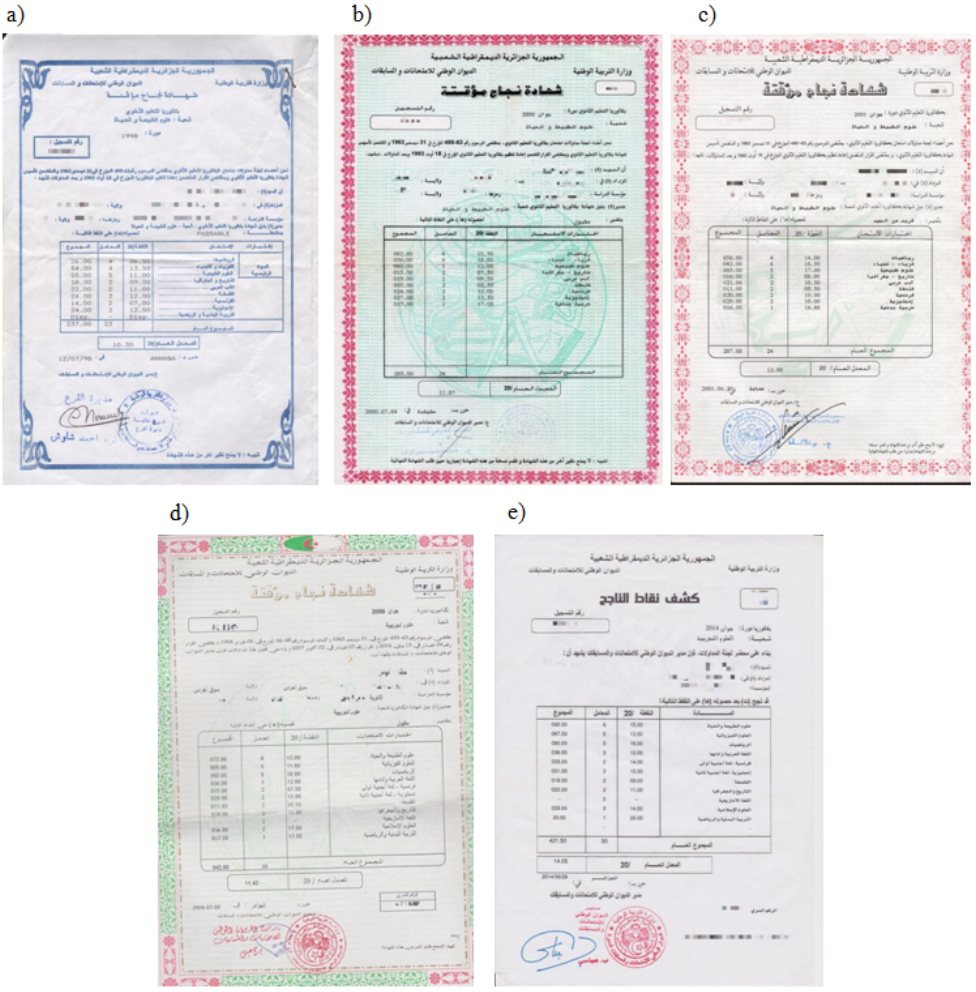



**Figure 1.** Examples of AHSTs of various formats: a) from 1990 to 1999; b) of 2000; c) from 2001 to 2004; d) from 2005 to 2012 and from 2015 to 2017; e) from 2013 to 2014

However, the variations are at several levels; for example, at the level of the paper quality (standard or special paper), the writing font, the language with which

the student's information is written (Arabic or French), the stamp and signature, the text and background colors,... etc.

According to Figure 1, it can be noted that the scores table is usually in the middle of the AHST and the average table is below it. It should also be noted that the frame of these two tables is a simple rectangle or a rectangle with rounded corners. Then, there are several branches of study in high schools in Algeria; the branches are different from each other by the number of courses and the contents of the courses.

Afterwards, we will present the different levels of structures of an AHST in Figure 2.

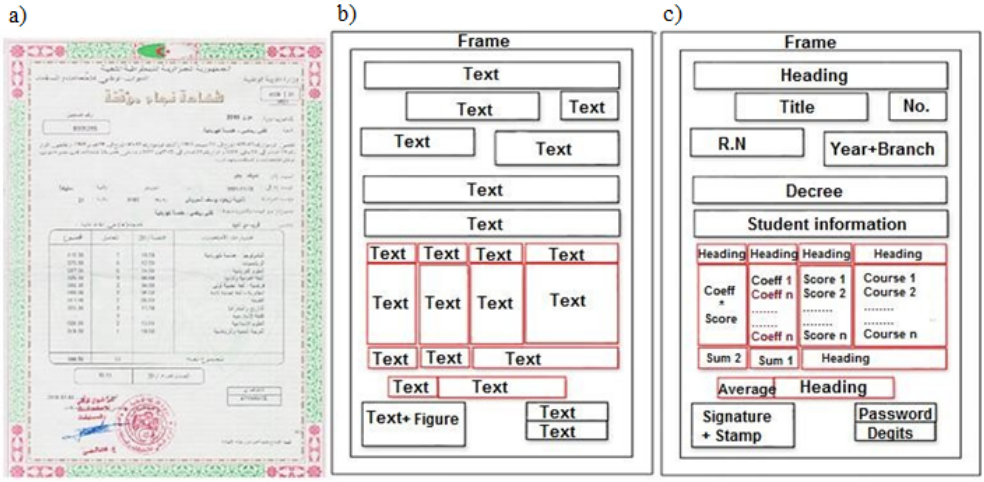



**Figure 2.** Different levels of structures of AHST: a) original AHST; b) physical structure; c) logical structure

## 4. Proposed approach

As we said previously, the scores and average to be located are grouped in two tables in the AHST; thus, the extraction of this information is relative to the localization of the two tables. However, the proposed approach for locating the two tables uses both ruling-line information as well as other textual information such as document structure, text, and spaces between the text portions. Therefore, the adopted localization approach is not dependent on the presence of all of the ruling lines in the table. It can work even in the absence of certain ruling lines of a table or the erasure and discontinuity of lines, even with a large gap between the segments of the line (and even in the presence of noise in the table image).

The conception of our system is done through several steps (which are summarized in Figure 3) that present the main components of the schematic of the proposed system.

This schematic consists of three essential parts:

a) *First part:* Digitization and preprocessing. Digitization allows one to convert the paper document into a pixel image. Preprocessing consists of eliminating the defects related to the scanned image in order to facilitate the next steps,

b) *Second part:* Segmentation. This part allows one to segment the AHST into information zones,

c) *Third part:* Localization of the two tables (of scores and average) according to their physical appearance and extraction of the desired information.

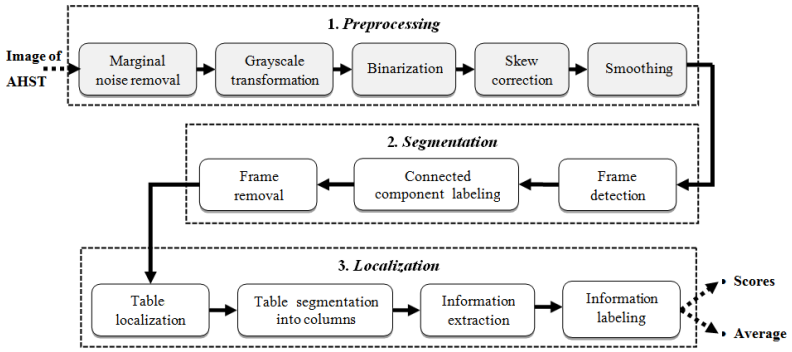



**Figure 3.** Schematic of general process of proposed system

## 4.1. Preprocessing of the AHST

Preprocessing gathers a set of techniques that lead to modifying a digital image in order to improve its quality or extract information from it. Several treatments may be included in the preprocessing. In our system, we have chosen the following steps: marginal noise elimination, grayscale transformation, binarization, skew correction, and noise removal by smoothing.

### 4.1.1. Marginal noise detection and elimination

Marginal noise is formed of the set of shadows that appear in black or a color close to black in the vertical or horizontal margins of an image. It results in many sources such as perforations, skews of the document, the scanning of thick documents, or the edges of the pages in books. The elimination of marginal noise is done as follows:

- Detect the marginal noise at the top, down, right, and left of the image. To find the marginal noise at the top, for example (respectively down, right, and left), we conduct a labeling of the connected components in the first lines of the original image and formed of pixels having a color that is close to black. The marginal noise will consist of all of the labeled connected components. In Figure 4.b, the detected marginal noise is colored in blue.
- Remove the detected marginal noise by coloring all of its pixels with the dominant color of the image (Figure 4.c).


**Figure 4.** Detection and elimination of marginal noise: a) original AHST; b) marginal noise detected in blue; c) marginal noise removed

### 4.1.2. Grayscale transformation

To convert a color image into grayscale, the three values representing the levels of red, green, and blue of each pixel must be replaced by a single value representing the brightness. This transformation is necessary because the binarization method that will be used is applicable only on grayscale images. This transformation is done simply by replacing the color of each pixel of the image by the average of its values of red, green, and blue.

### 4.1.3. Binarization

This allows us to separate the foreground from the background of the image, which produces two classes of pixels: a class representing the background of the image (in white), and another class representing the scene of the image (in black). In fact, a large number of binarization techniques have been proposed in the literature. In our system, we chose to use the method of Sari et al. described in [31], which is a hybrid thresholding method producing good results for images of degraded documents (according to the authors).

This technique runs in two passes. In the first pass, a global thresholding is applied to the image in order to classify the maximum of its pixels (whose gray level is between two global thresholds $T_1$ and $T_2$) into *foreground* or *background*. In the second pass, the remaining pixels are assigned to one of two classes: *foreground* or *background* based on a local analysis.

This method may be summarized in four steps as follows:

- Calculate global threshold $T$ using Otsu's method [26],
- Determine both thresholds $T_1$ and $T_2$, noting that $d_{min}$ is the minimum distance between the average intensity of the foreground and the average intensity of the background. $T_1$ and $T_2$ are given by: $T_1 = T - \frac{d_{min}}{2}$ and $T_2 = T + \frac{d_{min}}{2}$,
- Global thresholding: pixels that have a gray-level greater than $T_2$ are transformed into white, and those whose gray level are less than $T_1$ are colored in black. Noting that $I$ is the grayscale image and $I_b$ is the binarized image (with 0 denoting *black* and 255 denoting *white*), global thresholding using $T_1$ and $T_2$ is summarized by the following equation:

$$I_b(x,y) = \begin{cases} 0, & \text{if } I(x,y) < T_1 \\ 255, & \text{if } I(x,y) > T_2 \\ I(x,y), & \text{otherwise} \end{cases} \tag{1}$$

- Local thresholding of the remaining pixels: for each pixel not yet classified, we locally calculate its new binary values (black or white) obtained by the application of three local thresholding methods; namely, the Niblack [25], Sauvola [32], and Nick [19] methods. However, the local thresholding methods compute a local threshold for each pixel by sliding a square window over the entire image.
  - In Niblack's method, local threshold *T(x, y)* is calculated using mean m and standard deviation $\sigma$ of all pixels in the window. Thus, threshold *T(x, y)* is given by:

$$T(x,y) = m + k \times \sigma \tag{2}$$

    where $k$ is a parameter fixed to be equal to $-0.2$ by the author.
  - Sauvola's local threshold is calculated using the following formula:

$$T(x,y) = m \times \left(1 - k \times \left(1 - \frac{\sigma}{R}\right)\right) \tag{3}$$

    where $R$ is the dynamic range of standard deviation $\sigma$, and $k$ takes positives values in the interval of [0.2, 0.5].
  - Nick's method calculates the local threshold as follows:

$$T(x,y) = m + k \times \sqrt{\left(\frac{\sum p_i^2 - m^2}{NP}\right)} \tag{4}$$

  - By applying the three previous local methods, we obtain three binary images (noting that $I_1$, $I_2$, and $I_3$ are the resulting images). The final binary value of a pixel is that assigned by at least two of the three methods. This is given by the following equation:

$$I_b(x,y) = \begin{cases} 0, & \text{if } \sum_{i=1}^{3} I_i(x,y) \leqslant 255 \\ 255, & \text{Otherwise} \end{cases} \tag{5}$$

#### 4.1.4. Skew correction

Unfortunately, some of our documents are inclined, which makes the localization of the two tables difficult. It was therefore necessary to apply a step of skew correction for inclined documents. However, we used a simple and classic skew-correction method based on projection profile analysis.

The steps of this method may be summarized as follows:

- For each probable inclination angle $a$, do:
  - rotate the image with angle $(a)$,
  - calculate the histogram of horizontal projections of the rotated image (this histogram is displayed in blue in Figure 5a),
  - the projection value corresponds to the maximum value of the histogram.
- Inclination angle $\theta$ of the image is the one with which the value of projection is maximal.
- Rotate the binarized image with angle $(\theta)$.
- Rotation produces transparent pixels in the rotated image. The last step is then to color the transparent pixels of the binary image in white.

Figure 5b presents the final result of the skew correction.

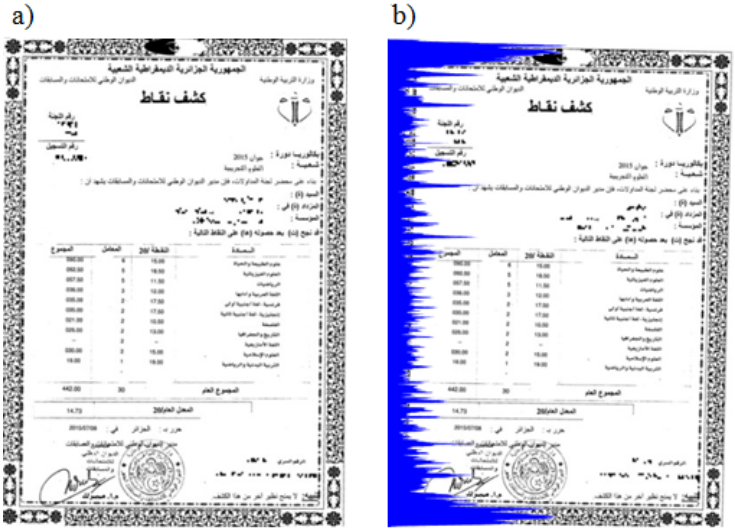



**Figure 5.** Skew correction of skewed AHST: a) skewed AHST with its horizontal projection histogram; b) AHST deskewed of angle 1.4°

### 4.1.5. Smoothing

Binarization and skew correction can introduce noise into an image, which is reflected in particular by the presence of irregularities along the characters' stroke. To overcome this problem, we apply smoothing using the algorithm of [23], which reduces the noise of a binary image by eliminating the isolated pixels on the one hand and by closing the empty holes on the other. This simple technique is based on a statistical decision when the new value of each pixel in a binarized image is calculated based on its initial value and those of its eight neighboring pixels. Thus, a white pixel becomes black if the majority of its neighboring pixels are black, and the same is true if a pixel

is black and the majority of its neighboring pixels are white. Noting that $I$ is the deskewed binary image and $I'$ is the smoothed image, the new value of pixel $(x\ y)$ can be given by the following equation (where 0 denotes black, 255 denotes white, and $s$ is a predetermined threshold):

$$I'(x,y) = \begin{cases} 0, & \text{if } I(x,y) = 255 \text{ and } \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} I(x+i)(y+j) > s \\ 255, & \text{if } I(x,y) = 0 \text{ and } \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} I(x+i)(y+j) > s \times 255 \\ I(x,y), & \text{otherwise} \end{cases} \tag{6}$$

## 4.2. Segmentation

In our system, we apply a mixed segmentation. First, we start by selecting the frame of the AHST. Then, we use a down-top segmentation technique to group pixels with the same properties into connected components. Finally, we eliminate the frame of the AHST because it does not matter in the document.

### 4.2.1. Frame detection

According to the physical study of AHSTs that we conducted, we noted that the transcripts from years 1990-2012 as well as those from 2015, 2016, and 2017 contain different formats of the frame surrounding the document information: the frame is in the form of a rectangle. Therefore, it is formed by a single connected component, framed in the form of a series of stars or other geometric shapes, etc. There are also other AHSTs that do not contain any borders (the 2013 and 2014 transcripts). Figure 6 presents some examples of the formats of existing frames.


**Figure 6.** Some examples of AHST frames: a) 1997 AHST; b) 2000 AHST; c) 2015 through 2017 AHSTs

The method that we applied for frame detection is based on the RunLength Smoothing Algorithm (RLSA). RLSA is used to connect those black pixels separated by fewer than $n$ white pixels according to the horizontal or vertical direction. We proposed not to apply the RLSA algorithm to the entire image but only to the part of the image containing the frame. In addition, based on the physical study of the AHST, it was found that the position of the frame in the document may differ slightly

from one document to another, but the thickness of the frame never exceeds the value (document width/10). However, the frame takes the form of a rectangle formed from four sides (top, bottom, left, right); after a set of tests, we set different values of threshold $n$ for each side of the rectangle. Thus, it has been found that the most suitable value of threshold $n$ for the two horizontal sides (the one at the top of the image and the other at the bottom) is 10% of the image width. In both vertical sides, $n$ is then set to be equal to 20% of the image width for the left side and 30% of the image width for the right side, respectively.

The application of the RLSA algorithm to the parts of the image containing the horizontal (or vertical) sides of the frame leads to connecting the black pixels of the frame that are near the horizontal (or vertical) direction. At the end, the frame becomes composed of a single object. The steps of the applied method are as follows:

- Calculate $l$ = the smoothed image width/10.
- Find the two horizontal sides of the frame. To do this:
  - Divide the image horizontally in sub-images of height equal to $l$. The two horizontal sides of the frame are in the first and last sub-images, respectively.
  - Apply the RLSA algorithm horizontally on the first and last sub-image, separately (Figure 7a) by taking $n = l \times 10\%$, which allows us to connect the black pixels of the frame close to the horizontal axis.
  - Refine the horizontal sides of the frame by coloring all of their pixels black. To do this, we start by calculating the histogram of the horizontal projections of the first and last sub-images (this histogram is in blue in Figure 7b). From this histogram, the starting and ending lines of each of the two sides (shown in dotted red in Figure 7b) are determined. Finally, all pixels between the starting and ending lines of both sides are colored black.
- Find the two vertical sides of the frame. To do this:
  - By dividing the image vertically into sub-images of a width equal to $l$, the two vertical sides of the frame are in the first and last sub-images.
  - Apply the RLSA algorithm vertically on the left of the frame by taking $n = l \times 20\%$ and then on the right by taking $n = l \times 30\%$, which allows us to connect the black pixels of the frame close to the vertical axis (Figure 7a).
  - Refine the vertical sides of the frame by coloring all of their pixels in black. This is done by calculating the histogram of the vertical projections of the first and last sub-images (this histogram is shown in blue in Figure 7b) and by finding the starting and ending columns of each of the two sides (shown in dotted red in Figure 7b). Finally, all pixels between the starting and ending columns of both sides are colored black.
- At the end of the algorithm, the frame becomes composed of a single object (as shown in Figure 7c).

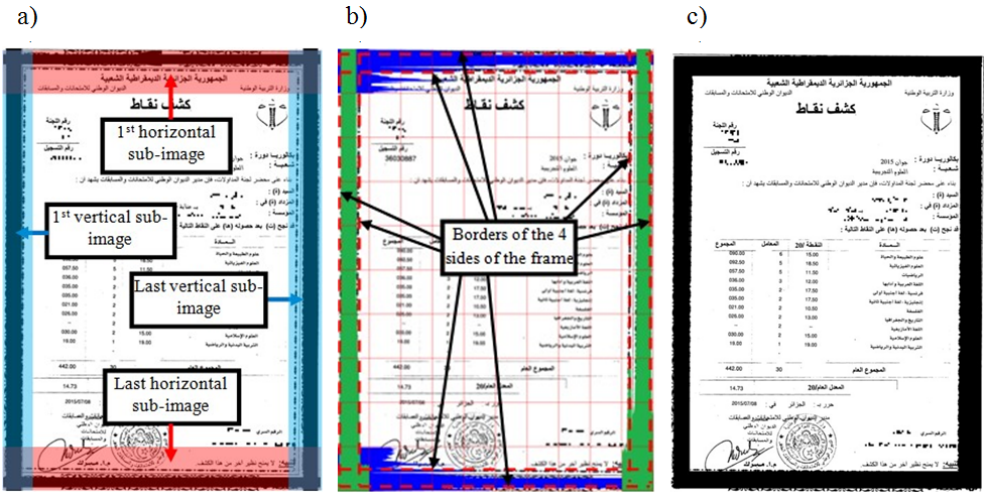



**Figure 7.** Frame detection: a) division into sub-images and application of RLSA on first and last ones; b) horizontal projections in blue and vertical projections in green; c) frame detected

### 4.2.2. Connected component labeling

This consists of grouping all neighboring black pixels into a separate unit; for this, we use the pixel aggregation method. The aim of this labeling for binary images is not to build a list of connected components but to assign each pixel a binary image to its connected component.

### 4.2.3. Frame removal

Simply, the frame represents the largest connected component; so, to eliminate the frame, we just look in the list of connected components that is the connected component that has the largest size and delete it. Indeed, frame removal is done in order to facilitate the detection of the score and average tables, which is the goal of our work.

## 4.3. Localization

The localization of an object consists of determining its position in a document image. This phase is composed of four stages: localization of the two tables (scores table and average table), table segmentation into columns, information detection, and finally the labeling of the detected information and the extraction of the scores and average.

### 4.3.1. Localization of scores table and average table

After the application of the preprocessing steps, we notice in some AHSTs that the border of the scores table and average table is almost removed, which makes it difficult to locate these tables based only on the tables' ruling lines. This is why we use a hybrid

localization method in our work that analyzes both the ruling lines that can exist and the text in order to detect tables. The proposed method is based on projection profile analysis.

The steps included in this method are as follows:

- Calculate the histogram of horizontal projections only for the portion of an image that may contain the score and average tables (the histogram is in blue in Figure 8a). This portion is determined approximately from the physical study of the documents; it is chosen more or less wide to make sure that it contains both tables.
- Look for the horizontal lines of the tables from the horizontal projection histogram; these correspond to the histogram peaks or the lines of the image whose projection values are greater than a certain threshold (red lines in Figure 8a).
- Find the beginning and ending lines of each of the two tables; the two low lines belong to the average table, while all of the other lines are the horizontal lines of the scores table. The starting and ending lines determine the vertical position of the two tables in the image.
- Find the starting and ending columns of each of the two tables. To find the beginning column of the scores table, we run vertically through the region containing the scores table from left to right until we find the first black pixel. The column of this pixel is considered to be the starting column of the scores table. The same principle is used to find the ending column of the scores table, but the run is from right to left. The column of the first black pixel encountered is the ending column of the scores table. The starting and ending columns of the average table can be found in the same way.
- Calculate the histogram of the vertical projections only for the part of image containing the scores table; the same method can be for the average table. The two histograms are displayed in green in Figure 8b.
- Look for the vertical lines of the two tables from the histograms of the vertical projections; these correspond to the histogram peaks or the columns of the image whose projection value is greater than a certain threshold.
- Filtering the list of vertical lines: if the starting column is very close to the first vertical line, it becomes the starting column; otherwise, the starting column is added to the list of vertical lines. Likewise, if the ending column is very close to the last vertical line, the latter becomes the ending column; otherwise, the ending column is added to the list of vertical lines. In addition, we eliminate one of the two very close vertical lines. The same treatment is done for the average table. The remaining vertical lines are displayed in blue in Figure 8b.
- Gather the pixels of all of the horizontal and vertical lines of the scores table into the same connected component representing the scores table, and do the same for the average table. Figure 8c shows the final result where the areas of the two tables are in yellow and the pixels of the table's frames are in red.

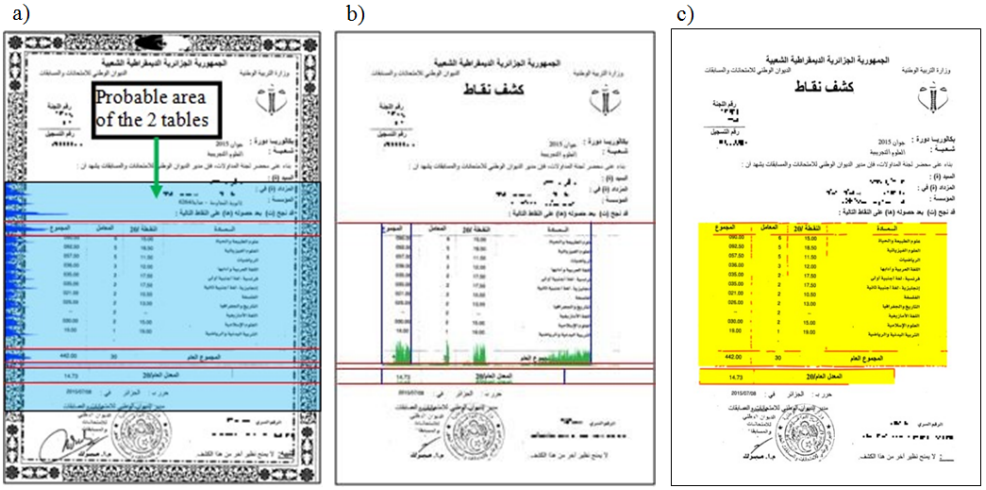



**Figure 8.** Localization of two tables: a) horizontal projection histogram of area containing two tables; b) vertical projection histogram and extraction of vertical lines; c) tables detected

### 4.3.2. Table segmentation into columns

In fact, this segmentation is necessary because each column in the two tables presents a particular type of information; for example, the columns present from right to left in the scores table: the names of the courses, the note in each course, the coefficient of each course, and the sum of the scores.

The segmentation into columns of the scores table is performed using the technique of vertical projections, and hereinafter the followed demarche in our system:

- The first step is to remove the table's border in order to have only the relevant information contained in the table. The scores table area will only contain background pixels in yellow, and the information contained in the table (scores, course names, etc.) is in black.
- Calculate the histogram of the vertical projections of the area containing the scores table.
- The local minima of the histogram or the image columns for which the vertical projection value is below a certain threshold are considered as separation spaces between the table columns. A column of the table is therefore between two successive minima.
- Filtering the found columns. Those columns in the table with few black pixels are removed and merged with the largest adjacent column. Similarly, those columns whose widths are lower than a threshold are considered to be false columns and are merged with the largest adjacent column. Finally, we merge every both very close columns.

The segmentation of the average table is performed in the same previous way, but the scanning is done on the region of the image that contains the average table.

Figure 9a displays the columns of the two tables in green, and the areas of the two tables remain in yellow.

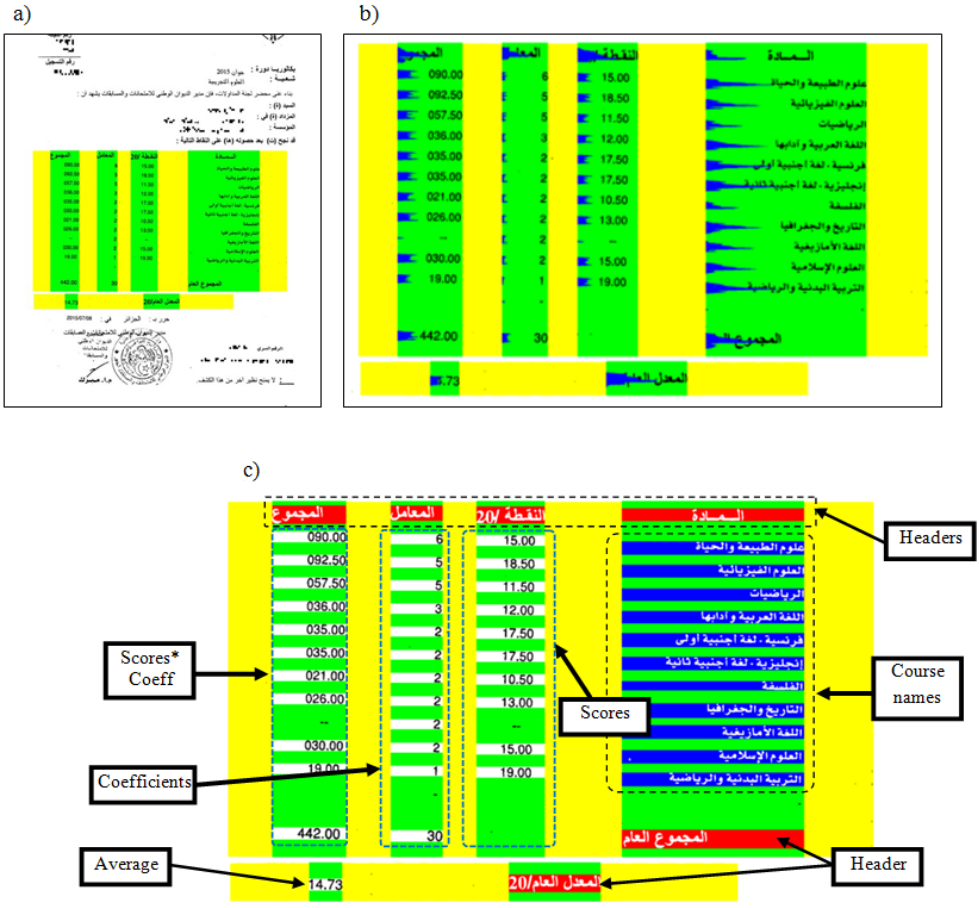



**Figure 9.** Segmentation into columns and extraction of information: a) table segmentation into columns; b) horizontal projection histogram of each column; c) extraction and labeling of information contained in tables

### 4.3.3. Extraction of information from columns

The third step in the localization process is the extraction of information (contained in the scores table and average table) from the segmented columns. Since each piece of information is contained in a cell of a table, the extraction of information is expressed by the localization of the cells of the two tables. In this step, we also use the famous technique of horizontal projections; the application of this technique is done on each column of the two tables separately. The steps followed to extract the information from the scores table are as follows:

- For each column $i$ of the table, do the following:
  - Calculate the histogram of the horizontal projections of Column $I$ (these histograms are in blue in Figure 9.b).
  - Analyze the histogram in order to extract the peaks and empty ranges (the entries in the histogram for which the horizontal projection value is nil). The peaks represent the baselines of the information (notes, course names, ...), while the minima correspond to the separation spaces between the information in Column $i$. A piece of information is therefore between two successive blank ranges that compose a cell.
  - Filtering the list of extracted information in order to eliminate the empty cells and decompose the merged cells. The first filtering consists of eliminating the information of small height, as they correspond to points or parasitic spots and are considered empty cells (because they do not provide important information). The second filtering is to decompose the information of very large height. A piece of information of large height is actually only two or more pieces of information pasted; this results from a bad extraction because of the lack of space between these pieces of information.

The extraction of averages from the average table is done in the same previous manner.

#### 4.3.4. Labeling of detected information and return scores and average

Once all of the information contained in the table is extracted, it remains only to label it based on the a priori knowledge on the arrangement of this information in the table. Thus, the following rules have been established to accomplish this labeling.

- R1: the reading direction is from right to left, so the rightmost column corresponds to the first column.
- R2: if the table consists of five columns, delete the first (because it has no information).
- R3: the first row of the scores table is a header row; all of its cells contain headers.
- R4: the last cell in the first column of the scores table is a header cell.
- R5: all cells except for the first and last of the first column in the scores table contain course names.
- R6: the cells except for the first of the second column in the scores table contain scores.
- R7: the cells except for the first of the third column in the scores table contain course coefficients.
- R8: the cells except for the first of the fourth column in the scores table include scores $\times$ coefficients.
- R9: the average table consists of two cells only.
- R10: the rightmost cell in the average table is a header cell.
- R11: the cell on the left in the average table includes the average.

Figure 9c shows the final extraction and labeling results, where each type of information is displayed in a distinct color: the scores, coefficients, score × coefficients, and average are in white; the headers are in red, and the course names are in blue.

## 5. Experiments and results

In this section, we present the elements introduced in the evaluation of the proposed system: the test dataset used during the experiments, the performance measures employed, and the obtained results. Indeed, we proceeded to the evaluation of our system at three different levels in order to take different aspects of the system into account and to be able to precisely localize the errors: at the table-detection level, at the level of information extraction from the tables, and at the level of functional analysis of the extracted information.

### 5.1. Test dataset

As our system was designed for a particular type of document (an AHST) for which there is no public dataset that we can use to evaluate the performance of our system, we have prepared a local dataset for the test. The prepared dataset is composed of 650 AHSTs scanned at a resolution of 360 dpi. The AHSTs have been chosen to cover all possible variations (see Figure 1). Thus, we have selected AHSTs from all years from 1990 through 2017 of all existing branches of study, all having different formats and structures (with or without frames, the frame shapes, the table locations, the shapes of scores table, etc.). Each AHST contains two tables: a scores table and an average table; as a result, our test collection consists of 1300 tables. Some AHSTs are of good quality, with a high contrast between the foreground and background, the writing is clear, and all of the ruling lines of the tables are present. Other AHSTs are of insufficient quality, lowly contrasted, degraded with different types of noise (stains, holes, transparency effects, etc.) resulting from their poor conservation, and a large part of the tables' ruling lines are deleted. Figure 10 shows some examples of such AHSTs. The AHSTs of our test dataset are distributed as can be seen in Table 1.

**Table 1**
Distribution of AHSTs of test dataset over specific years

| Years | Models | No. of AHSTs |
|---|---|---|
| 1990–1999 | Figure 1a | 230 |
| 2000 | Figure 1b | 25 |
| 2001–2004 | Figure 1c | 95 |
| 2005–2012, 2015–2017 | Figure 1d | 260 |
| 2013–2014 | Figure 1e | 40 |
| | **Sum** | **650** |

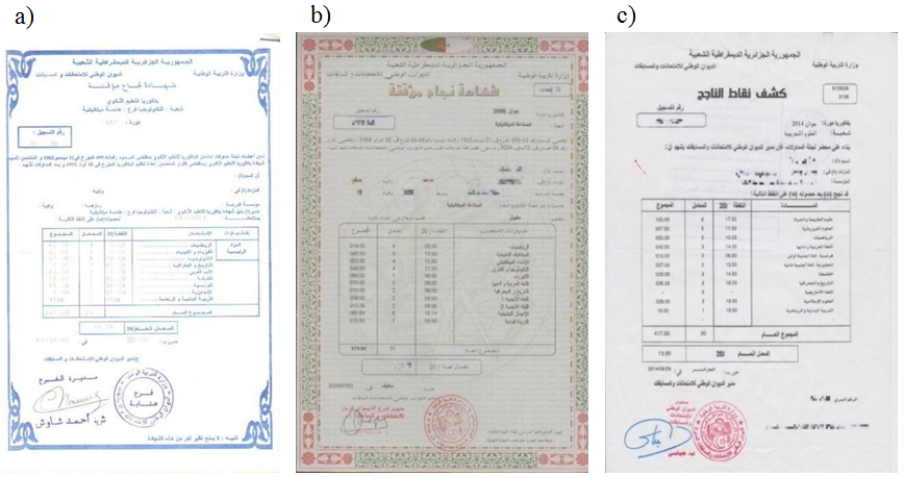



**Figure 10.** Examples of poor quality AHSTs: a) writing erased; b) lowly contrasted; c) stains and humidity

However, we have established a set of characteristics or ground truth data for each AHST from our test collection relating to our subject; namely, table location, table function, state of the ruling lines (present, totally erased, almost erased, little erased), number of columns, column location, column function, number of cells in each column, cell function, and cell position. This ground truth data is saved in an XML file corresponding to each AHST. Figure 11 presents an example of such a file.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <AHST path="D:/AHSTs/1-2002.jpg">
  - <Table id="1" Function="Scores-table" Bounding-Box="1555,310,2624,2109" State-Ruling-
  lines="Present" nb-Cols="5">
    - <Column id="1" Function ="Course-names" Bounding-Box="1555,1457,2624,1937" nb-Cels="11">
      <Cel id="0" Function ="Heading" Bounding-Box="1589,1457,1656,1937" />
      <Cel id="1" Function ="Couse-name" Bounding-Box="1809,1457,1848,1937" />
      <Cel id="2" Function ="Couse-name" Bounding-Box="1858,1457,1899,1937" />
      ..... .....
      <Cel id="10" Function ="Heading" Bounding-Box="2529,1457,2601,1937" />
    </Column>
    - <Column id="2" Function ="Scores" Bounding-Box="1555,1012,2624,1222" nb-Cels="10">
      <Cel id="0" Function ="Heading" Bounding-Box="1591,1012,1639,1222" />
      <Cel id="1" Function ="Score" Bounding-Box="1808,1012,1840,1222" />
      ..... .....
      <Cel id="9" Function ="Score" Bounding-Box="2207,1012,2243,1222" />
    </Column>
    - <Column id="3" Function ="Coefficients" Bounding-Box="1555,698,2624,884" nb-Cels="11">
      <Cel id="0" Function ="Heading" Bounding-Box="1588,698,1650,884" />
      <Cel id="1" Function ="Coefficient" Bounding-Box="1807,698,1839,884" />
      ..... .....
      <Cel id="10" Function ="Sum" Bounding-Box="2554,698,2587,884" />
    </Column>
    - <Column id="4" Function ="Sums" Bounding-Box="1555,372,2624,573" nb-Cels="11">
      <Cel id="0" Function ="Heading" Bounding-Box="1584,372,1655,573" />
      <Cel id="1" Function ="Sum" Bounding-Box="1806,372,1839,573" />
      ..... .....
      <Cel id="10" Function ="Sum" Bounding-Box="2552,372,2586,573" />
    </Column>
  </Table>
  -<Table id="2" Function ="Average-table" Bounding-Box="2639,566,2766,1682" State-Ruling-
  lines="Little-erased" nb- Cols="2">
    -<Column id="1" Function ="Heading" Bounding-Box="2639,1201,2766,1608" nb-Cels="1">
      <Cel id="0" Function ="Heading" Bounding-Box="2671,1201,2744,1608" />
    </Column>
    -<Column id="2" Function ="Average" Bounding-Box="2639,733,2766,849" nb-Cels="1">
      <Cel id="0" Function ="Average" Bounding-Box="2704,733,2737,849" />
    </Column>
  </Table>
</AHST>
```

**Figure 11.** XML file containing ground truth data of AHST of Figure 1c

## 5.2. Evaluation measures

Performance evaluation is done in the basis of recall and precision. In addition, we used other specific measures at each level of evaluation.

### 5.2.1. Table-detection measures

The evaluation of table-detection methods is complex, as it depends on the ground truth and its metrics [39]. However, several performance measures have been used by researchers in the literature for evaluating table detection. These measures vary (as discussed in [34] and [9]) from simple precision- and recall-based measures to more sophisticated measures for benchmarking complete table structure-extraction algorithms.

In the "ICDAR 2013 Table Competition" [8], two measures were used in addition to *recall*, *precision*, and *F1 score*: *completeness* and *purity* (defined in [7]); both are well-known in the context of page segmentation. A region is classified as complete if it includes all sub-objects in the ground truth region; a region is classified as pure if it does not include any sub-objects that are not also in the ground truth region.

Shafai et al. [34] proposed using standard measures for document image segmentation by focusing on the table regions. These measures are *correct detections*, *partial detections*, *over-segmented tables*, *under-segmented tables*, *missed tables*, and *false-positive detections* in addition to *recall*, *precision*, and *F1 score*. These measures have been used by several researchers (for example, in [9] and [39]).

The performance metrics developed in [21] (namely, *correct*, *splitting*, *merging*, *missing*, *false alarms*, and *spurious*) have also been employed by researchers (for example, in [1, 41, 42, 47]) for the evaluation of table-detection algorithms.

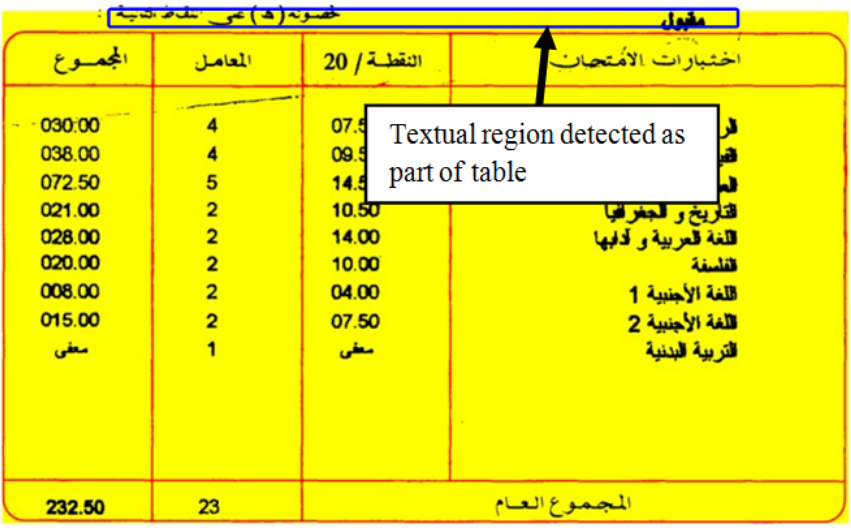
In our experiments, we used ten evaluation measures. Nine of these measures are those proposed in [34] and cited above that quantitatively evaluate different aspects of table-detection algorithms. These measures seem to us to be more adequate, as they may be quantified by explicit formulas. In addition, they are more detailed than those used in the ICDAR 2013 competition, and they make it possible to precisely locate detection errors.

Indeed, only one aspect is not taken into account by the preceding measures; the case where the region of a detected table corresponds to a region of a ground truth table in addition to a textual region. This table is considered detected but not pure. An example of such a case is when the legend of a table is detected as part of this table (Figure 12). To take this case into account, we added a tenth measure: *non-purity*.

Note that $G_i$ the bounding box of the $i^{th}$ ground-truth table and $D_j$ the bounding box of the $j^{th}$ detected table by a table-detection algorithm. The amount of overlap between the two bounding boxes is defined by [34] as:

$$A(G_i, D_j) = \frac{2|G_i \cap D_j|}{|G_i| + |D_j|} \tag{7}$$

where $|G_i \cap D_j|$ represents the area of intersection of the two zones, and $|G_i|$ and $|D_j|$ represent the individual areas of the ground-truth and detected tables. Amount of area overlap $A$ will vary between zero for non-overlapped tables and one when the two tables match perfectly.




| المجمـوع | المعامل | النقطـة / 20 | اختبارات الامتحان |
|---|---|---|---|
| 030.00 | 4 | 07.5 | |
| 038.00 | 4 | 09.5 | |
| 072.50 | 5 | 14.5 | |
| 021.00 | 2 | 10.50 | التاريخ و الجغرافيا |
| 028.00 | 2 | 14.00 | اللغة العربية و آدابها |
| 020.00 | 2 | 10.00 | الفلسفة |
| 008.00 | 2 | 04.00 | اللغة الأجنبية 1 |
| 015.00 | 2 | 07.50 | اللغة الأجنبية 2 |
| معفى | 1 | معفى | التربية البدنية |
| 232.50 | 23 | المجموع العام | |

**Figure 12.** Detection of non-pure table – table area is in yellow

The details of these measures are as follows:

1) *Correct Detections:* this is the number of detected tables that have a large overlap ($A \geqslant 0.9$) with one of the ground truth tables and that do not contain a textual region. We have added this last constraint to ensure that the correctly detected tables are pure.
2) *Partial Detections:* this is the number of ground truth tables that have a partial overlap ($0.1 < A < 0.9$) with one of the detected tables.
3) *Over-Segmented Tables:* this is the number of ground-truth tables that have a major overlap ($0.1 < A < 0.9$) with more than one of the detected tables. This indicates that different parts of the ground-truth table were detected as separate tables.
4) *Under-Segmented Tables:* this is the number of ground-truth tables that have a major overlap ($0.1 < A < 0.9$) with one of the detected tables; however, that detected table also has major overlaps with other ground truth tables. This means that more than one table was merged during detection and reported as a single table.
5) *Missed tables:* this is the number of ground truth tables that do not have a major overlap with any of the detected tables ($A \leqslant 0.1$). These tables are regarded to be missed by the detection algorithm.

6) *False Positive Detections:* this is the number of detected tables that do not have a major overlap with any of the ground-truth tables ($A \leqslant 0.1$). These tables are regarded as false-positive detections since the system mistook some non-table region as a table.
7) *Non-pure detections:* this is the number of detected tables that have a major overlap with one of the ground truth tables but also contain textual regions.
8) *Area Precision:* this measure summarizes the global performance of the table-detection algorithm by measuring the percentage of detected tables that actually belong to the table regions of a ground truth image. The precision is calculated by using the following formula:

$$Precision_{Area} = \frac{Area\ of\ Ground\ truth\ regions\ in\ Detected\ regions}{Area\ of\ all\ Detected\ table\ regions} \tag{8}$$

9) *Area Recall:* this measure evaluates the percentage of the ground-truth table regions that were marked as detected table regions. The formula for calculating recall is as follows:

$$Recall_{Area} = \frac{Area\ of\ Ground\ truth\ regions\ in\ Detected\ regions}{Area\ of\ all\ Ground\ truth\ table\ regions} \tag{9}$$

10) *F1 Score:* this considers both precision and recall to compute the accuracy of the methodology. It is calculated by the following:

$$F1 = \frac{2 \times Precision_{Area} \times Recall_{Area}}{Precision_{Area} + Recall_{Area}} \tag{10}$$

#### 5.2.2. Cell extraction measures

Each piece of information is contained in a cell of the table; the evaluation of information extraction from a table is, therefore, the evaluation of cell localization, which also named the evaluation of cell structure recognition. Thus, we used the evaluation strategy proposed in [7] for evaluating cell structure recognition, which is the one employed in the context of the "ICDAR 2013 Table Competition." However, the evaluation is performed by comparing the cell structure obtained using our approach with the ground truth cell structure. This comparison is done by generating a list of all adjacency relationships between each content cell and its nearest horizontal and vertical neighbors as well as to compare them with the ground truth adjacency relationships in terms of *recall*, *precision*, and *F1 score* thereafter. Note that no adjacency relationships are generated between blank cells or a blank cell and a content cell. An adjacency relationship is a tuple containing the textual content of both cells, the direction, and the number of blank cells (if any) in between.

In fact, this evaluation strategy seems adequate because it does not take blank cells into account; thus, it agrees with our vision of structuring tables. The second reason is that, as reported in [7] and [8], this evaluation strategy provides a simple and repeatable way to fairly account for a wide variety of errors in table structure

recognition (e.g., extra blank columns, split rows, undetected column-spans, etc.). In addition, using neighborhoods makes the comparison invariant to the absolute position of the table (e.g., if everything is shifted by one cell).

1.
$$Recall = \frac{Correct\ adjacency\ relationships}{Total\ adjacency\ relationships} \tag{11}$$

2.
$$Precision = \frac{Correct\ adjacency\ relationships}{Detected\ adjacency\ relationships} \tag{12}$$

3.
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

#### 5.2.3. Functional analysis measures

Functional analysis means determining the function of the extracted cells and their abstract logical relationships [8]. In our evaluation, we are limiting ourselves to determining the cell functions. Indeed, as described in [7], a table's functional representation cannot typically be fully discovered from the layout alone. Domain-specific knowledge is required to be able to assign functions.

However, the evaluation of functional analysis is done only in terms of the precision of the assignment of functions to the detected cells, as the system assigns a function (label) to each cell; therefore, the recall is 100%. To do this, the functions assigned by the system are compared with pre-established ground truth functions (which are integrated into the xml files corresponding to the AHSTs).

*Precision:* to highlight the different functional analysis errors that can be produced, we evaluate the assignment of each function separately and calculate the accuracy of the functional analysis as the average of the precisions of the assignment of each function:

$$Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{14}$$

where $n$ is the number of existing functions, and $Precision_i$ is the precision of assigning function $i$; this is calculated by the following:

$$Precision_i = \frac{Correctly\ assigned\ functions_i}{Assigned\ functions_i} \tag{15}$$

### 5.3. Obtained results

The proposed system was applied to all images in the test dataset in order to evaluate its performance on actual AHSTs; the obtained results are compared to the ground truth data using the evaluation measures previously presented. Thus, the evaluation was done on each AHST separately to be able to individually analyze the obtained results and determine the success and failure cases of the proposed algorithm. In

addition, we can globally quantify the overall average performances of our system and compare them with other systems on the whole AHST dataset.

Since the tests were conducted on a local and non-public dataset as that which was used in "the ICDAR 2013 Table Competition," we did not compare our system to the methods that were participated in the ICDAR 2013 competition nor to other methods from the literature using the competition dataset; therefore, it would not be a fair comparison with our approach. It is for this reason that we limited integrating a table-detection module to the comparison with public recognition engines. Thus, we used two commercial engines (Abby Fine Reader 14 Corporate and OmniPage 18 Professional) and an open source one (the table detector of the Tesseract 3.0 OCR system) [34].

In this section, we present the results obtained from the different systems compared at each evaluation level.

### 5.3.1. Table-detection evaluation

The evaluation results of the table detection obtained using our approach as well as the other public recognition systems on our dataset (consisting of 650 AHSTs containing 1300 tables) in terms of the measures described above are summarized in Table 2.

**Table 2**
Evaluation results of table detection

| | Our approach | Abby FineReader | OmniPage | Tesseract |
|---|---|---|---|---|
| Corrects | 1279 | 774 | 714 | 341 |
| Partials | 8 | 89 | 145 | 288 |
| Over-segmented | 0 | 51 | 62 | 139 |
| Under-segmented | 0 | 14 | 39 | 41 |
| Missed | 0 | 169 | 102 | 427 |
| False positives | 0 | 278 | 205 | 58 |
| Non-pures | 13 | 73 | 117 | 95 |
| Recall | 98.385% | 69.538% | 64.923% | 40.231% |
| Precision | 98.385% | 70.516% | 65.694% | 49.447% |
| F1 | 98.385% | 70.024% | 65.306% | 44.365% |

As we can see from Table 2, our approach achieved the best performances in terms of the correct detection, recall, accuracy, and F1 score, and it overcame the other systems. However, our system was able to perfectly detect 1279 of the 1300 tables present in the dataset, which showed that the system could locate table regions with a precision of 98.38%, a recall of 98.38%, and a high compromise between recall and precision (F1-score = 98.38%). Also, the performances accomplished by our system are far above those obtained using the other systems. The system ranked second is Abby FineReader, which allowed for the correct detection of 774 tables with a recall of 69.54% and precision of 70.52%. OmniPage was ranked third by correctly locating 714 tables with a recall of 64.923% and precision of 65.69%. Finally, Tesseract only

located the regions of 341 tables among the 1300 existing tables; it showed a recall of 40.23% and precision of 49.45%. On the other hand, our method did not generate false positives, over-segmentations, nor under-segmentations (unlike with the other systems). The table-detection errors made by our system concern only the partial detection of 8 tables and localization of 13 non-pure ones where the text line that was just above the table was considered to be part of the table. Examples of such errors are illustrated in Figure 13 and Figure 12, respectively.

| المجموع | المعامل | النقطة /20 | المـــادة |
|---|---|---|---|
| 102.00 | 6 | 17.00 | علوم الطبيعة والحياة |
| 082.50 | 5 | 16.50 | العلوم الفيزيائية |
| 087.50 | 5 | 17.50 | الرياضيات |
| 043.50 | 3 | 14.50 | اللغة العربية وآدابها |
| 026.00 | 2 | 13.00 | فرنسية - لغة أجنبية أولى |
| 020.00 | 2 | 10.00 | إنجليزية - لغة أجنبية ثانية |
| 015.00 | 2 | 07.50 | الفلسفة |
| 026.00 | 2 | 13.00 | التاريخ والجغرافيا |
| -- | 2 | -- | اللغة الأمازيغية |
| 025.00 | 2 | 12.50 | العلوم الإسلامية |
| | 1 | | التربية البدنية والرياضية |
| | - | | - |
| 427.50 | 29 | المجموع العام | |

**Figure 13.** Partial detection of table – table area is in yellow

In fact, the good results achieved by our system on real AHST images are reasonable, as our system has been specifically designed to process AHSTs (unlike the other systems that are generic recognition engines). Therefore, our system takes the structure and the characteristics of the AHSTs into account (language, writing orientation, approximate position of tables, etc.) to reach the correct detection of the tables. Then, the strategy adopted by our system (which does not rely only on the presence of ruling lines or the text or spaces between text analysis but on a combination of all) has given more flexibility and adaptation to our system to recognize table regions, even in the absence of some of their ruling lines or the erasure and discontinuity of lines (and even in the presence of noise). In addition, the prior knowledge of the number of tables in each AHST and their spatial arrangement allowed us to avoid errors of missed detections, over-segmentations, under-segmentations, and false positives; it also allowed us to minimize the errors of partial and non-pure detections. Another strength point of our system is its ability to process skewed and noisy AHSTs thanks to the skew-correction and noise-elimination modules (cases where the other systems failed).

In addition, the errors occurred by our system are of two types: partial detection errors and non-pure detection errors. Non-pure detections are usually produced when the table and the text line that is just above it touch each other because of the presence of noise in this area, a printing problem in the original document, or as a result of preprocessing steps. In this case, the text line is detected as part of the table. For partial detections, they are caused by the poor quality of the original AHSTs, leading to the presence of too much noise in the table area or the erasure of all ruling lines and several pieces of information in the table. In this case, a detection technique based on horizontal and vertical projections will not be able to correctly locate the entire area of the table.

Regarding the other systems, the number of correct detections obtained with Abby FineReader and OmniPage is reduced, and a large number of detection errors occurred (missed detections, false positive detections, etc). This is because these commercial systems fail to detect tables in complex layout documents that contain page borders, have multiple scripts, and are composed of large white spaces. In accordance with [17], the Tesseract system (which relies only on text information) also resulted in several detection errors, especially partial detections and missed detection errors. This is justified by the fact that it is difficult to localize all types of tables present in the dataset using only text information. This is also reflected in the relatively low values of average precision and recall.

An analysis of the individual results shows that the table-detection errors relate more precisely to the scores tables and that the average tables have been located perfectly. This result can be justified by the fact that the average table in all of the AHSTs is a table of a simple structure and that we have a priori knowledge about its location. Thus, this table is composed of a single row, which is itself separated into two cells: one is a header cell, and the other contains the general average of the student in the "Baccalauréat" exam (and it is just below the scores table, which makes its localization easier). The structure of the scores table, on the other hand, may differ from one AHST to another; there are tables with four columns and others with five, and some have differing number of rows, the presence or absence of all ruling lines, etc. In addition, the presence of noise or erasures in the area that may contain the scores table (which is a large area) may complicate the detection of this table and produce erroneous results by widening or decreasing the detected area of the table.

In the final analysis, testing on a real-image dataset such as ours is the confident way to evaluate the performance of table-detection systems. As a result, any system with good performance on this dataset is considered effective. For example, although they are well-known and widely used commercial recognition systems, the Abby FineReader and OmniPage systems' performances on our image dataset are weak; therefore, they are not effective for table detection in AHSTs.

### 5.3.2. Cell-extraction evaluation

After the table-level evaluation, we evaluated our system for cell extraction and compared its performance to that of the Abby Fine Reader 14 Corporate and OmniPage

18 Professional engines. However, for the evaluation to be fair and unaffected by the table-localization errors, it must be done for the cell-detection module in isolation. To do this, we evaluated the cell detection only on the correctly detected tables; this was compared for the different systems. The assessment was made in terms of the recall, precision, and F1-score as described above. The evaluation results obtained by our system, Abby Fine Reader 14 Corporate, and OmniPage 18 Professional are summarized in Table 3.

**Table 3**
Evaluation results of cell extraction

| Method | No. of tables | Recall | Precision | F1 |
|---|---|---|---|---|
| Our approach | 1279 | 98.62% | 99.17% | 98.89% |
| Abby Fine Reader | 774 | 72.09% | 83.45% | 77.35% |
| OmniPage | 714 | 69.73% | 84.66% | 76.47% |

The evaluation results summarized in Table 3 show that our system also overcame the other systems in this second level of evaluation and was able to extract the cells from the correctly detected tables, with a recall of 98.62% and precision of 99.17%. The performances of the compared commercial engines (expressed in terms of recall and precision) increased slightly as compared to their performances when detecting tables.

Provided that the tables were correctly detected, the good performances obtained by our cell-extraction module are especially due to the fact that the localization of a table's rows and columns is not sensitive to the presence or absence of the ruling lines that divide the table into rows and columns. On the other hand, the segmentation of a table into columns and then into rows is done by using an analysis of the horizontal and vertical projection profiles, which allows us to detect the beginning and ending of each column or row (even in the absence of all horizontal and vertical ruling lines). In addition, the filtering step employed (which relies on a statistical analysis) allowed us to reduce cell-detection errors by eliminating the blank cells and decomposing the merged ones. Regarding the other systems, although the details of the algorithms used by these commercial engines are not known to the public, it seems to us from the individual analysis of each table that they rely much more on the analysis of ruling lines and, therefore, fail when these lines (or parts of them) are missing. The absence of a vertical line, for example, leads to the merging of several couples of horizontally adjacent cells.

The individual cell-detection results for each table separately show that the number of tables for which the cells were perfectly extracted using our system (F1-score = 100%) were 1244 among the 1279 tables tested, 4 tables for which the detection precision was between 79% and 82%, and the rest had precision levels from 95% to 97%.

In fact, the cell-localization errors encountered in some tables were caused by the fact that certain information (notes, for example) in these tables touch other information or ruling lines in the table containing this information; therefore, they

are considered to be part of the table border when labeling the connected components and not as full components. This kind of touching usually results from problems with printing the original document or the presence of too much noise in the table area, which leads to linking two or more components together. Figure 14 shows an example of such a case where two scores were not detected because they were pasted to the border of the scores table and are colored red (the same color of the border) during the table detection.


**Figure 14.** Example of cell-localization problem: a) two scores pasted to border; b) erroneous result of cell detection

### 5.3.3. Functional analysis evaluation

The last level of evaluation is the functional level. Here, we evaluate the performance of our system to assign functions or assign a sense to the detected cells and finally extract the scores and average (which is the ultimate goal of our work). Thus, we have six functions for the extracted cells: “Header”, “Course name”, “Score”, “Coefficient”, “Sum” and ”Average”. Since our objective is the extraction of the scores and average, we are much more interested in those cells whose functions are “Score” or “Average”.

The evaluation was performed on the correctly located cells of the 1279 tables detected in terms of the function of assignment precision. However, the functions assigned by the system were compared to pre- established ground truth functions; the obtained results may be presented in the confusion matrix of Table 4. The function assignment precisions are summarized in Table 5.

From Tables 4 and 5, we note that the average precision of the functional analysis completed by our system on the 1279 tables correctly detected is 99.766%. Thus, the averages were extracted perfectly (precision = 100%) for all 650 AHSTs of the test dataset; the scores were labeled with a precision of 99.678%.

In fact, our system succeeded to a precision of 100% to assign the functions to the cells of the 1244 tables for which the cells were extracted perfectly. Functional analysis errors were produced in the other tables. However, this high precision in the functional analysis is due to the robust structure of the tables and the a priori knowledge of this structure, which made it easy to correctly assign the functions to the extracted cells.

**Table 4**
Functions assigned by our system *vs* ground truth functions

| No. of functions assigned by our system | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| No. of ground truth functions | | Header | Course name | Score | Coefficient | Sum | Average | total |
| | Header | 3776 | 3 | 2 | 4 | 0 | 0 | 3785 |
| | Course name | 17 | 6203 | 0 | 0 | 0 | 0 | 6220 |
| | Score | 20 | 0 | 6200 | 0 | 0 | 0 | 6220 |
| | Coefficient | 19 | 0 | 0 | 6827 | 0 | 0 | 6846 |
| | Sum | 20 | 0 | 0 | 0 | 6826 | 0 | 6846 |
| | Average | 0 | 0 | 0 | 0 | 0 | 650 | 650 |

**Table 5**
Function assignment precisions

| Function | Precision [%] |
|---|---|
| Header | 99.762 |
| Course name | 99.727 |
| Score | 99.678 |
| Coefficient | 99.722 |
| Sum | 99.708 |
| Average | 100 |
| Average precision | 99.766 |

The errors committed in the other tables were caused by the bad detection of some cells in these tables. Thus, since the assignment of functions to cells is based on the number and positions of these cells, the absence of a cell in the table may influence the functions assigned to its adjacent cells. For example, if the header of the first column in the scores table is not detected, the first course name will be labeled as a header. Similarly, if a noise is detected as a cell before the header of the second column, for example, this noise will be considered to be a header and the header as a score.

## 6. Conclusion

The present work is made into the context of the electronic archiving and understanding of AHSTs. The objective was to analyze digitized AHSTs in order to extract important information (namely, the scores and average of the students), which could allow the sharing, retrieval, and reuse of these AHSTs.

Since the information to be extracted is gathered in two tables in the AHST (scores table and average table), the extraction of this information is relative to the localization of the two tables. Therefore, we realized a system that integrates various processings that leads to the localization of the scores and average tables from AHSTs

of different styles and formats as well as extracts the desired information in order to offer an easier handling of the data: archiving, indexing, searching, etc. Thus, the adopted localization method does not only rely on an analysis of the table ruling lines but also on text information, which makes it applicable even with the absence of certain ruling lines. The segmentation into columns is then performed, and the cells are detected from the segmented columns.

Finally, the labeling of the detected cells is done based on prior knowledge of the table structure, and the notes and averages are extracted.

Experiments were performed on a local test dataset consisting of 650 images of AHSTs in order to evaluate the performances of our system at three different levels; table-detection, cell-extraction, and functional-analysis. A comparison with public systems has also been completed, and the obtained results show the reliability of the proposed system.

## References

[1] Cesarini F., Marinai S., Sarti L., Soda G.: Trainable table location in document images. In: *16th International Conference on Pattern Recognition*, vol. 3, pp. 236–240, 2002.

[2] Chen J., Lopresti D.: Table detection in noisy offline handwritten documents. In: *International Conference on Document Analysis and Recognition*, pp. 399–403, 2011.

[3] Coüasnon B., Lemaitre A.: Recognition of tables and forms. In: Doermann D., Tombre K. (eds.), *Handbook of Document Image Processing and Recognition*, pp. 647–677, Springer, London, 2014.

[4] Embley D.W., Hurst M., Lopresti D., Nagy G.: Table-processing paradigms: a research survey, *International Journal on Document Analysis and Recognition*, vol. 8(2–3), pp. 66–86, 2006.

[5] Embley D.W., Tao C., Liddle S.W.: Automating the extraction of data from HTML tables with unknown structure, *Data & Knowledge Engineering*, vol. 54(1), pp. 3–28, 2005.

[6] Gatos B., Danatsas D., Pratikakis I., Perantonis S.J.: Automatic table detection in document images. In: *International Conference on Advances in Pattern Recognition and Image Analysis, ICAPR 2005*, pp. 609–618, 2005.

[7] Göbel M., Hassan T., Oro E., Orsi G.: A methodology for evaluating algorithms for table understanding in PDF documents. In: *ACM Symposium on Document Engineering*, pp. 45–48, 2012.

[8] Göbel M., Hassan T., Oro E., Orsi G.: ICDAR 2013 table competition. In: *12th International Conference on Document Analysis and Recognition*, pp. 1449–1453, 2013.

[9] Gilani A., Qasim S.R., Malik I., Shafait F.: Table Detection using Deep Learning. In: *14th IAPR International Conference on Document Analysis and Recognition*, vol. 1, pp. 771–776, 2017.

[10] Green E.A., Krishnamoorthy M.S.: Model-Based Analysis of Printed Tables. In: *International Conference on Document Analysis and Recognition*, pp. 214–217, 1995.

[11] Handley J.C.: Document recognition. In: E.R. Doughert, (ed.) *Electronic Imaging Technology, chapter 8*, pp. 289–316. SPIE-The International Society for Optical Engineering, 1999.

[12] Harit G., Bansal A.: Table detection in document images using header and trailer patterns. In: *8th Indian Conference on Computer Vision, Graphics and Image Processing*, p. 62, 2012.

[13] Hori O., Doermann D.S.: Robust table-form structure analysis based on box-driven reasoning. In: *International Conference on Document Analysis and Recognition*, pp. 218–221, 1995.

[14] Hu J., Kashi R.S., Lopresti D.P., Wilfong G.: Medium-independent table detection. In: Lopresti D.P., Zhou J. (eds.), *Proceedings of Document Recognition and Retrieval VII*, vol. 3967, pp. 291–302, International Society for Optics and Photonics, SPIE, 2000.

[15] Hurst M.: *The Interpretation of Tables in Texts*. PhD thesis, University of Edinburgh, 2000.

[16] Huynh-Van T., Nguyen-An K., Khanh T.L.B., Yang H.J., Tran T.A., Kim S.H.: Learning to detect tables in document images using line and text information. In: *2nd International Conference on Machine Learning and Soft Computing*, pp. 151–155, 2018.

[17] Kasar T., Barlas P., Adam S., Chatelain C., Paquet T.: Learning to detect tables in scanned document images using line information. In: *International Conference on Document Analysis and Recognition*, pp. 1185–1189, 2013.

[18] Kasar T., Bhowmik T.K., Belaïd A.: Table information extraction and structure recognition using query patterns. In: *13th International Conference on Document Analysis and Recognition*, pp. 1086–1090, 2015.

[19] Khurshid K., Siddiqi I., Faure C., Vincent N.: Comparison of Niblack inspired Binarization methods for ancient documents. In: *SPIE 7247, Document Recognition and Retrieval XVI*, pp. 267–275. San Jose, California, United States, 2009.

[20] Laurentini A., Viada P.: Identifying and understanding tabular material in compound documents. In: *11th IAPR International Conference on Pattern Recognition, vol. II. Conference B: Pattern Recognition Methodology and Systems*, pp. 405–409, 1992.

[21] Liang J.: *Document Structure Analysis and Performance Evaluation*. Phd thesis, University of Washington, Seattle, 1999.

[22] Lopresti D., Nagy G.: A tabular survey of automated table processing. In: A.K. Chhabra, D. Dori (eds.), *International Workshop on Graphics Recognition*, *Lecture Notes in Computer Science*, vol. 1941, pp. 93–120. Springer, Berlin, Heidelberg, 1999.

[23] Mahmoud A.S.: Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding, *Pattern Recognition*, vol. 27(6), pp. 815–824, 1994.

[24] Mandal S., Chowdhury S.P., Das A.K., Chanda B.: A simple and effective table detection system from document images, *International Journal on Document Analysis and Recognition*, vol. 8(2), pp. 172–182, 2006.

[25] Niblack W.: *An Introduction to Digital Image Processing*. Strandberg Publishing Company, Birkeroed, Denmark, 1985.

[26] Otsu N.: A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9(1), pp. 62–66, 1979.

[27] Pinto D., McCallum A., Wei X., Croft W.B.: Table extraction using conditional random fields. In: *26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 235–242, 2003.

[28] Rahgozar M.A., Cooperman R.: A graph-based table recognition system. In: L.M. Vincent, J.H. Jonathan (eds.), *Document Recognition III*, vol. 2660, pp. 192–203, International Society for Optics and Photonics, SPIE, 1996.

[29] Ramel J.Y., Crucianu M., Vincent N., Faure C.: Detection, extraction and representation of tables. In: *7th International Conference on Document Analysis and Recognition*, pp. 374–378, 2003.

[30] Santosh K.C.: g-DICE: Graph mining based document information content exploitation, *International Journal on Document Analysis and Recognition*, vol. 18(4), pp. 337–355, 2015.

[31] Sari T., Kefali A., Bahi H.: Text extraction from historical document images by the combination of several thresholding techniques, *Advances in Multimedia*, vol. 2014, p. 11, 2014.

[32] Sauvola J., Pietikäinen M.: Adaptive document image binarization, *Pattern Recognition*, vol. 33(2), pp. 225–236, 2000.

[33] Seo W., Koo H.I., Cho N.I.: Junction-based table detection in camera-captured document images, *International Journal on Document Analysis and Recognition*, vol. 18(1), pp. 47–57, 2015.

[34] Shafait F., Smith R.: Table Detection in Heterogeneous Documents. In: *9th IAPR International Workshop on Document Analysis Systems*, pp. 65–72, 2010.

[35] Shahab A., Shafait F., Kieninger T., Dengel A.: An open aproach towards the benchmarking of table structure recognition systems. In: *9th IAPR International Workshop on Document Analysis Systems*, pp. 113–120, 2010.

[36] e Silva A.C., Jorge A.M., Torgo L.: Design of an end-to-end method to extract information from tables, *International Journal on Document Analysis and Recognition*, vol. 8(2–3), pp. 144–171, 2006.

[37] Tapsoba L.: *La contribution des projets de gestion electronique des documents (GED) a la performance organisationnelle de Ouagadougou (CAO)*. PhD thesis, University Aube Nouvelle, Switzerland, 2017.

[38] Tran D.N., Tran T.A., Oh A., Kim S.H., Na I.S.: Table detection from document image using vertical arrangement of text blocks, *International Journal of Contents*, vol. 11(4), pp. 77–85, 2015.

[39] Tran T.A., Tran H.T., Na I.S., Lee G.S., Yang H.J., Kim S.H.: A mixture model using Random Rotation Bounding Box to detect table region in document image, *Journal of Visual Communication and Image Representation*, vol. 39, pp. 196–208, 2016.

[40] Wang Y., Haralick R., Phillips I.T.: Automatic table ground truth generation and a background-analysis-based table structure extraction method. In: *International Conference on Document Analysis and Recognition*, pp. 528–532, 2001.

[41] Wang Y., Phillips I.T., Haralick R.M.: Table detection via probability optimization. In: D. Lopresti, J. Hu, R. Kashi (eds.) *International Workshop on Document Analysis Systems*, *Lecture Notes in Computer Science*, vol. 2423, pp. 272–282, Springer, Berlin, Heidelberg, 2002.

[42] Wang Y., Phillips I.T., Haralick R.M.: Table structure understanding and its performance evaluation, *Pattern Recognition*, vol. 37(7), pp. 1479–1497, 2004.

[43] Watanabe T., Naruse H., Luo Q., Sugie N.: Structure analysis of table-form document on the basis of the recognition of vertical and horizontal line segments. In: *1st International Conference on Document Analysis and Recognition*, pp. 638–646, 1991.

[44] Watanabe T., Luo Q., Sugie N.: Towards a practical document understanding of table-form documents: its framework and knowledge representation. In: *2nd International Conference on Document Analysis and Recognition*, pp. 510–515, 1993.

[45] Watanabe T., Luo Q., Sugie N.: Layout recognition of multi-kinds of table-form documents, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17(4), pp. 432–445, 1995.

[46] Zanibbi R., Blostein D., Cordy J.R.: A survey of table recognition: Models, observations, transformations, and inferences, *International Journal on Document Analysis and Recognition*, vol. 7(1), pp. 1–16, 2004.

[47] Zhouchen L., He J., Zhong Z., Wang R., Shum H.Y.: Table detection in online ink notes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(8), pp. 1341–1346, 2006.

## Affiliations

**Abderrahmane Kefali**
Université 8 Mai 1945 Guelma, Département d'Informatique, BP 401, Guelma 24000, Algeria, kefali.abderrahmane@univ-guelma.dz; Badji Mokhtar Annaba University, LabGED Laboratory, B.P. 12, Annaba 23000, Algeria, kefali@labged.net, ORCID ID: https://orcid.org/0000-0002-3802-7482

**Soumia Drabsia**
Université 8 Mai 1945 Guelma, Département d'Informatique, BP 401, Guelma 24000, Algeria, miyasoumia5@gmail.com

**Toufik Sari**
Badji Mokhtar Annaba University, LabGED Laboratory, B.P. 12, Annaba 23000, Algeria, sari@labged.net; Badji Mokhtar Annaba University, Computer Science Department, B.P. 12, Annaba 23000, Algeria, toufik.sari@univ-annaba.dz, ORCID ID: https://orcid.org/0000-0002-1591-6885

**Mohammed Chaoui**
Université 8 Mai 1945 Guelma, Département d'Informatique, BP 401, Guelma 24000, Algeria,chaoui.mohammed@univ-guelma.dz, ORCID ID: https://orcid.org/0000-0001-7716-0860

**Chokri Ferkous**
Université 8 Mai 1945 Guelma, LabSTIC, Département d'Informatique, BP 401, Guelma 24000, Algeria, ferkous.chokri@univ-guelma.dz, ORCID ID: https://orcid.org/0000-0002-7408-0803

**Received**: 31.07.2019
**Revised**: 09.12.2019
**Accepted**: 09.12.2019