**Optical Table Generator**

Vishal Sharma1, Аnkit Dаhiya2, Chаndаn Mаllа3, Sаnjeev Shаrmа4, Himаnshu Bаnsаl5

Department of computer Science Engineering

Bharati Vidyapeeth College of Engineering, Paschim Vihar

Guru Gobind Sing Indraprastha University, Dwarka, Delhi

[1vishalmtr@gmail.com](mailto:1vishalmtr@gmail.com)

[2ankitsub96@gmail.com](mailto:2ankitsub96@gmail.com)

[3chandanmalla@gmail.com](mailto:3chandanmalla@gmail.com)

[4ssanjeev2424@gmail.com](mailto:4ssanjeev2424@gmail.com)

[5himanshu0137.hb@gmail.com](mailto:5himanshu0137.hb@gmail.com)

**Abstract:** Tables are a ubiquitous form of communication. Tables represent the information present in the easiest and comprehensible way. OCR (Optical Character Recognition) can be described as Mechanical or electronic conversion of scanned images where images can be handwritten, typewritten or printed text.

In this paper, a complete OCR methodology for recognizing tables either Printed or handwritten without any knowledge of the font, is presented. OCR technology used is already developed. The methodology consists of three steps: The first two step refers to pre-processing and recognizing characters using OCR, while the third one refers to segmentation of scanned document. First, a pre-processing step that includes image enhancement and reducing noise. At second step OCR recognizes all the characters from its training set of data. Finally, in third step, segmentation is done in order to determine number of rows, columns and cells. Software used for OCR is OpenCVopencv.org

**Keywords**: Table Recognition, Scanned Images, Segmentation

1. **Introduction**
   1. **Why tables?**

Tables are the prevalent means of representing and communicating structured data. They may contain words, numbers, formulae, and even graphics. Developed originally in the days of printed or handwritten documents (indeed, tables may pre-date sentential text [Hurst, M.: The Interpretation of Tables in Texts.

Ph.D. thesis, University of Edinburgh (2000)]), they have been adapted to word processors and page composition languages, and form the underlying paradigm for spreadsheets and relational

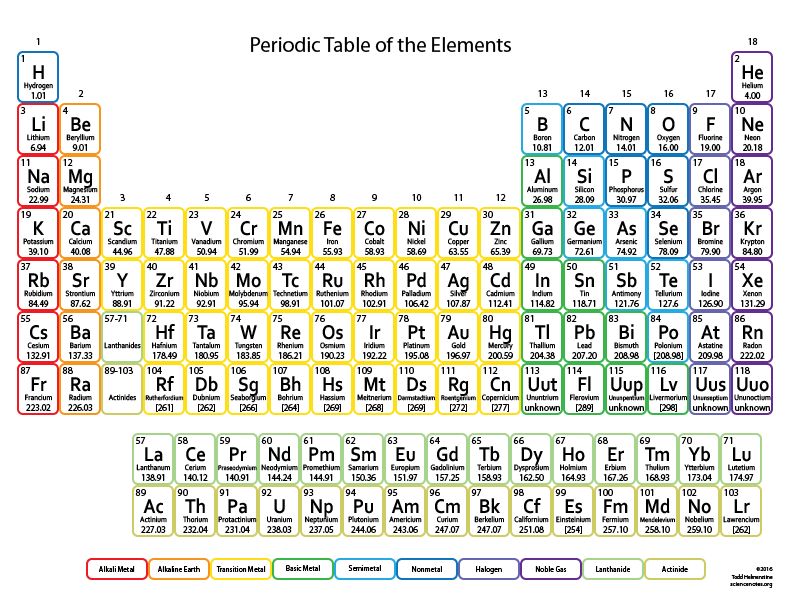
database systems.

The other common representation for structured data is a list. If we consider ordered lists analogous to vectors, then we can think of tables as analogous to matrices. Unlike vectors and matrices, lists and tables may contain non-numeric data items. Graphs are required for relationships more complex than can be represented by tables and are used primarily for *inter*-document structure. Trees are often used to represent *intra*-document structure.

Note that not all tables can be easily interpreted using

only common sense: consider, for instance, the Periodic Table of the Elements (see Figure 1), which requires substantial domain knowledge to understand.

A common objective of finding and delimiting tables, equations and illustrations, is to clear the path for optical character recognition (OCR) or, if the document is already in electronic form, for text analysis. Tables are between text and graphics with regard to the relative proportion of alphanumeric symbols, linear components and white space.

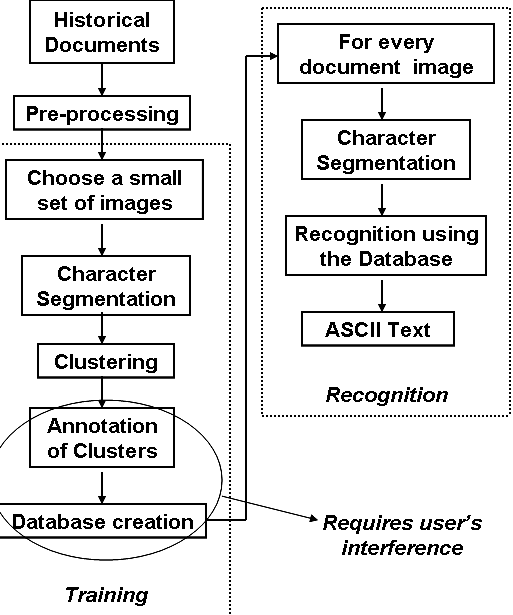


**Figure 1**

**1.2 Using OCR**

Optical character recognition (also optical character reader, OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast).[[ HPE Haven.["OCR Document"](https://dev.havenondemand.com/apis/ocrdocument#overview)]](https://en.wikipedia.org/wiki/Optical_character_recognition#cite_note-1) It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision. [https://en.wikipedia.org/wiki/Optical\_character\_recognition] OCR is used here to extract readable information from tables and represent them into digitized format. It consists of a pre-processing stage where documents are converted into binary images, a top – down segmentation technique that extracts the characters, the creation of a database by the extracted characters and a recognition stage where the database is used for converting any document into text file. To better understand the OCR methodology please look at Figure 2.

The main advantage of this methodology is the fact that neither any knowledge of the fonts in advance nor the existence of a standard database is needed. So, it can be applied to different types of documents and even deal with characters or ligatures that do not appear frequently. Depending on the type of historical documents that we want to process a database that assists the recognition procedure can be created.



**Figure 2**

1. **Literature Review**
2. **Methodology**

The subtasks behind building the complete OCR application are listed below:

1. Preprocessing Document Image (Handwritten or Electronic)
2. Performing Segmentation
3. Feeding it to OpenCV OCR
4. Post-processing the generated output

Among the sub tasks number 1 is independent than others. Tasks 2 to 4 are sequentially dependent on the success of the previous step.

**3.1 Preprocessing Document Image**

The image is taken and is converted to gray scale image. The gray scale image is then converted to binary image. This process is called Digitization of image. Practically any scanner is not perfect; the scanned image may have some noise. This noise may be due to some unnecessary details present in the image. By applying suitable methods, the denoised image is produced. The denoised image thus obtained is saved for further processing. At the end of this step we feed the preprocessed document to OpenCV

**3.2 Feeding it to OpenCV OCR**

The pre-processed image serves as the input to this and each single character in the image is found out [“α-Soft: An English Language OCR”, 2010 Second International Conference on Computer Engineering and Applications. Junaid Tariq, Umar Nauman Muhammad Umair Naru.] The image from the extraction stage is correlated with all the templates which are preloaded into the system. Once the correlation is completed, the template with the maximum correlated value is declared as the character present in the image.

**3.3 Performing Segmentation**

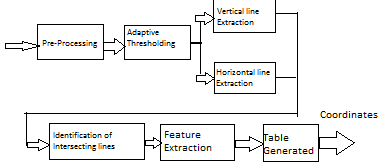
The OCR image serves as the input to this step and using Segmentation Software each row, column and cells is recognized. This step requires lot of complex equations and computer calculations. Language used to create Segmentation Software is Python

**3.4 Post Processing**

After the recognition stage, if there are some unrecognized characters/cells found, those characters/cells are given their meaning in the post -processing stage. This step also ensures that there should be none unwarranted symbols at the end of output and also all the rows, columns and cells are present at the end of the output.

**4. System Description**

Our method does not rely on the text information as it is handled by learned OCR system hence it is insensitive to the script or layout of the page and can handle multi column documents. In our approach, we seek to identify horizontal and vertical lines present in the image and use a learned classifier to locate tables based on the properties of the detected lines. A schematic block- diagram of the proposed method is shown in Figure 3.

****

**Figure 3**

**4.1 Extraction of Horizontal and vertical lines**

We employ a run-length approach to extract lines along the horizontal and vertical directions. In this work, it is assumed that the table is printed on a white background. So, in order to enhance dark and thin line-like structures, the input image *I* is first smoothed using pre- processing and automated functions from OpenCV. The pre-processed image *Ip* is then threshold adaptively with *k*1 times its maximum intensity as the threshold value. The method does not require accurate binarization since it relies only on the line information and not on the textual components. The threshold *k*1 is fixed to a low value of 0.05 so that even ‘weak’ lines show up after thresholding. The

resulting binary image *Ibw* is subjected to a run-length count along the rows and columns to obtain horizontal and vertical lines. If the count of ‘ON’ pixels in a particular direction starting at a pixel location exceeds a threshold value *l*, the segment is accepted as a line. All pixels with run-lengths less than the specified threshold value are ignored. The threshold decides the shortest line that can be detected by the system and is adaptively set to 1/20 times the width of the input image. Since tables normally occupy a significant area of the image, the method is insensitive to the choice of this threshold. It may be mentioned here that the document is assumed to have no skew, and hence a skew detection and correction step may be invoked, if necessary. Combining the output of the line segments obtained from the two directions, we get a composite image *IL* that contains all the detected horizontal and vertical lines.