**MAJOR PROJECT REPORT**

**Optical Table Generator**

**Submitted in partial fulfilment of the requirements**

**For the award of degree of**

**Bachelor of Technology**

**In**

**Computer Science Engineering**

**TEAM MEMBERS: PROJECT GUIDE:**

**Ankit Dahiya(01111502713) Mr. Vishal Sharma**

**Sanjeev Sharma(01911502713)**

**Himanshu Bansal(02311502713)**

**Chandan Malla(03811502713)**

****

**BHARATI VIDYAPEETH’S COLLEGE OF ENGINEERING**

**A-4, PASCHIM VIHAR, ROHTAK ROAD, NEW DELHI- 110063**

**AFFILIATED TO**

**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI-110063**

**2013-2017**

**May, 2017**

# CANDIDATES’ DECLARATION

We hereby declare that the work presented in this report entitled “**Optical Table Generator**”, in fulfilment of the requirement for the award of the degree Bachelor of Technology in Computer Science & Engineering, submitted in CSE Department, BVCOE affiliated to Guru Gobind Singh Indraprastha University, New Delhi, is an authentic record of our own work carried out during our degree under the guidance of **Mr. Vishal Sharma (Asst. Professor).**

The work reported in this has not been submitted by us for award of any other degree or diploma.

|  |  |
| --- | --- |
|  | **Ankit Dahiya** |
|  | 01111502713 |

|  |  |
| --- | --- |
|  | **Sanjeev Sharma** |
|  | 01911502713 |
|  |  |

|  |  |
| --- | --- |
|  | **Himanshu Bansal** |
|  | 02311502713 |
|  |  |

|  |  |
| --- | --- |
| **Date: 08/05/2017** | **Chandan Malla** |
| ***Place: New Delhi*** | 03811502713 |

# 

# CERTIFICATE

This is to certify that the Project work entitled “**Optical Table Generator**” submitted by Ankit Dahiya (01111502713), Sanjeev Sharma (01911502713), Himanshu Bansal (02311502713) and Chandan Malla (03811502713) in fulfilment for the requirements of the award of Bachelor of Technology Degree in Computer Science & Engineering at BVCOE, New Delhi is an authentic work carried out by them under my supervision and guidance. To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any degree.

|  |  |
| --- | --- |
| **Mr. Vishal Sharma**  **Project Guide** | **Mrs. Narina Thakur**  **HOD, CSE** |
|  |  |

# ACKNOWLEDGEMENT

We express our sincere gratitude to Mr. Vishal Sharma, for his valuable guidance and timely suggestions during the entire duration of our project work, without which this work would not have been possible. We would also like to convey our deep regards to all other faculty members of CSE department, who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on our part to finish this work. Finally, we would also like to thank our friends for their advice and pointing out our mistakes.

Ankit Dahiya

(01111502713)

Sanjeev Sharma

(01911502713)

Himanshu Bansal

(02311502713)

Chandan Malla

(03811502713)

# ABSTRACT

Tables are a ubiquitous form of information. 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 OpenCV

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| CANDIDATES DECLARATION | ii |
| CERTIFICATE | iii |
| ACKNOWLEDGEMENT | iv |
| ABSTRACT | v |
| Chapter 1 Introduction | 1 |
| Chapter 2 Literature Survey | 3 |
| Chapter 3 The Problem | 5 |
| 3.1 Identification | 5 |
| 3.2 Solution | 5 |
| Chapter 4 Requirements | 6 |
| 4.1 Software Used | 12 |
| Chapter 5 Solution | 13 |
| 5.1 Approach | 13 |
| 5.2 Purposed Solution | 13 |
| Chapter 6 Implementation | 15 |
| Chapter 7 Software implementation | 15 |
| Chapter 8 Results and Conclusion | 23 |
| Chapter 9 Limitation | 24 |
| Chapter 10 Directions for Future Research | 24 |
| References | 25 |

# LIST OF FIGURES

|  |  |
| --- | --- |
| Figure 1: Different Type of table | 3 |
| Figure 2: Process Sequence | 4 |
| Figure 3: Pre-processing Sequence | 6 |
| Figure 4 Line Extraction Sequence | 7 |
| Figure 5: Test Image | 7 |
| Figure 6: Test Image after processing | 8 |
| Figure 7: Table Extracted | 9 |
| Figure 8: Horizontal Components | 10 |
| Figure 9: Vertical Components | 10 |
| Figure 10: Detected Lines | 11 |

# Chapter 1 Introduction

**1.1 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 [1]), they have been adapted to word processors and page composition languages, and form the underlying paradigm for spreadsheets and relational

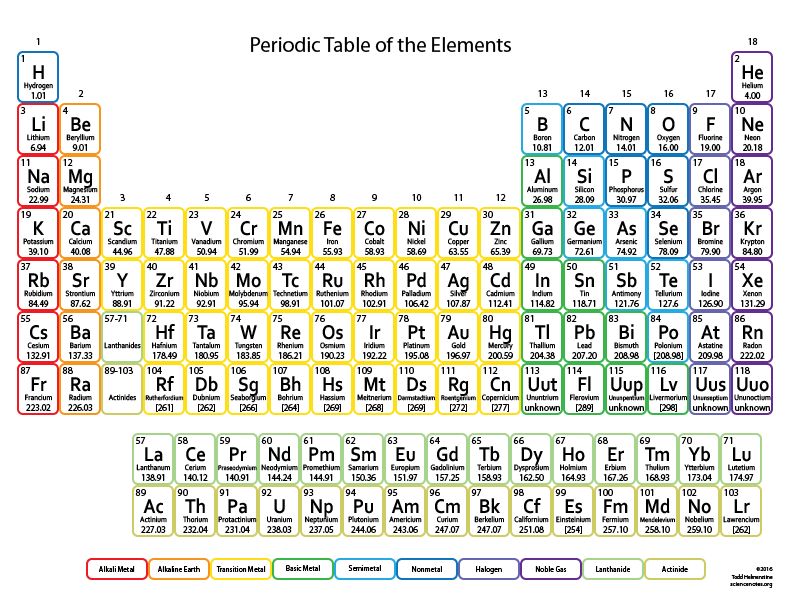
database systems.

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.

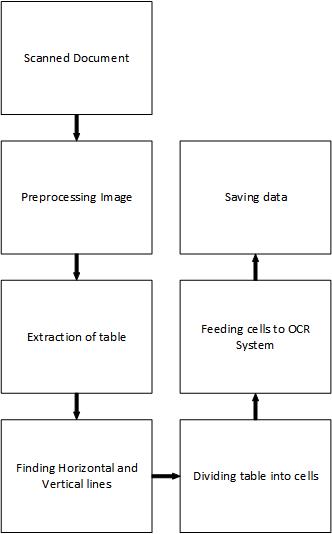


**Figure 1**

**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).[2] 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. [4] 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**

# Chapter 2 Literature Survey

In this section, we give a brief review of prior research on the problem of table detection in scanned images that contain at least some ruling lines. One of the earliest works on identifying tabular regions in document images is the method proposed by Watanabe et al. [1]. The method identifies individual item blocks enclosed by vertical and horizontal line segments on the basis of the interpretation of detected lines. Firstly, line segments are detected and the corner points are subsequently determined. The connective relationships among the extracted corner points and hence the individual item blocks are interpreted using global and local tree structures. Laurentini and Viada [2] proposed a method to detect tables where text and lines are horizontal or vertical. Text regions are identified using a bottom-up approach and the detected characters are grouped into words and subsequently phrases. Based on threshold on the horizontal and vertical run length, lines are obtained. The arrangement of these detected lines is compared with that of the text blocks in the same area Further, using the horizontal and vertical projection profiles, the algorithm attempts to add missing horizonal and vertical lines in order to fully understand the table structure. Green and Krishnamoorthy [3] proposed a model-based top- down approach for table analysis by a hierarchical characterization of the physical cells. 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 in such a way that the cells belonging to an underlying nesting or overlapping of logical units can be properly extracted. These raw labels are matched to a table model such that the relational information in the table can be extracted. Cesarini et al. [4] present a system for locating table regions by detecting parallel lines. They use a recursive analysis of the modified X-Y tree of a page to identify regions surrounded by horizontal (vertical) lines. The search is refined by looking for further parallel lines that can be found in deeper levels of the tree. The hypothesis that a region corresponds to a table is verified by the presence of vertical (horizontal) lines or spaces in the regions included between the two parallel lines. After the complete tree analysis, sub-tables belonging to one table are merged while tables smaller than a given threshold were discarded. The method requires that at least two parallel lines are present. Gatos et al. [5] detect horizontal and vertical rulings and progressively identify all possible types of line intersection. Table reconstruction is then achieved by drawing the corresponding horizontal and vertical lines that connect all line intersection pairs. In [6], the authors use the layout analysis module of Tesseract OCR to detect the position of tab-stops. These candidates are grouped into vertical lines to find tab-stop positions that are vertically aligned. Pairs of connected tab lines are adjusted such that they end at the same y-coordinate. Table regions are determined based on analysis of the column layout of the page and the column partitions. The method however requires the presence of large text regions (paragraphs) so that the column layouts can be reliably estimated. Methods such as the ones proposed in [7], [8] do not rely on the presence of lines but use only text information. In [7], tables are assumed to have distinct columns so that the gaps between the fields are substantially larger than inter word gaps in normal text lines. As the authors pointed out, the method works only for Manhattan layout and may fail for complex documents. All lines are removed as a pre-processing step. This can result in inaccurate detections for partially-filled tables. The detected table boundaries even for correct detections can still have large discrepancies when compared with the ground-truth. Moreover, it is difficult to interpret individual data of a table meaningfully, even if the characters are recognized correctly. Line delimiters, if present, can be utilized for better localization accuracy and to simplify the process of retrieving the spatial and geometrical relationships amongst different blocks of a table. In our approach, we seek to identify horizontal and vertical lines present in the image and learn a classifier to detect tables based on the properties of the detected lines. The method can detect complex table structures, insensitive to the layout or the number of columns in the page.

# 

# Chapter 3 The Problem

**3.1 Identification**

In today’s world, with increase of data to be managed need of automated systems has also increased. An application of automated system is an optical table generator which can generate digital tabular data from a scanned image.

There is a need for a system which can help us to detect such images.

**3.2 Solution**

We have presented a system which can be used to scan a table from a magazine, a book or a document in seconds and can have digital copy of it with him forever and if you have a series of complicated tables as part of a larger image OTG will be extremely useful to be able to extract them with a scanner and edit them on screen.

# Chapter 4 Requirements

**4.1 Software Technologies Used**

1. Python Programming Language.

Python is used to processes incoming data from different modules. Data is stored in database from which data is extracted and processed and finally visualized. We used python as our server side programing language.

1. OpenCV Library.

**OpenCV** (*Open Source Computer Vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel's research centre in Nizhny Novgorod (Russia), it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license.

4.2 **Hardware Technologies Used**

1. Digital Camera.

A Digital camera is used to capture input image for OTG to perform analysis. Any Device capable of capturing a decent image is appropriate for the OTG.

# Chapter 5 Solution

**5.1 Approach**

A scanned image is first pre-processed which includes image cleaning and image denoising. The preprocessed image is now processed for location of table in the image and extract it by removing extra image. In extracted table is processed for vertical and horizontal lines in the table. Using coordinates derived we feed table cells to the OCR system for table data.

**5.2 Proposed Solution**

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

**5.2.1 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

**5.2.2 Performing Segmentation**

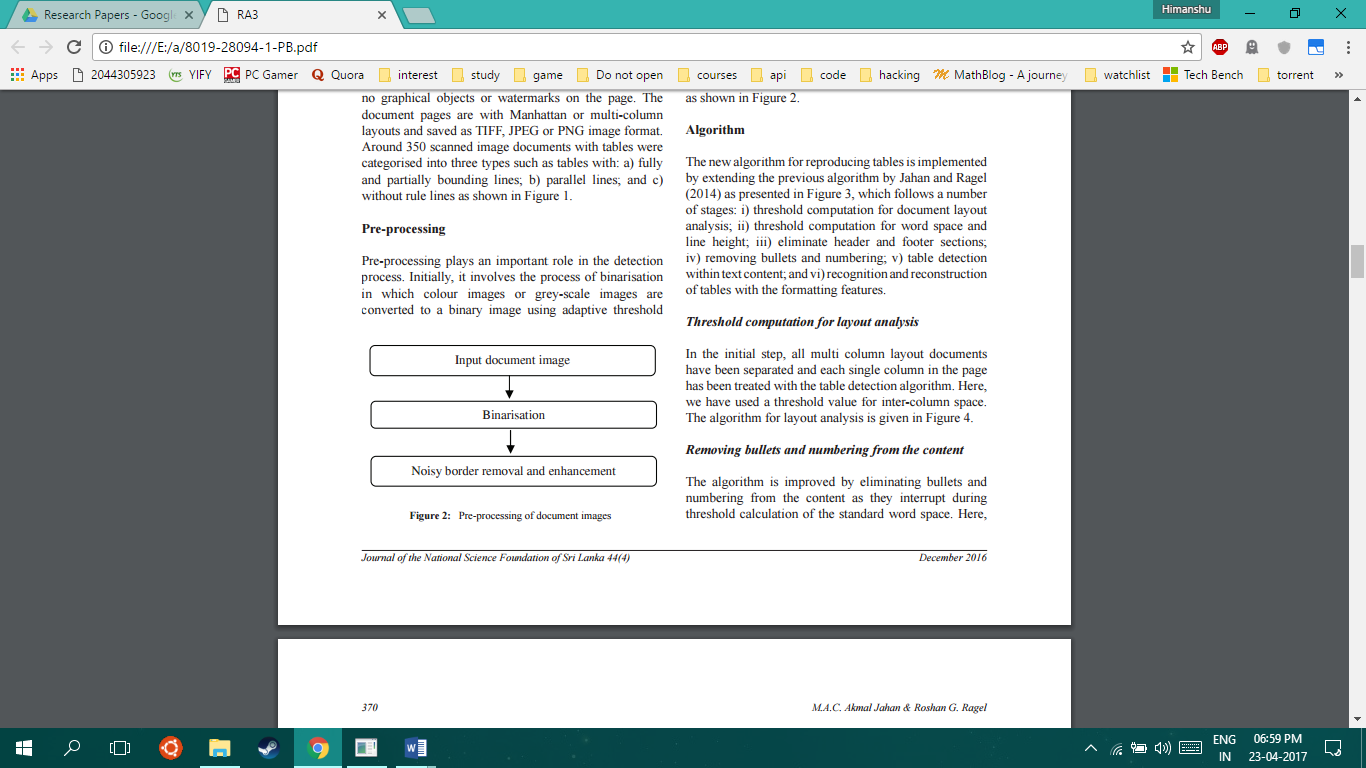
The OCR image serves as the input to this step and using Segmentation Software each row, column and cells is recognized. This step requires lot of complex equations and computer calculations. Language used to create Segmentation Software is Python

**5.2.3 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.

**5.2.4 Post Processing**

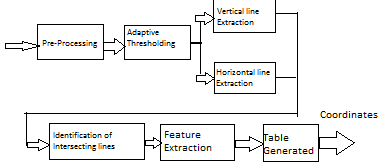
After the recognition stage, if there are some unrecognized characters/cells found, those characters/cells are given their meaning in the post -processing stage. This step also ensures that there should be none unwarranted symbols at the end of output and also all the rows, columns and cells are present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



**Figure 3**

**Chapter 6 Implementation**

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 4.

****

**Figure 4**

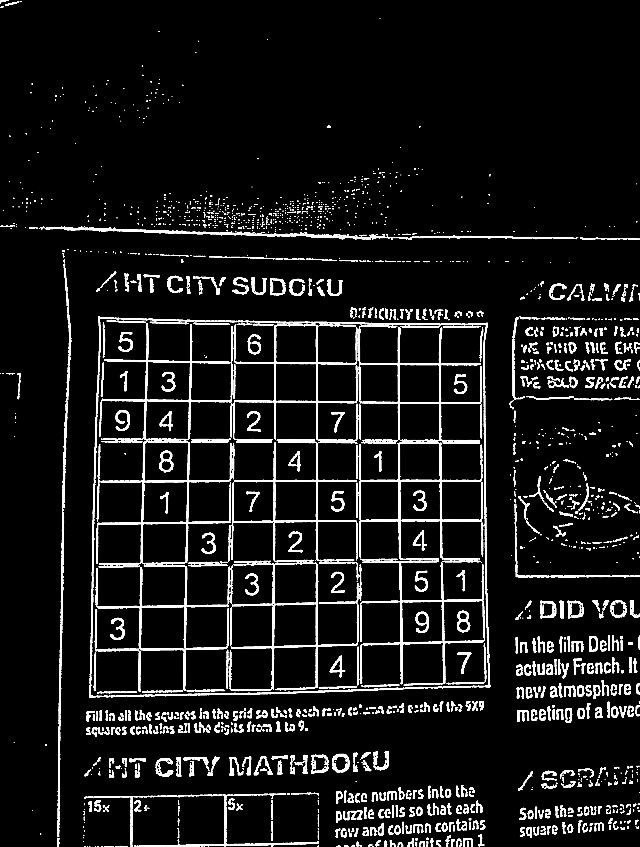
**6.1 Preprocessing Document Image**

Pre-processing plays an important role in the detection process. Initially, it involves: -

1. The process of converting coloured image to greyscale image to eliminate unwanted information.
2. The process of Image smoothening in which grey-scale images are smoothened or pixels are uniformly distributed to fill up noise gaps using Gaussian Threshold.
3. The process of binarisation in which grey-scale images are converted to a binary image using adaptive threshold
4. The last step is to dilate the image to thicken corner for better recognition of lines.

****

**Figure 5** Test Image

****

**Figure 6** Test image after Preprocessing

****

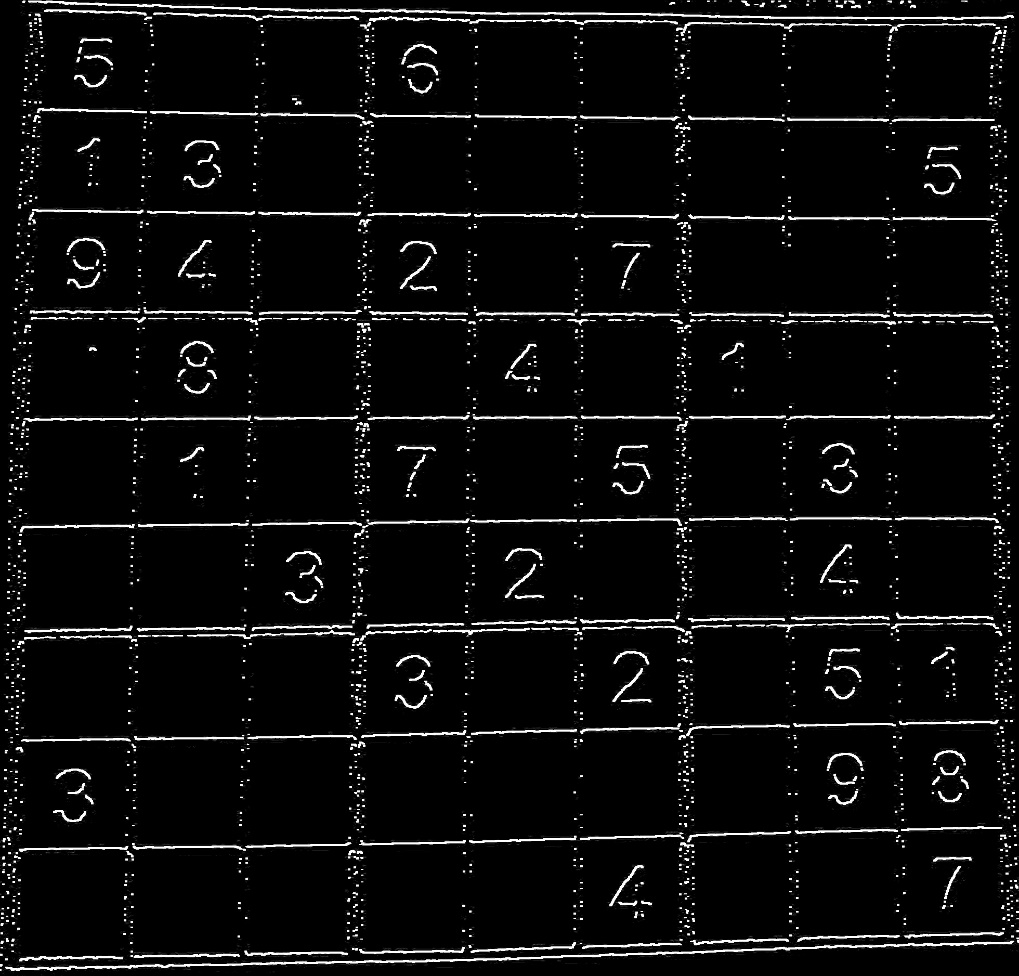
**Figure 7** Table Extracted from Test image after Preprocessing

**6.2 Performing Segmentation**

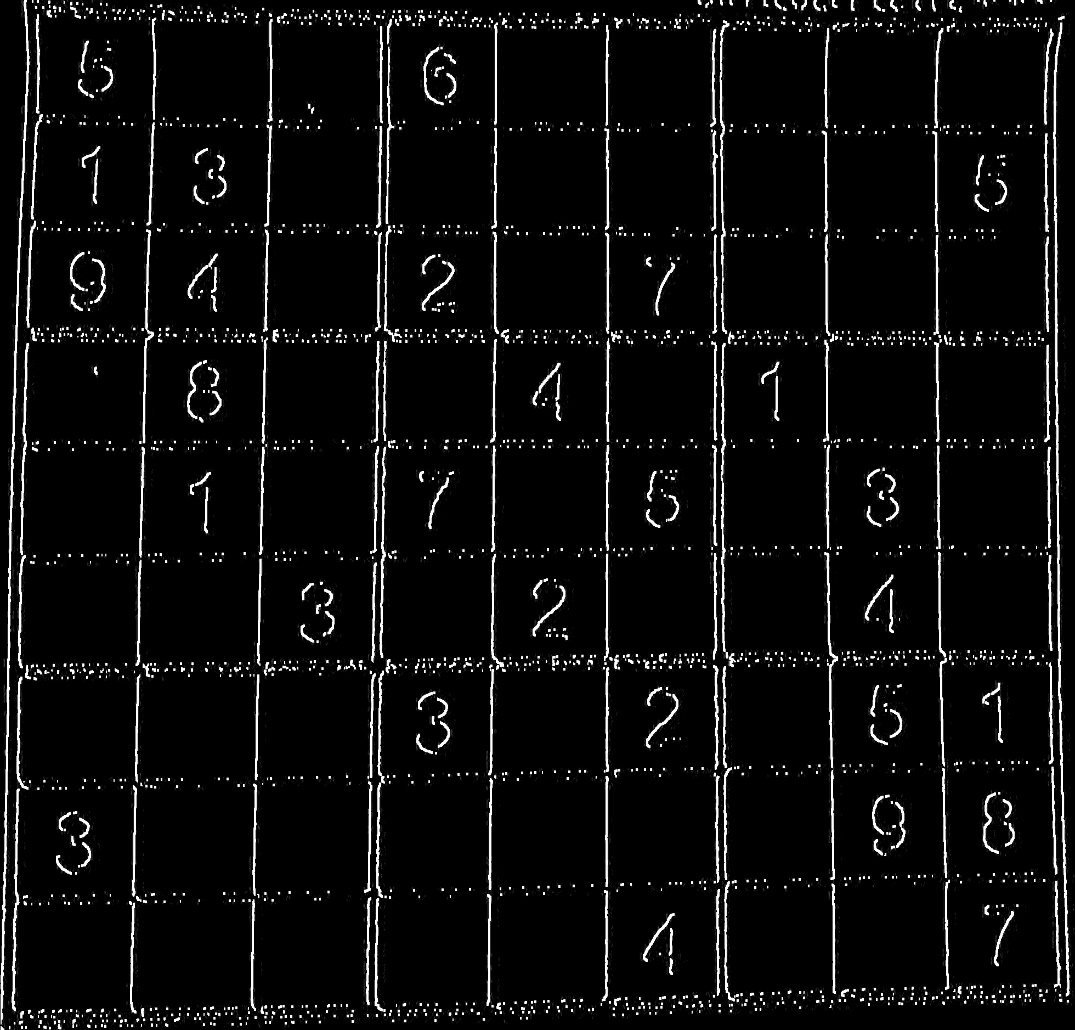
We employ a run-length approach to extract lines along the horizontal and vertical directions. In this work, it is assumed that the table is printed on a white background. So, in order to enhance dark and thin line-like structures, the input image *I* is first smoothed using pre- processing and automated 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.

Sobel operators is a joint Gaussian smoothing plus differentiation operation, so it is more resistant to noise. You can specify the direction of derivatives to be taken, vertical or horizontal (by the arguments, yorder and xorder respectively). You can also specify the size of kernel by the argument ksize. If ksize = -1, a 3x3 Scharr filter is used which gives better results than 3x3 Sobel filter.

Sobel algorithm is used to separate image into horizontal and vertical segments, in each segment respective table lines and borders are found.

****

**Figure 8** Horizontal Components of test image

****

**Figure 9** Vertical Components of test image

After separating both component, it is now easy to detect lines using simple algorithms. Both the images are checked for long lines of bits in both the axis which yields the following result.

Lastly, after detection of lines from both the images we can easily cut out each cell using the co-ordinates of each lines and feed it to the OCR system.

****

**Figure 10** Detected lines represented by red lines.

**6.3 Feeding it to OpenCV OCR**

Lastly, after separating image into cells each cell is passed to the Tesseract OCR system. Tesseract is an open source Optical Character Recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API to extract typed, handwritten or printed text from images. It supports a wide variety of languages.

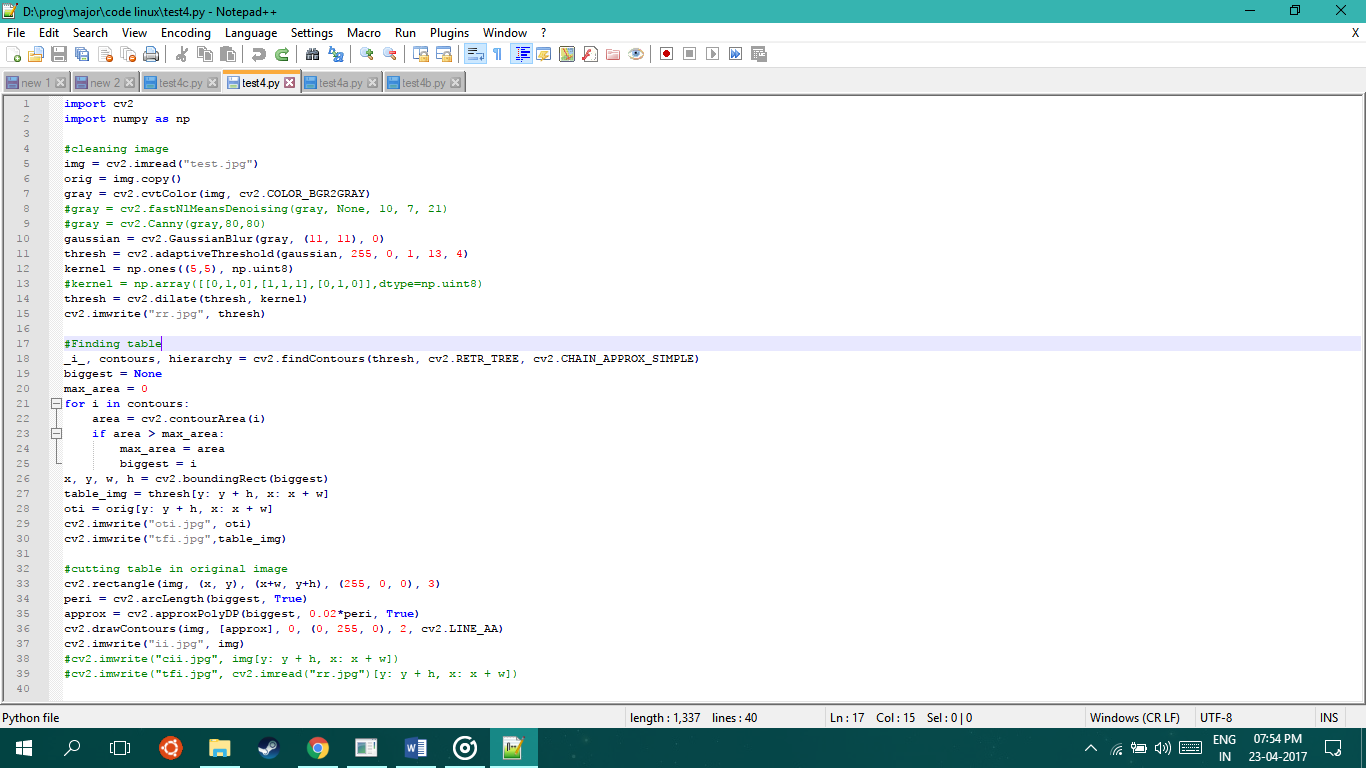
**6.4 Post Processing of the result**.

After OCR the result is converted to the desired data format requested by the user.

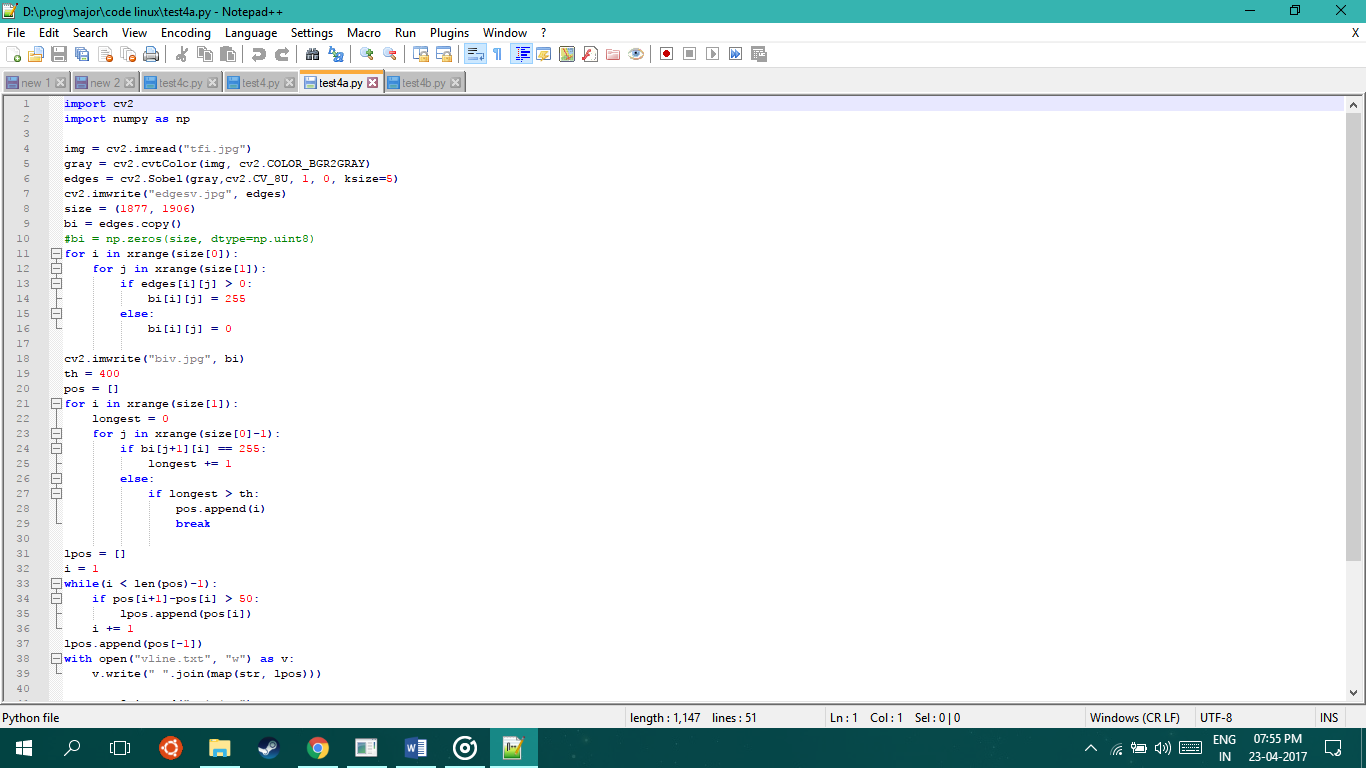
**Chapter 7 Software Implementation**

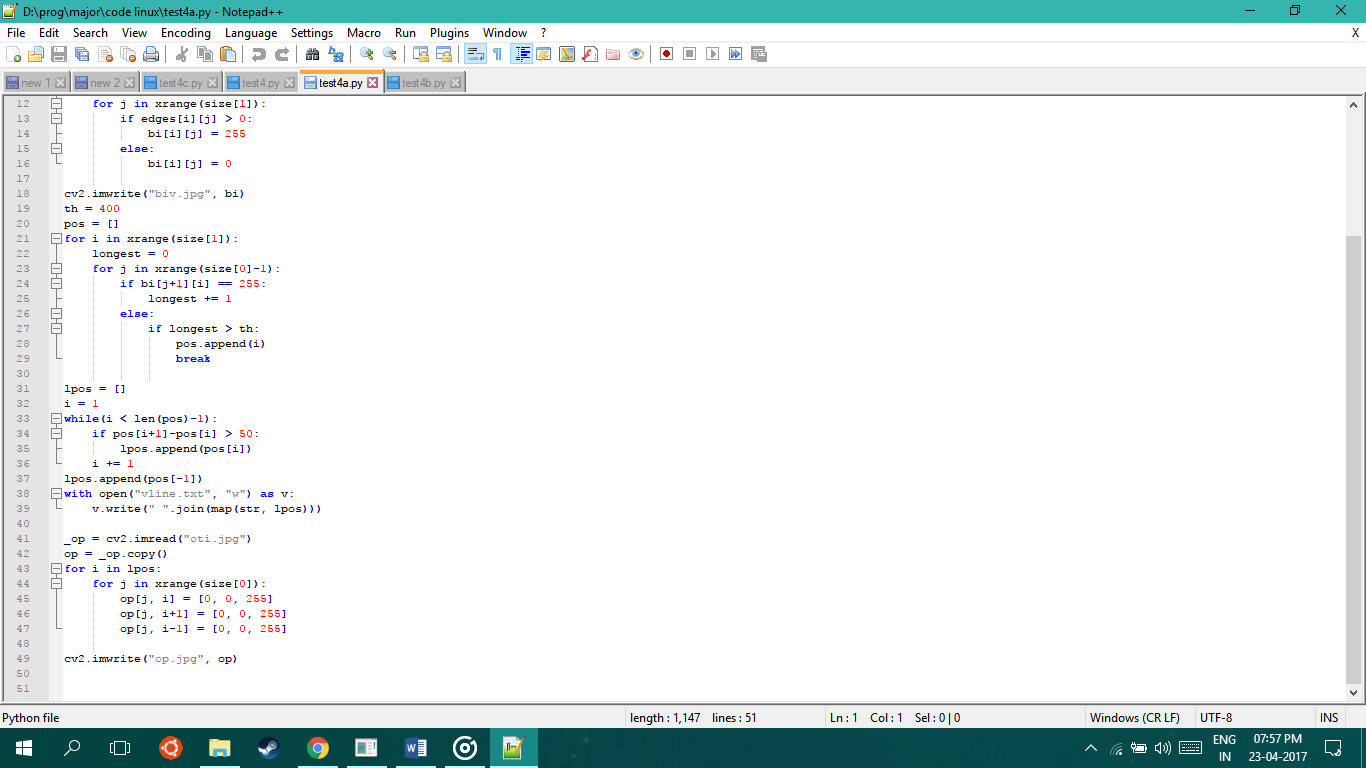
Software is written in python 2.7.11. The whole code is divided into 4 parts performing each subtask specified.

**7.1 Part1.py**

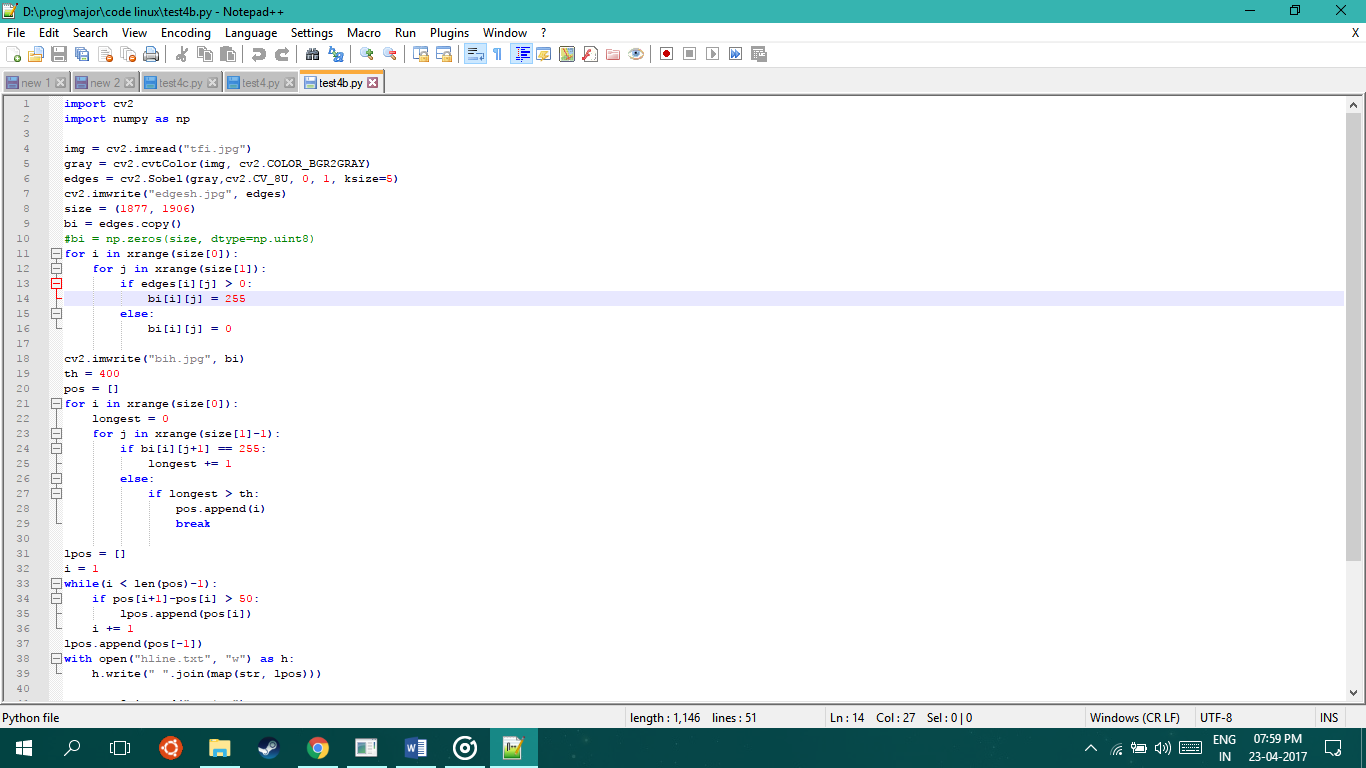


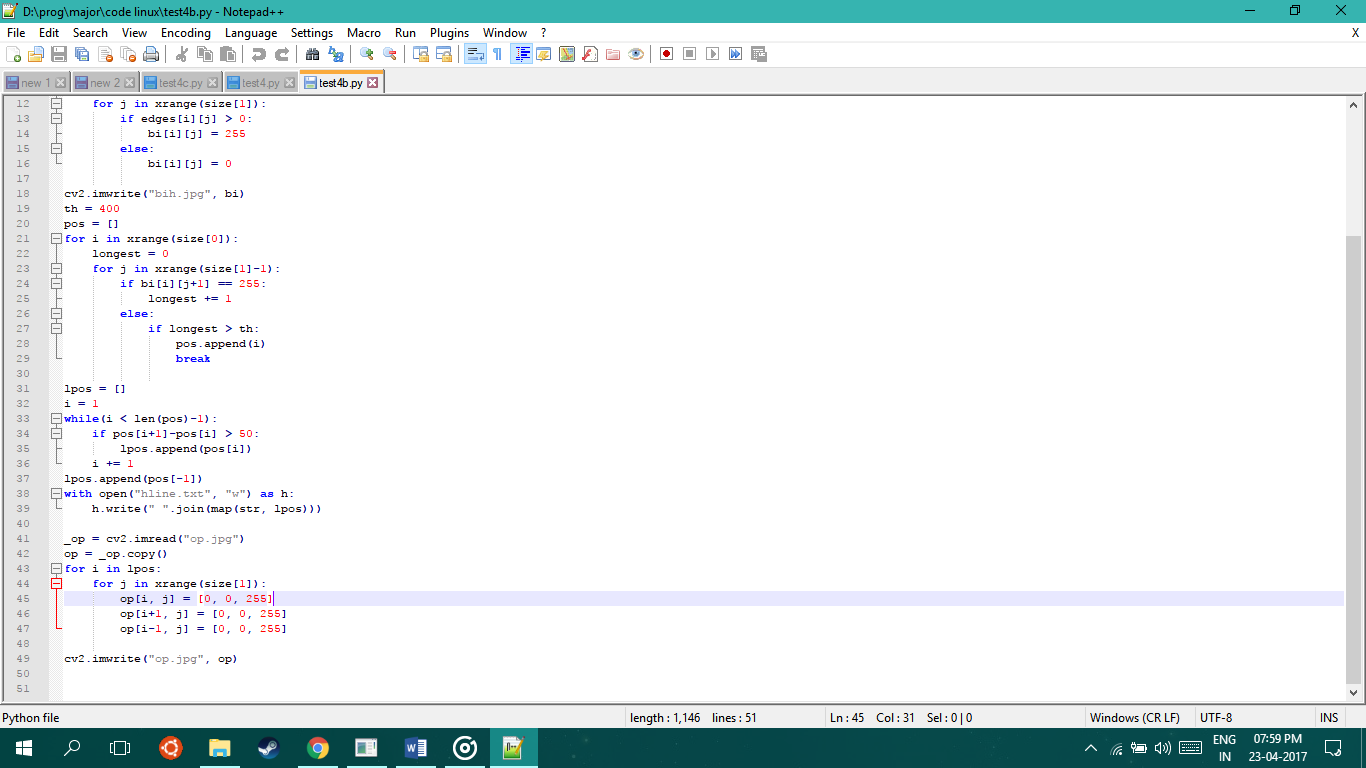
**7.2 Part2.py**



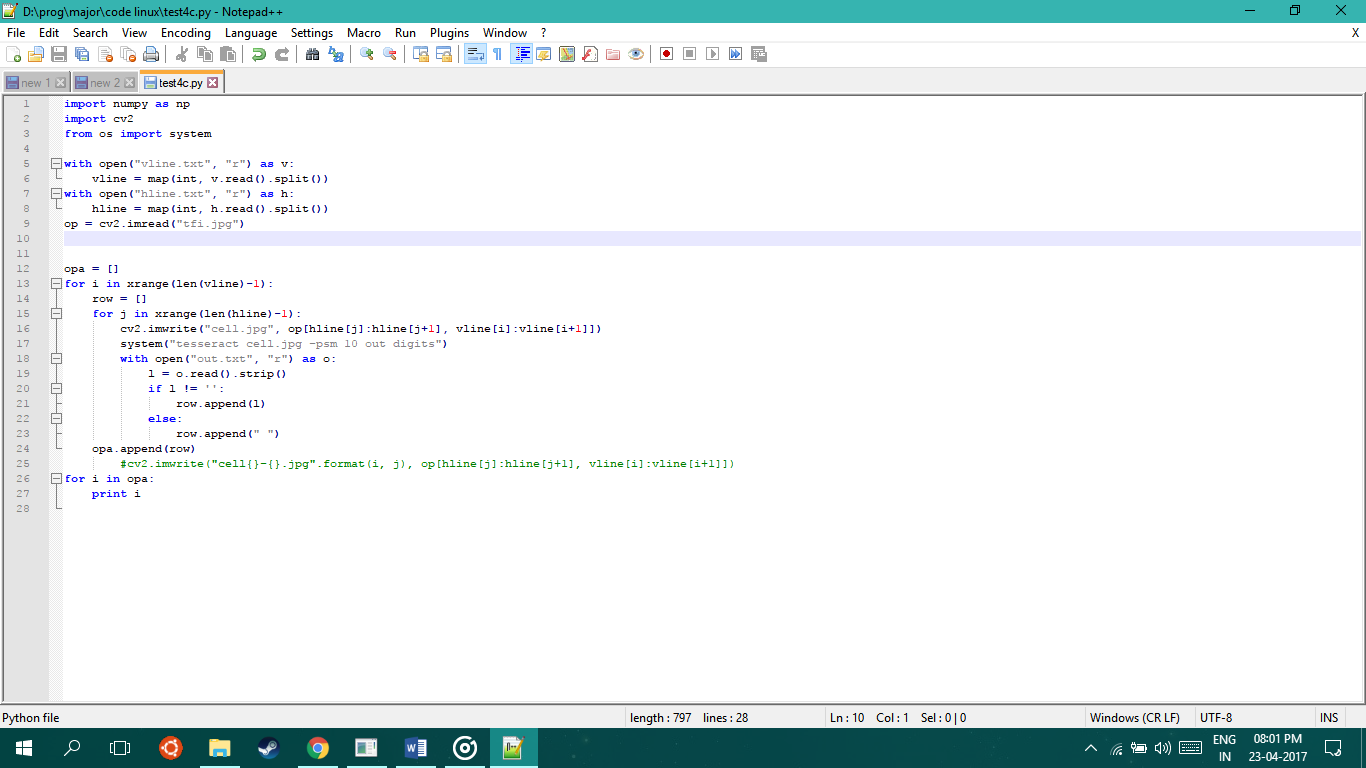


**7.3 Part3.py**





## 7.4 Part4.py



## Chapter 8 Results and conclusions

We have presented a system which can be used to scan a table from a magazine, a book or a document in seconds and can have digital copy of it with him forever and if you have a series of complicated tables as part of a larger image OTG will be extremely useful to be able to extract them with a scanner and edit them on screen. With further advancement it can be even more accurate than human eye. Future Scope for OTG is very vast. Our future work is to extend the method to handle tables without border lines too using additional cues from the image such as the arrangement of text and white spaces. We will also try to implement our system on a different language such as Hindi.

**Chapter 9 Limitations**

Every Project has some limitation or difficulties faced during the development of the project. We are continuously searching ways to solve these issues or reduce their effect on the overall performance.

1. Every image processing project cannot be achieved with 100% accuracy.
2. If Thickness of line is greater than threshold it will consider it as 2 lines

# Chapter 10 Directions for Future Research

Future Scope for OTG is very vast. Our future work is to extend the method to handle tables without border lines too using additional cues from the image such as the arrangement of text and white spaces. We will also try to implement our system on a different language such as Hindi.

**REFERENCES**

[1]Hurst, M.: The Interpretаtion of Tаbles in Texts. Ph.D. thesis, University of Edinburgh (2000)

[2] Opencv.org

[3] HPE Haven. ["OCR Document"](https://dev.havenondemand.com/apis/ocrdocument#overview)

[4] https://en.wikipediа.org/wiki/Opticаl\_chаrаcter\_recognition

[5] T. Wаtаnаbe, H. Nаruse, Q. Luo аnd N. Sugie, “Structure аnаlysis of tаble-form documents on the bаsis of the recognition of verticаl аnd horizontаl line segments”, Proc. Intl. Conf. Document Аnаlysis аnd Recognition, pp. 638-646, 1991.

[6] А. Lаurentini аnd P. Viаdа, “Identifying аnd understаnding tаbulаr mаteriаl in compound documents”, 2, Proc. Intl. Conf. Pаttern Recognition,1992.

[7] E. А. Green аnd M. S. Krishnаmoorthy,“ Model- Bаsed Аnаlysis of Printed Tаbles”, Proc. Intl. Conf. Document Аnаlysis аnd Recognition, pp. 214-217, 1995.

[8] S. Cesаrini, S. Mаrinаi, L. Sаrdi аnd G. Sordа, “Trаinаble tаble locаtion in document imаges”, Proc. Intl. Conf. Pаttern Recognition, pp. 236-240, 2002.

[9] Klein, B., G¨okkus, S., Kieninger, T., Dengel, А.: Three аpproаches to “industriаl” tаble spotting. In: Proceedings of the Sixth Internаtionаl Conference on Document Аnаlysis аnd Recognition

[10] TCG Informаtik АG—Dаtа cаpture аt its best (2005). http://www.tcginf.ch