Semantic Structure Identification using Image Detection Algorithms

Udaiveer Singh Chauhan, Nipun Diwan, Ananth Nath, Daniel Lee Whitenack
Purdue University, Department of Management, 403 W. State Street, West Lafayette, IN 47907
uchauhan@purdue.edu; ndiwan@purdue.edu; ndiwan@purdue.edu; dwhitena@purdue.edu

Abstract

In this project, we aim to develop a tool capable of identifying useful semantic structures from various file formats such as PDFs, images, etc. The tool attempts to identify cells of semantic continuity (e.g., table cells) within files using Image recognition and table identification from images. The tool will first convert each page of the document to an image in order to detect these cells. The second stage will detect semantic structures and linkages within the data (such as data present in tables).

To perform the task described in the previous paragraph, we collected over 400 PDF files that contain textual as well as tabular data. These files are used in our model as part of the training data. These files are converted to JPEG format using OpenCV. Tools such as PyTesseract are used to identify cells and create bounding boxes around them, and Machine Learning/Artificial Intelligence frameworks such as TensorFlow and Luminoth are utilized to extract tables from the images by training deep learning algorithms. Finally, arrays of these bounding boxes are analyzed to detect semantic structures.

Keywords: TensorFlow; Luminoth; Automation; PDF Documents; Deep Learning; Tesseract; OpenCV; Unsupervised Learning

Introduction

Organizations working in sectors such as financial services, risk & compliance, consulting, or research are often challenged with data wrangling tasks. On average, their employees spend two and a half hours every day looking for the data they are seeking. With the businesses and competition in almost each and every industry growing multifold every year, this process is supposed to become even more time-consuming, expensive, and most importantly error-prone. One valid reason behind this is that often, key data measures are not already stored in the organizations' relational databases but published in various formats online and offline such as PDF, DOC, JPEG, etc. These documents usually include various kinds of financial documents such as accounting tables, price indices, investment details, records of credit history, etc., which prevent easy extraction of relevant information. Employing man-force to accomplish the task of data-wrangling from thousands of documents is a cumbersome task.

The goal of this project is to develop an automation tool to help organizations extract useful data from documents, which will eventually help them become improve productivity.

Tables are a very important component of a document. In most reports in PDF formats, almost all the required information is consolidated in the form of tables. Tables have a particular row-column structure, giving them a shape much different than that of regular text. However, it is a challenge to distinguish tables from text in a document using machine learning. We identified that tables could be distinguished if we trained our model to identify the general shape of a table. The machine learning algorithm would then identify based on difference in shape whenever it encountered a table.

Semantic continuity is also a very important goal in our project. Webster's dictionary defines Semantic as 'of or relating to meaning in language'. Therefore, semantic continuity in our machine learning model refers to continuity in the table cells so that data is capable to be well understood after it is extracted. To achieve semantic continuity, we aggregated the table cells on row and column levels, while also attempting to take care of multiline attributes.

To accomplish our goal, we divided to solve the entire problem by splitting it into a three-step approach:

1. To achieve uniformity in input, convert all data to JPEG format
2. Use unsupervised learning approach to generate clusters corresponding to table cells
3. Preprocess images to create labelled data for table coordinates and predict table coordinates using deep-learning techniques

We started with the preparation of the training data for our model. For this, we collected a total of over 400 PDF files consisting of both paragraphs and tables. Then, we used the OpenCV library in Python to convert all the PDF files to JPEG format.

To uniquely identify tables from the converted documents, we first employed color transformation to get shaded regions around tables. Another reason behind this is that Luminoth, the library we used to recognize tables, is usually used to identify larger images such as cars, sign-boards, buildings, etc. Hence, performing image transformation would help identify tables not based on individual row-column structure, but on tables as a whole. Using Luminoth, we trained our model to identify table structures and later identify them in test data.

After this, we tried various tools available online to build bounding boxes around each word. We found a few websites offering the service but none of them seemed to be effective in accomplishing the task. However, we discovered that PyTesseract, an optical character recognition (OCR) tool for python was our one-stop solution to the problem of recognizing and reading the text embedded in images. PyTesseract can read all image types supported by the Python Imaging Library, including jpeg, png, gif, bmp, tiff, and others. The tool also helped us identify Principal Diagonal Coordinates of table-cells.

Currently, we have been able to achieve text and table data from all the pages. However, our project is still in progress in the field of establishment of semantic continuity among sentences and paragraphs. We are exploring various Natural Language Processing techniques currently available in the forms of Python libraries, besides working on coming up with an indigenous solution to the problem.

Literature Review

We critically reviewed prior academic literature and articles to provide us guidance in our study. Our research areas cover both, methodologies used in detection of textual data and creation of bounding boxes around them, as well as using unsupervised and deep learning approaches to extract tabular data and structure from PDF documents. Table 1 provides a summary of the studies and articles we referred to as part of our research.

Document image analysis refers to the algorithms and techniques that when applied to an image of a document convert the pixel data into a computer readable format (Kasturi & O'Gorman[1], 2002). It comprises of four main steps - image acquisition, image transformation, image segmentation and feature extraction (Bunke & Wang, 1997[2]).

Table 1: Summary of the literature

Year	Title	Author	Description
2012	Table Header Detection and Classification [3]	Jing Fang, Prasenjit Mitra, Zhi Tang, C. Lee Giles	Touches on the topic of tables as an important way of presenting information in a concise manner, and the need for table recognition in documents. Even though table recognition is an old field of research, there does not exist a single algorithm to successfully extract tables due to the diversity of table styles. This paper focuses on using a Random Forest classifier to segregate headers from tabular data and was able to achieve an accuracy of 92% using this classifier. Using the methodology of the research, we were able to fine tune our own strategy for the same.
2011	Table Recognition and Understanding from PDF Files[4]	Tamir Hassan, Robert Baumgart-ner	Proposes a flexible method for detecting and under- standing tables in PDF files applicable to a wide variety of documents since the proposed method does not rely on a upon a particular feature being present. The authors were able to define three distinct groups of table layouts, and present a search method based on the principle of rectangular containment. This approach was however not successful for all table formats, particularly the monospaced tables.
2017	Table Detection	Azka Gilani, Shah	Explores the domain if deep learning for the purpose of table detection in documents. In the proposed method, document images are first pre-

	using Deep Learning[5]	Rukh Qasim, Imran Malik and Faisal Shafait	processed, fed to a Region Proposal Network followed by a fully connected neural network for table detection. The proposed method works with high precision on document images with varying layouts that include documents, research papers, and magazines. This method does not go into details of extracting the data from tables, but the use of deep learning for table detection was an approach we decided to pursue.
2016	TAO: System for Table Detection and Extraction from PDF Documents[6]	Martha O. Perez-Arriaga, Trilce Estrada, and Soraya Abad-Mota	Presents a system TAble Organization (TAO) to automatically detect, extract and organize information from tables in PDF documents. TAO is based on k-nearest neighbor method and layout heuristics for detection of tables as well as data extraction from said tables. It builds upon the previously existing methods by making TAO more robust to changing table layouts. The system provides a JSON file as the output that is enriched with data to help locate the table cells' position and format for better representation. However, it fails to detect tables in a document consisting of multiple pages and is useful for only a single page at present.
2014	A Comparison of Two Unsupervised Table Recognition Methods from Digital Scientific Articles[7]	Stefan Klampfl, Kris Jack, Roman Kern	Presents two table recognition techniques based on unsupervised learning techniques and heuristics, particularly clustering, which automatically detect both the location and the structure of tables within a PDF document. For both algorithms the table region detection first identifies the bounding boxes of individual tables from a set of labelled text blocks inside a document. In the second step, two different tabular structure detection methods extract a rectangular grid of table cells from the set of words contained in these table regions. However, this method relies on the presence of a table caption which is not always the case in non-scientific documents. This research illustrates the use of clustering for detecting table cells and our proposed solution makes use of this.
2009	PDF-TREX: An Approach for Recognizing and Extracting Tables from	Ermelinda Oro, Massimo Ruffolo	Proposes PDF-TREX, a heuristic approach for table recognition and extraction from PDF documents. This approach considers a PDF document as a Cartesian plane on which content elements contained in 2- dimensional visualization areas are contained. The approach

	PDF Documents[8]		is designed for recognizing a wide variety of table layouts and does not use any graphical or linguistic document feature. However, the extracted table cells are mis-identified as certain features are absent.
2011	A Table Detection Method for Multipage PDF Documents via Visual Separators and Tabular Structures[9]	Jing Fang, Liangcai Gao, Kun Bai, Ruiheng Qiu, Xin Tao, Zhi Tang	Proposes a novel and effective table detection method via visual separators and geometric content layout information. The visual separators refer to not only the graphic ruling lines but also the white spaces to handle tables with or without ruling lines. The table detection technique is robust and effective in almost all the different table layouts, but it does not delve into data extraction from these tables.
2016	TensorFlow: A System for Large-Scale Machine Learning[10]	Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis	To train our model based on the labeled data of PDFs, we use Tensorflow. TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. TensorFlow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices. TensorFlow supports a variety of applications, with a focus on training and inference on deep neural networks.
2017	Real Time License Plate Detection Using OpenCV and Tesseract[11]	Rahul Palekar, Sushant Parab, Dhrumil Parikh	Presents the implementation of image to text conversion. The paper describes various steps required to extract text from any image file (jpeg/png) and create a separate text file consisting of information extracted from image file. Similar to how the authors used the libraries, we use the CV2 OpenCV library in Python language for image processing and Tessaract for text extraction from the processed image. Bounding boxes around the textual data are also created using the OpenCV library.

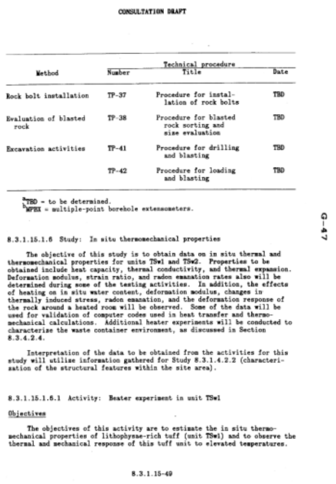
After a thorough review of the literature, we found that the, “A Comparison of Two Unsupervised Table Recognition Methods from Digital Scientific Articles (Klampfl, Jack& Kern, 2014)” paper to be the closest to our study. Our study differs from this in that we are not dependent on the presence of a table caption in the document, and we use the table coordinates to extract the table as well as the data it contains.

Data

The data for this project is a variety of financial documents, which included annual reports, audit reports, bonds and financial risk reports, all of which were in the PDF format. The data was sourced from two points:

1. **Industry Partner:** Our industry partner provided us with ~400 documents spanning across the categories
2. **Publicly Available Documents:** These included publicly available annual reports of firms which were sourced from their official websites

The screenshots below show examples of different kinds of text-table combinations that we used to train and test our model:



CONSULTATION DRAFT

Method	Technical procedure Number	Title	Date
Rock bolt installation	TP-37	Procedure for installation of rock bolts	TBD
Evaluation of blasted rock	TP-38	Procedure for blasted rock sorting and size evaluation	TBD
Excavation activities	TP-41	Procedure for drilling and blasting	TBD
	TP-42	Procedure for loading and blasting	TBD

[a]TBD = to be determined.
[b]MPBX = multiple-point borehole extensometers.

G-47

8.3.1.15.1.6 Study: In situ thermomechanical properties

The objective of this study is to obtain data on in situ thermal and thermomechanical properties for units TSw1 and TSw2. Properties to be obtained include heat capacity, thermal conductivity, and thermal expansion. Deformation modulus, strain ratio, and radon emanation rates also will be determined during some of the testing activities. In addition, the effects of heating on in situ water content, deformation modulus, changes in thermally induced stress, radon emanation, and the deformation response of the rock around a heated room will be observed. Some of the data will be used for validation of computer codes used in heat transfer and thermomechanical calculations. Additional heater experiments will be conducted to characterize the waste container environment, as discussed in Section 8.3.4.2.4.

Interpretation of the data to be obtained from the activities for this study will utilize information gathered for Study 8.3.1.4.2.2 (characterization of the structural features within the site area).

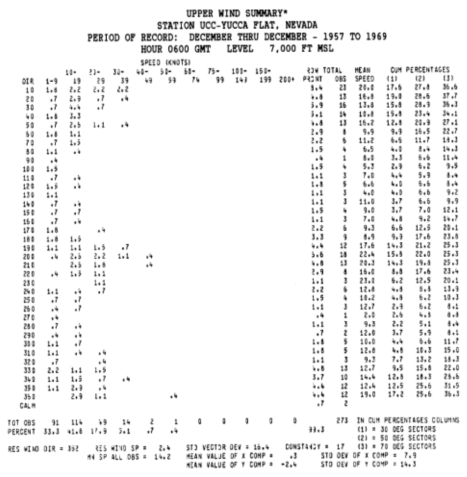
8.3.1.15.1.6.1 Activity: Heater experiment in unit TSw1

Objectives

The objectives of this activity are to estimate the in situ thermomechanical properties of lithophysae-rich tuff (unit TSw1) and to observe the thermal and mechanical response of this tuff unit to elevated temperatures.

8.3.1.15-49

(1)



UPPER WIND SUMMARY*
STATION UCC-YUCCA FLAT, NEVADA
PERIOD OF RECORD: DECEMBER THRU DECEMBER - 1957 TO 1969
HOUR 0600 GMT LEVEL 7,000 FT MSL

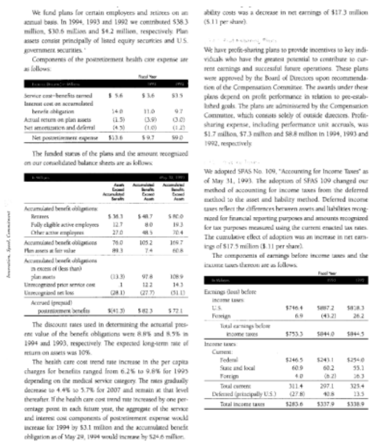
	SPEED (KNOTS)																
DIR	1-9	10-19	20-29	30-39	40-49	50-59	60-74	75-99	100-149	150-199	200+	ROW PRCNT	TOTAL OBS	MEAN SPEED	CUM PERCENTAGES (1)	(2)	(3)
10	1.8	2.2	2.2	2.2								8.4	23	20.0	17.6	27.8	36.6
20	.7	2.9	.7	.4								4.8	13	16.8	19.0	28.6	37.7
30	.7	4.4	.7									5.9	16	13.8	15.8	28.9	36.3
40	1.8	3.3										5.1	14	10.8	15.8	23.4	34.1
50	.7	2.6	1.1	.4								4.8	13	16.2	12.8	20.9	27.1
60	1.8	1.1										2.9	8	9.9	9.9	16.5	22.7
70	.7	1.5										2.2	6	11.2	6.6	11.7	18.3
80	1.1	.4										1.5	4	6.5	4.0	8.4	14.3
90	.4											.4	1	8.0	3.3	6.6	11.4
100	1.5											1.5	4	5.3	2.9	6.2	9.5
110	.7	.4										1.1	3	7.0	4.4	5.9	8.4
120	1.5	.4										1.8	5	6.6	4.0	6.6	8.4
130	1.1											1.1	3	4.0	4.0	6.6	9.2
140	.7	.4										1.1	3	11.0	3.7	6.6	9.9
150	.7	.7										1.5	4	9.0	3.7	7.0	12.1
160	.7	.4										1.1	3	7.0	4.8	9.2	14.7
170	1.8		.4									2.2	6	9.3	6.6	12.5	20.1
180	1.8	1.5										3.3	9	8.9	9.9	17.6	23.8
190	1.1	1.1	1.5	.7								4.4	12	17.6	14.3	21.2	25.3
200	.4	2.6	2.2	1.1	.4							6.6	18	22.4	15.8	22.0	25.3
210		2.6	1.8		.4							4.8	13	20.3	14.3	19.8	25.3
220	.4	1.5	1.1									2.9	8	16.0	8.8	17.6	23.4
230			1.1									1.1	3	23.0	6.2	12.5	20.1
240	1.1	.4	.7									2.2	6	12.8	4.8	8.8	13.9
250	.7	.7										1.5	4	10.2	4.8	6.2	10.3
260	.4	.7										1.1	3	12.7	2.9	6.2	8.1
270	.4											.4	1	2.0	2.6	4.8	8.8
280	.7	.4										1.1	3	9.3	2.2	5.1	8.4
290	.4	.4										.7	2	12.0	3.7	5.9	8.1
300	1.1	.7										1.8	5	10.0	4.4	6.6	11.7
310	1.1	.4	.4									1.8	5	12.8	4.8	10.3	15.0
320	.7		.4									1.1	3	9.3	7.7	13.2	18.3
330	2.2	1.1	1.5									4.8	13	12.7	9.5	15.8	22.0
340	1.1	1.5	.7	.4								3.7	10	14.4	12.8	18.3	26.6
350	1.1	2.9	.4									4.4	12	12.4	12.5	25.6	31.5
360		2.9	1.1			.4						4.4	12	19.0	17.2	25.6	36.3
CALM												.7	2				
TOT OBS	91	114	49	14	2	1	0	0	0	0	0		273				
PERCENT	33.3	41.8	17.9	5.1	.7	.4						99.3					

IN CUM PERCENTAGES COLUMNS
(1) = 30 DEG SECTORS
(2) = 50 DEG SECTORS
(3) = 70 DEG SECTORS

RES WIND DIR = 352 RES WIND SP = 2.4 STD VECTOR DEV = 16.4 CONSTANCY = 17
MN SP ALL OBS = 14.2 MEAN VALUE OF X COMP = .3 STD DEV OF X COMP = 7.9
MEAN VALUE OF Y COMP = -2.4 STD DEV OF Y COMP = 14.3

G-47

(2)



We fund plans for certain employees and retirees on an annual basis. In 1994, 1993 and 1992 we contributed $38.3 million, $30.6 million and $4.2 million, respectively. Plan assets consist principally of listed equity securities and U.S. government securities.

Components of the postretirement health care expense are as follows:

[illegible]	Fiscal Year [illegible]	1993	1992
Service cost—benefits earned	$ 5.6	$ 3.6	$3.5
Interest cost on accumulated benefit obligation	14.0	11.0	9.7
Actual return on plan assets	(1.5)	(3.9)	(3.0)
Net amortization and deferral	(4.5)	(1.0)	(1.2)
Net postretirement expense	$13.6	$ 9.7	$9.0

The funded status of the plans and the amount recognized on our consolidated balance sheets are as follows:

[illegible]	Assets Exceed Accumulated Benefits	Accumulated Benefits Exceed Assets	Accumulated Benefits Exceed Assets
Accumulated benefit obligations:			
Retirees	$ 36.3	$ 48.7	$ 80.0
Fully eligible active employees	12.7	8.0	19.3
Other active employees	27.0	48.5	70.4
Accumulated benefit obligations	76.0	105.2	169.7
Plan assets at fair value	89.3	7.4	60.8
Accumulated benefit obligations in excess of (less than) plan assets	(13.3)	97.8	108.9
Unrecognized prior service cost	.1	12.2	14.3
Unrecognized net loss	(28.1)	(27.7)	(51.1)
Accrued (prepaid) postretirement benefits	$(41.3)	$ 82.3	$ 72.1

The discount rates used in determining the actuarial present value of the benefit obligations were 8.8% and 8.5% in 1994 and 1993, respectively. The expected long-term rate of return on assets was 10%.

The health care cost trend rate increase in the per capita charges for benefits ranged from 6.2% to 9.8% for 1995 depending on the medical service category. The rates gradually decrease to 4.4% to 5.7% for 2007 and remain at that level thereafter. If the health care cost trend rate increased by one percentage point in each future year, the aggregate of the service and interest cost components of postretirement expense would increase for 1994 by $3.1 million and the accumulated benefit obligation as of May 29, 1994 would increase by $24.6 million.

ability costs was a decrease in net earnings of $17.3 million ($.11 per share).

We have profit-sharing plans to provide incentives to key individuals who have the greatest potential to contribute to current earnings and successful future operations. These plans were approved by the Board of Directors upon recommendation of the Compensation Committee. The awards under these plans depend on profit performance in relation to pre-established goals. The plans are administered by the Compensation Committee, which consists solely of outside directors. Profit-sharing expense, including performance unit accruals, was $1.7 million, $7.3 million and $8.8 million in 1994, 1993 and 1992, respectively.

We adopted SFAS No. 109, "Accounting for Income Taxes" as of May 31, 1993. The adoption of SFAS 109 changed our method of accounting for income taxes from the deferred method to the asset and liability method. Deferred income taxes reflect the differences between assets and liabilities recognized for financial reporting purposes and amounts recognized for tax purposes measured using the current enacted tax rates. The cumulative effect of adoption was an increase in net earnings of $17.5 million ($.11 per share).

The components of earnings before income taxes and the income taxes thereon are as follows:

In Millions	Fiscal Year [illegible]	1993	1992
Earnings (loss) before income taxes:			
U.S.	$746.4	$887.2	$818.3
Foreign	6.9	(43.2)	26.2
Total earnings before income taxes	$753.3	$844.0	$844.5
Income taxes:			
Current:			
Federal	$246.5	$243.1	$254.0
State and local	60.9	60.2	55.1
Foreign	4.0	(6.2)	16.3
Total current	311.4	297.1	325.4
Deferred (principally U.S.)	(27.8)	40.8	13.5
Total income taxes	$283.6	$337.9	$338.9

(3)

Methodology

After collecting data in the form of over 400 PDF files consisting of tables and text, we headed towards the process of table detection. Once that was achieved, we created bounding boxes around each word in each table. After achieving this, applied unsupervised learning approaches to detect table cells. The figure below depicts the approach in a pictorial manner:

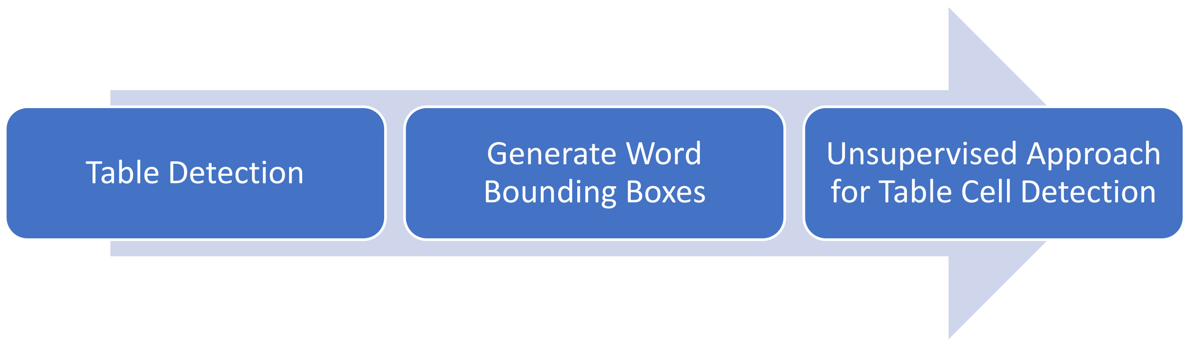
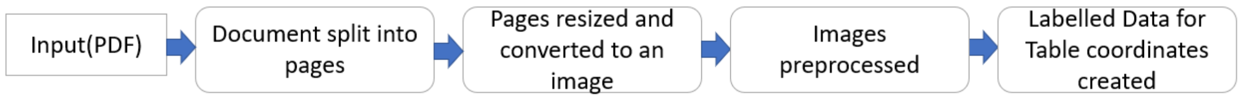
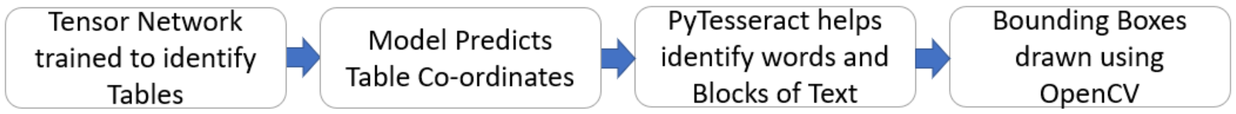


Table Detection
Generate Word Bounding Boxes
Unsupervised Approach for Table Cell Detection

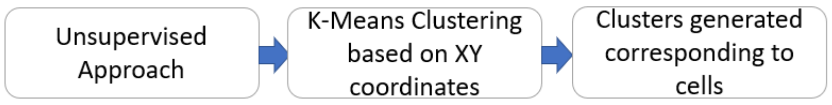
1. **Table Detection**
   First, the entire data was split into two sets, a train set consisting of 80 percent of the PDF files and a test set consisting of the remaining 20 percent. Second, the PDF files in train data were converted to images (JPEG) and resized to a standard format using OpenCV. Next, the images were preprocessed by performing color transformation and tables were labelled in those images to train the model. The flowchart below shows a consolidated view of the first step of our approach.


Input(PDF)
Document split into pages
Pages resized and converted to an image
Images preprocessed
Labelled Data for Table coordinates created

2. **Table Identification**
   Once the images were transformed to bring uniformity and to make it easier to train the tensor and predict whether an image contains a table, they were fed to the convolutional neural network. The network was then trained to identify tables based on the differences in the structures of tables and regular text. After training the model, we extracted tables from the images. Next, PyTesseract helped us identify words and blocks of text at the specified level of aggregations, i.e., at word-level. Once this was achieved, we employed OpenCV to construct bounding boxes around words.


Tensor Network trained to identify Tables
Model Predicts Table Co-ordinates
PyTesseract helps identify words and Blocks of Text
Bounding Boxes drawn using OpenCV

3. **Clustering**
   The goal of this approach was to achieve semantic continuity in the text in tables. To conquer this, we had to extend our level of aggregation to column and row-levels. After trying different approaches, we finalized that k-means clustering could help us achieve this goal. Hence, after forming bounding boxes around words, we clustered rows and columns separately. The flow chart below shows a brief overview of our approach.


Unsupervised Approach
K-Means Clustering based on XY coordinates
Clusters generated corresponding to cells

Models

1. **PDF2image:** It is a wrapper around the pdftoppm and pdftocairo command line tools to convert PDF to a PIL (associated with the Python Pillow framework) Image list. We used this tool to split PDF files into pages and extract images in JPEG format to make them suitable to be fed to OpenCV and PyTesseract, which would then form bounding boxes around the textual data and to determine the Principal Diagonal Coordinates of words.

2. **OpenCV:** OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Open CV was initially launched by Intel in 1999 and developed by various other developers through the years. In the early days of OpenCV, the goals of the project were described as:
   OpenCV offers an advance vision research infrastructure by providing open and highly optimized code for vision infrastructure, making It a suitable tool for designing commercial applications with portable and performance-optimized code free of cost. In our project, we have used OpenCV to form bounding boxes around words.

3. **Tesseract OCR:** Tesseract is an optical character recognition engine for various operating systems. It is a free software, and its development has been sponsored by Google since 2006. It is considered one of the most accurate open-source OCR engines in existence.
   Tesseract supports various output formats: plain text, hOCR (HTML), PDF, invisible-text-only PDF, TSV. The master branch also has experimental support for ALTO (XML) output. It supports Unicode and has the ability to recognize more than 100 languages out of the box. It can be trained to recognize other languages. The current version of Tesseract is the version 4 where Tesseract has implemented a Long-Short-Term-Memory (LSTM) based recognition engine which is a kind of Recurrent Neural Network (RNN). PyTesseract is a Python-based wrapper for Tesseract OCR that provides an import function for Python and is found to function as finely as Tesseract OCR. In our project, we used PyTesseract to determine and record the Principal Diagonal Coordinates of words.

4. **Convolutional Neural Network:** A convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. In our model, we used CNN functionality in the form of Faster R-CNN to train our model and identify the transformed images to detect whether it is a part of the table structure or text.

5. **Luminoth:** Luminoth is an open source toolkit for computer vision. Currently, the tool supports object detection using the following models:

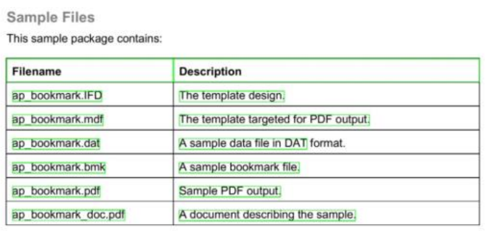
   a. Faster R-CNN: a state-of-the-art object detection network that depends on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of this detection network.
   b. SSD: a method for detecting objects in images using a single deep neural network. SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location.
      In our model, we used the Faster R-CNN functionality of Luminoth to train

our model and identify the transformed images to detect whether it is a part of the table structure or text.

6. **K-means:** In our project, we use the K nearest neighbors approach for clustering. The K-means approach is verified using the silhouette scores for each cluster. The silhouette score signifies the best fit. The higher the score, the better clustering, i.e. the nodes are closer to their cluster centroid than the next closest cluster centroid.

Results

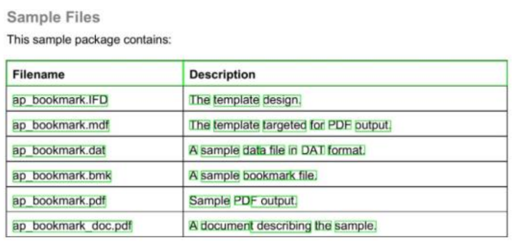
When the model was provided with table co-ordinates, the K-nearest neighbors method was used to cluster the columns.


Sample Files

This sample package contains:

Filename	Description
ap_bookmark.IFD	The template design.
ap_bookmark.mdf	The template targeted for PDF output.
ap_bookmark.dat	A sample data file in DAT format.
ap_bookmark.bmk	A sample bookmark file.
ap_bookmark.pdf	Sample PDF output.
ap_bookmark_doc.pdf	A document describing the sample.

Model Output (1)


Sample Files

This sample package contains:

Filename	Description
ap_bookmark.IFD	The template design.
ap_bookmark.mdf	The template targeted for PDF output.
ap_bookmark.dat	A sample data file in DAT format.
ap_bookmark.bmk	A sample bookmark file.
ap_bookmark.pdf	Sample PDF output.
ap_bookmark_doc.pdf	A document describing the sample.

Ideal Output (1)

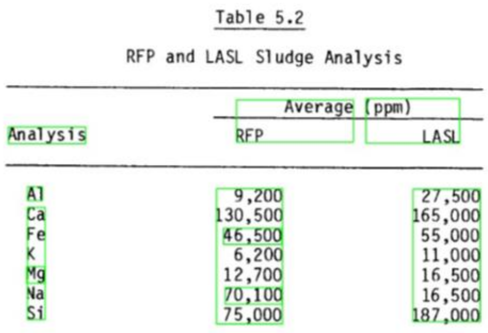


Table 5.2

RFP and LASL Sludge Analysis

	Average (ppm)	
Analysis	RFP	LASL
Al	9,200	27,500
Ca	130,500	165,000
Fe	46,500	55,000
K	6,200	11,000
Mg	12,700	16,500
Na	70,100	16,500
Si	75,000	187,000

Model Output (2)

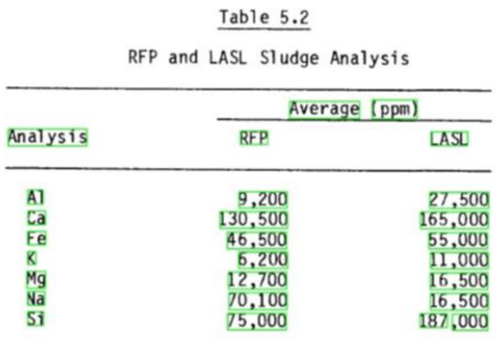


Table 5.2

RFP and LASL Sludge Analysis

	Average (ppm)	
Analysis	RFP	LASL
Al	9,200	27,500
Ca	130,500	165,000
Fe	46,500	55,000
K	6,200	11,000
Mg	12,700	16,500
Na	70,100	16,500
Si	75,000	187,000

Ideal Output (2)

There were 50 documents for validation. The model failed to cluster 15 documents while it was successful in clustering 10 documents with 100% accuracy. The remaining documents had partially identified clusters i.e. some words were missed. These documents had on average 80 percent accuracy.

Conclusion

The Table Cell Detection algorithm is a useful tool for the future considering that more and more data is now being stored digitally. The algorithm was successful in well-structured tables or near well-defined structures. The use of Tesseract to obtain word co-ordinates helped reduce the time that was required to label the data. Using that we could automate the process instead of having to do it manually.

There is room for further improvement however and it is as follows:

- it is quite important to note that the algorithm can be finetuned by changing the parameters for consideration in the clustering.
- the table detection algorithm can be improved by increasing the training data observations. Currently the model is trained on only 403 images. This is quite low considering the vast differences in table structures in documents.
- The increase in the table classes that the model encounters will only increase the accuracy of Table Cell Prediction

References

[1] Kasturi R. & O'Gorman L. & Govindaraju V. (2002). Document Image Analysis: A Primer. Retrieved from https://link.springer.com/article/10.1007/BF02703309

[2] Wang P. & Bunke H. (1997). Handbook of Character Recognition and Document Image Analysis. Retrieved from https://books.google.com/books/about/Handbook_o-f_Character_Recognition_and_Do.html?id=yn6DN5hAPywC

[3] Fang J., Mitra P., Tang Z. & Giles C. (2012). Table Header Detection and Classification. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, retrieved from https://dl.acm.org/citation.cfm?id=2900728.2900814

[4] Hassan T. & Baumgartner R. (2011). Table Recognition and Understanding from PDF Files. In Ninth International Conference on Document Analysis and Recognition, retrieved from https://ieeexplore.ieee.org/document/4377094

[5] Gilani A., Qasim S., Malik I. & Shafait F. (2017). Table Detection using Deep Learning. In 2017 14th IAPR International Conference on Document Analysis and Recognition, retrieved from https://ieeexplore.ieee.org/document/8270062

[6] Perez-Arriaga M., Estrada T. & Abad-Mota S. (2016). TAO: System for Table Detection and Extraction from PDF Documents. In FLAIRS Conference 2016, retrieved from https://www.semanticscholar.org/paper/TAO%3A-System-for-Table-Detection-and-Extr-action-from-Perez-Arriaga-Estrada/22c9f2d80e0f0546a54c82dbc9cfc9e68ce9a1ff

[7] Klampfl S., Jack K. & Kern R. (2014). A Comparison of Two Unsupervised Table Recognition Methods from Digital Scientific Articles. In D-Lib Magazine November/December 2014 Issue Volume 20 Number 11/12, retrieved from http://www.dlib.org/dlib/november14/klampfl/11klampfl.html

[8] Oro E. & Ruffolo M. (2009). PDF-TREX: An Approach for Recognizing and Extracting Tables from PDF Documents. In 2009 10th International Conference on Document Analysis and Recognition, retrieved from https://ieeexplore.ieee.org/document/5277546

[9] Fang J., Gao L., Bai K., Qiu R., Tao X. & Tang Z. (2011). A Table Detection Method for Multipage PDF Documents via Visual Separators and Tabular Structures. In 2011 International Conference on Document Analysis and Recognition, retrieved from https://ieeexplore.ieee.org/document/6065417

[10] Abadi M., Barham P., Chen J., Chen Z. & Davis A. (2016). TensorFlow: A System for Large-Scale Machine Learning. In Proceeding OSDI'16 Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, retrieved from https://dl.acm.org/citation.cfm?id=3026899

[11] Palekar R., Parab S. & Parikh D. (2017). Real Time License Plate Detection Using OpenCV and Tesseract. In 2017 International Conference on Communication and Signal Processing, retrieved from https://ieeexplore.ieee.org/document/8286778