Semantic Structure Identification using Image Detection Algorithms

Udaiveer Singh Chauhan, Nipun Diwan, Ananth Nath, Daniel Lee Whitenack

Purdue University Krannert School of Management

uchauhan@purdue.edu; ndiwan@purdue.edu; natha@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 within files using Image recognition Unsupervised Learning. 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 originating in tables).

Introduction

- Organizations operating in sectors such as financial, risk & compliance, consulting, or research are often challenged with **data wrangling** tasks
- On average, employees spend **two and a half hours** every day seeking data
- Key Data measures are usually **not directly stored** in companies' databases
- Data is usually found in a variety of formats such as **PDF, DOC, JPEG**, etc.
- Common data include accounting **tables, price indices, investment details, records of credit history**, etc.
- Extraction process is **manual, error-prone and time-consuming**.

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.

Figure 1. Screenshot from a PDF file containing a sample table

Goal

The main aim and motivation for this project is to:

1. Identify Tables in PDFs and other documents
2. Obtaining Word Clusters to get Semantically linked text in both Tables and Paragraphs of text

Challenges in sight

- To achieve semantic continuity* in the text extracted from documents
- To increase accuracy of the model in order to achieve a completely error prone tool that can solve problems for organizations

**The establishment of meaning and continuity in a model dealing with semantics*

Literature Review

Literature Title	Motivation for Research	Takeaway from Research
Real Time License Plate Detection Using OpenCV and Tesseract (Palekar & Parab, 2017)	Implementation of image to text conversion has been tricky due to shortcomings of previous image processing applications.	Used 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.
Table Detection in Document Images using Foreground and Background Features (Arif & Shafait, 2018)	Due to a variety of table layouts, it is difficult to design encoding techniques for extracting tables without any defined borders.	Proposed a deep learning based Faster R-CNN model for detection of tabular regions from document images based on the fact that tables usually contain more numerical data..
Table Recognition and Understanding from PDF Files (Hassan, 2007)	Presents a flexible method for detecting tables in PDF files not reliant upon one particular feature being present.	Defined a search method based on the principle of rectangular containment. Found a number of useful pre-processing steps useful in analyzing PDF files.
A Case Study on TensorFlow and Artificial Neural Networks (Vivekanandan, 2017)	TensorFlow performs very well on recognition problems, and the performance can be further improved by having more iterations.	Run our CNN models in TensorFlow and train them with hundreds of iterations.

Methodology

Methodology/Approach

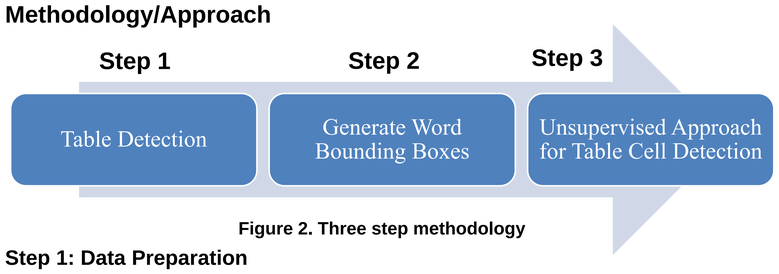
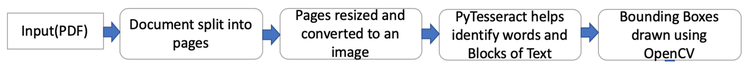


Figure 2. Three step methodology

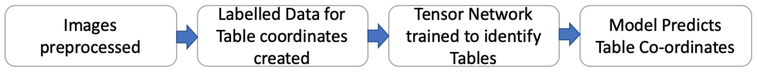
Step 1: Data Preparation

We have a total of 418 documents in PDF format with approximately 50 pages in each. Each document is a combination of text in the forms of paragraphs & tables, and images, which might further consist of useful data.



- PDF Files are converted into images
- Blocks of text identified and bounding boxes drawn using OpenCV

Step 2: Table Identification



- Images are transformed to bring uniformity and make it easier to train the tensor and predict whether an image contains a table

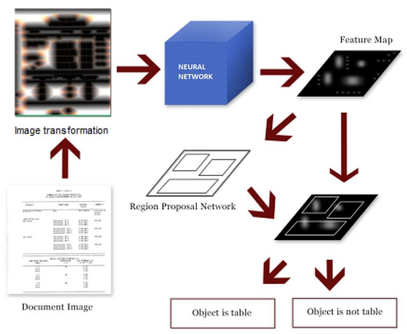
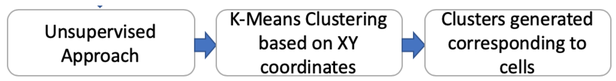


Figure 3. Preprocessing flow to differentiate tables from text

Step 3: Unsupervised Approach for Table Cell Detection



- PyTesseract is used to label the words within the coordinates predicted by the deep learning model
- K-Means Clustering is done in two steps:
  - Row wise Clustering: All words within the same row are clustered based on the X co-ordinates and horizontal distance between them
  - Column wise Clustering: All words within the same column are clustered based on their Y co-ordinates and their vertical distance

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.

First, all the words are bounded in boxes in **Green**
Second, sentence, line, and paragraph level aggregations are achieved

Results

The model designed will be able to detect tables in the pdfs and provide the end coordinates of the principal diagonal.

Step 1

The Table Detection algorithm was based on this paper “Table Detection using Deep Learning Azka Gilani, Shah Rukh Qasim, Imran Malik and Faisal Shafait”. We have used this approach for the initial phase of our project. The algorithm returns the output for the table coordinates in the below format:

"file": "data/val/9541_023.png", "objects": [{"bbox": [26, 900, 2451, 2184], "label": "table", "prob": 0.8744}, {"bbox": [133, 544, 2390, 1476], "label": "table", "prob": 0.536}]

Step 2

Using the output coordinates obtained from the previous step we were able to draw bounding boxes for each word within the table.

Step 3

From the above image, we can observe that the coordinates can be obtained for each word within the table.

- Using these coordinates we tried to cluster the words based on their coordinates using DBSCAN. This approach failed as it could not deal with single word clusters
- We then tried using K-means with the midpoint of the bounding boxes of each word. In this approach, the clusters failed as the distance between words of separate rows was also considered.
- We are currently working on another approach where we generate clusters based on columns and then on rows to obtain unique cells by using an intersection of the two sets.

Model Evaluation

There were 50 documents for validation. The model failed to cluster 15 documents while it was successful in clustering 10 documents. The remaining documents had partially identified clusters i.e. some words were missed. These documents had on average 70 percent accuracy.

Using the clustering on row approach, we were able to divide the tables into columns as per the below figures:

Ideal Output

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

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.

Conclusions

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.

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

Acknowledgements

We thank Professor Daniel Lee Whitenack for his constant guidance on this project.