Robust PDF Table Locator Using CNN/K-Means

Todd Macdonald, Nick Pether
Dept. of Computer Science, Stanford University
CS 229 Final Project

ABSTRACT

For our project, we sought to use computer vision and machine learning techniques to locate tables in PDF's with a better recall than existing software. Our approach consisted of three main steps:

1. Use OCR contour analysis to identify the characters in the image.
2. Run k-means on the locations of the characters to generate bounding boxes that might contain a table.
3. Identify the bounding boxes that contain tables using a CNN

Based on visual inspection, we achieved 100% recall with k-means for the bounding boxes, though low precision. Since we have several thousand images as testing and training examples, we have not quantified the precision for this intermediate step of our approach, though we will include quantitative results from our CNN in our final report. In comparison to our baseline, an open-source table extractor called Tabula, which has 99% precision but 50% recall, we see much promise.

MOTIVATION

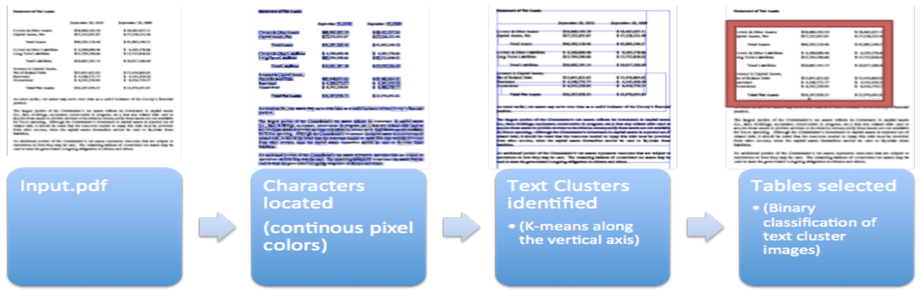
Machine learning and Data Science applications typically require large amounts of data stored in a computer readable format (e.g. a .csv file). Unfortunately, a significant portion of highly valuable and publicly available data, including most government documents and scientific papers, are published as tables embedded in Adobe PDF documents. This means that data scientists are regularly must endure re-enter existing data by hand. In order to bring about a future in which publically available data is not only 'available' but also 'useful' for research purposes, we set out to build use object detection techniques from computer vision to improve the way that tables are identified in PDF's.

DATASET & FEATURES

We trained and tested our model on PDF's obtained through California Common Sense's (CACS) Open Records Initiative (ori.cacs.org), a repository of Comprehensive Annual Financial Statements from different county and state governments. From this repository, we have scraped PDF financial statements from over two dozen counties, for an estimated total of 60,000 images of potential tables. We are currently training our classifier on about 2,000 example images. Every image that we use for train or test data must be hand-labeled using our labeling program.

Before the images are given as input to our k-means clustering algorithm, we use OCR contour analysis to preprocess the images. By detecting continuous contours of color within our images, we can identify the locations of all of the characters within an image. All of the features of the images are determined in the layers of our CNN (the CNN is provided scaled images as input).

METHODOLOGY



CHALLENGE #1: IDENTIFY "POTENTIAL" TABLES

Problem:
- A table could be anywhere on a page.
- We could check every possible cropped section of a given page using sliding windows of different sizes, but this would be slow and computationally expensive.

Solution:
- To solve this problem, we use an optimized version of k-means to reduce our 'potential tables' search space to only include blocks of text.
- In this approach, we represent each character as a point in 2-d or 1-d space.
- Next, we group together these characters into clusters and then put a bounding box around each cluster. Each of these bounding boxes is a 'potential' table. The partial images within these bounding boxes form the inputs to our image classifier.
- To efficiently consider different sized bounding boxes, we use the silhouette score of k-means over different numbers of clusters to identify the most ideal number of clusters to use for k-means on a given page.

CHALLENGE #2: CLASSIFY TABLES FROM CANDIDATES

Problem:
- There is a lack of useful images of "tables" vs "not tables" for machine learning.

Solution:
- We train an image classifier on 'potential' tables generated in the above step.
- Based on our research, we chose to use a 3-layer CNN classifier since it is shift invariant, excellent for image classification, and does not require extra pre-processing to extract features.
- The CNN labels all images of partial or full tables as positive. Thus, after this step, we have a classifier that recognizes table fragments!
- To classify entire tables, we then return the bounding box on all overlapping table fragments for a given page. These are your tables.

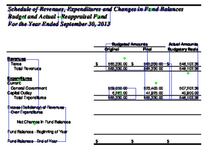
RESULTS & ANALYSIS

Part 1:Identification of Potential tables via K-clustering:

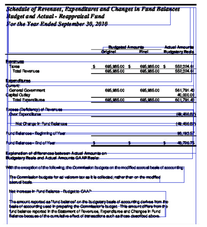
	Tabula (baseline)	K-means (2-d)	K-means (1-d)
Precision	0.99	0.32	0.48
Recall	0.55	1.0	1.0
F1	0.71	0.48	0.65

We compare the results of our algorithm for generating potential tables against those of Tabula, an existing open source tool for finding tables.

- K-means clustering of characters in two dimensions (x,y) did not generate useful results as tables would usually end up split between horizontally aligned but non-overlapping bounding boxes



- K-means clustering along the vertical axis avoided this problem



Our purpose in this step was simply to achieve high recall (success!) without *ridiculously* low precision. We just needed our potential tables to include all actual tables

CONCLUSION

Based on the high recall of our k-means algorithm, our approach shows great promise in how to identify tables from within scanned pdf's. Currently available tools, such as the open-source table extractor Tabula, do not work for scanned pdf's. Therefore, our tool shows how object detection techniques in computer vision can be used to make a more robust solution to the table identification problem.

In terms of future work, we have still to finish training our CNN on all of our labeled images by the time of our final report. Once this step is complete, we will perform analysis to tune our hyperparameters and other aspects of our current approach.