Transforming Legacy Engineering Reports into Useful Machine Readable Information

Sandya Wijaya,[1] Giovanni Celis Hernandez,[1] Peiying Guan,[1] Kimberly Baldwin,[2] Kevin Smith,[2] Nadia Mondegari [2]
[1]Data Science Discovery, University of California, Berkeley, CA 94720
[2]Bureau of Ocean Energy Management, Washington, DC, 20240

Abstract

The Bureau of Ocean Management (BOEM)'s well test reports record a formation's fluids and pressures, and is used to evaluate reservoir properties for oil & gas and carbon storage. These reports are legacy data – it will likely never be collected in the same location again. To ensure future availability in serving its important function, our project objective is to extract data from historical well test reports (low-quality PDFs) and convert them into a machine-readable format (DataFrame).

- Divergent styling, fonts, headers, and subsections are prevalent – making it difficult to standardize. This project builds a robust model adaptable to the variety.
- We leverage OCR, machine learning, computer vision to extract data reliably and enhance interpretability.

Background

Existing Preprocessing & OCR

Preprocess images of tables in a higher quality well test report using OpenCV

Inverted images, binarization, noise removal, dilation and erosion

Only binarization and grayscaling increased the quality of the image

Testing Optical Character Recognition (OCR) using Pytesseract

Poor performance in terms of how much characters are detected (NaNs) & accuracy ('FIELD' → 'FIELO')

```
   level  page_num  block_num  par_num  line_num  word_num  left  top  \
0      1         1          0        0         0         0     0    0
1      2         1          1        0         0         0     9   25
2      3         1          1        1         0         0     9   25
3      4         1          1        1         1         0     9   25
4      5         1          1        1         1         1     9   25
..   ...       ...        ...      ...       ...       ...   ...  ...
287    5         1         28        1         1         1   606   28
288    2         1         29        0         0         0   977  266
289    3         1         29        1         0         0   977  266
290    4         1         29        1         1         0   977  266
291    5         1         29        1         1         1   977  266

     width  height       conf   text
0     1024     789  -1.000000    NaN
1      173       9  -1.000000    NaN
2      173       9  -1.000000    NaN
3      173       9  -1.000000    NaN
4       36       9  44.691105  FIELO
..     ...     ...        ...    ...
287      5       8  94.684380      a
288     28      13  -1.000000    NaN
289     28      13  -1.000000    NaN
290     28      13  -1.000000    NaN
291     28      13  95.000000

[292 rows x 12 columns]
```

FIG 1. Previous OCR results using a grayscale binarized image

Technical Terms

Drill Stem Test (Well Test):

- A drill stem test measures a formations' pressure and fluids and returns data on formation properties (permeability, flow rate, and estimates of productivity)
- This data is extremely useful for oil and gas resource evaluation, potential carbon storage targets.

Optical Character Recognition (OCR):

- OCR is a technological process that utilizes algorithms and image processing techniques to identify individual characters or words within the image, discerns their patterns, and translates them into digital text.
- General Steps: Preprocesses, to enhance text recognition accuracy → Segmentation, to isolate individual characters or words from the image → Recognition, which uses pattern recognition or machine learning algorithms to interpret the shapes and patterns of the characters → Compare identified characters against a database of known characters

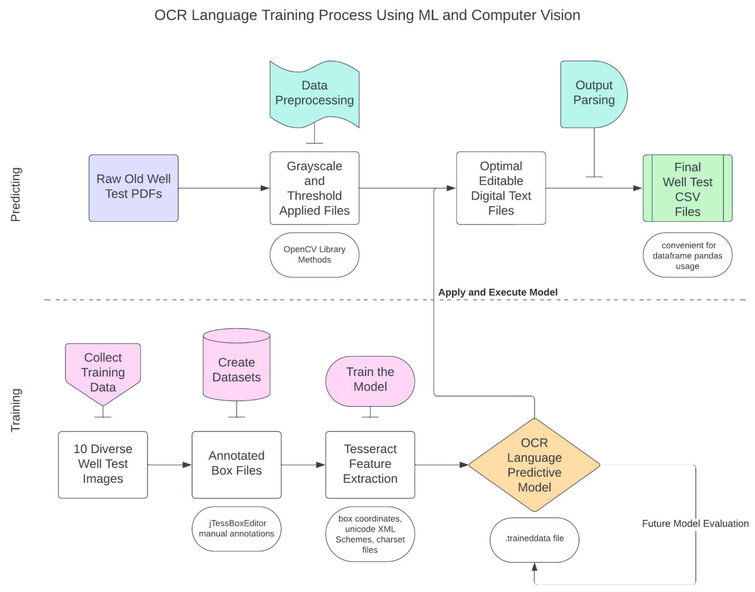
Research Goals

- Develop an **automated** system using OCR and diverse data processing techniques to convert these historical PDFs into DataFrames.
- Enhance robustness of **preprocessing** step so it accounts for style and quality variations to allow this model to be scalable for other PDFs.
- Integrate **quality checks** into the model to ensure the accuracy and reliability of the extracted data.

Methods

Custom Box Training and Computer Vision

- This method involves creating a pre-trained Tesseract model that is passed in as a parameter to an OCR system and returns an optimized output.
- By gathering and structuring each individual image file, we generate and annotate box files that represents identifiable characters (letters, numbers, symbols).
- After manually correcting these box files, we execute a multi-step training process using Tesseract commands.
- The final output is the obtained table as a dataframe.



Tabula

- Tabula is a free third-party tool to digitize tables from PDFs.
- Process: Upload PDF to Tabula, manually select areas to extract from, export in desired format (CSV, JSON, ZIP, etc.)

"pdftools" library in R

- The 'pdftools' package is an R interface to PDF Toolkit (PDFtk) that can extract text and manipulate PDFs.
- Process: Extract all text in a PDF using 'pdf_text( )', eliminate extra spaces, use Regex to isolate specific information, then convert cleaned list into a DataFrame.

Miscellaneous Achievements

Quantifying accuracy on keywords

- 'accuracy_on_keywords(ocr_result)' function takes in an OCR result and outputs accuracy percentage in detected keywords (e.g. 'REPORT' 'PAGE').
- It utilizes a set of predetermined keywords and their frequencies, iterates through these keywords, and employs Regex to count their occurrences in the input.

Choosing relevant pages from PDF

- As our final method takes in images (not PDFs), we developed steps to identify useful pages in a PDF. The output lists page paths that contain relevant tables.
- First, pdf2image converts every page into an image.
- Then, we identify if a page contains a table using Black content assessment (45% accuracy) or Canny Edge Detection (62% accuracy).
- Lastly, we filter for *relevant* tables – field reports, not summaries – using simple OCR (83% accuracy) or Template Matching by OpenCV (55% accuracy).

```
wanted_table_paths

['Data/P00296_B014.pdf-page11.jpg',
 'Data/P00296_B014.pdf-page12.jpg',
 'Data/P00296_B014.pdf-page13.jpg',
 'Data/P00296_B014.pdf-page15.jpg',
 'Data/P00296_B014.pdf-page16.jpg'
```

FIG 3. Example output; a list of paths that indicate pages with relevant tables.

Challenges & Considerations

Problem	*Solutions*
Multiple-line headers get mistaken for observations.	Manually add headers for subset of PDFs
Hard to evaluate model accuracy as data does not exist.	Manually create "correct answers" for subset of PDFs
Low quality of scanned PDF files cause misreadings for Regex	Enhance preprocessing, manually fix last details
Accurate selection of desired pages need high computation.	Dumbed-down OCR, template matching, Regex

Final Product Results

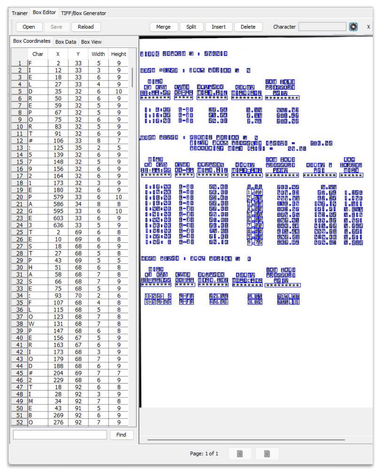


FIG 4. Box annotations in jTessBox Editor for model and dataset training

	Time Of Day HH : MM : SS	Date DD-MM	Elapsed Time:Min	Delta Time:Min	Bot Hole Pressure(PSIA)	Delta PSI	LOG Horner Time
0	0:21:59	9-FB	0.01	0.00	0.00	0.00	NaN
1	0:23:59	9-FB	0.01	0.05	1097.74	0.00	NaN
2	0:24:59	9-FB	0.99	1.00	1165.12	67.38	NaN
3	0:25:59	9-FB	1.99	2.00	1209.76	44.64	NaN
4	0:26:59	9-FB	2.99	3.00	1244.36	34.60	NaN
5	00:27:59	9-FB	3.99	4.00	1266.41	22.05	NaN
6	0:28:59	9-FB	4.99	5.00	1285.66	19.25	NaN
7	0:29:59	9-FB	5.99	6.00	1302.12	16.46	NaN
8	00:30:59	9-FB	6.99	7.00	1314.68	12.56	NaN
9	00:31:59	9-FB	7.99	8.00	1325.98	11.30	NaN
10	00:32:59	9-FB	8.99	9.00	1336.30	10.32	NaN
11	00:33:59	9-FB	9.99	10.00	1345.23	8.93	NaN
12	0:35:59	9-FB	11.99	12.00	1360.16	14.93	NaN
13	0:37:59	9-FB	13.99	14.00	1372.01	11.85	NaN
14	00:39:59	9-FB	15.99	16.00	1383.18	11.17	NaN
15	0:41:59	9-FB	17.99	18.00	1392.94	9.76	NaN
16	0:43:59	9-FB	19.99	20.00	1401.59	8.65	NaN
17	0:45:59	9-FB	21.99	22.00	1408.84	7.25	NaN
18	00:47:59	9-FB	23.99	24.00	1415.54	6.70	NaN
19	0:49:59	9-FB	25.99	26.00	1421.96	6.42	NaN
20	0:51:59	9-FB	27.99	28.00	1427.96	6.00	NaN
21	0:53:59	9-FB	29.99	30.00	1433.96	6.00	NaN
22	0:58:59	9-FB	34.99	35.00	1446.05	12.09	NaN
23	1:3:59	9-FB	39.99	40.00	1457.95	11.90	NaN
24	1:8:59	9-FB	44.99	45.00	1367.45	9.50	NaN
25	1:10:8	9-FB	46.13	46.14	1000.81	2.36	NaN
26	1:12:00	9-FB	0.01	0.00	0.00	0.00	NaN

english.NA.exp00.txt.csv

FIG 5. Optimized Pandas DataFrame for Single Well Test Results Post-Modeling

Custom box files are created for each identifiable characters on the scanned PDF and manually fixed in Fig 1, then the trained model outputs the detected DataFrame in Fig 2.

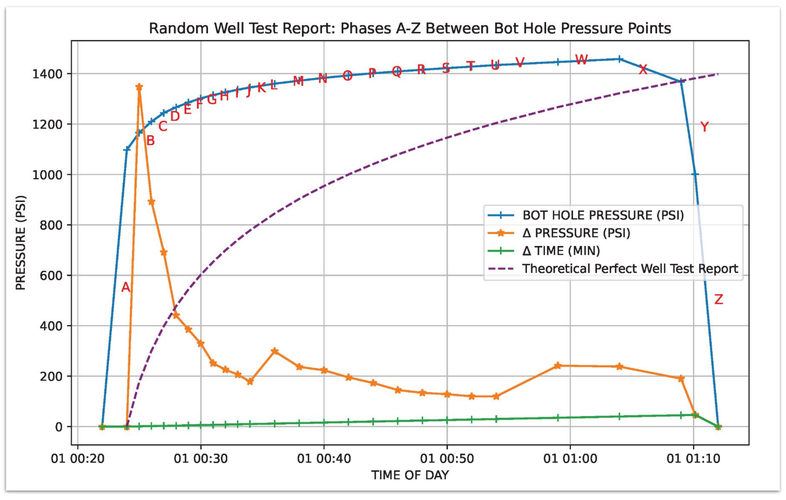


FIG 6. Single Well Test Report DataFrame to Visualization

Phase Transition Recognition in Fig 6.

- Represents distinct phases within the well testing process. (A-C) running in hole; (D-M) initial flow period; (N-S) initial buildup; (T-W) final buildup; (X-Z) pulling out of hole.

Perfect Well Test Reference in Fig 6.

- Logarithmic line as a reference to compare observed phases against an ideal stretched out scenario.

Early Anomaly Indicators in Fig 6.

- Analyzing derivative of pressure (delta pressure) provides insights of rate of change of pressure during a well drill test.
- Sudden spikes/drops indicate critical events such as fluid influx or wellbore instability. Detecting such rapid pressure changes allows early indication of anomalies & irregularities.

Conclusion and Future Work

- Make the process of choosing which pages to extract tables from more seamless and accurate – explore feasibility to change the model so it accepts a PDF.
- Train the model on more PDFs, and continue to fine tune model based on discoveries made from this.
- Expand model to allow classifications of outputs for insight on how pressure changes with well tests over time.

Acknowledgements: The first authors would like to thank their peers in the project for their guidance, collaboration, and support throughout the project. We would also like to thank the second authors, Kimberly Baldwin, Kevin Smith, and Nadia Mondegari for their constant support and guidance during this research project. Finally, we also want to thank the Data Science Discovery Program 2023 at UC Berkeley for the opportunity to work on this project through the program.

Bradski, G. (2000). The OpenCV Library. *Journal of Software Tools.*
Frerebeau N (2019). tabula: An R Package for Analysis, Seriation, and Visualization of Archaeological Count Data. *Journal of Open Source Software.*
Ooms, J. (2023). tesseract: Open Source OCR Engine. Python package version 5.2.1. Retrieved from docs.ropensci.org/tesseract
Xie Y (2014). pdftools: Text Extraction, Rendering and Converting of PDF Documents. R package version 2.3.1. CRAN.R-project.org/package=pdftools