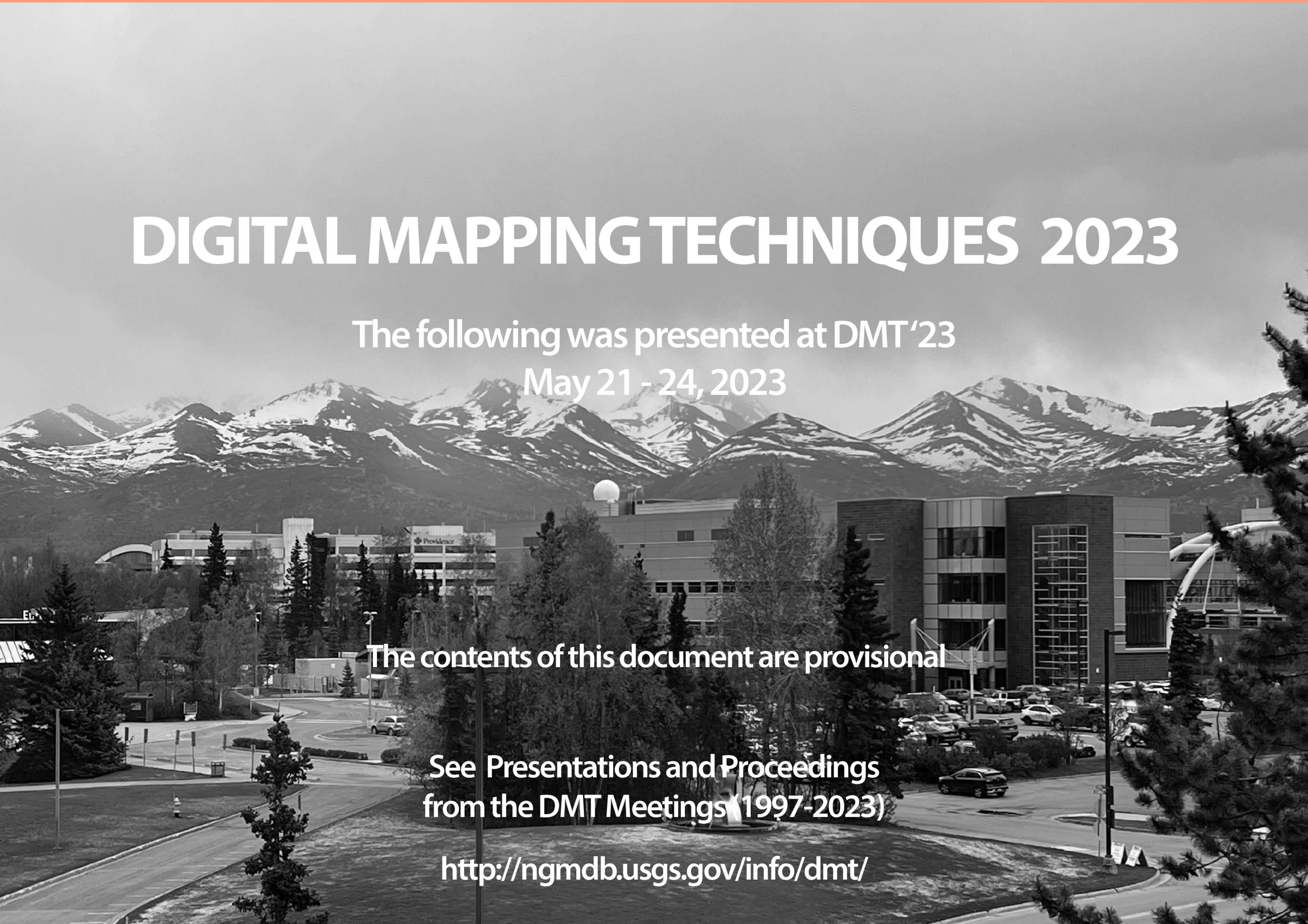


DIGITAL MAPPING TECHNIQUES 2023

The following was presented at DMT '23
May 21 - 24, 2023

The contents of this document are provisional

See Presentations and Proceedings
from the DMT Meetings (1997-2023)

http://ngmdb.usgs.gov/info/dmt/

<u>Title</u>: From Paper to AI: Optimizing geologic mapping workflows

<u>Authors</u>: Darby DeBruhl, Gerald Krieger, Robert Clark

<u>Affiliation</u>: South Carolina Geological Survey

As the South Carolina Geological Survey (SCGS) enters its 198th year of operation, the importance of addressing its backlog of historical maps and drill logs has become increasingly evident. The process of scanning and digitizing paper products can be incredibly time-consuming, depending on the complexity of the material at hand. Up until this point, all historic map digitization at SCGS has been done heads-up, by hand, click-by-click; traditional raster-to-vector conversion tools have presented their own challenges in their inability to distinguish the desired features of a document (*ex. contact lines*) from the undesired ones (*ex. topographic lines, such as those that may be present in a basemap*). The process of digitizing drill logs has similar obstacles; of the logs that are handwritten, many are sloppy enough to be undetectable by traditional optical character recognition (OCR), and the wide variety of differently-formatted log sheets over the years rules out many options for automation. Spurred by the ongoing conversation surrounding artificial intelligence (AI) online and in the media, SCGS has explored a range of AI-driven tools with the potential to enhance our digitization workflow. The results, while imperfect, remain promising nonetheless.

1. **Slide 1:** Title

2. **Slide 2:** Introduction

3. **Slide 3:** Artificial intelligence is advancing at a rapid pace. Just a decade ago, the idea of AI being a part of everyday life was almost inconceivable. Fast forward to the present day, and the term is nearly ubiquitous in every facet of the American economy. From the impact it's expected to have on the nation's GDP to the striking growth projected in the value of the AI industry, we are seeing rapid advancements in the development of this groundbreaking technology.

4. **Slide 4:** In the Earth Science fields, such as Geology, Natural Hazards, Climatology, Remote Sensing, and Hydrology we are already seeing its impact in countless ways.

5. **Slide 5:** As this technology evolves, we wanted to see how it could be used in our organization. This presentation highlights the use of AI and ML in the enhancement of our geologic mapping workflows. Our research leverages ArcGIS Pro's Deep Learning toolset, Google's Cloud Vision API, Amazon Textract, and online LLMs like ChatGPT and Bard to improve the digitization of scanned maps and the cataloging of historic drill hole logs. Our findings demonstrate that the incorporation of AI tools into our digitization workflow has the potential to expedite the conversion process significantly.

6. **Slide 6: Deep Learning at SCGS (1/2)**
    a. We trained a pixel classification model on the Lowndesville quadrangle, visible on the left. We chose the Lowndesville quad because 1), it had already been digitized, so we didn't need to expend any additional effort to create training data, and 2) because it's

part of a larger map series that we were interested in digitizing the rest of, which would be a good way to test the efficacy of the final model anyway.

b. Our workflow is comprised of 3 major steps, each of which is discussed more thoroughly in the Deep Learning section of our poster:
    i. **1. Processing the scanned image** -- This involved things like tweaking the contrast, brightness, etc. according to the needs of the scanned image to make the contacts pop as much as possible. These things could be done in virtually any image editing program, but we used Photoshop. This step takes less than 5 minutes, but it hugely improves the quality of the final model.
    ii. **2. Training and running the pixel classification model** -- After processing the image and tidying up the pre-vectorized data that we had already created for Lowndesville, it's ready to be used as training data for the deep learning model. This was by far the most computationally intensive part of the process – it can take hours to days and even weeks depending on the size of your training dataset and your hardware limitations. We had unique requirements to keep in mind, because most applications of deep learning in ArcGIS are heavily focused on multispectral imagery, like satellite imagery or aerial photos. There were not a lot of resources on dealing with single-band, monochromatic imagery like a scanned map. Despite this, we were able to find some parameters that worked for us (DeepLabV3 model type + ResNet34 architecture), and the middle image of this sequence was the result of running the finished model on its own training data. The model went through each pixel in the image and sorted it into one of the classes that we defined, creating a cleanly classified raster. It was able to discern the features we wanted, which are the contact lines, from the features we didn't, like the contour lines, road lines, labels, etc. That distinction previously would not have been possible for us without human intervention, and all of those extra features would have made traditional raster to vector conversation tools an impractical option for this process.
    iii. **3. Classified raster to vector lines** -- This step is just a matter of getting the now-classified raster back into vector format. Any number of raster-to-vector tools can be used to convert this back into line features, but we used the ArcScan extension for ArcMap. In the third image of this sequence, the red lines are what was done automatically, and the green lines are the bits that we had to do manually to fill in the gaps. QAQC is important -- you can see that there are still places where a human has to fix some mistakes -- but they're few and far between enough for this workflow to still be a massive improvement in efficiency.

7. **Slide 7: Deep Learning at SCGS (2/2)**
Final results on the rest of the map series -- Lowndesville on the far left, with three adjacent quads to the right. Once again, the red lines are the parts done autonomously by the computer, and the green lines are edits made by a human reviewer. While it may have taken several hours or days to do this amount of work the old-fashioned way, with this new workflow, the contact lines for the entire rest of the map series were digitized in around an hour.

8. **Slide 8: OCR at SCGS (1/7)**
   The first of two methods we used for OCR was Google Cloud Vision, an AI-based deep image analysis tool to help digitize map units and their corresponding explanations. This service provided us with deep image analysis capabilities, which we used to detect specific text within scans of our old paper geologic maps. The information extracted by the tool was then programmatically stored in the appropriate fields of a corresponding DescriptionOfMapUnits table, located in a GeMS-compliant geodatabase that was created for the quadrangle at hand.
9. **Slide 9: OCR at SCGS (2/7)**
   This slide shows screenshots from the original scanned paper map we used during this project, as well as a screenshot with digital DMU data that has been input via a combination of ArcPy and OCR from Google Cloud Vision.
   We chose to test this workflow on the Lowndesville quadrangle, mostly due to how it returned favorable results in the digitization portion of the project, as outlined earlier in the presentation.
10. **Slide 10: OCR at SCGS (3/7)**
    These are snippets from the two codes we used to perform the operation, with the one on the left performing the Google Vision text extraction, and the one to the right making use of ArcPy to populate the geodatabase with that data.
    Google provides support for their service through a wide variety of programming languages, including Java, Python, and C#; however, we chose to use Python due to the ease of integration with ArcGIS's ArcPy Python module.
11. **Slide 11: OCR at SCGS (4/7)**
    To compensate for the lack of an in-house programmer, we heavily relied on web-based AI chatbots powered by large language models (LLMs) such as Google's Bard and OpenAI's ChatGPT. The screenshots featured on this slide show some examples of these tools making modifications and improving code, helping to debug and detect errors, and simply writing whole blocks of code by themselves. We were overall surprised by the accuracy of these results; despite OpenAI's lack of association with Google, and ArcPy's status as a proprietary package, ChatGPT demonstrated a strong "knowledge" of coding and module-specific tasks.
12. **Slide 12: OCR at SCGS (5/7)**
    The second software we used in this project when it came to Optical Character Recognition was Amazon Textract. Amazon Textract is an Optical Character Recognition service provided as part of the Amazon Web Services (AWS) suite of cloud computing platforms. Textract uses machine learning algorithms pooled from various AWS services to help extract text and structured data for various types of documents. What sets Textract apart is its ability to analyze documents that do not follow a predefined template, such as a form or simple set of paragraphs. Instead, it leverages machine learning to extract text and data that align with a user-specified query, including typed or handwritten text. This feature is particularly useful for scenarios like processing the handwritten drill logs we have collected at the Survey.

13. **Slide 13: OCR at SCGS (6/7)**
    One very unique feature of Textract we really took advantage of is its ability to use queries. As you can see in this slide's demo screenshots, you can feed in a document, ask the tool to answer a wide range of questions within the document, and it will quickly return answers, without

requiring the user to specify any context or background information. We asked the queries shown here, in the middle, and the answers unique to this drill log appear, despite the log being handwritten and in questionable handwriting. Additionally, in the snippet on the right, you can see how you can also search for a specific word across drill logs -- in this example, sand -- and Textract will be able to pull out where this occurs and in what context.

14. **Slide 14: OCR at SCGS (7/7)**
    Once the process is complete, Textract can create a csv including all the data requested through the queries into one easy-to-use, tabular format, shown in the screenshot above. This data can then be processed in a variety of ways, including within ArcGIS through ArcPy.
    As you can see in the image of the spreadsheet we created, we asked Textract the following queries:
    What is the drill hole number?
    What is the date?
    Who was it drilled by?
    What is the elevation?
    What are the UTM coordinate numbers?
    Who was it logged by?
    What is the description of the location?

15. **Slide 15:** As we move forward into what many experts call “The Age of Artificial Intelligence,” we must remain cognizant of the potential dangers that could result from becoming overly reliant on this technology.

16. **Slide 16:** While there are things to be wary of moving forward, we would also be remiss not to acknowledge the positives. In industries ranging from Healthcare to Transportation, Manufacturing, Financial Services, Real Estate, and Scientific Research, AI has already proven to be a very useful tool. The use of AI in geologic research will only continue to grow, and through this presentation, we aim to provide a valuable framework for other State Surveys and to extend an open invitation to collaborate in this endeavor.

17. **Slide 17:** Closing

SOUTH CAROLINA
GEOLOGICAL SURVEY

From Paper to AI: Optimizing Geologic Mapping Workflows

MEET THE TEAM

Robert Clark

GIS Manager - LWC

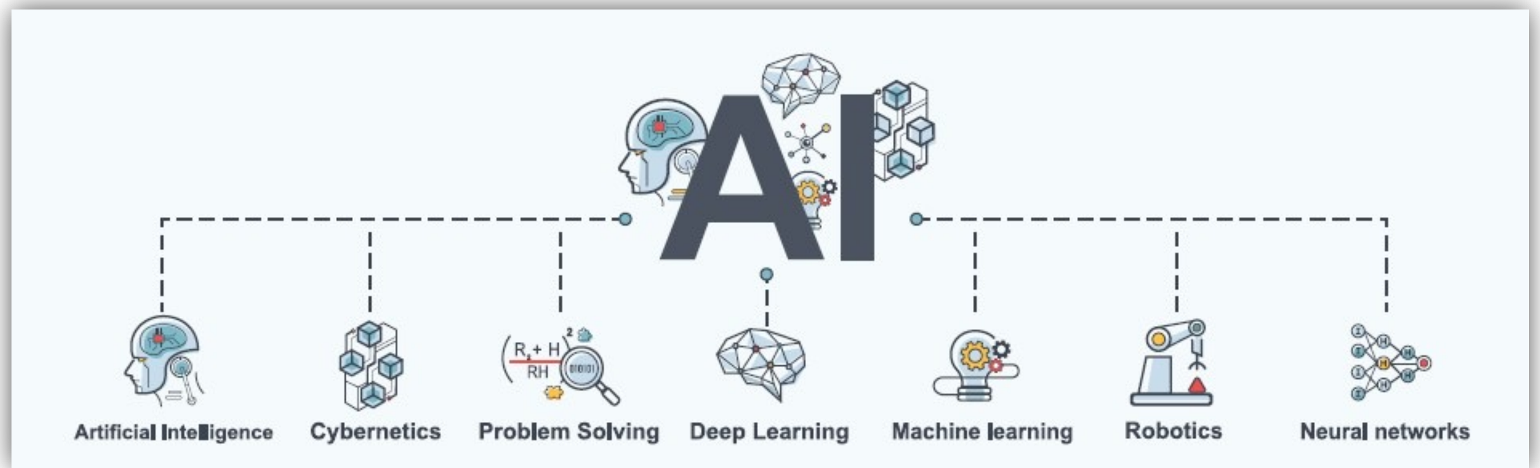
Darby DeBruhl

GIS Analyst - SCGS

Gerald Krieger

Cartographer - SCGS

ARTIFICIAL INTELLIGENCE



GROWTH

Expected growth of 21% to the nation's GDP by 2030.

PRODUCTIVITY

64% of businesses expect AI to increase productivity in their business operations.

VALUE

Expected to reach a $407 billion dollar value by 2027, up from $86 billion in 2022.

Source: Forbes Advisor

ARTIFICIAL INTELLIGENCE
IN THE GEOSCIENCES

GEOLOGY

ENERGY

NATURAL HAZARDS

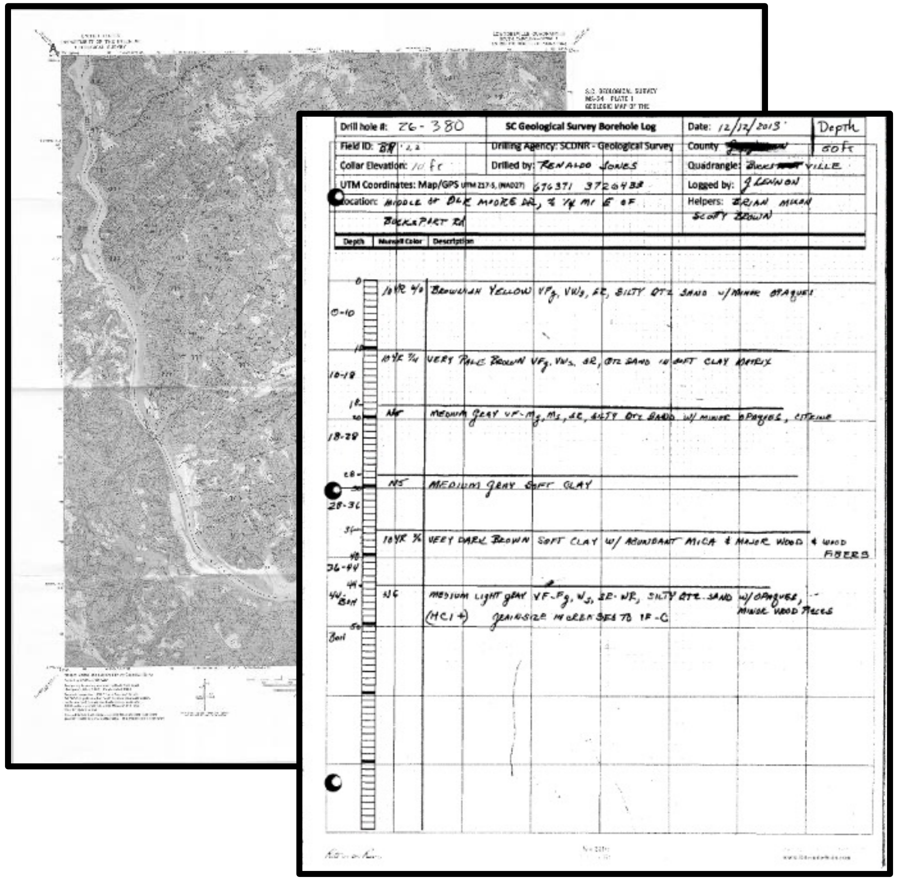
CLIMATOLOGY

REMOTE SENSING

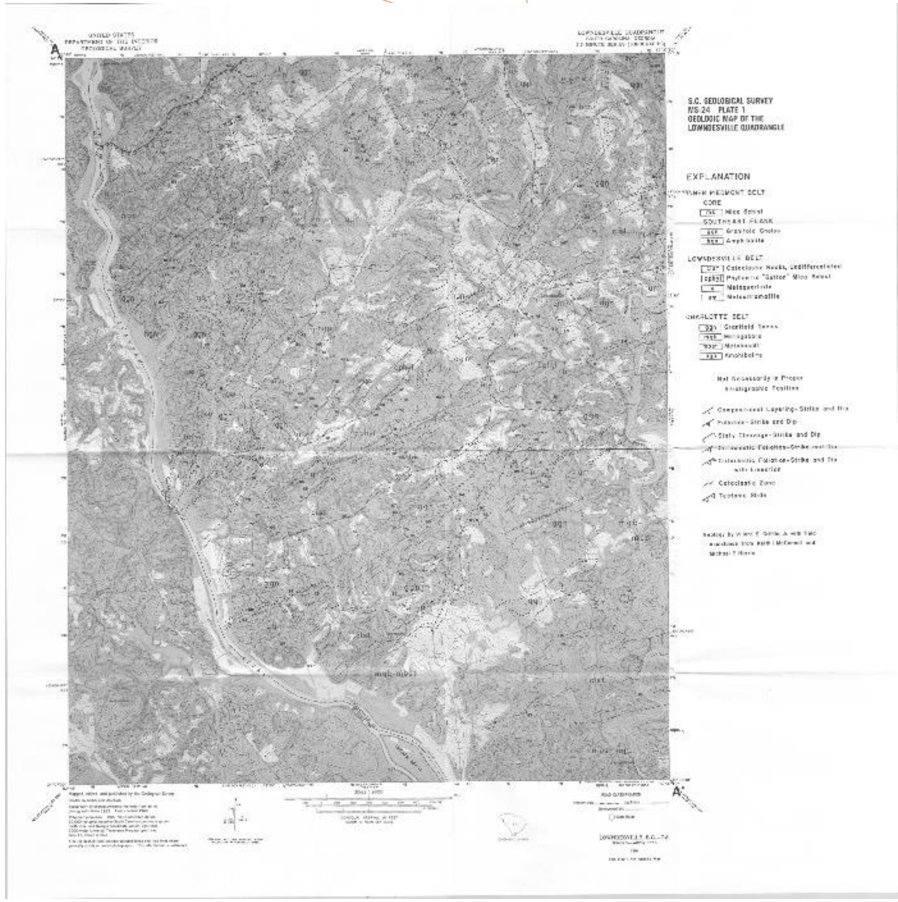
HYDROLOGY

ARTIFICIAL INTELLIGENCE
AT THE SOUTH CAROLINA GEOLOGICAL SURVEY

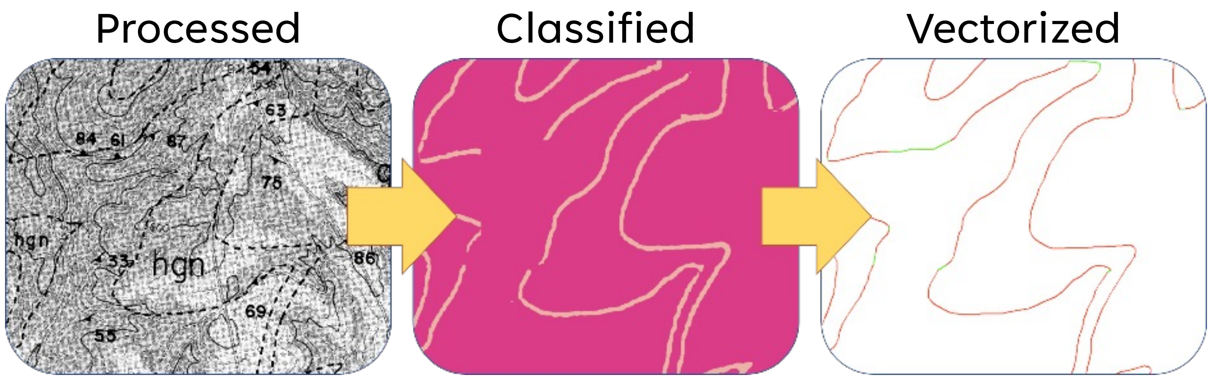
- Why are we using it?
  - Legacy Data
  - Historic Maps
  - Historic Drill Logs
- AI Tools we used…
  - ChatGPT
  - Google Cloud Vision
  - Amazon Textract
  - ESRI Deep Learning
- Benefits provided…
  - Saves time
  - Improves Accuracy

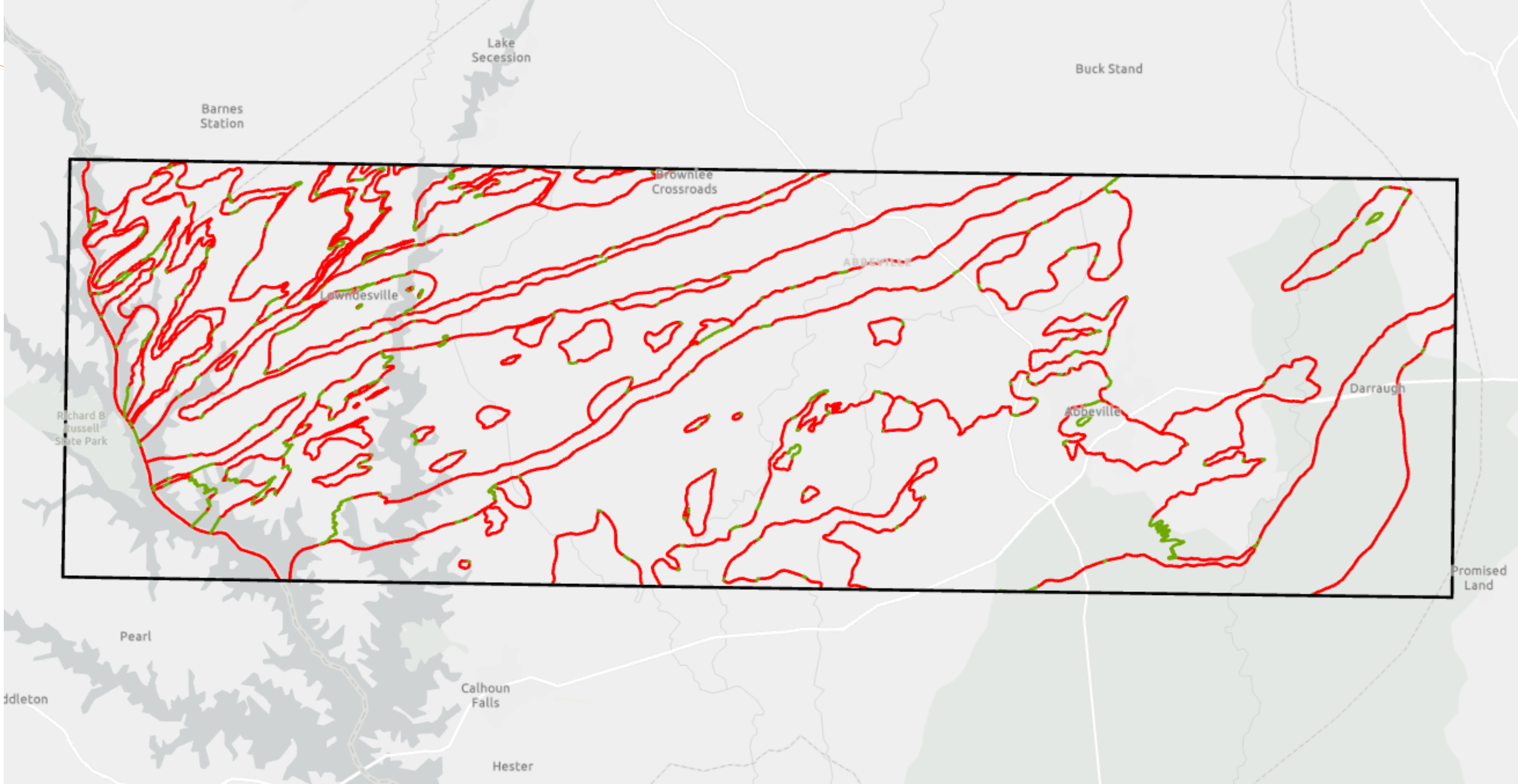


DEEP LEARNING



- AI > Machine Learning > Deep Learning
- Neural networks – based on the structure of the human brain
- 2019: Esri releases Deep Learning capabilities in ArcGIS Pro 2.3



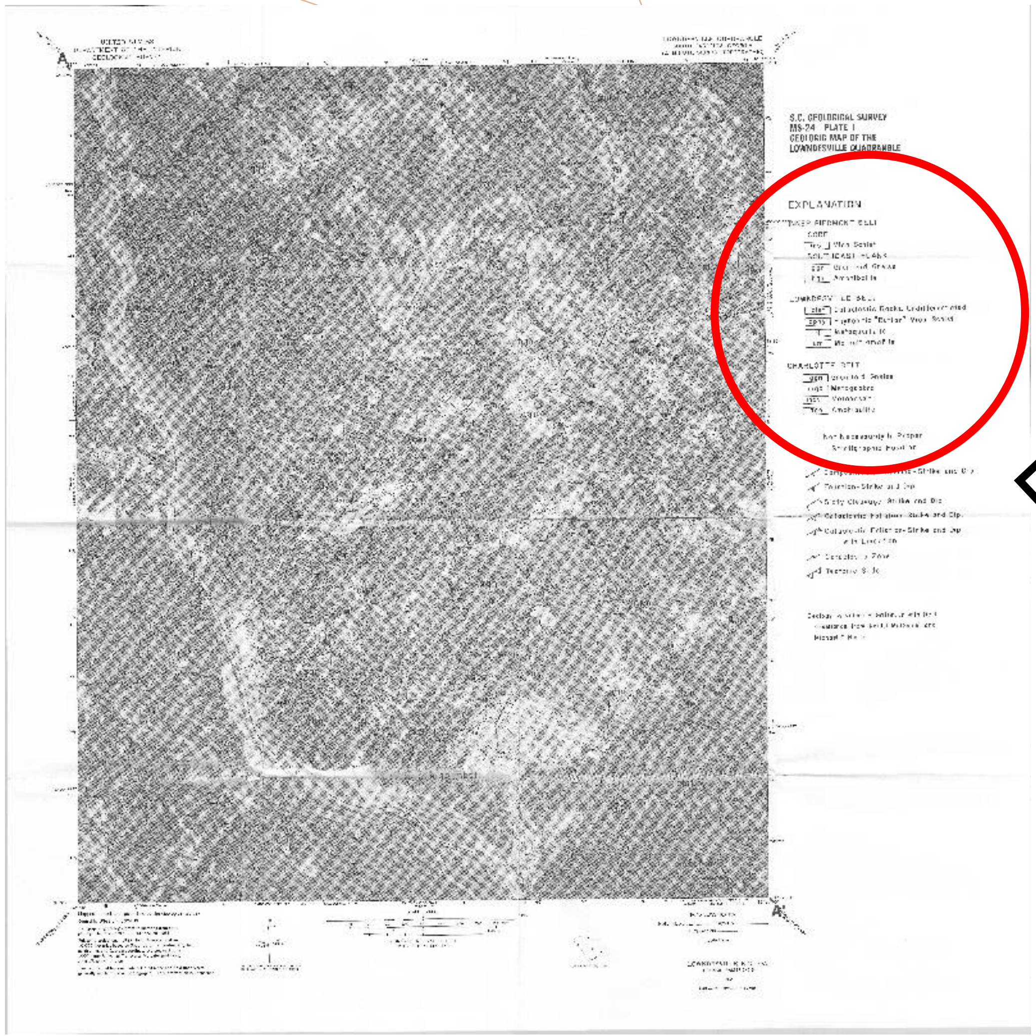


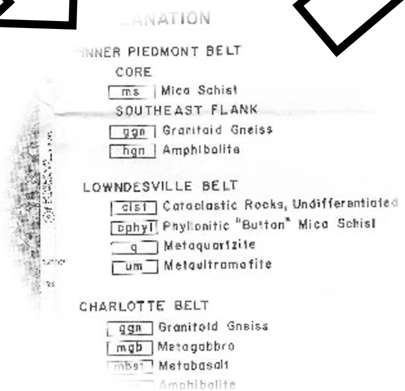
Lake Secession
Buck Stand
Barnes Station
Brownlee Crossroads
Lowndesville
Darraugh
Abbeville
Promised Land
Pearl
Calhoun Falls
Hester

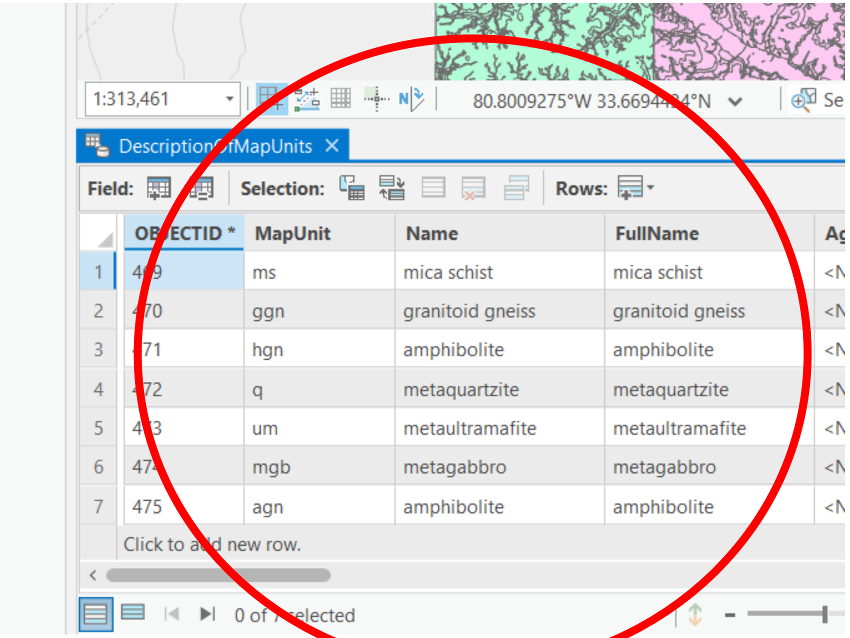
OPTICAL CHARACTER RECOGNITION

- Google Cloud Vision API
  - An AI-based tool within the Google Cloud Suite of products
  - Includes deep image analysis capabilities allowing for the extraction of Map Units as well as their corresponding explanations
  - A .csv file with the relevant data was created by the tool, allowing for easy input into a file geodatabase table using ArcPy

OPTICAL CHARACTER RECOGNITION

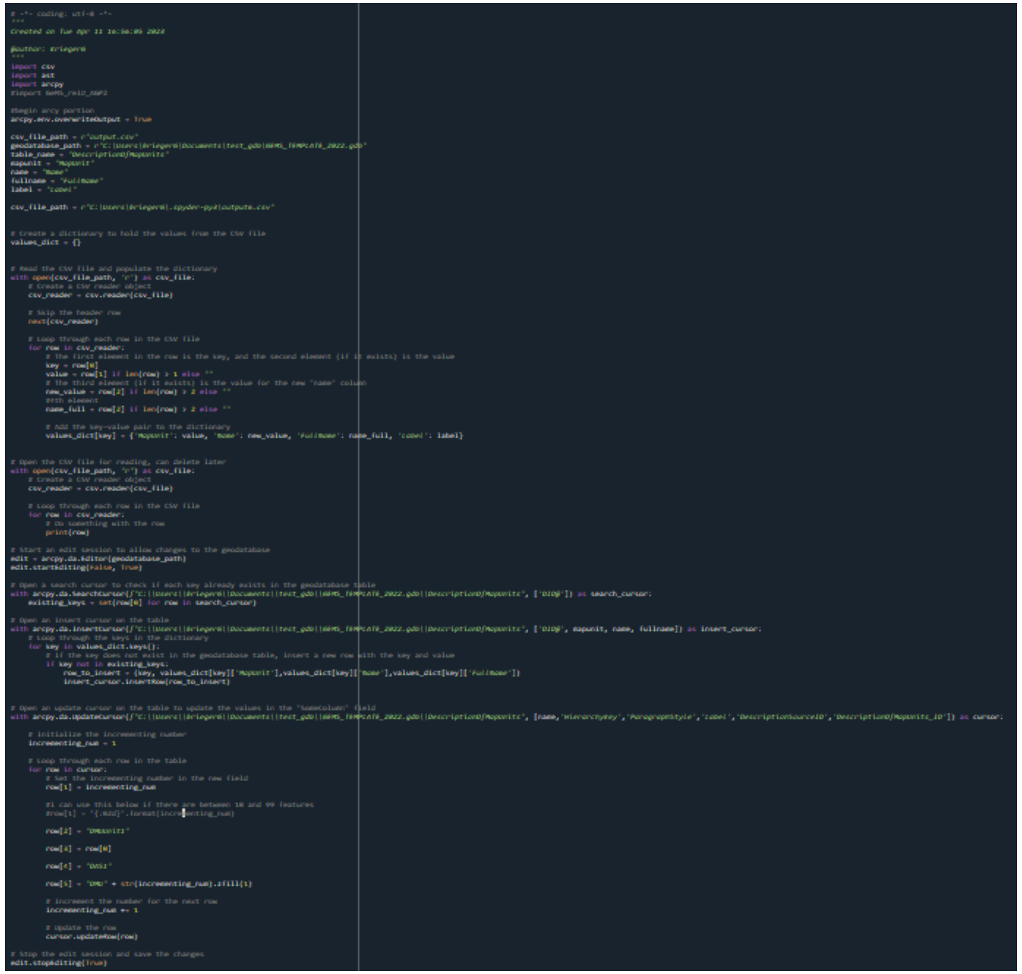






OPTICAL CHARACTER RECOGNITION

ArcPy



ChatGPT

OPTICAL CHARACTER RECOGNITION

Improvements / Modifications

can you modify this code to also print the names of the columns? # remove duplicates with open('output4.csv', 'r', newline='') as input_file, open('output5.csv', 'w', newline='') as

Sure! Here's the modified code that also prints the names of the columns:

```python
# remove duplicates
with open('output4.csv', 'r', newline='') as input_file, open('output5.csv',
    # Set up the CSV reader and writer
    reader = csv.reader(input_file)
    writer = csv.writer(output_file)

    # Initialize a set to keep track of rows that have already been written
    seen = set()

    # Initialize a counter to keep track of the row number
    row_num = 0

    # Loop through the rows of the input file
    for row in reader:
        # Convert the row to a tuple so that it can be added to the set
        row_tuple = tuple(row)

        # Check if the row has already been written to the output file
        if row_tuple in seen:
            continue

        row_num += 1

        # Delete rows 1, 2, and 4
        if row_num in [1, 2, 4, 8]:
            continue

        # If the row has not been seen before, add it to the set and write i
        seen.add(row_tuple)

        # If this is the first row, print the column names
        if row_num == 1:
            print("Column Names: ", row)

        writer.writerow(row)
```

This code adds an `if` statement to check if the current row being processed is the first row (i.e., the row that contains the column names). If it is the first row, it prints the column names to the console.

Whole Blocks of Code

can you create a code that can put values from a csv into a column of a feature class of a geodatabase?

Yes, I can provide you with sample Python code that uses the ArcPy library to read values from a CSV file and insert them into a column of a feature class in a geodatabase.

Before we start, make sure to replace the variables `csv_file_path`, `feature_class_path`, and `column_name` with the path to your CSV file, the path to your feature class, and the name of the column you want to insert values into.

Here's the code:

```python
import arcpy

csv_file_path = r"C:\data\input.csv"
feature_class_path = r"C:\data\my_geodatabase.gdb\my_feature_class"
column_name = "my_column"

# Create a dictionary to hold the values from the CSV file
values_dict = {}

# Read the CSV file and populate the dictionary
with open(csv_file_path, 'r') as csv_file:
    for line in csv_file:
        # Split the line by comma and strip whitespace
        parts = line.strip().split(',')

        # The first part is the key, and the second part is the value
        key = parts[0]
        value = parts[1]

        # Add the key-value pair to the dictionary
        values_dict[key] = value

# Start an edit session to allow changes to the geodatabase
edit = arcpy.da.Editor(feature_class_path)
edit.startEditing(False, True)

# Open an update cursor on the feature class
with arcpy.da.UpdateCursor(feature_class_path, ['OID@', column_name]) as cur
    for row in cursor:
        # Get the key for this row (assumes the OID is the key)
        key = row[0]

        # If the key exists in the dictionary, update the value
        if key in values_dict:
            row[1] = values_dict[key]
            cursor.updateRow(row)

# Stop the edit session and save the changes
edit.stopEditing(True)
```

This code reads the CSV file and populates a dictionary with key-value pairs. It then starts an edit session on the geodatabase and opens an update cursor on the feature class. For each row in the cursor, it gets the key (which assumes the ObjectID is the key) and checks if it exists in the dictionary. If it does, it updates the value in the specified column. Finally, it stops the edit session and saves the changes.

Error Debugging

Traceback (most recent call last): File "C:\Users\kriegerG\.spyder-py3\word_detection_draft.py", line 101, in <module> cell = ws['A'].find(str(word.upper())) AttributeError: 'tuple' object has no attribute 'find'

View other drafts

The error you are getting is because you are trying to call the `find` method on a tuple. A tuple is an immutable sequence of objects, and does not have any methods.

To fix this error, you need to change the way you are accessing the cell in the worksheet. Instead of using the `find` method, you can use the `cell_value` property.

The following code shows how to fix the error:

```python
cell = ws['A'].cell_value
```

Use code with caution. Learn more

This will return the value of the cell in the first column of the worksheet.

Here is a more detailed explanation of the error:

- The `AttributeError` exception is raised when an object does not have the attribute that is being accessed.
- The error message `'tuple' object has no attribute 'find'` tells you that the `find` method does not exist on tuples.
- The `find` method is a method of strings, and is used to find the first occurrence of a substring in a string.
- To access the value of a cell in a worksheet, you can use the `cell_value` property.
- The `cell_value` property returns the value of the cell as a string.

I hope this helps! Let me know if you have any other questions.

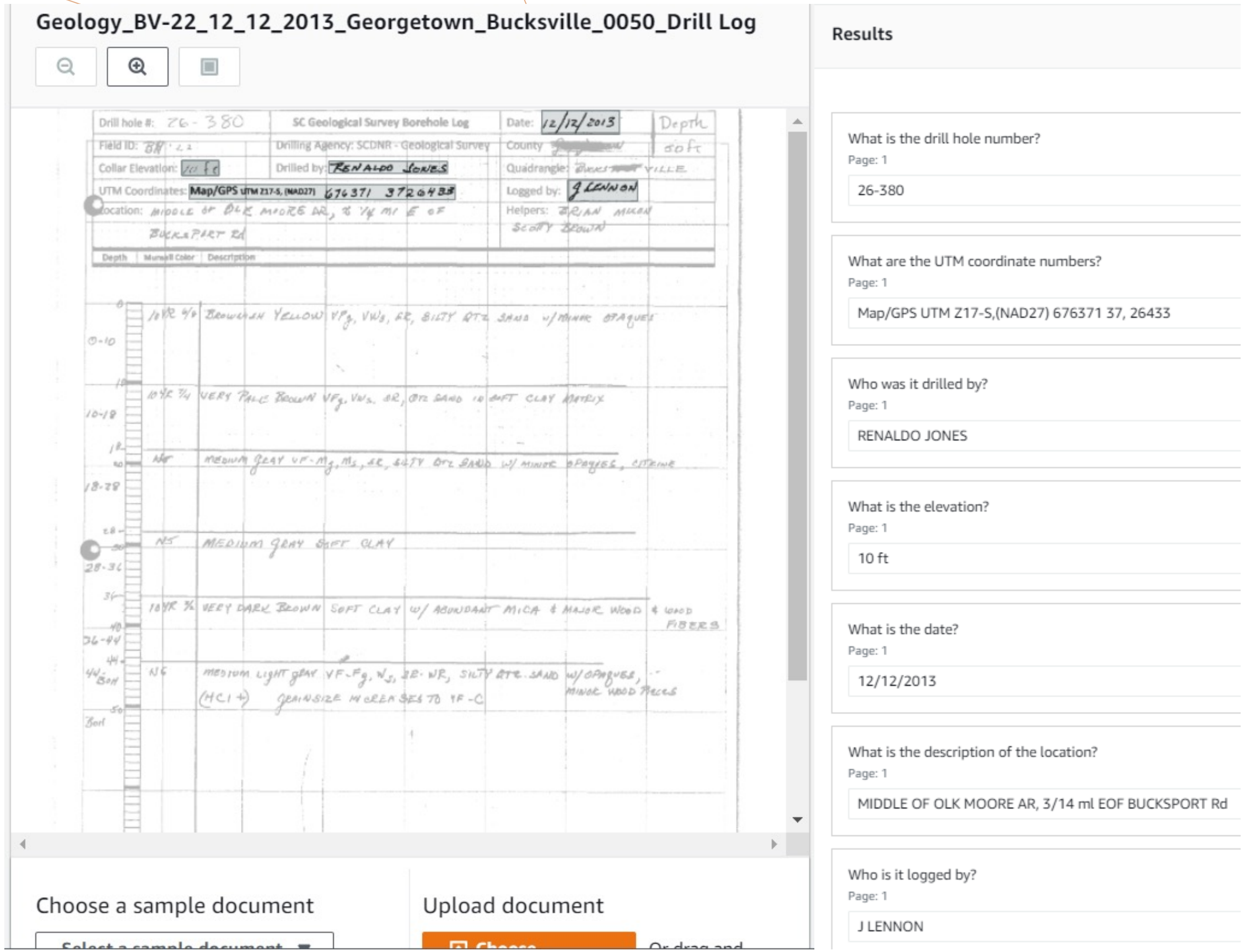
Google it

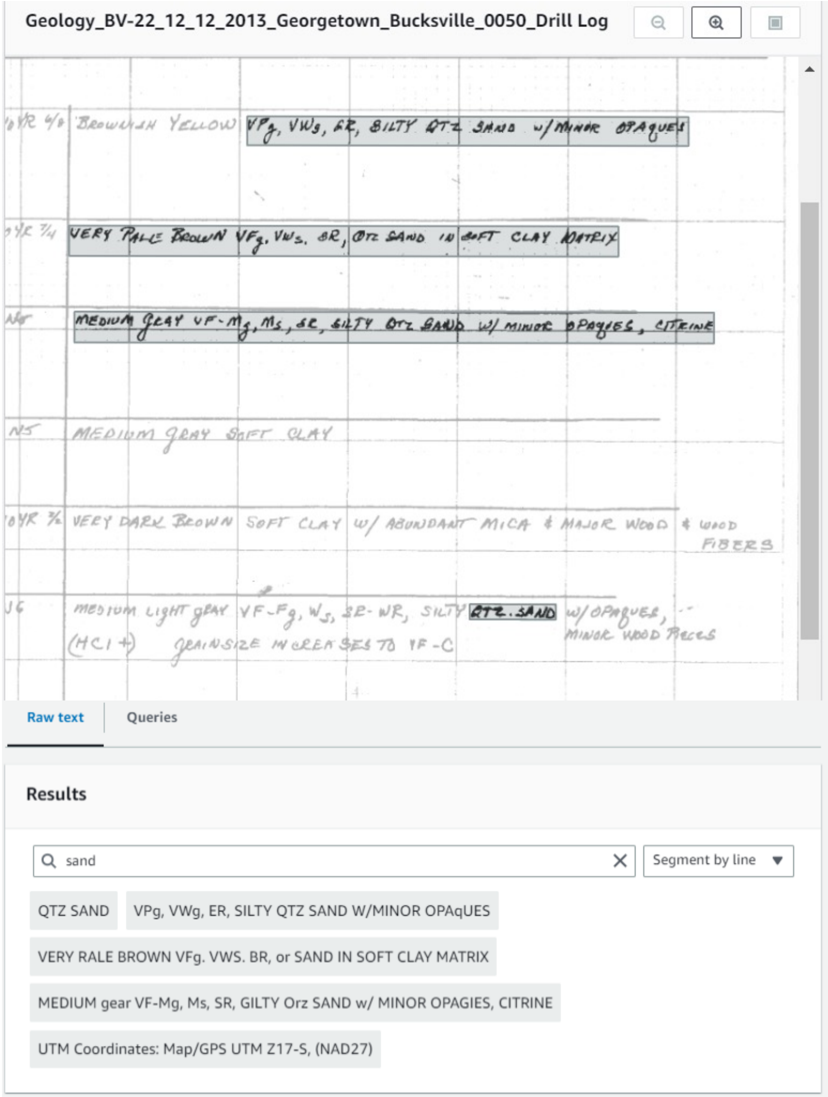
Bard

OCR – AMAZON TEXTRACT

- Amazon Textract
    - An Optical Character Recognition service provided as part of the Amazon Web Services (AWS) suite of cloud computing platforms
    - Uses a variety of machine learning algorithms pooled from various AWS services, and uses them to help extract text and structured data for various types of documents.
    - We used this service to extract, sort, and organize data from our database of thousands of drill log scans.

OCR – AMAZON TEXTRACT





OCR – AMAZON TEXTRACT

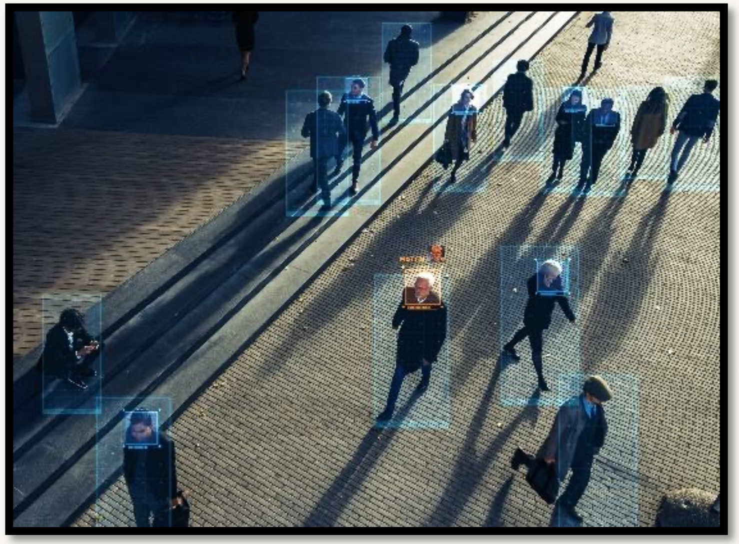
	A	B	C	D	E	F	G
1	What is the drill hole number?	What is the date?	Who was it drilled by?	What is the elevation?	What are the UTM coordinate	Who was it logged by?	What is the description of the location?
2	09-260	2/19/2014	Renaldo Jones	93.6 m 307 ft	524085 e 3719893 n	W.R.Doar, III	Sikes Rd at longview rd, SW corner.
3	09-261	3/31/2014	Joe Koch	53m 174 ft	533664 E 3713207 N	W.R.Doar, III	Bobwhite rd at Goshawk rd, NW corner just inside field.
4	09-261	2/20/2014	Renaldo Jones	70m 230 ft	3720064 N	W.R.Doar, III	Belleville Rd, 1/4 SW of??? rd,east side, 1.25miles NE of Mossdale rd.
5	09 - 254	2/10/2014	Renaldo Jones	75.4 m 247 ft	525242 E 3718615 N	W. R. Doar, III	Messengale rd, east side, 1,200 ft south of Belleville rd.
6	09-256	2/18/2014	Renaldo Jones	68 m 223 ft	526091 E, 3717860 N	W.R. Doar, III	Kings Grant road at Mossdale Road, SE corner.
7	09-259	2/19/2014	Renaldo Jones	65 m 213 ft	527657 E, 3718977 N	W.R. Doar, III	Kings grant Rd at Morning Dove rd, NE corner.
8	09-262	2/20/2014	Renaldo Jones	65 m 213 ft	528388, E, 3719057 N	W.R. Doar, III	Kings Grant Rd at Cotton Pointe Rd, SE corner.
9	09-263	2/20/2014	Renaldo Jones	66m 217 ft	3719630 N	W.R.Doar, III	Cotton Point Rd, west side, 1/4 mile north of Kings grant rd.
10	09-264	2/20/2014	Renaldo Jones	67 m 220 ft	527675 E 3719830 N	W.R. Doar, III	Morning dove rd, 100 ft north of Cotton Point rd, east side
11	9	4/3/2014	Renaldo Jones	49m 160 ft	530395 E, 3708191 N	W.R. Doar, III	Houcks Gin Rd, 200 feet west of Old State Rd, north side.
12	09-273	3/11/2014	Renaldo Jones	58 m 190 ft	3717981 N	W.R.Doar, III	Cameron Rd, 1000 ft NE of Hwy 6, north side.
13	09-264	4/4/2014	Renaldo Jones	49.5 m 162 ft	3707006 N	W.R. Doar, III	Cadinal road at Pickett Level Road, NE corner
14	09-253	2/7/2014	Renaldo Jones	51.5 m 169 ft	529462 E 3713338 N	W.R. Doar, III	Winding Brook Drive at John Rd, NW corner, 100 feet west, north side of road
15	38-396	4/1/2014	Renaldo Jones	52.5 m 172 ft	524733 E, 3711798 N	W.R.Doar, III	Bonepart Rd at Cameron rd, 200 ft south, west side.
16	09-758	3/14/2014	Renaldo Jones	56 m 183 ft	527486 E, 3713984 N	W.R. Doar, III	Cameron rd, 1 mile NE of Cameron, south side of road
17	38-395	3/25/2014	Joe Koch	50.6 m 166 ft	526059 E 3707946 N	W.R. Doar, III	Gramlin Road, south side, 100 ft east of Whisper wood road.
18	09-272	3/5/2014	Renaldo Jones	51 m 167 ft	3718875 N	W.R. Doar, III	Cameron Rd, 1/4 mle SW of Vice rd, east side in pull off
19	09-271	3/4/2014	Renaldo Jones	56 m 184 ft	3720059 n	W.R.Doar, III	Vice Rd., north side, 1/2 mile west of Cameron Rd.
20	38-398	4/3/2014	Renaldo Jones	53 m 174 ft	524980 e 3710092 N	W.R.Doar, III	Intersection of Whisperwood rd and Wild Turkey rd, SW corner.
21	09-200	3/27/2014	Joe Koch	51.9 m 170 ft	3710796 n	W.R.Doar, III	Nate Store road 1/2mile south of Jerico Road, east side.
22	Clarendon County #57	December 10, 1996l	Gary Taylor, Will Doar a	79 ft	3717460 m N, 548320 m E	Ralph H. Willoughby	sediment of ancient Santee River terrace (Upper Pleistocene)
23	159	18-Jan-02	Gary Taylor, Joe Koch a	79 ft	3719669, 0547523	C. W. Clendenin, Jr.	E 1/2 of NW 1/9 of NW 1/9 of Saint Paul 7.5-minute quadrangle, on woods road in lowland adjoini
24	09-259	3/21/2014	Renaldo Jones	50m 164 ft	534623, E, 3709491 N	W.R. Doar, III	Nate Store road at Midway Road, east side 200 ft north of Midway rd.
25	09-260	3/21/2014	Renaldo Jones	59 m 193 ft	524205 E, 3717069 N	W.R.Doar, III	US HWY 176 Old State Road, west side, across from Nighthawk Lane, 1/2 mile south of Bellevile R
26	09-252	1/12/2014	Renaldo Jones	68.5 m 225 ft	523749 E 3717968 N	W.R. Doar, III	NW corner of US 176/Old state road, and Belleville Road.
27	09-258	2/19/2014	Renaldo Jones	76.5 m 250 ft	524351 E, 3718554 N	W.R.Doar, III	Sikes Rd, 800 north of Belleville Rd, east side.

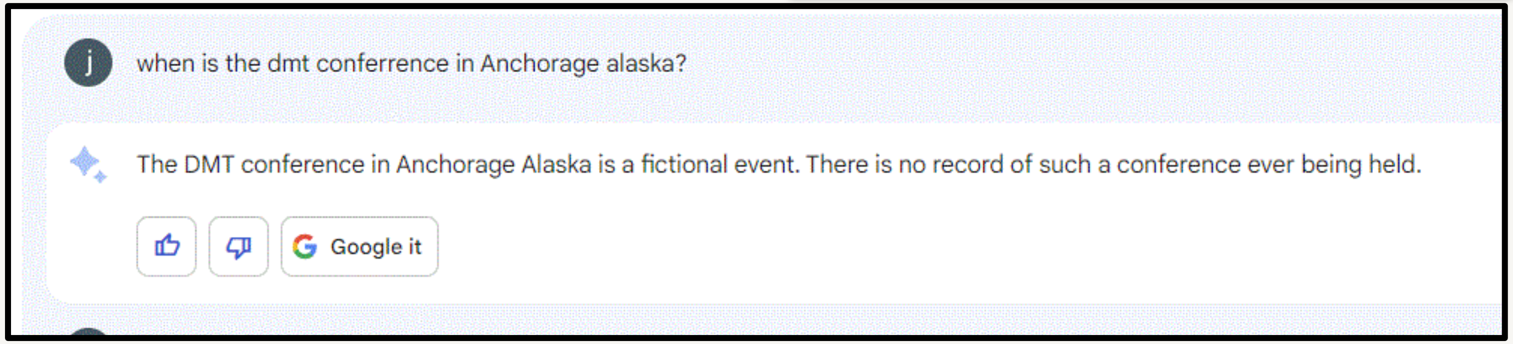
output6

THE FUTURE OF AI

THE BAD

- Consumer Privacy
- AI Bias
- Job Loss
- Disinformation





THE FUTURE OF AI
THE GOOD

Scientific Research

Healthcare

Transportation

Manufacturing

Financial Services

Real Estate

Decision Making

Cost-Savings

Task Automation

24/7 Operations

Human Error

Productivity

THANK YOU

Robert Clark, Darby DeBruhl, Gerald Krieger

ClarkRL@dnr.sc.gov

DeBruhlD@dnr.sc.gov

KriegerG@dnr.sc.gov

South Carolina Geological Survey