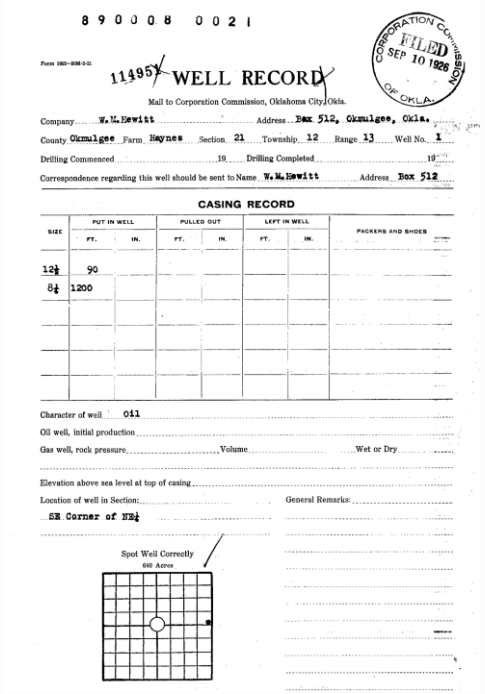
PDF to Structured Well Location Data

End-to-End Pipeline: Data Flow • Transformations • ML Improvements

Akshay Kumar Mundrathi

Feb 10, 2026



890008 0021

CORPORATION COMMISSION FILED SEP 10 1926 OF OKLA.

11495 WELL RECORD

Mail to Corporation Commission, Oklahoma City, Okla.

Company W. M. Hewitt Address Box 512, Okmulgee, Okla.

County Okmulgee Farm Haynes Section 21 Township 12 Range 13 Well No. 1

Drilling Commenced 19 Drilling Completed 19

Correspondence regarding this well should be sent to Name W. M. Hewitt Address Box 512

CASING RECORD

SIZE	PUT IN WELL FT.	PUT IN WELL IN.	PULLED OUT FT.	PULLED OUT IN.	LEFT IN WELL FT.	LEFT IN WELL IN.	PACKERS AND SHOES
12½	90						
8¼	1200						

Character of well Oil

Oil well, initial production

Gas well, rock pressure Volume Wet or Dry

Elevation above sea level at top of casing

Location of well in Section: SE Corner of NE¼

General Remarks:

Spot Well Correctly
640 Acres

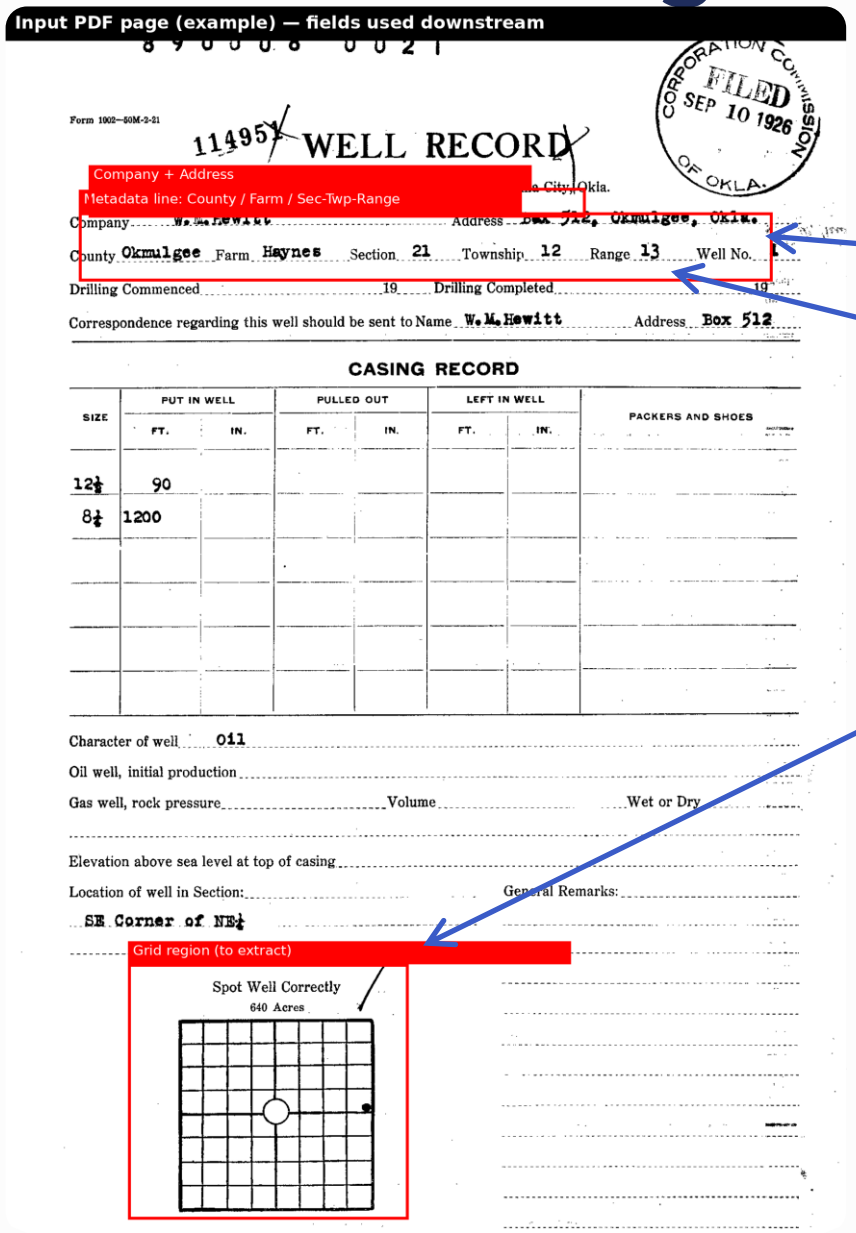
- Raw document of well record in pdf format.
- Converted it into pixels.
- Extracted necessary information by various techniques.
- Segregated information such as Section, Township, Range, precise dot location (U-Net model) with respect to the grid corners into excel sheets.
- Transformed into query loads.
- Performed SQL queries on AWS RDS to get latitude and longitude from Section, Township, Range, precise dot location with respect to the grid corners.
- Populate map with the well locations.

What We Will Cover

1. Inputs and targets: Defining extraction goals
2. PDF rendering: High-res CV image generation
3. Grid Extraction: Multi-method region cropping
4. Location crop: STR bounding box grouping
5. OCR/LLM extraction: Transforming pixels to text

- Normalization: Grid click to NW mapping
- Data Merging: Joining extraction branches
- Final Dataset: CSV schema and status fields
- QA Strategy: Confidence and failure signals
- Future Roadmap: U-Net segmentation plan

Input PDF and Target Fields



Core Extraction Targets:

- **County:** Identifies the regional administrative area.
- **STR (S-T-R):** Section, Township, and Range identifiers.
- **Well Grid:** The "Spot well correctly" coordinate grid.

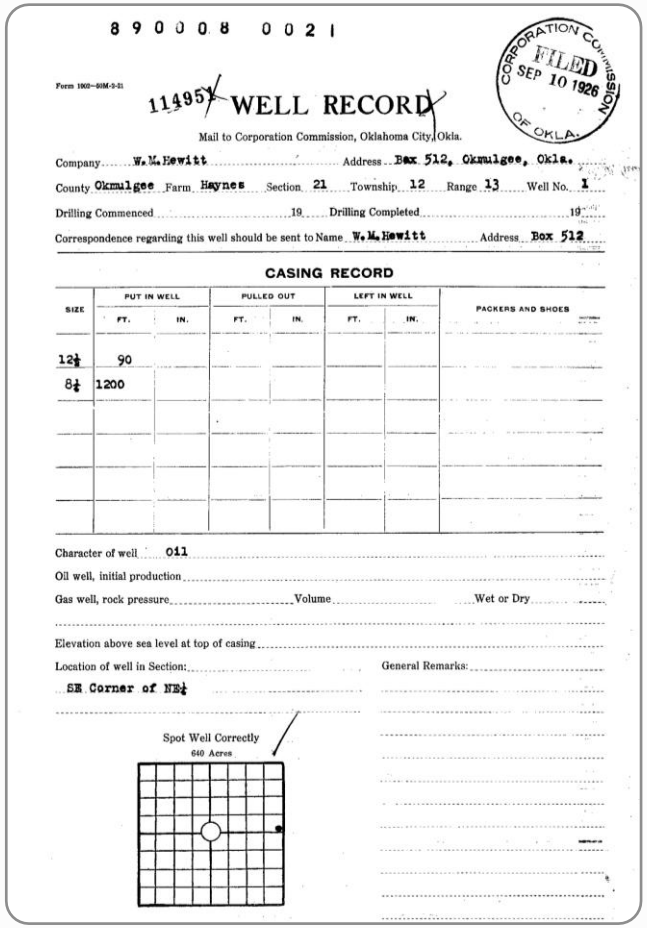
The pipeline transforms unstructured scans into *machine-readable* fields for GIS ingestion.

Stage A — PDF Rendering to Images

Rendering Strategy

- Multiplier: 2x resolution boost for clarity.
- Output: Per-page OpenCV BGR images.
- Logic: Sequential processing of multipage files.

Note: Pipeline identifies relevant pages automatically; non-grid pages are filtered later.



8 9 0 0 0 8 0 0 2 1

CORPORATION COMMISSION FILED SEP 10 1926 OF OKLA.

Form 1002—50M-9-21

11495 WELL RECORD

Mail to Corporation Commission, Oklahoma City, Okla.

Company W. M. Hewitt Address Box 512, Okmulgee, Okla.

County Okmulgee Farm Haynes Section 21 Township 12 Range 13 Well No. 1

Drilling Commenced 19 Drilling Completed 19

Correspondence regarding this well should be sent to Name W. M. Hewitt Address Box 512

CASING RECORD

SIZE	PUT IN WELL FT.	PUT IN WELL IN.	PULLED OUT FT.	PULLED OUT IN.	LEFT IN WELL FT.	LEFT IN WELL IN.	PACKERS AND SHOES
12½	90						
8¼	1200						

Character of well Oil

Oil well, initial production

Gas well, rock pressure Volume Wet or Dry

Elevation above sea level at top of casing

Location of well in Section:

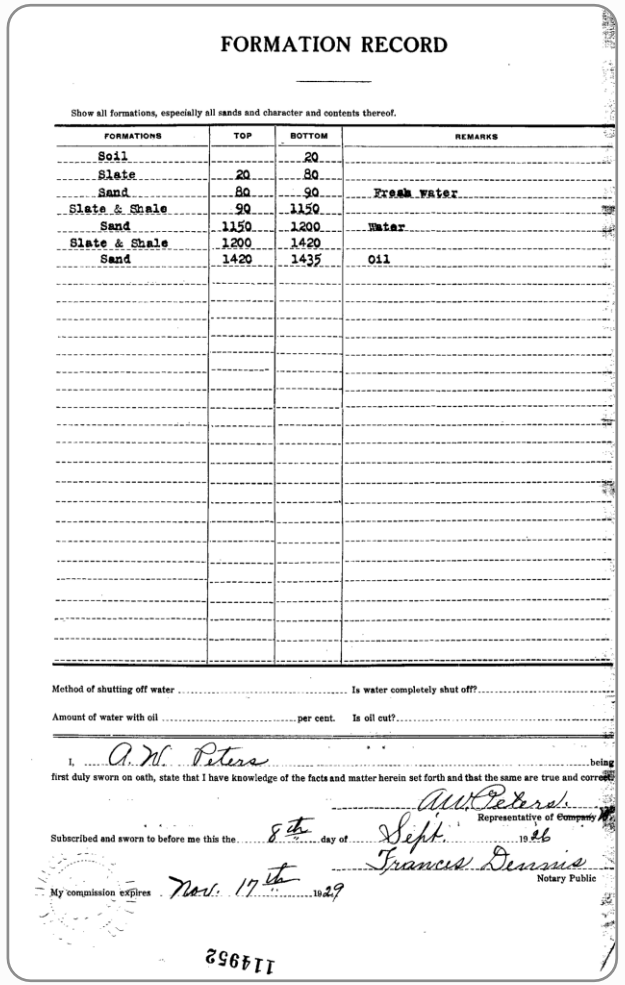
SE Corner of NE¼

General Remarks:

Spot Well Correctly

640 Acres

Page 1: Usually contains grid



FORMATION RECORD

Show all formations, especially all sands and character and contents thereof.

FORMATIONS	TOP	BOTTOM	REMARKS
Soil		20	
Slate	20	80	
Sand	80	90	Fresh water
Slate & Shale	90	1150	
Sand	1150	1200	Water
Slate & Shale	1200	1420	
Sand	1420	1435	Oil

Method of shutting off water Is water completely shut off?

Amount of water with oil per cent. Is oil cut?

I, A. W. Peters being first duly sworn on oath, state that I have knowledge of the facts and matter herein set forth and that the same are true and correct.

A. W. Peters Representative of Company

Subscribed and sworn to before me this the 8th day of Sept. 1926

Frances Dennis Notary Public

My commission expires Nov. 17th 1929

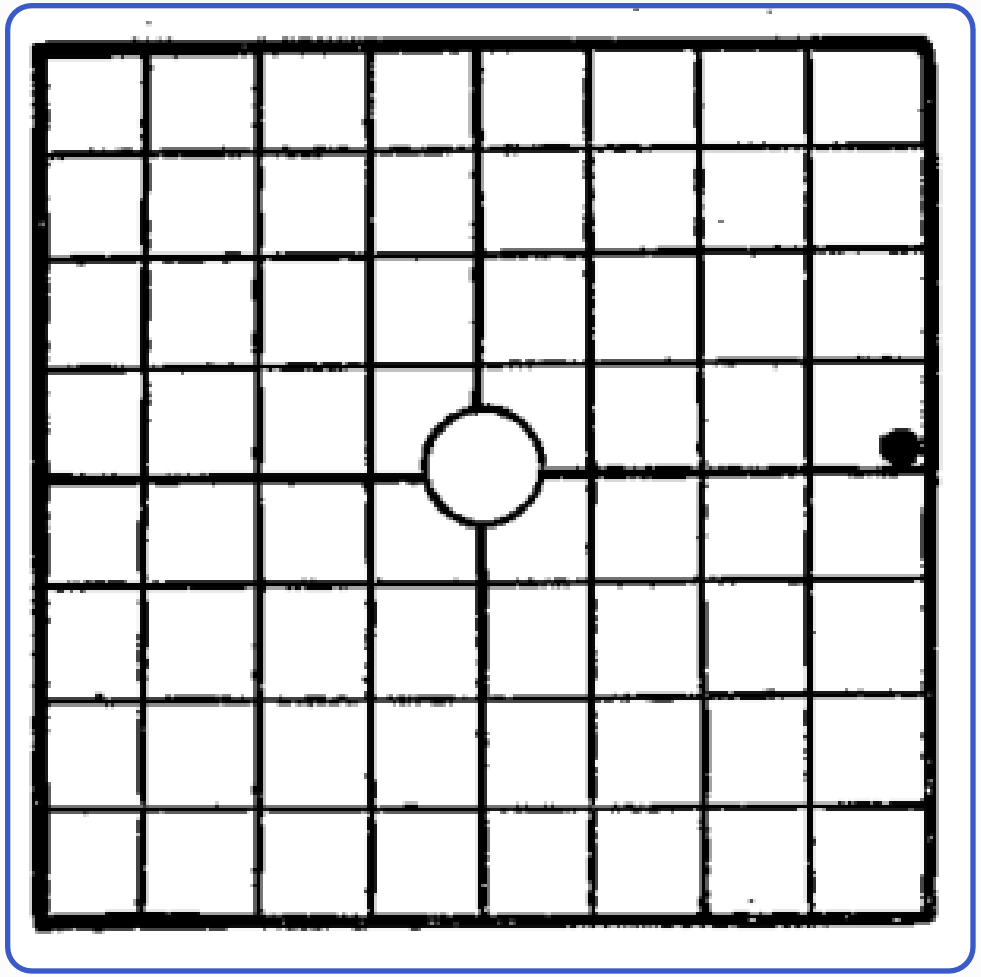
11495Z

Page 2: Supporting info only

Stage B — Grid Extraction & Logging

Process Objective

- Locate and isolate the square grid region from the full page.
- Handle variability in scan quality using a multi-method approach.
- Output: Individual PNG crops for localized dot detection.



[LOG EXCERPT]

```
--- Page 1 ---
✅ Grid extracted and saved: ..._page_1_grid.png
--- Page 2 ---
⚠️ No grid detected on this page.
```

How Grid Detection Chooses the Best Crop

The extract_grid_region_combined Algorithm:

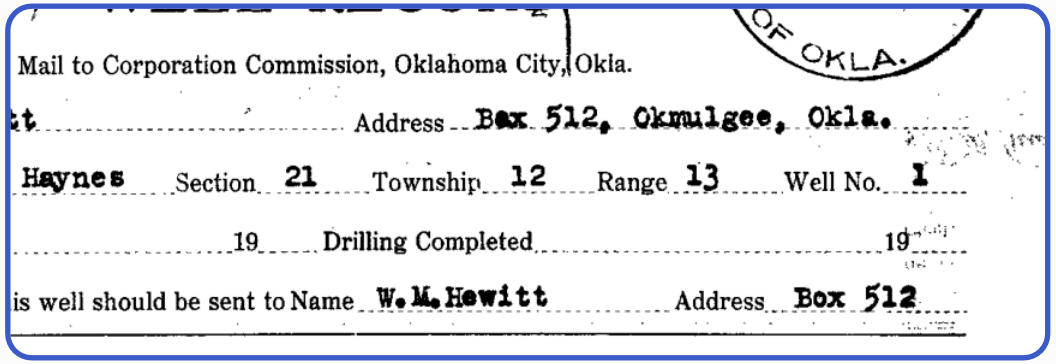
1. Parallel Candidate Detection: Run Adaptive Threshold, Otsu, Canny, Hough, Rotated, and Corners methods simultaneously.
2. Geometric Filtering: Validate each candidate against strict bounds:
   - Aspect Ratio: 0.85 – 1.15 (Targeting square grids)
- Dimensions: 280px to 850px width/height
- Minimum Area: 78,400 pixels

1. Selection: Rank all valid candidates and choose the one with the maximum area.
2. Fail-Safe: If no candidates pass filters, return "No grid detected".

Stage C — Unified Bounding Box Crop for STR

Unified Location Crop Logic

- Keyword Detection: Scan page for "Section", "Township", and "Range".
- Box Grouping: Identify the bounding boxes for detected keywords and their adjacent value fields.
- Union Operation: Compute a single Unified Bounding Box that encapsulates the entire STR block.
- Persistent Output: Save the region as a "unified_value" image for OCR processing.



Implementation Details: Uses get_unified_bounding_box to find the smallest rectangle containing all relevant keywords and values on the page.

Stage D — OCR the Cropped STR Region

Pixels to Data: Google Vision OCR

- Source: Cropped PNG of the STR region.
- Process: PNG → Byte Stream → Google Vision Cloud Vision API.
- Result: Raw JSON text detection blocks.

The Challenge of Noise:

Scanned documents often yield "noisy" raw text (e.g., "Sec: 19 | T: 2ON | R: 14"). Later LLM parsing is required to normalize these values.

Example Raw OCR Response:

"Section 21 Township 12 Range 13 Spotted Correct: YES"

Note: OCR provides the raw text; structure is added in Stage E.

Stage E — STR Extraction Using LLM Vision

Intelligent Normalization with Gemini:

- **Iterative Processing:** run_str_csv_extraction iterates through all cropped unified images.
- **Extraction Logic:** extract_str_from_image calls Gemini to identify discrete Section, Township, and Range fields.
- **Handling Formats:** Normalizes complex formats like **17N / 12E** or **27-N** into standardized numeric/text fields.
- **Intermediate Storage:** Outputs are saved to a preliminary STR CSV for the final merge stage.

This stage bridges the gap between raw OCR noise and high-quality structured database fields.

Stage F — County Detection & Confidence

County Extraction Pipeline:

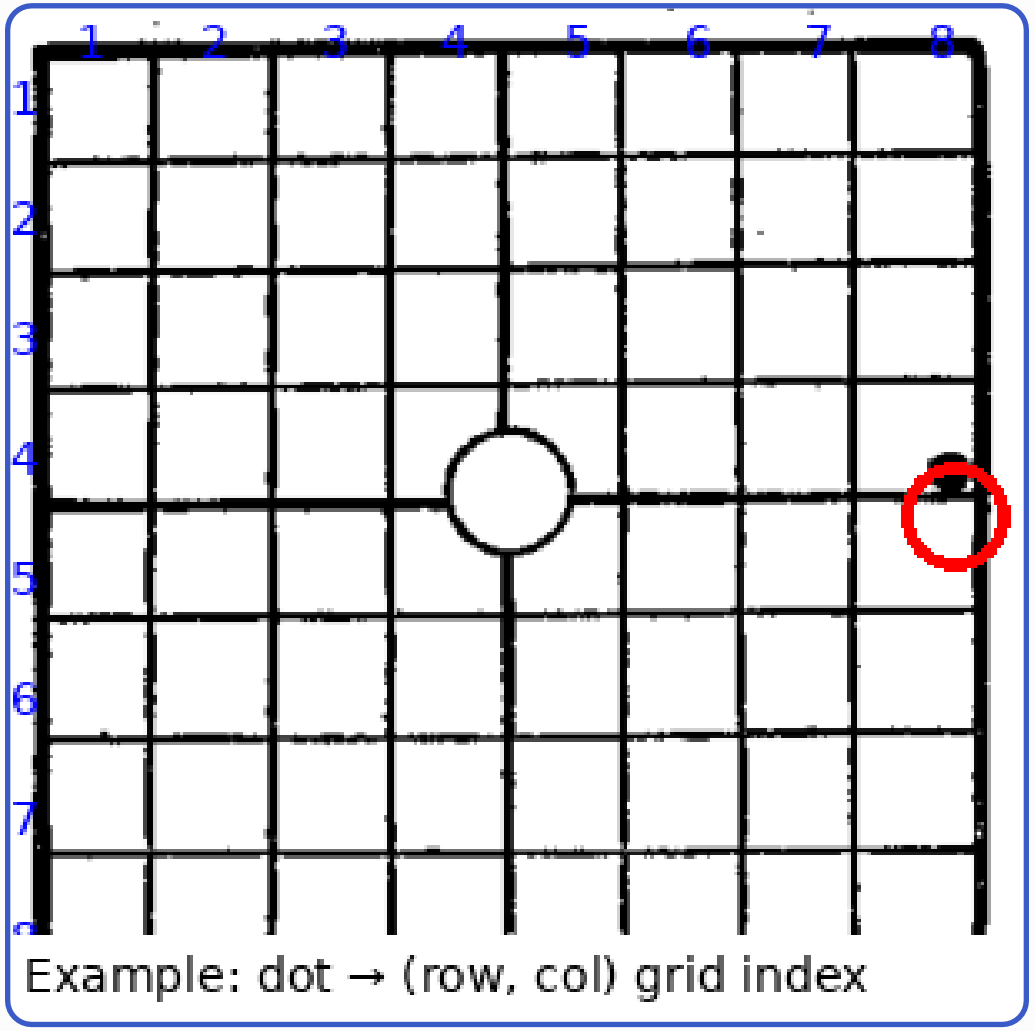
- LLM Pass 1: Gemini detects the county name from the localized header crop.
- Fuzzy Matching: Raw response is compared against a *canonical list* of known counties.
- Confidence Scoring: Based on match quality and LLM certainty.
- Fallback Strategy: The Pass2_Used flag tracks if a secondary analysis was required.

Data Field Management:

- **Detected County:** Standardized name.
- **Confidence:** 0-100 score.
- **Pass2_Used:** Boolean flag.
- **Raw Response:** Traceability for audits.

Matches are validated via parse_county_base_name_response to prevent hallucination.

Stage G — Dot to Grid Coordinates



Example: dot → (row, col) grid index

Coordinate Mapping Process:

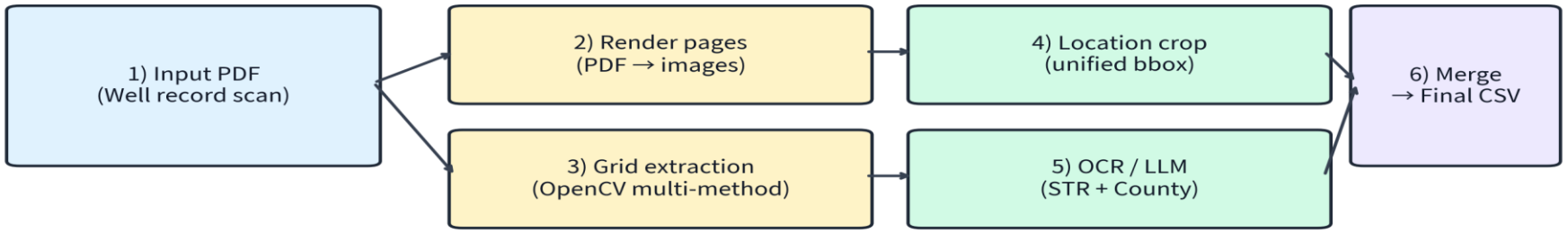
- Step 1: Identify dot centroid within the isolated grid crop.
- Step 2: Map pixel coordinates to a 8x8 matrix (row, col).
- Step 3: Convert discrete (row, col) to a PLSS quarter-quarter label.

Example Mapping:

A dot at Row 4, Col 8 translates to the directional code NE-SE-SE.

Function get_nw_coordinates handles the multi-level quadrant logic.

End-to-End Flow and Merge Points



High-level data flow (as implemented in OSU_WELL_CHECKPOINT2.py)

Dual-Branch Parallel Processing:
Branch A: Text Extraction (County + STR) via Google Vision & Gemini LLM.
Branch B: Grid Extraction & Dot Detection via Computer Vision (CV2).
The Merge: Both branches join using the image_id (base filename) to create the combined final CSV.

Final Output Dataset (CSV) — Column Meanings

The final_combined_output.csv delivers structured spatial data:

- image_id: The unique identifier linked to the source file.
- Section/Township/Range: Standardized location identifiers.
- Detected County & Confidence: The identified county and its match score.
- Pass2_Used: Indicates if fallback logic was triggered.
- row/col: 1-based indices for the 8x8 well grid.
- nw: The hierarchical PLSS directional label (e.g., SE-NW-SW).
- Status: Current processing state (e.g., "Cropped").

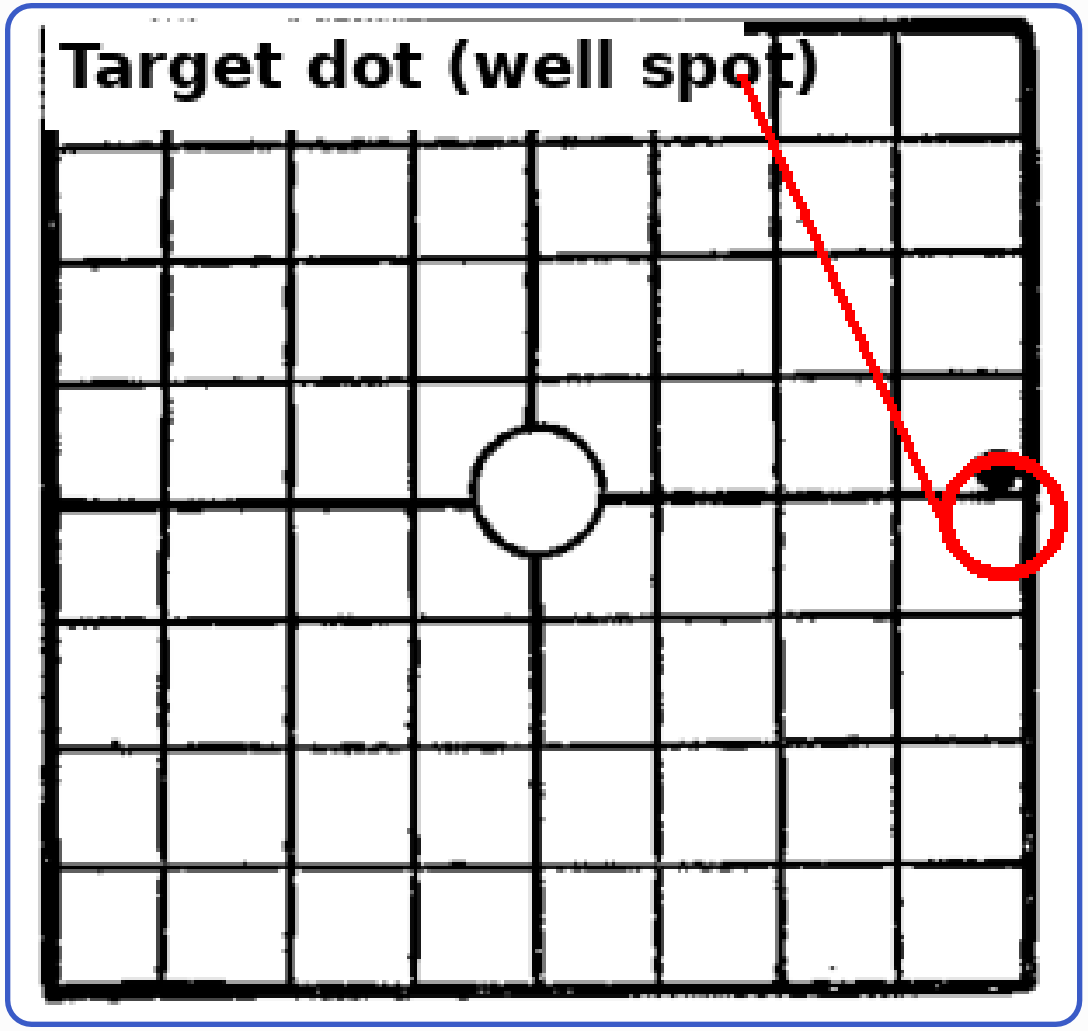
Final CSV Example Rows

image_id	Section	Township	Range	Detected County	row	col	nw
00000000_BUNCH 1_1400935	19	20	14	Washington County	4.0	4.0	NW-SE-SE
00000000_No Data_1030447	11	14	12	Okmulgee County	6.0	5.0	SE-NW-SW
00000000_No Data_1220780	21	17N	12E	Creek County	nan	nan	nan
00000000_No Data_1220867	34	19	12	Tulsa County	8.0	2.0	SW-SW-SE
00000000_No Data_548897	21	12	13	Okmulgee County	4.0	8.0	NE-SE-SE

Practical Insights:

- Handling Missing Data: Fields marked 'nan' indicate no detection was possible for that specific image.
- Data Complementarity: The *row/col* and *nw* fields provide a precise spatial overlay to the STR text data.
- Validation: Notice high confidence (100) for standard county matches.

Why Dot Detection Is Hard on Scanned Grids



The Limitations of Classic CV:

- Noise & Speckles: Dust and artifacts mimic dots.
- Contrast Issues: Faded ink vs. thick grid lines.
- Geometric Distortions: Scan skew and rotation affect alignment.
- Occlusions: Stamps or handwriting crossing into the grid.

The Solution: Deep learning segmentation provides a robust alternative to pixel-level thresholding, learning to distinguish the true dot from surrounding noise.

QA Signals and Common Failure Modes

Failure Mode	Target Signal	Impact
Missing Grid	Logs: "No grid detected"	Missing row/col & nw fields
County Misread	Confidence Score < 90	Possible Pass2_Used fallback
Dot Ambiguity	Status != "Cropped"	Coordinate mismatch in GIS

Actionable QA Ideas:

- **Thresholding:** Auto-flag any Confidence score below 85 for human review.
- **Review Queue:** Route "No grid" or "Pass2_Used=True" cases to a manual QA dashboard.
- **Sampling:** Periodically sample 5% of "Cropped" records to verify dot placement.

How Latitude and Longitude Are Computed from PLSS + Grid Data

Problem Statement:
- Lat/lon are NOT directly stored in the source data.
- Computed from PLSS metadata + human-selected grid.

Township/Range/Section/County → SQL lookup in PLSS table/view → section corner coordinates (4 corners).

Inputs Used:

- Section number, Township (N/S), Range (E/W)
- County name
- Grid position: row (1-8), col (1-8)
- Section corners: top_left, top_right, bottom_left, bottom_right (lat, lon)

How the Location Is Calculated:

1. Township/Range info → PLSS database lookup.
2. PLSS lookup → section boundary corner coordinates.
3. Grid row/col → relative position (x,y) on 0-1 scale.
4. Relative position + corners → bilinear interpolation.
5. Interpolate lat vertically then horizontally; lon horizontally then vertically.

(row, col) → (x, y)

(x, y) + corners → (lat, lon)

image_id	sect	twp	N/S	rng	E/W	county	top_left	top_right	bottom_left	bottom_right
00000000_No Data_1030447	11	14	N	12	E	Okmulgee County	(35.70996, -95.99769)	(35.71182, -95.99769)	(35.70996, -95.99993)	(35.71182, -95.99993)
00000000_No Data_1220867	34	19	N	12	E	Tulsa County	(36.08808, -96.01162)	(36.08990, -96.01162)	(36.08808, -96.01386)	(36.08990, -96.01386)
00000000_No Data_548897	21	12	N	13	E	Okmulgee County	(35.50561, -95.92831)	(35.50746, -95.92831)	(35.50561, -95.93055)	(35.50746, -95.93055)
00000000_No Data_548922	25	27	N	13	E	Washington County	(36.79865, -95.86316)	(36.80045, -95.86316)	(36.79865, -95.86542)	(36.80045, -95.86542)

Example corner coordinates (from SQL match)

Post-Processing: SQL Query and Lat/Lon Calculation

Start:
final_combined_output.csv
(Section, Township, Range, County, row, col, image_id)

→

SQL Query Step:
plss_query_matches.csv
Output of SQL query / PLSS match

→

- Read 4 corners
- Compute relative (x,y)
- Bilinear interpolation

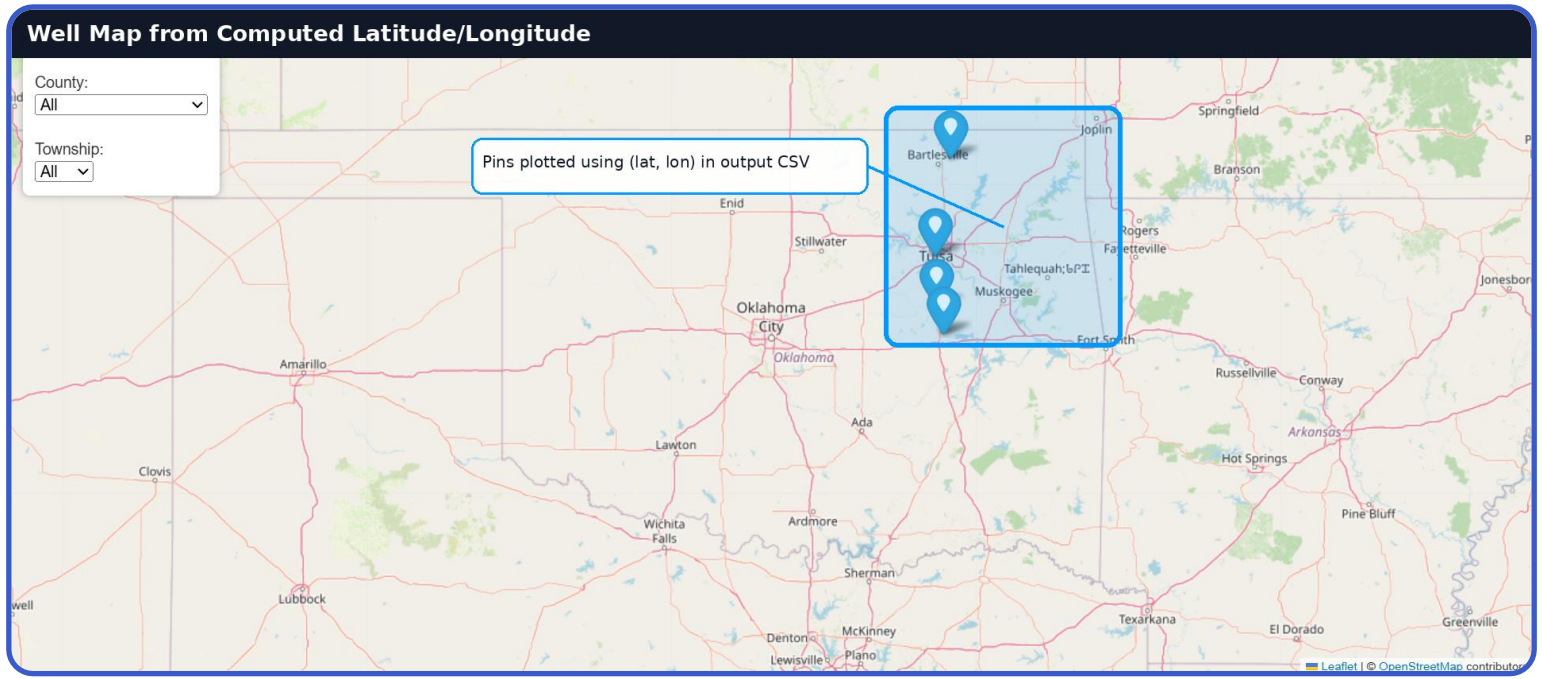
→

Write Back:
final_combined_output_with_latlon.csv
(Adds latitude, longitude columns)

image_id	sect	twp	N/S	rng	E/W	county	top_left	top_right	bottom_left	bottom_right
00000000_No Data_1030447	11	14	N	12	E	Okmulgee County	(35.70996, -95.99769)	(35.71182, -95.99769)	(35.70996, -95.99993)	(35.71182, -95.99993)
00000000_No Data_1220867	34	19	N	12	E	Tulsa County	(36.08808, -96.01162)	(36.08990, -96.01162)	(36.08808, -96.01386)	(36.08990, -96.01386)
00000000_No Data_548897	21	12	N	13	E	Okmulgee County	(35.50561, -95.92831)	(35.50746, -95.92831)	(35.50561, -95.93055)	(35.50746, -95.93055)
00000000_No Data_548922	25	27	N	13	E	Washington County	(36.79865, -95.86316)	(36.80045, -95.86316)	(36.79865, -95.86542)	(36.80045, -95.86542)

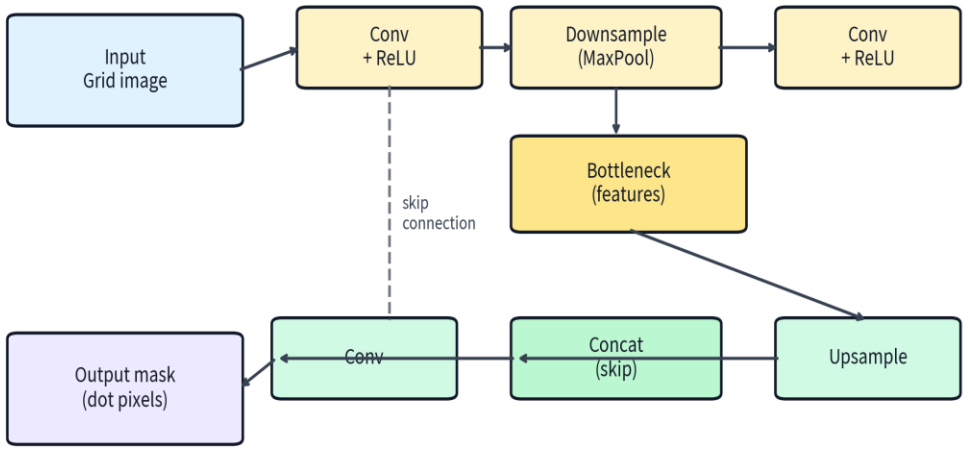
```
# Compute lat/lon
corners = plss_db.lookup(county, township, range, section)
lat, lon = interpolate(row, col, corners)
```

Well Map from Computed Latitude Longitude



- **Input:**
  Output CSV with latitude, longitude
- **Plot:**
  One marker per well (filter by County/Township)
- **Outcome:**
  Spatial QA + quick discovery of outliers

Future — U-Net Dot Segmentation



Future improvement: train a U-Net to segment the dot directly

- Note: After we can compute lat/lon reliably, the main remaining bottleneck is robust dot detection.

Developing a Robust Model:

- Labeling: Generate binary masks for dot centroids.
- Augmentation: Inject noise, blur, and rotation to train for low-quality scans.
- Loss Function: Combined Dice + BCE for precise boundary segmentation.
- Post-Processing: Centroid extraction → 8x8 Row/Col mapping.

Integration: Use the U-Net as a high-confidence assist or replacement for the current CV2 stage.

Next Steps and Q&A

Pipeline Summary:

- Inputs: Scanned multipage PDFs.
- Transforms: Parallel CV + LLM processing.
- Lat/Lon: Computed via SQL PLSS corners + bilinear interpolation → map visualization.
- Outputs: Validated, GIS-ready CSV data.

Immediate Roadmap:

- Stabilize classic CV dot detection.
- Automate ground-truth labeling.
- Train & Deploy U-Net Segmentation.
- Map the latitude and longitude locations of well

Q&A