

Quick Start Guide

5-Minute Setup

1. Install Dependencies (One-Time)

Windows:



cmd

```
# Install Tesseract OCR
# Download from: https://github.com/UB-Mannheim/tesseract/wiki
# Run installer and add to PATH

# Install Poppler
# Download from: http://blog.alivate.com.au/poppler-windows/
# Extract and add bin folder to PATH

# Install Python packages
pip install -r requirements.txt
```

Linux:



bash

```
sudo apt-get install tesseract-ocr poppler-utils
pip install -r requirements.txt
```

macOS:



bash

```
brew install tesseract poppler
pip install -r requirements.txt
```

2. Run Your First Analysis



bash

```
python well_rag_pipeline.py --pdf your_report.pdf --output ./results
```

3. View Results



bash

```
cd results
cat summary.md          # View summary
python -m json.tool analysis_report.json # View detailed JSON
```

Common Use Cases

Case 1: Quick Summary

Generate 250-word summary (default):



bash

```
python well_rag_pipeline.py --pdf report.pdf --output ./results
```

Case 2: Detailed Summary

Generate 500-word summary:



bash

```
python well_rag_pipeline.py --pdf report.pdf --output ./results --words 500
```

Case 3: Custom Nodal Analysis

Use your own parameters:



bash

```
python well_rag_pipeline.py --pdf report.pdf --nodal-json my_inputs.json --output ./results
```

Case 4: Image Processing (BONUS)

Extract from diagram/image:



bash

```
python well_rag_pipeline.py --image nodal_diagram.png --output ./results
```

Understanding the Output

analysis_report.json

Complete structured data:



json

```
{
  "metadata": {
    "pdf_file": "report.pdf",
    "analysis_date": "2025-11-09T...",
    "word_limit": 250,
    "actual_words": 247
  },
  "extracted_parameters": {
    "well_name": "NLW-GT-02-S1",
    "operation": "GRE workover",
    ...
  },
  "nodal_analysis_results": {
    "status": "success",
    "results": {
      "operating_point": {...},
      "productivity": {...}
    }
  },
  "summary": "Well: NLW-GT-02-S1..."
}
```

summary.md

Human-readable report with:

- Executive summary
- Nodal analysis results (if successful)
- Extracted parameters
- Operating conditions

summary.pdf

Professional PDF version of the Markdown report.

Troubleshooting

"Command not found: tesseract"

Fix: Add Tesseract to your PATH or reinstall

"No module named 'sklearn'"

Fix: Run `pip install scikit-learn`

Very slow processing

Cause: OCR is running (scanned PDF) **Solution:** Normal for scanned documents. Wait or reduce DPI in code.

Missing parameters in output

Cause: PDF format not recognized by regex patterns **Solution:** Use `--nodal-json` to provide parameters manually

Tips for Best Results

- 1. **Use high-quality PDFs:** Text-layer PDFs process 10x faster than scanned
- 2. **Check OCR quality:** If