This report compares three optical character recognition (OCR) engines—Tesseract OCR, EasyOCR, and PaddleOCR—evaluated on a single-page pharmaceutical Certificate of Quality (pharmaceutical-sdf-page3-certificate-quality.pdf) rendered at 300 DPI as page_image.png in Google Colab. In the test run, Tesseract OCR produced the cleanest and most accurate text extraction. EasyOCR extracted most of the text on CPU but introduced character misreads, layout splits, and stray tokens. PaddleOCR loaded its models (PP-LCNet_x1_0_doc_ori, UVDoc, PP-LCNet_x1_0_textline_ori, PP-OCRv6_medium_det, and PP-OCRv6_medium_rec) but threw an error during execution because PaddleOCR.predict() received an unexpected keyword argument 'cls'.

Document and Test Setup

- Input File: pharmaceutical-sdf-page3-certificate-quality.pdf (Page 1 converted to page_image.png at 300 DPI)
- Document Type: Pharmaceutical Certificate of Quality for AKTA™ ready Gradient Flow Section With Inlets (Lot Number 17242818, Product Article Number 29-1846-12)
- Environment: Google Colab Python 3 runtime on CPU (gpu=False for EasyOCR)
- OCR Engines Tested:
    1. Tesseract OCR (pytesseract.image_to_string)
    2. EasyOCR (easyocr.Reader(['en'], gpu=False) with detail=0)
    3. PaddleOCR (PaddleOCR(use_angle_cls=True, lang='en') calling paddle_ocr.ocr(image_path, cls=True))

Engine Findings

Tesseract OCR

- Status: Completed successfully.
- Results: Most accurate and readable output overall. Correctly extracted product details, lot number (17242818), article number (29-1846-12), manufacture date (20210126), expiration date (20230126), approval timestamp (13-Jul-2022 20:47 (GMT)), release criteria (Autoclave: Pump tubing 121°C > 15 min; Gamma Irradiation: Inlets 25.0 — 40.0 kGy), regulatory conformance (USP <88> Biological Reactivity Test Class VI, EMA/410/01, ISO 9001), and address information (14 Walkup Drive, Westborough, MA 01581 USA).
- Issues: Minor header noise at the top and small stray characters near logos at the bottom.

EasyOCR

- Status: Completed successfully on CPU (with PyTorch/DataLoader warnings).
- Results: Captured most of the text, but had noticeable character and formatting errors. Examples include "AKTAT ready" instead of "AKTA™ ready", "1212C > 15 min" instead of "121°C > 15 min", "25.0", "S", and "40.0 kGy" split across lines instead of "25.0 —

40.0 kGy", "EMA/410/O1" instead of "EMA/410/01", and extra stray tokens/lines (such as "5", "8", "1", "|", "fbiseroatCrigmaustactured", and "Westborough; MA 01581 USA").

PaddleOCR

- Status: Failed during execution with "PaddleOCR Error: PaddleOCR.predict() got an unexpected keyword argument 'cls'".
- Cause: use_angle_cls=True is deprecated in favor of use_textline_orientation, and passing cls=True caused PaddleOCR.predict() to raise an unexpected keyword argument error.
- Fix: Initialize with PaddleOCR(use_textline_orientation=True, lang='en') and call the updated predict/ocr method without cls=True, updating result parsing as needed.

Summary Comparison

Tesseract OCR

- Execution: Succeeded
- Key Accuracy: High (clean units, codes, dates, and specifications)
- Main Drawback: Minor top/bottom noise

EasyOCR

- Execution: Succeeded (CPU)
- Key Accuracy: Moderate (character misreads and split lines)
- Main Drawback: Extra stray tokens and formatting artifacts

PaddleOCR

- Execution: Failed ('cls' argument error)
- Key Accuracy: Not evaluated due to runtime error
- Main Drawback: API deprecation/argument mismatch

Conclusion For this document and setup, Tesseract OCR provided the cleanest and most reliable extraction out of the box. Updating the PaddleOCR call to match the current API will allow a complete three-way accuracy comparison, and adding validation checks or manual review is recommended for critical quality and compliance fields.