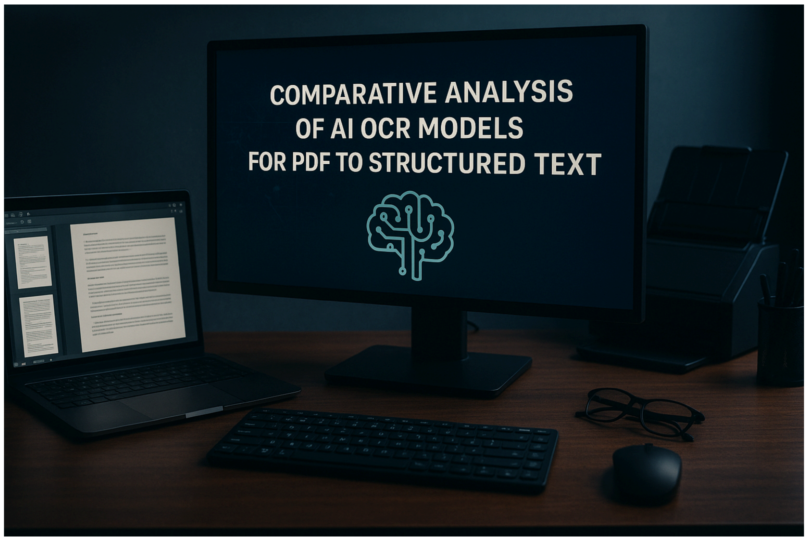
# Comparative Analysis of AI OCR Models for PDF to Structured Text

By IntuitionLabs • 5/27/2025 • 30 min read

ocr ai models pdf to text structured data text extraction deep learning document processing tesseract layoutlm form recognition



# Best AI Models for OCR and PDF-to-Text Conversion

## Introduction

Optical Character Recognition (OCR) is the technology that converts images of text (such as scanned paper documents or photos) into machine-readable text. In the context of PDFs, OCR is crucial for transforming scanned PDF files (which are essentially images) into selectable, searchable text. Modern AI-driven OCR tools go beyond simple text extraction – they strive to **preserve layout and structure**, output data in structured formats (like JSON or XML), and handle a variety of languages and fonts. This report provides an in-depth look at leading OCR models and tools (both open-source and commercial) as of 2025, comparing their capabilities and performance in converting PDFs into structured text. We will examine open-source engines (like **Tesseract** and newer Transformer-based models such as **TrOCR**, **Donut**, and the **LayoutLM** family) alongside commercial services (Google Document AI, Amazon Textract, Azure Form Recognizer, Adobe PDF Services API, and the new **Mistral OCR**). Key considerations – from PDF types (scanned vs. digital) to multilingual support, output formats, accuracy, speed, and integration – are discussed in detail, with examples for use cases including invoice parsing, academic papers, books, and historical archives.

## Leading OCR Tools and AI Models

### Open-Source and AI-Driven OCR Solutions

#### Tesseract OCR (Open Source Classic)

Tesseract is a well-established open-source OCR engine originally developed in the 1980s and later maintained by Google. It is known for its reliability on clean printed text and its support for a wide range of languages (over 100 out of the box) unstract.com. Tesseract uses an LSTM-based deep learning engine (since version 4) and can achieve **solid accuracy on high-quality scans** unstract.com. For example, with clear black-on-white printed documents, Tesseract's character recognition accuracy can reach 95% or higher, comparable to some commercial engines in straightforward cases. It also allows custom training, so users can fine-tune it on specific fonts or languages to improve recognition unstract.com.

However, Tesseract has limitations. It outputs results primarily as plain text (or HTML-based hOCR/ALTO with bounding boxes), meaning it doesn't inherently preserve complex layout structure like tables or forms – those require post-processing. It can struggle with **noisy or complex layouts** (e.g. multi-column documents, rotated text, or dense forms) and is not robust for handwriting. In comparisons, Tesseract often lagged behind newer AI models in handling low-quality scans and hand-written text source.opennews.org research.aimultiple.com. Nonetheless, Tesseract remains a **popular free choice** for basic OCR needs, especially where budget or on-premise deployment is a priority, and it's widely used in many projects (from digitizing books to powering search in archives).

### TrOCR – Transformer OCR by Microsoft

TrOCR (Transformer OCR) is an end-to-end OCR model introduced by Microsoft Research that leverages Transformer neural networks for text recognition medium.com. Unlike Tesseract's two-step approach (layout analysis then character recognition), TrOCR uses a Vision Transformer encoder to process the image and an autoregressive text decoder to directly generate the recognized text huggingface.co. This deep learning approach achieves **higher accuracy than traditional OCR engines** on difficult text, thanks to its powerful sequence modeling learnopencv.com. For instance, TrOCR has been shown to **surpass previous OCR techniques in accuracy** on printed text benchmarks learnopencv.com. It comes pretrained on large datasets (including printed text and optionally handwritten text) and can be fine-tuned for specific domains.

TrOCR works best on segmented text lines or regions – it excels at reading focused text in an image, such as a line on a sign or a snippet of a document inference.roboflow.com. It can handle multi-line passages as well, though processing very complex full pages may require breaking the image into regions. The model supports English well (pretrained on English documents) and has versions for Chinese and handwritten English (through separate fine-tuned weights) huggingface.co. In terms of output, TrOCR returns recognized text as a sequence (plain text string). It does not inherently preserve layout or structure beyond the text content – if structured output is needed, another system must interpret the text or one must fine-tune TrOCR to output tokens with formatting. Integration of TrOCR is done via Python (e.g. through Hugging Face Transformers library) and requires a GPU for efficient inference. As an open model, it can be customized extensively. Overall, TrOCR represents the trend of applying Transformer-based sequence models to OCR, resulting in **excellent character accuracy** (especially on difficult fonts or low-quality images) in research settings.

### Document Understanding Transformers: LayoutLM Family

While OCR extracts text, making sense of that text in context (for example, understanding which text is a header vs a table cell vs a form field) is a separate challenge. The LayoutLM family (LayoutLM, LayoutLMv2, LayoutXLM) consists of Transformer models that incorporate not just the text content, but also the **layout information** (coordinates on page) – and in later versions, the image pixels – to enable higher-level document understanding. These models (from Microsoft Research) are pretrained on large corpora of scanned documents and can be fine-tuned for tasks like form field extraction, invoice parsing, document classification, or question-answering on documents huggingface.co huggingface.co. LayoutLMv2 in particular is a multimodal model that models the interaction of text, layout, and image in a single framework arxiv.org. It achieved state-of-the-art results on benchmarks such as forms (e.g. the FUNSD dataset) and receipts (CORD, SROIE datasets) huggingface.co. For example, a fine-tuned LayoutLMv2 model can automatically label and extract fields like "Invoice Number" or "Total Amount" from an invoice, by understanding the spatial layout and neighboring text, rather than just looking for keywords.

LayoutLM models are **not OCR engines themselves** – they typically require an OCR step first to provide text and bounding boxes as input. Once OCR (e.g. using an engine like Microsoft's OCR or Tesseract) has provided text snippets with their coordinates, LayoutLM can be applied to classify each snippet or predict relationships (like which snippets form a key-value pair). The **LayoutXLM** variant is a multilingual extension that was pretrained on documents in multiple languages, allowing cross-language document understanding huggingface.co. This is useful for global use cases – e.g., the same model can handle an English invoice or a French invoice if fine-tuned appropriately. In summary, the LayoutLM family is critical when converting OCR'd text into truly structured, labeled data. They shine in **use cases like form and invoice parsing**, where context and position inform meaning. However, using them requires ML expertise: one must fine-tune these models on labeled examples of the target task, and they require computing resources for training/inference. When combined with a reliable OCR, LayoutLM-based solutions provide **state-of-the-art document understanding** beyond plain text extraction.

### Donut – OCR-Free Document Understanding Transformernderstanding Transformer) is an innovative model from NAVER Clova AI that takes a radical approach: it **does not use a separate OCR component at all** huggingface.co. Instead, Donut is an end-to-end Transformer that directly takes document images and outputs the desired content or information in structured form. In other words, it treats document parsing like a translation task – translating from an image to a target text (which could be plain text, JSON, or other format). Donut's architecture consists of a vision encoder and a text decoder, similar in spirit to TrOCR, but it's trained specifically on document tasks and often fine-tuned to output structured data github.com aimodels.fyi. Because Donut is **OCR-free**, it can learn to handle layout and structure implicitly: for example, if fine

### Donut – OCR-Free Document Understanding Transformer

Donut (Document Understanding Transformer) is an innovative model that takes an **end-to-end approach to document OCR**. Developed by NAVER Clova, Donut utilizes a Transformer encoder-decoder architecture to directly parse document images into target text, **without using any separate OCR engine* huggingface.co. In essence, Donut "translates" a document image into a textual representation (which can be structured). For example, if fine-tuned on invoices, Donut can directly output a JSON string of fields like invoice number, date, totals, etc., from the image. By avoiding a distinct OCR step, Donut can mitigate error propagation and **learn layout and content jointly**. It has achieved state-of-the-art results on tasks like receipt understanding (e.g. on the **SROIE** receipt dataset) and other form parsing benchmarks, demonstrating that an OCR-free, fully trainable approach can rival traditional pipelines.

Donut's output format is flexible – it's essentially learned from the training data. In practice, researchers have fine-tuned Donut to output structured JSON for forms, or plain text (with special markers) for general documents. It **preserves structure implicitly** by generating tokens in the correct order and hierarchy (for instance, it might learn to output table rows sequentially or label fields with JSON keys). One key benefit is that Donut can be tailored to very specific use cases: for a given document type (say, a customs form), one can train the model to output exactly the fields of interest in the desired format. Since Donut uses a deep Transformer, it requires a GPU for efficient inference and a substantial amount of training data to cover variations of documents. It currently supports whatever languages it's trained on – the original model was primarily English (and Korean), but it can handle other languages if trained (though multi-language in one model is limited compared to OCR engines like Google's). Overall, Donut represents a cutting-edge approach where **the model learns to read and structure documents in one g github.com aimodels.fyi**. This makes it powerful for specialized document processing (high accuracy when trained well), but it's less plug-and-play than OCR engines – essentially, Donut is a framework one can use to build a bespoke OCR+structure model for a given task.

### Mistral OCR – Multimodal Document AI Model

**Mistral OCR** is a next-generation OCR and document understanding model introduced by Mistral AI in 2025. Marketed as "the world's best document understanding API mistral.ai, Mistral OCR pushes the boundaries of what an OCR engine can extract. It doesn't just find text – it *comprehends each element of a document: text, images, tables, even mathematical equations* mistral.ai. The model takes PDFs or images as input and produces output in a structured, composite format (specifically, an ordered, interleaved stream of text and images, often provided as a Markdown document). This means if the input PDF had embedded images (like charts or figures) alongside text, Mistral will extract the text in reading order and also extract the images, placing references to them in the output where they appeared. The result is a **rich, faithful representation of the original document's content** – for example, a scientific paper with figures and formulas can be converted into a Markdown file where the paragraphs of text are in order, the formulas are preserved (often as images or LaTeX if available), and the figures are included in-lin mistral.ai mistral.ai. Such "document as prompt, structured output" capability is ideal for feeding downstream AI systems; indeed, Mistral notes that this can be used with large language models in retrieval-augmented QA systems for answering questions from document mistral.ai.

Under the hood, Mistral OCR is a proprietary model (accessible via API or on-prem deployment) that is *multilingual and multimodal by design* mistral.ai. It has been trained on a vast array of documents in different languages, which allows it to natively handle documents in many languages without extra configuration. It has achieved **top-tier benchmark results** on OCR accuracy and even unusual tasks (like reading dense scientific content) while also being one of the **fastest** OCR models availabl mistral.ai. Early independent evaluations confirm its high accuracy: one report found Mistral OCR's accuracy to be "High" on structured text, outperforming traditional OCR engines on complex layout parsio.io. It is particularly recommended for **mixed-content documents** – e.g. research papers or reports with text, tables, and images – where it was observed to yield superior result parsio.io.

That said, as a bleeding-edge system, Mistral OCR can exhibit some quirks. Reviewers noted a few cases of misclassification (e.g., a page with mostly text being mistakenly treated as an image, causing some text to be missed) and minor formatting issues (such as spurious newlines or an empty image reference appearing in the output parsio.io parsio.io. These issues were not common, but users processing highly varied documents might need to implement checks for such cases. Mistral OCR is offered as a commercial API (with pricing around 1000 pages per dollar) and is also available for self-hosting in sensitive environment mistral.ai mistral.ai. In summary, Mistral OCR represents the **state of the art in document OCR**: it not only reads text with excellent accuracy but also preserves the context (layout, images, structure) in a machine-friendly forma mistral.ai. This makes it a powerful option for advanced use cases like academic paper analysis, complex report digitization, and any scenario where retaining the original document structure is as important as the text itself.

## Commercial Cloud OCR Services

### Google Document AI (Google Cloud Vision OCR)

Google's OCR offering is part of its **Document AI** platform on Google Cloud. It encompasses a general OCR capability (often referred to as Google Cloud Vision OCR or "Document OCR") as well as specialized models for certain document types. **Google's OCR engine is highly accurate and supports a vast number of languages** – over 200 languages, including non-Latin scripts (Chinese, Japanese, Arabic, Cyrillic, etc. cloud.google.com source.opennews.org. This broad multilingual support is a major strength; in fact, a 2023 evaluation noted that Google's language coverage is on par with Microsoft's and open-source Tesseract, and notably **better than Amazon Textract's** for non-Latin alphabet source.opennews.org. The OCR is not just character extraction; it also **preserves layout structure** to a degree. Google's engine can identify paragraphs, blocks, lines, and even basic document structure like headings versus body tex cloud.google.com. It also has advanced capabilities, such as detecting *handwriting* (in ~50 languages) and even recognizing *math formulas and checkboxes* on form cloud.google.com – features that reflect Google's decades of research in OCR and document analysis. The output from Google's Document OCR API is typically a JSON (or Protocol Buffer) containing all detected text elements with their coordinates, along with the structural hierarchy (page->block->paragraph->line->word) and confidence scores.

Beyond raw OCR, Google Document AI provides **pre-trained document parsers** for common use cases. For example, there is a Form Parser that can automatically extract key-value pairs from form-like documents, and specialized models for invoices, receipts, W2 tax forms, passports, etc. These models leverage Google's OCR under the hood but add a layer of semantic understanding to output structured data (like an "Invoice Total" field with a dollar amount). They work out-of-the-box without user trainin cloud.google.com cloud.google.com. For instance, the invoice parser will return a JSON with fields such as Invoice ID, Vendor Name, Invoice Date, line items, totals, etc., populated by the model. This greatly simplifies integration for those specific document types. Google has also introduced a Document AI Workbench that allows users to fine-tune custom models (including using a few sample documents to adapt the AI to new document types cloud.google.com.

In terms of performance, Google's OCR is at the top tier. Benchmarks show it achieves extremely high text recognition accuracy – in one 2025 study, Google Vision OCR had the highest overall accuracy (~98% accuracy on a mixed dataset of printed, media, and handwritten documents research.aimultiple.com. It handled difficult multi-language documents and complex layouts admirably in independent test source.opennews.org. Google's service is cloud-based and highly scalable: it can process PDFs (including multi-page files, via an asynchronous batch API) and large volumes of images, with pricing roughly on par with other cloud providers (approximately $1.50 per 1000 pages for OCR, with volume discounts source.opennews.org. A consideration with Google is that its setup can be a bit complex for new users – as one review noted, using the API may involve handling Google Cloud Storage buckets for large files and the learning curve of Google Cloud's consol source.opennews.org. But once set up, it is a robust solution. In summary, **Google Document AI provides industry-leading OCR accuracy**, superb language support, and end-to-end solutions for converting PDFs to structured data (especially with its pretrained parsers), making it a go-to choice for many enterprise document processing need source.opennews.org.

### Amazon Textract (AWS)

Amazon Textract is AWS's cloud OCR and document analysis service. It is a fully managed service that can **extract printed text and handwriting** from scanned documents, and goes a step further by identifying the structure in forms and table unstract.com. Textract's OCR engine is powerful: it was benchmarked to have accuracy on par with Google's in many scenarios, achieving over 95–99% text recognition accuracy on standard printed document research.aimultiple.com research.aimultiple.com. In difficult cases (like handwriting), Textract also performs well – it was specifically noted to handle cursive and printed handwriting, which Tesseract and some others struggle wit source.opennews.org. As of recent updates, Textract supports multiple languages. Initially it focused on English, but now it can process documents in **Spanish, German, Italian, French, Portuguese, and English** (with automatic language detection among these aws.amazon.com unstract.com. It does not yet natively support East Asian scripts like Chinese or Japanese, which is a limitation for global use (Google and Azure cover those, whereas Textract is oriented towards Latin-script languages).

One of Textract's standout features is its ability to extract **structured data from forms and tables**. When Textract analyzes a form (through its AnalyzeDocument API with the "FORMS" feature), it doesn't just dump text – it identifies key-value pairs. For example, on an application form, Textract can return a key "Name:" with the value that was filled in, or "Date:" with the corresponding date, etc. It preserves the relationship between fields and their filled value unstract.com. This is invaluable for converting form PDFs directly into JSON where each field is labeled. Similarly, with the "TABLES" feature, Textract will detect tables and output the cells in a structured format (each cell's row and column coordinates, and the text inside). It **preserves table structure** so that you could directly reconstruct the table in CSV or Excel from the outpu unstract.com. Textract can even capture selection marks (checkboxes), indicating whether a checkbox is filled or not. More recently, Amazon added an expense/invoice analysis feature (AnalyzeExpense API) that is tailored to invoices and receipts – it automatically identifies common fields like vendor, total, tax, etc., similar to other specialized solutions.

Textract is designed for **scalability and integration** within the AWS ecosystem. Being cloud-native, it automatically scales to handle large volumes – one can process millions of pages by parallelizing calls or using asynchronous batch processing. AWS provides convenient integration points: for instance, you can have documents in an S3 bucket and trigger Textract processing via Lambda functions. The results can be stored back to S3 or passed to Amazon Comprehend (for NLP on the extracted text unstract.com. Many AWS customers incorporate Textract into data pipelines (for example, ingesting scanned contracts, using Textract to get text and fields, then loading that into databases). Textract also emphasizes security – it inherits AWS's security measures (encryption, access controls) and is certified for handling sensitive data (HIPAA eligible, etc. unstract.com.

In summary, Amazon Textract offers **high-quality OCR with additional intelligence to understand forms and tables**, making it a strong choice for business documents. Its drawbacks are mainly the relatively limited language support (focused on Western languages) and the complexity of handling documents with very unconventional layouts (where you might need to combine its output with custom logic). For typical use cases like invoices, forms, or receipts in supported languages, Textract provides accurate results and structured outputs out-of-the-box, and it easily plugs into AWS-based workflow unstract.com unstract.com.

### Azure Form Recognizer (Azure Document Intelligence)

Azure's OCR and document processing solution is known as **Form Recognizer**, recently rebranded under Azure's AI Document Intelligence services. It combines strong OCR capabilities with both prebuilt and custom models for deeper extraction. **At its core, Azure's OCR (the "Read" and "Layout" APIs)** can extract text from images/PDFs with high accuracy, including both printed text and handwritten conten unstract.com. It supports a wide array of languages – at least 10 languages are explicitly supported for full document processing (English, French, German, Italian, Spanish, Portuguese, Dutch, Chinese, Japanese, Korean, etc. azure.microsoft.com, and the underlying OCR engine in Azure's Cognitive Services can handle over 100 languages for text extraction. This includes support for Cyrillic, Arabic, and other scripts (Azure's OCR is comparable to Google's in multilingual reach, and significantly broader than Textract's source.opennews.org. Azure's OCR also handles mixed languages and can auto-detect language within the document. For handwriting, Microsoft's research gives it a strong base: Azure can read cursive or print handwriting (especially in English and some European languages) reasonably wel unstract.com, though very messy handwriting may still pose challenges.

Where Azure Form Recognizer shines is in its **prebuilt models and custom training** options. The service provides ready-to-use models for common document types: **invoices, receipts, business cards, and ID cards** are among the prebuilt model unstract.com. Using the prebuilt invoice model, for example, you can upload an invoice PDF and get back structured JSON with vendor name, invoice date, invoice number, line items, totals, and so forth – no training or configuration needed. These prebuilt models are trained on diverse samples of those document types, so they work out-of-the-box for many scenarios, greatly reducing manual parsing. In tests, the invoice model showed high accuracy, correctly recognizing the majority of fields on sample invoice unstract.com unstract.com.

Beyond prebuilt models, Azure offers a **custom model training** capability that is a major draw for enterprises with unique documents. With as few as 5 sample documents, a user can train a custom model by labeling fields or tables in their documents (using Azure's Form Recognizer Studio or SDK unstract.com. The custom model will then learn the structure and be able to extract those specific fields from new documents of the same layout. This is extremely useful if you have, say, a proprietary form or an unusual document format not covered by prebuilt models. The custom models use Azure's machine learning on the layout and text, so they can generalize to some degree even if the exact positions shift, etc. Azure also provides a general Layout model that simply returns all text with bounding boxes and a structured hierarchy (similar to how Google's OCR output works), which is useful if you want to manually parse the layout or feed it into something like LayoutLM.

Another strength of Azure Form Recognizer is **table extraction**. It not only detects tables but keeps the cell structure intact, including row/column relationship unstract.com. If you have a financial report PDF with tables, Form Recognizer can output those tables as an array of rows and columns in JSON, or you can even have it generate an Excel file. It correctly splits rows and columns even without explicit lines/borders (by using spacing heuristics and ML), which saves a lot of time compared to manually fixing OCR'd tables.

Azure's service, being cloud-based, scales well and integrates with other Microsoft tools. For example, it can be used in Power Automate flows for document processing. Security is enterprise-grade (Azure AD, role-based access, and compliance with standards unstract.com.

Pricing is similar to competitors (some free pages monthly, then per-page charges). One notable offering is that Azure often includes a certain number of free pages per month (e.g., 500) for Form Recognizer in the free tier, which is friendly for small-scale usag unstract.com.

In summary, **Azure Form Recognizer (Document Intelligence)** provides robust OCR and layout extraction, broad language support, and the flexibility of *both prebuilt and custom-trained models* unstract.com unstract.com. It is particularly well-suited for organizations that have a variety of document types – some standard (which the prebuilt models handle) and some unique (which can be tackled with custom models). Its performance on printed text is excellent (on par with Google/AWS), and while historically it had some trouble with certain handwritten case research.aimultiple.com, Microsoft continually improves the models. The ability to accurately extract tables and forms and integrate into automated workflows makes Azure a strong choice for end-to-end document processing solutions.

### Adobe PDF Extract API (Adobe Document Services)

Adobe, the company behind the PDF format, offers its own cloud service for extracting content from PDFs: the **PDF Extract API**, part of Adobe PDF Services. This tool is specialized for PDFs (as opposed to images in general) and is designed to **retrieve text and structure from any PDF, whether it's a native PDF or a scanned document**. One key advantage of Adobe's approach is that if a PDF already contains a text layer (i.e. it's digitally generated, not scanned), Adobe will *use the native text directly rather than performing OCR* community.adobe.com. This preserves 100% accuracy for digital-text PDFs (no OCR errors on those) and retains things like correct reading order through the PDF's internal content order or tags. If the PDF is scanned (image-only), then the service will apply OCR automaticall blog.developer.adobe.com – essentially giving the best of both worlds in a seamless manner.

The output from Adobe PDF Extract API is a **structured JSON** that aims to capture the full document structure and conten blog.developer.adobe.com blog.developer.adobe.com. This JSON includes elements like paragraphs, headings, lists, tables, and figures in reading orde blog.developer.adobe.com. Each element comes with its bounding coordinates on the page and often styling information (font, etc.), which can be used to infer hierarchy or importanc blog.developer.adobe.com. For example, if a section title is in bold and larger font, the JSON might label it as a heading element. The service also can **extract tables** separately – it will identify tables and you have the option to get those tables exported as CSV or XLSX files directly, preserving rows and column blog.developer.adobe.com. Similarly, it can extract images (figures) from the PDF and provide them as image files, with references in the JSON. In effect, Adobe's solution is creating a near-complete DOM-like representation of the PDF's content, which is very useful for downstream processing or reflowing the content into other formats.

What sets Adobe apart is its focus on **maintaining reading order and meaningful structure**, which has historically been a challenge. PDF content can be fragmented or not stored in logical order internall blog.developer.adobe.com, so simply extracting text in code order often yields jumbled sentences (especially in multi-column layouts). Adobe's PDF Extract employs AI to **determine the correct reading order** of text block blog.developer.adobe.com, and group text into coherent paragraphs. It also provides metadata like detected headings or section breaks which other OCR APIs typically don't output. This makes it easier to, say, identify the title of a document, or break an article into sections.

In terms of OCR accuracy on scanned content, Adobe doesn't publish specific metrics, but their OCR engine (likely leveraging Adobe Sensei and possibly licensed OCR tech) is regarded as high quality. They support multiple languages as well (generally, Adobe Acrobat's OCR supports a wide range of languages, so the cloud service presumably inherits that capability). A community comment noted that unlike some competitors, Adobe's approach does not convert everything to image for OCR if it's unnecessary, which can improve accuracy on tricky PDF community.adobe.com.

The PDF Extract API is a cloud service – you send a PDF and receive a zip package containing the JSON and any extracted asset file blog.developer.adobe.com. It is part of Adobe's Document Services and requires an API key (with a paid license for high volume use, though a limited free trial exists). It's aimed at developers who need to incorporate PDF ingestion in applications, especially when the exact structure is needed (for example, analyzing academic papers, legal contracts, etc., where sections and formatting matter).

In summary, **Adobe's PDF Extract API excels at preserving the rich structure of PDFs**. It outputs text in the correct reading order, identifies elements like tables and figures, and provides a comprehensive JSON representation of the documen blog.developer.adobe.com blog.developer.adobe.com. It seamlessly handles both digital text PDFs and scanned images, applying OCR only when needed. This makes it an excellent choice when you need more than just plain text – for instance, converting a PDF into an HTML or into data fields. Adobe leverages its deep knowledge of the PDF format to offer something beyond what general OCR engines provide: a true structured extraction that can save countless hours of post-processing to rebuild document structure.

## Comparison of OCR Models and Tools

To summarize the above, the following table compares key features of the mentioned OCR solutions:

**OCR Tool / Model**

**Type**

**Languages**

**Notable Capabilities**

**Output Structure**

**Use Case Highlights**

**Tesseract OCR**

Open-source (on-prem)

100+ (multilingual)

Printed text OCR; configurable; trainable

Plain text or hOCR (HTML+coords)

Basic OCR for clean scans; offline use

**TrOCR (Microsoft)**

Open-source Transformer

Pretrained English (others with finetune learnopencv.com

High accuracy on difficult text using deep learnin learnopencv.com; can fine-tune for handwriting

Plain text sequence output

Scenarios needing extra accuracy on print or specific model fine-tunes (e.g., OCR in an ML pipeline)

**LayoutLMv2 / LayoutXLM**

Open-source Transformer

Multilingual (LayoutXLM)

Understands text+layout for forms, tables

Labeled fields or tokens (after ML model fine-tuning)

Form understanding; field extraction (requires OCR input)

**Donut (Clova)**

Open-source Transformer

Primarily English (task-specific)

End-to-end OCR-free document parsing; can output JSON directly

Whatever format it's trained for (e.g., JSON, text)

Custom document parsing tasks; e.g. end-to-end invoice -> JSON extraction

**Mistral OCR**

Proprietary API (cloud or on-prem)

Multilingual (built-in mistral.ai

SOTA accuracy on complex layouts; extracts images, tables, formula mistral.ai; very fast

Markdown with text+image in sequence

Complex PDFs (scientific, reports) where preserving content structure is ke parsio.io

**Google Document AI**

Cloud API (Google Cloud)

200+ language cloud.google.com

Vision OCR with layout; handwriting (50 langs cloud.google.com; specialized parsers for invoices, etc.

JSON with text, layout, and entity fields (for specialized models)

General-purpose OCR at scale; multi-language documents; form and receipt processing with prebuilt models

**Amazon Textract**

Cloud API (AWS)

~6 languages (Latin aws.amazon.com

Text & handwriting OCR; form key-value extractio unstract.com; table structure output; integrates with AWS workflow

JSON with blocks (lines, words, tables, forms); also key-value pairs in JSON

Business documents (forms, invoices) in supported langs; easy AWS integration; needs custom handling for other languages

**Azure Form Recognizer**

Cloud API (Azure)

~10+ (incl. CJK azure.microsoft.com, 100+ OCR support

Strong OCR for print/handwritin unstract.com; prebuilt models for invoices/receipt unstract.com; custom model training with few sample unstract.com

JSON with text lines, tables, and extracted fields (for prebuilt models)

Enterprise forms and invoices; scenarios requiring custom model training (unique forms); broad language needs

**Adobe PDF Extract API**

Cloud API (Adobe)

Multilingual (matches Acrobat OCR capabilities)

Leverages native PDF text if presen community.adobe.com; detailed reading order; identifies paragraphs, headings, table blog.developer.adobe.com

JSON with full document structure; plus optional CSV for tables, images extracted

Converting PDF to richly structured data; pulling content for analysis or reformatting (especially for design-heavy PDFs)