pdfOCR

Harness the power of PDF

What is iText pdfOCR?

iText pdfOCR offers Optical Character Recognition functionality to convert your scanned documents, PDFs and images into fully ISO-compliant PDF or PDF/A-3u files making it possible to access and process the text they contain. It makes use of the powerful open-source Tesseract 4 engine, featuring neural net (LSTM) based OCR. Every day we receive scanned documents or images containing printed text. But without machine-readable text, the content cannot be edited, searched, indexed or processed.

Smart tip: Want to recognize documents in “Vulcan” or any other language not in the default set of Tesseract dictionaries? No problem, simply tell iText pdfOCR to use your own “Vulcan” dictionary for character recognition. Live long and prosper.

How does iText pdfOCR work?

Simply pass to iText pdfOCR the image(s) you need to be recognized as text, followed by the language(s) the text was written in. This will automatically select the right Tesseract dictionaries, or you can supply your own dictionary.

At the end of the OCR process, either a text file or a searchable PDF (optionally PDF/A-3u) will be created. You can select whether PDFs are created with separate layers for the original image content and the recognized text, or as flattened PDFs with the layers merged.

```
final Tesseract4LibOcrEngine tesseractReader = new Tesseract4LibOcrEngine(tesseract4OcrEngineProperties);
tesseract4OcrEngineProperties.setPathToTessData(new File(TESS_DATA_FOLDER));

OcrPdfCreator ocrPdfCreator = new OcrPdfCreator(tesseractReader);
try (PdfWriter writer = new PdfWriter(OUTPUT_PDF)){
   ocrPDFCreator.createPdf(LIST_IMAGES_OCR, writer).close();
}
```

Java code example

```
var tesseractReader = new Tesseract4LibOcrEngine(tesseract4OcrEngineProperties);
tessract4OcrEngineProperties.SetPathToTessData(new FileInfo(TESS_DATA_FOLDER));
var ocrPdfCreator = new OcrPdfCreator(tesseractReader);
using(var writer = new PdfWriter(OUTPUT_PDF)) {
     ocrPdfCreator.CreatePdf(LIST_IMAGES_OCR, writer).Close();
}
```

.NET (C#) code example

Smart tip: If you’re starting with a scanned PDF document, you can use iText 7 Core to first extract the images to use with iText pdfOCR.

pdfOCR | iTEXT Harness the power of PDF

Features

Automate text recognition

Enables the automation of text recognition into a document workflow process.

Input types: BMP, PNM, PNG, JFIF, JPEG or TIFF

Process single images, or list of images at once, export results as text or PDF. You can also use PDF as an input to search from.

Ideal for long-term archiving

Can generate PDF/A-3u compliant files, the accepted standard for long-term archiving and preservation of PDF electronic documents.

Multiple language and advanced scripts support

Includes dictionaries for over 100 languages; custom dictionaries can be supplied for additional support. Use with iText pdfCalligraph to ensure reproduced and recognized texts in multiple languages are rendered correctly.

Powered by Tesseract 4

iText pdfOCR uses the latest stable version of the Tesseract OCR engine which features improved speed, accuracy and training tools.

Simple, yet flexible API

It is also abstracted, to allow support for different OCR engines with minimal effort from users.

As pdfOCR is an iText 7 add-on, you also get all the benefits and features of iText 7 Core.

We’re also developers!

In our 20 years of code, we know how important it is to have good documentation, and good processes in place. You can always find comprehensive documentation and code examples online. Commercial customers can also count on our world-class in-house customer support, offering direct access to expert assistance from our development team.

Get iText pdfOCR

Scan the QR-code or go to the link below:

itextpdf.com/itext7/pdfOCR