www.bytescout.com

OCR with mean dataset in C# and ByteScout PDF Extractor SDK

OCR with mean dataset in C#

On this page you will learn from code samples for programming in C#. ByteScout PDF Extractor SDK was made to help with OCR with mean dataset in C#. ByteScout PDF Extractor SDK is the SDK is designed to help developers with pdf tables and pdf data extraction from unstructured documents like pdf, tiff, scans, images, scanned and electronic forms. The library is powered by OCR, computer vision and AI to provide unique functionality like table detection, automatic table structure extraction, data restoration, data restructuring and reconstruction. Supports PDF, TIFF, PNG, JPG images as input and can output CSV, XML, JSON formatted data. Includes full set of utilities like pdf splitter, pdf merger, searchable pdf maker and other utilities.

This rich sample source code in C# for ByteScout PDF Extractor SDK includes the number of functions and options you should do calling the API to implement OCR with mean dataset. To do OCR with mean dataset in your C# project or application you may simply copy & paste the code and then run your app! Enhanced documentation and tutorials are available along with installed ByteScout PDF Extractor SDK if you'd like to dive deeper into the topic and the details of the API.

Trial version can be downloaded from our website. Source code samples for C# and documentation are included.

Source Code Files:

Program.cs

```
using System.Diagnostics;
using Bytescout.PDFExtractor;

// This example demonstrates the use of Optical Character Recognition (OCR) to extract
// from scanned PDF documents and raster images.

// To make OCR work you should add the following references to your project:
// 'Bytescout.PDFExtractor.dll', 'Bytescout.PDFExtractor.OCRExtension.dll'.

namespace OCRExample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create Bytescout.PDFExtractor.TextExtractor instance
            TextExtractor extractor = new TextExtractor();
            extractor.RegistrationName = "demo";
            extractor.RegistrationKey = "demo";

            // Load sample PDF document
            extractor.LoadDocumentFromFile("sample_ocr.pdf");

            // Enable Optical Character Recognition (OCR)
            // in .Auto mode (SDK automatically checks if needs to use OCR or not)
            extractor.OCRMode = OCRMode.Auto;

            // Set the location of OCR language data files
            extractor.OCRLanguageDataFolder = @"c:\Program Files\Bytescout PDF Extracto

            // Set OCR language
            extractor.OCRLanguage = "eng"; // "eng" for english, "deu" for German, "fr
            // Find more language files at https://github.com/bytescout/ocrdata/tree/m

            // Set PDF document rendering resolution
            extractor.OCRResolution = 300;


            // You can also apply various preprocessing filters
            // to improve the recognition on low-quality scans.

            // Automatically deskew skewed scans
            //extractor.OCRImagePreprocessingFilters.AddDeskew();

            // Remove vertical or horizontal lines (sometimes helps to avoid OCR engin
            //extractor.OCRImagePreprocessingFilters.AddVerticalLinesRemover();
            //extractor.OCRImagePreprocessingFilters.AddHorizontalLinesRemover();

            // Repair broken letters
            //extractor.OCRImagePreprocessingFilters.AddDilate();

            // Remove noise
            //extractor.OCRImagePreprocessingFilters.AddMedian();
```

```
            // Apply Gamma Correction
            //extractor.OCRImagePreprocessingFilters.AddGammaCorrection();

            // Add Contrast
            //extractor.OCRImagePreprocessingFilters.AddContrast(20);


            // (!) You can use new OCRAnalyser class to find an optimal set of image pr
            // filters for your specific document.
            // See "OCR Analyser" example.


            // Save extracted text to file
            extractor.SaveTextToFile("output.txt");

            // Cleanup
            extractor.Dispose();

            // Open result document in default associated application (for demo purpose
            ProcessStartInfo processStartInfo = new ProcessStartInfo("output.txt");
            processStartInfo.UseShellExecute = true;
            Process.Start(processStartInfo);
        }
    }
}
```

VIDEO

https://www.youtube.com/watch?v=s28W3_KMraU

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout PDF Extractor SDK Home Page
Explore ByteScout PDF Extractor SDK Documentation
Explore Samples
Sign Up for ByteScout PDF Extractor SDK Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co