以下に、**Python**および\*\*C#\*\*での「OCR付きPDF読み取り → 伝票番号抽出 → PDF分割 → CSV出力 → PostgreSQLへのINSERT」のサンプルコードを提示します。

**✅ 前提**

* 各ページに「伝票番号（例：DEN12345678）」が含まれている前提
* 出力先フォルダ：output/{伝票番号}.pdf
* CSV形式：denno,text
* PostgreSQLの接続情報は仮定（ホスト・DB名等は適宜変更）

**🐍 Python 版**

**🔧 必要ライブラリ**

pip install pytesseract pdf2image PyMuPDF psycopg2

※ Windowsで pdf2image を使う場合、[Poppler](https://github.com/oschwartz10612/poppler-windows/releases/) をインストールし、PATHを通す必要があります。

**📄 Pythonコード（pdf\_process.py）**

import os, re, csv

import pytesseract

from pdf2image import convert\_from\_path

import fitz # PyMuPDF

import psycopg2

PDF\_PATH = "input/orders.pdf"

OUTPUT\_DIR = "output"

CSV\_PATH = "output/data.csv"

TESS\_LANG = "jpn" # 日本語の場合は "jpn"

# PostgreSQL接続情報

PG\_CONN = {

"host": "localhost",

"port": 5432,

"database": "your\_db",

"user": "your\_user",

"password": "your\_password"

}

os.makedirs(OUTPUT\_DIR, exist\_ok=True)

conn = psycopg2.connect(\*\*PG\_CONN)

cur = conn.cursor()

csv\_file = open(CSV\_PATH, "w", newline="", encoding="utf-8")

writer = csv.writer(csv\_file)

writer.writerow(["denno", "text"])

doc = fitz.open(PDF\_PATH)

denno\_pages = {}

# OCR + 伝票番号抽出

for i in range(len(doc)):

page = doc.load\_page(i)

pix = page.get\_pixmap(dpi=300)

img\_path = f"{OUTPUT\_DIR}/page\_{i+1}.png"

pix.save(img\_path)

text = pytesseract.image\_to\_string(img\_path, lang=TESS\_LANG)

match = re.search(r"(DEN\d{8})", text)

if match:

denno = match.group(1)

denno\_pages.setdefault(denno, []).append(i)

writer.writerow([denno, text])

cur.execute("INSERT INTO orders (denno, content) VALUES (%s, %s)", (denno, text))

# 分割保存

for denno, pages in denno\_pages.items():

new\_pdf = fitz.open()

for page\_num in pages:

new\_pdf.insert\_pdf(doc, from\_page=page\_num, to\_page=page\_num)

new\_pdf.save(f"{OUTPUT\_DIR}/{denno}.pdf")

csv\_file.close()

conn.commit()

cur.close()

conn.close()

print("完了しました。")

**💠 C# 版**

**🔧 使用ライブラリ（NuGet）**

* Tesseract (Tesseract OCR wrapper)
* PdfSharpCore または PdfPig（分割処理用）
* Npgsql（PostgreSQL接続）

Install-Package Tesseract

Install-Package PdfPig

Install-Package Npgsql

**📄 C#コード概要（OCR・PDF分割・DB挿入）**

using System;

using System.IO;

using System.Text.RegularExpressions;

using Tesseract;

using PdfPig;

using PdfPig.Writer;

using Npgsql;

using System.Collections.Generic;

class Program

{

static void Main()

{

string pdfPath = "input\\orders.pdf";

string outputDir = "output";

string csvPath = Path.Combine(outputDir, "data.csv");

Directory.CreateDirectory(outputDir);

var dennoPages = new Dictionary<string, List<int>>();

using var document = PdfDocument.Open(pdfPath);

using var csvWriter = new StreamWriter(csvPath);

csvWriter.WriteLine("denno,text");

using var conn = new NpgsqlConnection("Host=localhost;Username=your\_user;Password=your\_password;Database=your\_db");

conn.Open();

for (int i = 0; i < document.NumberOfPages; i++)

{

var page = document.GetPage(i + 1);

var text = page.Text;

var match = Regex.Match(text, @"(DEN\d{8})");

if (match.Success)

{

string denno = match.Value;

if (!dennoPages.ContainsKey(denno)) dennoPages[denno] = new List<int>();

dennoPages[denno].Add(i + 1);

csvWriter.WriteLine($"{denno},\"{text.Replace("\"", "\"\"")}\"");

using var cmd = new NpgsqlCommand("INSERT INTO orders (denno, content) VALUES (@d, @c)", conn);

cmd.Parameters.AddWithValue("d", denno);

cmd.Parameters.AddWithValue("c", text);

cmd.ExecuteNonQuery();

}

}

// 分割して保存

foreach (var kvp in dennoPages)

{

var denno = kvp.Key;

var outputPdfPath = Path.Combine(outputDir, $"{denno}.pdf");

using var newPdf = new PdfDocumentBuilder();

foreach (int pageIndex in kvp.Value)

{

newPdf.AddPage(document.GetPage(pageIndex));

}

using var fs = File.OpenWrite(outputPdfPath);

newPdf.Build(fs);

}

conn.Close();

Console.WriteLine("処理完了。");

}

}

**✅ まとめ**

| **処理** | **Python** | **C#** |
| --- | --- | --- |
| OCR | pytesseract + pdf2image | Tesseract |
| PDF読み書き | PyMuPDF | PdfPig, PdfSharpCoreなど |
| PostgreSQL操作 | psycopg2 | Npgsql |
| 実装難易度 | やや簡単 | やや複雑（型の扱い・環境構築） |

ご希望であれば、**VBAからPythonまたはC#を呼び出す連携方法**も案内可能です。  
さらに、\*\*テンプレプロジェクトや処理の高速化案（マルチプロセスなど）\*\*もご希望があれば対応可能です。