**OCR Text Extraction and Image Generation: pytesseract**

**Objective**

**This script extracts text from an image using OCR (pytesseract), filters it based on keywords, and generates individual images for the filtered information.**

**Prerequisites**

1. **Install Python libraries:**
   * **pytesseract**
   * **Pillow**
2. **pip install pytesseract pillow**
3. **Install Tesseract-OCR from** [**Tesseract OCR**](https://github.com/tesseract-ocr/tesseract) **and set its executable path in the script.**

**How It Works**

1. **Image Loading: Reads the input image using Pillow.**
2. **OCR Processing: Extracts text with pytesseract.**
3. **Filtering: Filters lines containing keywords like "Account Name," "Account Number," "Balance," etc.**
4. **Image Creation: Saves each filtered text snippet as an individual image in a specified output folder.**

**Output**

1. **Filtered information is displayed in the console.**
2. **Separate images for each detail are saved in the extracted\_images folder.**

**Usage**

1. **Update the image file path and Tesseract-OCR executable path.**
2. **Run the script and find the output in the extracted\_images directory.**

**This script simplifies OCR tasks for extracting and organizing text-based details visually.**

Code

# Import necessary libraries

import pytesseract

from PIL import Image, ImageDraw, ImageFont

import os

pytesseract.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

# Load the uploaded image

image\_path = "C:/Users/DELLL/Downloads/65.jpg"

image = Image.open(image\_path)

# Convert image to text using pytesseract

extracted\_text = pytesseract.image\_to\_string(image)

# Filter important details based on specific keywords

important\_info = []

keywords = ["Account Name", "Account Number", "Balance", "Txn Date", "Debit", "Credit"]

for line in extracted\_text.split('\n'):

if any(keyword in line for keyword in keywords):

important\_info.append(line)

# Create separate images for each piece of important information

output\_dir = 'extracted\_images'

os.makedirs(output\_dir, exist\_ok=True)

font = ImageFont.load\_default()

for i, info in enumerate(important\_info):

output\_image = Image.new('RGB', (600, 100), color=(255, 255, 255))

draw = ImageDraw.Draw(output\_image)

draw.text((10, 20), info, font=font, fill=(0, 0, 0))

output\_image\_path = os.path.join(output\_dir, f'extracted\_info\_{i+1}.jpg')

output\_image.save(output\_image\_path)

print(f"Saved: {output\_image\_path}")

print(f"All extracted information images are saved in the '{output\_dir}' directory.")