

OCR Menu Extraction Project

In this project, we have not scraped any website because I couldn't find any website that has menu images. Instead, I downloaded the image and worked on it.

Overview

This project demonstrates how to use Optical Character Recognition (OCR) to extract menu items and their prices from an image and store the data in a SQLite3 database. The extracted data is then loaded into a pandas DataFrame for further analysis.

Packages Used

- **easyocr**: For text extraction from the image.
- **cv2 (OpenCV)**: For image processing.
- **sqlite3**: For storing the extracted data in a database.
- **pandas**: For loading and analyzing the data.

Steps Taken

1. Image Loading and OCR:

- Loaded the image using OpenCV.
- Used easyocr to extract text from the image.

2. Text Parsing:

- Parsed the extracted text to identify menu items and their prices.
- Stored the parsed data in a dictionary.

3. Database Operations:

- Created a SQLite3 database and a table to store the menu items and prices.
- Inserted the parsed data into the database.

4. Data Retrieval and Analysis:

- Reopened the database connection and fetched all data.
- Loaded the data into a pandas DataFrame for further analysis.

Code

```
import easyocr
import cv2
import sqlite3
import pandas as pd
import os
import re

# Path to the image
path = 'menu.jpg'

# Check if the file exists
if not os.path.exists(path):
    raise FileNotFoundError(f"The file {path} does not exist.")

# Initialize easyocr reader
reader = easyocr.Reader(['en'])

try:
    # Read the image using OpenCV
    img = cv2.imread(path)
    if img is None:
        raise ValueError("The image file could not be read.")

    # Convert the image to RGB format
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Extract text using easyocr
    result = reader.readtext(img_rgb)

    # Concatenate the extracted text
    extracted_text = ""
    for detection in result:
        extracted_text += detection[1] + "\n"

    print(extracted_text)

except Exception as e:
    print(f"An error occurred: {e}")

# Split text into lines
```

```

lines = extracted_text.split("\n")
menu_dict = {}
price_pattern = re.compile(r'(\d+)\$')

# Parse the lines to extract menu items and their prices
current_item = ""
for line in lines:
    match = price_pattern.search(line)
    if match:
        item = line[:match.start()].strip()
        price = int(match.group())
        if current_item:
            item = current_item + " " + item
            current_item = ""
        if item:
            menu_dict[item] = price
    else:
        if line.strip():
            if current_item:
                current_item += " " + line.strip()
            else:
                current_item = line.strip()

print(menu_dict)

# Connect to SQLite database (or create it if it doesn't exist)
conn = sqlite3.connect('menu.db')
cursor = conn.cursor()

# Create a table to store the menu items and prices
cursor.execute('''
CREATE TABLE IF NOT EXISTS menu (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    item TEXT NOT NULL,
    price INTEGER NOT NULL
)
''')

# Insert the parsed data into the table
for item, price in menu_dict.items():
    cursor.execute('INSERT INTO menu (item, price) VALUES (?, ?)', (item, price))

# Commit the changes and close the connection

```

```
conn.commit()
conn.close()

print("Data has been successfully inserted into the database.")

# Reconnect to the SQLite database
conn = sqlite3.connect('menu.db')
cursor = conn.cursor()

# Fetch all data from the 'menu' table
cursor.execute('SELECT * FROM menu')
data = cursor.fetchall()

# Close the connection
conn.close()

# Create a pandas DataFrame from the fetched data
df = pd.DataFrame(data, columns=['id', 'item', 'price'])
print(df.tail(5))
```