Detailed Report

TASK-1(Medical Data)

Introduction

The provided code is a Python script that processes DICOM (Digital Imaging and Communications in Medicine) files. It extracts metadata and converts the pixel data of DICOM images into a visually simplified form, saving the result as a PNG image. Additionally, it exports the metadata of each DICOM file to a separate JSON file.

Architecture

Script Entry Point:

The script has a clear entry point at the end where the process_dicom_folder function is called with specific input and output paths.

process_dicom_folder Function:

- **Inputs**:
    1. **input_folder**: Path to the folder containing DICOM files.
    2. **output_folder**: Path to the folder where simplified images will be saved.
    3. **json_output_file**: Path to the folder where JSON metadata will be saved.

- **Functionality**:

  Iterates through DICOM files in the specified input folder.

  For each file, creates an instance of the dicomData class.

  Fetches metadata, extracts simplified image data, saves the processed image, and exports metadata to JSON.

dicomData Class:

- **Attributes**:
    1. **image_dicom**: Path to the DICOM file.
    2. **dicom_metadata**: A dictionary to store DICOM metadata.
    3. **simple_image_data**: Processed image data in a simplified format.

- **Methods**:
    1. **__init__**(self, image_dicom): Initializes the class with the provided DICOM file path.
    2. **fetch_metadata**(self): Reads the DICOM file and converts its metadata to a JSON format.

3. **extract_simple_image**(self): Processes the pixel data of the DICOM image.
4. **save_image**(self, output_folder): Saves the simplified image to the specified output folder.
5. **export_metadata_to_json**(self, output_json): Exports the DICOM metadata to a JSON file.

Interaction and Dependencies

- **External Libraries:**
  1. **pydicom**: Used for reading DICOM files and converting metadata to JSON.
  2. **numpy**: Utilized for numerical operations, particularly in processing pixel data.
  3. **PIL** (Pillow): Employed for creating and saving images.
- **File Handling:**

  The script interacts with the file system using the os module to create folders, join paths, and manage file names.

Approach and Technical Understanding

1. Metadata Extraction

The fetch_metadata method uses the pydicom library to read the DICOM file and convert its metadata to a JSON format. This allows for easy serialization and storage.

2. Image Processing

The extract_simple_image method processes the pixel data of the DICOM image using numpy. It rescales the pixel values to fit within the range of 0 to 255 and converts them to an unsigned 8-bit integer. The processed image is then converted to a PIL Image for saving in PNG format.

3. File Handling

The save_image method ensures the existence of the output folder and saves the processed image with a name derived from the original DICOM file.

The export_metadata_to_json method exports the DICOM metadata to a JSON file with a name derived from the original DICOM file.

4. Folder Processing

The process_dicom_folder function walks through the specified input folder and processes each DICOM file found, creating an instance of dicomData for each file.

5. Assumptions

The input folder can have multiple images , we have to pick only the images with ".dicom" extension.

TASK-2(ETL data from different Sources)

Introduction

The provided code is a Python script that is performing the ETL (Extract Transform Load) on different file types like .xlsx and .csv and then merging the all data and saving the data into a new file(.csv)

Architecture:

1. Data Loading:

- Energy data is loaded from the "Energy_Indicators.xls" Excel file using pandas.
- World bank data for GDP is loaded from the "API_NY.GDP.MKTP.CD_DS2_en_csv_v2_5871885.csv" file.
- ScimEn data, representing country rankings, is loaded from the "scimagojr country rank 1996-2022.xlsx" file.

2. Data Transformation and Cleaning:

- Energy data is transformed and cleaned:
- World bank GDP data is transformed and cleaned
- ScimEn data is transformed and cleaned

All three datasets are indexed by the "Country" column.

3. Merging Datasets:

- The ScimEn dataset is filtered to include only the top 15 ranked countries.
- Energy and ScimEn datasets are merged on the "Country" column.
- The merged dataset is then merged with the GDP dataset on the "Country" column.

4. Final Output:

The final merged dataset is printed and saved to a CSV file named "final.csv" in the "task2/output/" directory.

Country	Rank	Documents	Citable documents	Citations	...	2012	2013	2014	2015
					...				
China	1	360468	358777	3947871	...	8.532186e+12	9.570471e+12	1.047562e+13	1.106157e+13
United States	2	199442	195042	3068926	...	1.625397e+13	1.684319e+13	1.755068e+13	1.820602e+13
India	3	76103	74167	760964	...	1.827638e+12	1.856721e+12	2.039126e+12	2.103588e+12
Japan	4	56249	55680	633294	...	6.272363e+12	5.212328e+12	4.896994e+12	4.444931e+12
United Kingdom	5	52572	51156	909276	...	2.706341e+12	2.786315e+12	3.065223e+12	2.934858e+12
Germany	6	47781	46767	641717	...	3.527143e+12	3.733805e+12	3.889093e+12	3.357586e+12
Russian Federation	7	43567	43290	175721	...	2.208294e+12	2.292470e+12	2.059242e+12	1.363482e+12
Canada	8	39036	38276	787010	...	1.828366e+12	1.846597e+12	1.805750e+12	1.556509e+12
Italy	9	35991	34424	529459	...	2.086958e+12	2.141924e+12	2.162010e+12	1.836638e+12
South Korea	10	35294	35005	503147	...	1.278428e+12	1.370795e+12	1.484318e+12	1.465773e+12
Iran	11	29776	29448	511199	...	6.440355e+11	4.927756e+11	4.603828e+11	4.082129e+11
France	12	29351	28759	471469	...	2.683672e+12	2.811877e+12	2.855964e+12	2.439189e+12
Spain	13	27880	27272	515485	...	1.324751e+12	1.355580e+12	1.371821e+12	1.196157e+12
Australia	14	25906	25237	495278	...	1.546953e+12	1.576330e+12	1.467590e+12	1.350580e+12
Brazil	15	25887	25493	260540	...	2.465228e+12	2.472819e+12	2.456044e+12	1.802212e+12

[15 rows x 20 columns]
PS C:\project>

Approach and Technical Understanding:

1. Energy Data Processing:

- Initial columns of the energy dataset are selected and renamed.
- Units are standardized (Petajoules to Gigajoules).
- Missing values ("...") are replaced with NaN.
- Country names are cleaned and some are replaced for consistency.

2. World Bank GDP Data Processing:

- Unnecessary rows are skipped.
- Country names are standardized for consistency.

3. ScimEn Data Processing:

- The "Region" column is dropped from the ScimEn dataset.

4. Merging Datasets:

- The ScimEn dataset is filtered to include only the top 15 ranked countries.
- Datasets are merged based on the "Country" column.

5. Handling Missing Data:

- The final dataset is checked for missing values, and appropriate handling is performed.

6. Data Export:

- The final merged dataset is saved to a CSV file for further analysis.

Interactions and Dependencies:

- **Pandas** (pd):Used for loading, transforming, and cleaning data, as well as merging datasets.
- **NumPy** (np):Used for numerical operations and handling missing values.
- **Regular** Expressions (re):Used for text cleaning (removing parenthesized substrings from country names).
- **Datetime** (datetime):Used for obtaining the current year and defining a range of years.

TASK-3(Read Data from the Image)

Introduction

The provided code is a Python script that process the given image and read the content or data within the image or a specific part of image and then store the data into an excel file.

Architecture and Approach:

1. Initialization:

The script initializes an OCR reader using the easyocr library, focusing on the English language and utilizing GPU support.

Paths are defined for the input image, output Excel file, and a temporary cropped image.

2. Image Processing:

The script uses OpenCV to read an image and convert it from BGR to RGB format.

A binary mask is created to identify regions with red color intensity within specified limits.

Contours are found in the binary mask using OpenCV's findContours function.

Assuming the first contour corresponds to a red rectangle, the script crops the image based on the bounding box of this contour.

3. OCR Processing:

The easyocr.Reader is utilized to perform Optical Character Recognition (OCR) on the cropped image.

Extracted text is returned from the extract_text function.

4. Excel Output:

The extracted text is saved to a DataFrame using pandas.

The DataFrame is then saved to an Excel file at the specified path (excel_path).

5. Cleanup:

After processing, the temporary cropped image is deleted.

6. Assumption:

The image has only one red rectangular box, although code can be modified to run for more than one red box. Only image has to be processed, therefore the format of the of the output data is a single list, if the input is a pdf file data can be stored in a tabular format.

Interaction and Dependencies

- **OpenCV (cv2):** It is a computer vision library used for image processing tasks.
- **PIL (Pillow) (Image)**: It is the Python Imaging Library, and Pillow is a modern version of it.
- **EasyOCR (easyocr):** It is a deep learning-based OCR library.

```
                              0
0                         PO; #
1                Sales Rep: Name
2                      Ship Date
3                       Ship Via
4                          Tenms
5                       Due Date
6                      3/18/2015
7                      #ITaxable
8                    Description
9                       Quantity
10                   Unit Price
11                   Lime Total
12                         P1002
13  Test Product 3 (Non-taxable_
14                        300.00
15                        300.00
16                         P1001
17       Test Product 2 (Service
18                        200.00
19                        200.00
20                         P1ooo
21                Test Product -
22                        100.00
23                        100.00
File 'task3/input_data/cropped_image.png' deleted successfully.
```