1.Image Slicing and Rotation

This project demonstrates image manipulation by dividing an image into four quadrants, rotating each quadrant by specific angles, and finally reassembling the modified quadrants back into a single image. The project uses Python, leveraging libraries such as numpy, PIL, and matplotlib for image handling and visualization.

Table of Contents

Requirements

This project requires the following libraries:

- numpy for array manipulation.
- Pillow (PIL) for image processing.
- matplotlib for displaying images.

Installation

1. Clone the repository:

   bash
   Copy code

   ```
   git clone https://github.com/your-username/image-slicing-rotation.git
   cd image-slicing-rotation
   ```

2. Install the dependencies:

   bash
   copy code

   ```
   pip install numpy pillow matplotlib
   ```

3. Place an image named tavo_nuotrauka.jpg in the project directory. This image will be used as the input.

Usage

Run the script to slice, rotate, and reassemble the image:

bash
Copy code

```
python main.py
```

The script will:

1. **Load** tavo_nuotrauka.jpg.
2. Slice the image into four quadrants.
3. Display each quadrant individually.
4. Rotate each quadrant by 90, 180, and 270 degrees, displaying each rotated version.
5. Reassemble the rotated quadrants and display the final result.

Example Output

Each quadrant is displayed with the original and rotated versions, followed by the final reassembled image.

Features

- Slices an image into four quadrants.
- Rotates each quadrant by various angles (90°, 180°, 270°).
- Displays the final reassembled image with rotated quadrants.

License

This project is licensed under the MIT License.

Web Scraper for Pitbull Characteristics Article

This project is a web scraper designed to extract and save text content from a specific article on pitbull characteristics, hosted on the website "https://pitbuliai.lt/".

Requirements

This script requires the following Python libraries:

- requests for fetching webpage content.
- BeautifulSoup from bs4 for parsing HTML and extracting content.

How It Works

1. **Fetch the Page Content**: The script uses requests to retrieve the HTML of the specified URL.
2. **Parse the HTML**: BeautifulSoup is used to parse the HTML and locate the main content.
3. **Extract Text Content**: The content within the specified div element with class entry-content is extracted.
4. **Save the Content**: The extracted text is saved in a file named output.txt.

Instructions

1. Clone the repository or copy the script to your working directory.
2. Install the required libraries, if not already installed:

   ```
   bash, copy code
   pip install requests beautifulsoup4
   ```

3. Run the script:

   ```
   bash
   Kopijuoti kodą
   python main.py
   ```

4. Upon successful completion, the article content will be saved in output.txt.

Error Handling

- If the page cannot be retrieved, an error message with the HTTP status code is displayed.
- If the content is not found within the specified HTML structure, a message will indicate that the content was not found.

Notes

- This script is designed specifically for the page structure on "https://pitbuliai.lt/". If the page structure changes, the script may need adjustments.