Object Recognition and Feature Matching

Overview

This project demonstrates how to perform object recognition and feature matching using OpenCV's SIFT (Scale-Invariant Feature Transform) algorithm in Python. The script identifies objects in a base image by matching them with provided object images. It uses feature detection and descriptor matching to locate and annotate objects in the base image.

Features

- **SIFT Feature Detection**: Detect and compute features in images.

- **Feature Matching**: Match features between object images and the base image using FLANN-based matcher.

- **Homography Transformation**: Compute homography to align object images with the base image.

- **Visualization**: Display results with matched features and annotations.

Requirements

- Python 3.6 or later

- OpenCV 4.x

- NumPy

Installation

Set Up a Python Environment

It’s recommended to use a virtual environment to manage dependencies. Create and activate a virtual environment:

```
python -m venv env

source env/bin/activate  # On Windows use: env\Scripts\activate
```

Install Dependencies

Create a requirements.txt file with the following content:

```
numpy

opencv-python
```

Install the dependencies using pip:

pip install -r requirements.txt

Usage

Running the Script

1. **Prepare Your Images**:

   - ✓ **Base Image**: Choose an image in which you want to search for objects. For example, `desk.jpg`.
   - ✓ **Object Images**: Select images of the objects you want to find in the base image. For example, `book.jpg` and `pen.jpg`.

2. **Execute the Script**:

   Run the script by passing the path to the base image followed by the paths to the object images. Here's an example command:

   python run_matching.py C:/path/to/base_image/desk.jpg C:/path/to/object_images/book.jpg C:/path/to/object_images/pen.jpg

- ➢ **`C:/path/to/base_image/desk.jpg`**: The path to the base image where objects will be detected.
- ➢ **`C:/path/to/object_images/book.jpg`**: The path to the first object image (e.g., a book).
- ➢ **`C:/path/to/object_images/pen.jpg`**: The path to the second object image (e.g., a pen).

This command will process the base image and attempt to find and mark the objects specified in the object images.

Files

- **run_matching.py**: The main Python script for performing object recognition and feature matching.
- **requirements.txt**: Lists the required Python packages for the project.

Troubleshooting

- **Error loading image**: Ensure that the image paths are correct and that the files exist and are accessible. Check that the image files are not corrupted.
- **OpenCV errors**: Verify that OpenCV and its dependencies are correctly installed. Check compatibility issues if errors occur.

Contact

For questions or support, please contact my email…

Divya Priya, "sdivyapriya1521@gmail.com". This documentation provides a clear and structured guide for users to set up and use your project, including installation steps, usage instructions, and troubleshooting tips.