

User Manual

Project Overview

This project serves as an example demonstrating how to use the custom Python image preprocessing library `image_preprocessing` to convert a color image into a black-and-white image. The library is installed via a GitHub repository and leverages the `to_black_and_white` function to process images.

Features

- Reads a color image named `original.jpg` from a specified directory.
- Converts the color image to a black-and-white (grayscale) image.
- Saves the processed image as `output.jpg`.

Prerequisites

- Python 3.6+ (Python 3.8 or later is recommended)
- Pip package manager

Project Dependencies

The project depends on the following libraries:

- **image_preprocessing**: A custom image preprocessing library installed from a specified GitHub repository.
- **Pillow**: Used for image loading and transformation. It is a dependency of the `image_preprocessing` library.

When installing `requirements.txt`, these dependencies will be automatically installed.

Installation Steps

1. **Clone the project to your local machine:**

```
git clone https://github.com/8086X/image-preprocessing-example.git
```

2. **Navigate to the project directory:**

```
cd image-preprocessing-example
```

3. **Install dependencies:**

```
pip install -r requirements.txt
```

This command will install the `image_preprocessing` library from the specified GitHub repository and automatically install the required Pillow library.

Usage Instructions

1. Ensure that there is a color image named `original.jpg` inside the `examples` folder. If not, prepare a color image and place it in the `examples` folder.
2. Run the following command from the project root directory:

```
python main.py
```

3. Once the program completes, the processed image will be saved as `output.jpg` inside the `examples` folder. Open the file to view the black-and-white converted image.

Troubleshooting

1. **Encoding Issues:**

- If you encounter encoding errors during installation or execution, ensure the `__init__.py` file in the `image_preprocessing` library is not empty and the file encoding is set to UTF-8.

2. **Slow or Failed Dependency Installation:**

- Verify your network connection.
- Consider using a domestic mirror source for faster installation.

3. **Module Not Found Error:**

- Ensure that you have installed dependencies in the correct virtual environment or Python environment.
- Re-run the installation command:

```
pip install -r requirements.txt
```

Future Extensions

- Modify or extend the `image_preprocessing` library to add more image preprocessing functionalities, such as cropping, rotating, scaling, or adding filters.
- Add test cases using `pytest` or `unittest` to automate testing.
- Optimize the code structure and comments, and enable the **Issues** feature on GitHub to track problems and suggestions.