

Template Matching with OpenCV

This project demonstrates the use of template matching in OpenCV to find specific patterns or objects within an image. It utilizes the `matchTemplate` function from the OpenCV library to compare a template image against a target frame to find the best match.

Prerequisites

Before you begin, ensure you have met the following requirements:

- Python 3.x
- OpenCV library (cv2)
- NumPy

You can install OpenCV and NumPy using pip:

```
```bash
pip install opencv-python numpy
```
```

Usage

To use this template matching script, you need to have two images:

1. The `template` image, which is the pattern you want to find.
2. The `target` image, where you want to find the pattern.

Place these images in a suitable directory and update the file paths in the script accordingly.

Script Overview

The script performs the following steps:

1. Reads the `template` and `target` images in grayscale.
2. Displays the original images.
3. Performs template matching using `cv2.matchTemplate`.
4. Finds the location with the highest matching score.
5. Displays the matching result with a visual indicator on the location.

Running the Script

To run the script, navigate to the script's directory and run:

```
```bash
python template_matching.py
```
```

Ensure that the image paths in the script are correct. The script will display the original images and the result of the template matching.

Contributing

Contributions to this project are welcome. To contribute:

1. Fork the project (<https://github.com/your-username/template-matching-opencv>).
2. Create a branch for your feature (``git checkout -b feature/AmazingFeature``).
3. Commit your changes (``git commit -m 'Add some AmazingFeature'``).
4. Push to the branch (``git push origin feature/AmazingFeature``).
5. Open a pull request.

Contact

If you have any questions or comments about the project, please feel free to open an issue or a pull request.