



Факультет программной инженерии и компьютерной техники

**Лабораторная работа №2 по
«Компьютерное зрение»**

Преподаватель: Денисов Алексей Константинович

Выполнили: Ахмаров Руслан

Бейбитхан Халифа

Лизиков Никита

Группа: P4144, P4155

Цель работы:

Реализовать простейшие алгоритмы сопоставления изображений.

Задание:

1. Реализовать программу согласно описанию. Можно использовать языки C++ или Python и любые библиотеки.
2. Сравнить качество работы двух вариантов реализации по точности детектирования.
3. Сделать отчёт в виде readme на GitHub, там же должен быть выложен исходный код.

Отчёт должен содержать следующие пункты:

1. Теоретическая база
2. Описание разработанной системы (алгоритмы, принципы работы, архитектура)
3. Результаты работы и тестирования системы (скриншоты, изображения, графики, закономерности)
4. Выводы по работе
5. Используемые источники

1. Теоретическая база

Template Matching — метод прямого сопоставления шаблона с изображением. Использует корреляцию между пикселями шаблона и входного изображения. Подходит для задач, где масштаб, ориентация и освещенность объектов не меняются.

SIFT (Scale-Invariant Feature Transform) — алгоритм выделения и сопоставления ключевых точек. Он устойчив к изменению масштаба, вращению и освещению, что делает его более универсальным, чем Template Matching.

2. Описание системы

Код на Python с использованием OpenCV реализует два метода:

1. Template Matching:

- Используется функция `cv2.matchTemplate` для сопоставления.
- Выводится прямоугольная рамка вокруг области с наибольшим совпадением.

2. SIFT:

- Генерация ключевых точек и дескрипторов с помощью `cv2.SIFT_create()`.
- Сопоставление точек между шаблоном и изображением через `cv2.BFMatcher`.
- Построение гомографии для определения положения шаблона.

3. Результаты тестирования

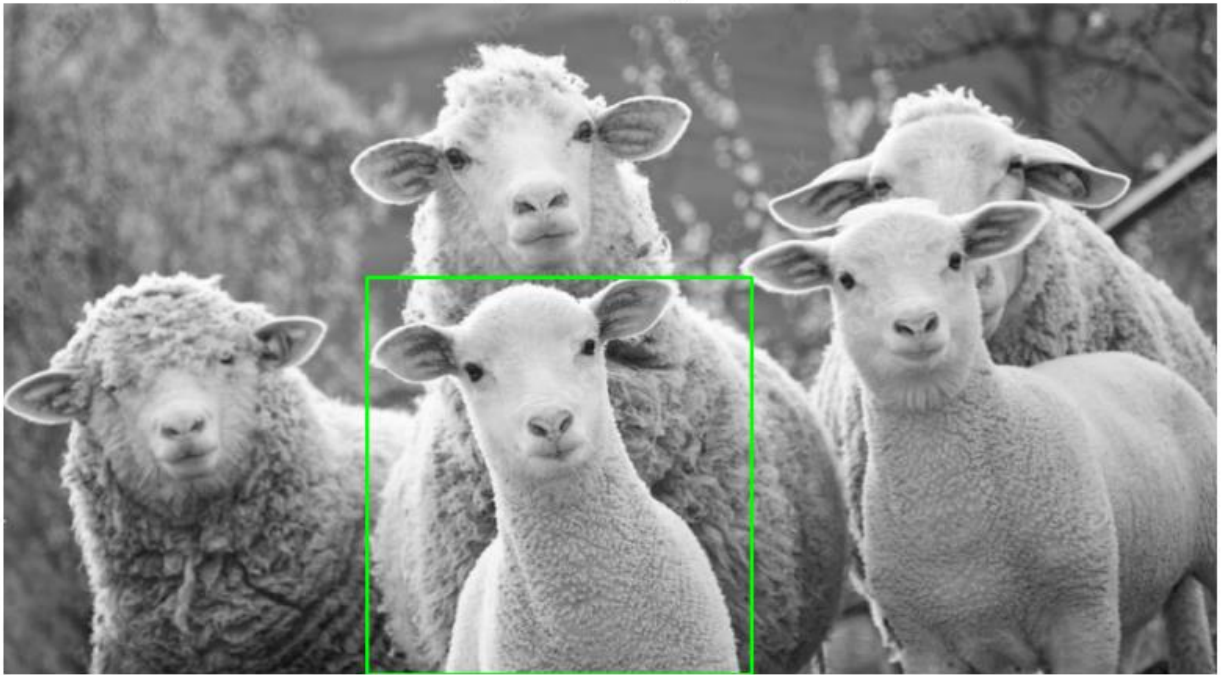
Тестовые данные:

Использованы изображения:

- **Шаблон:** часть основного изображения.
- **Входное изображение:** сцены с объектами под разным углом или масштабом.

1. Template Matching

Template Matching Result

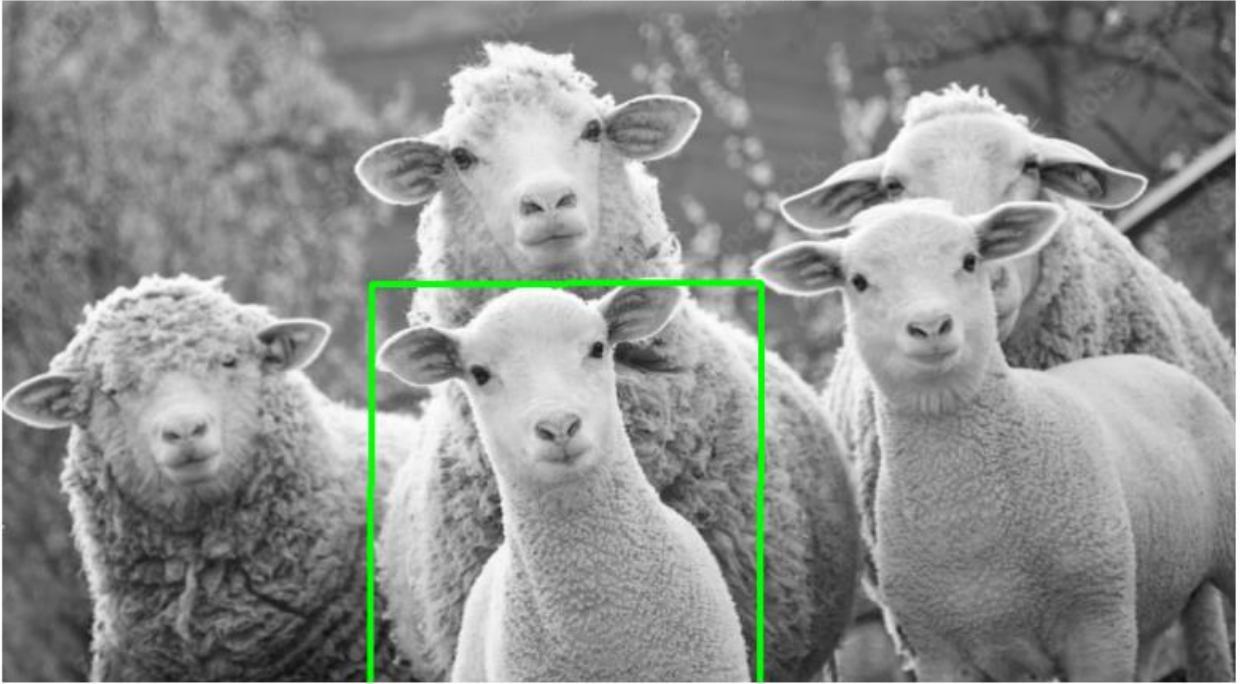


2. SIFT Matching

Filtered Keypoints Matching



SIFT Matching with Homography



Template Matching Result



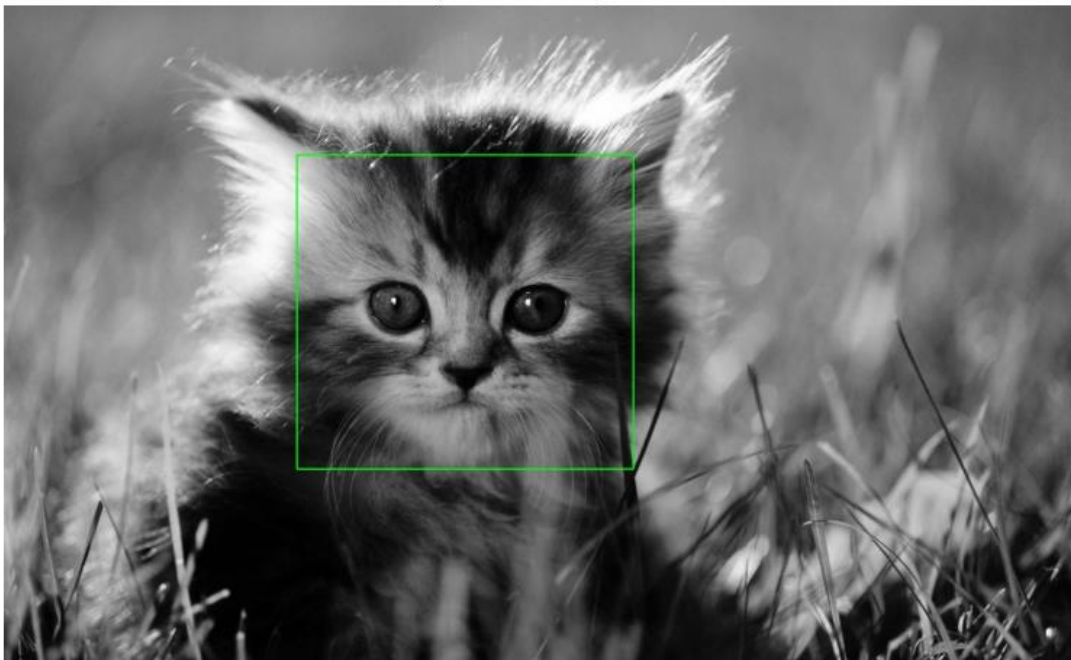
Filtered Keypoints Matching



SIFT Matching with Homography



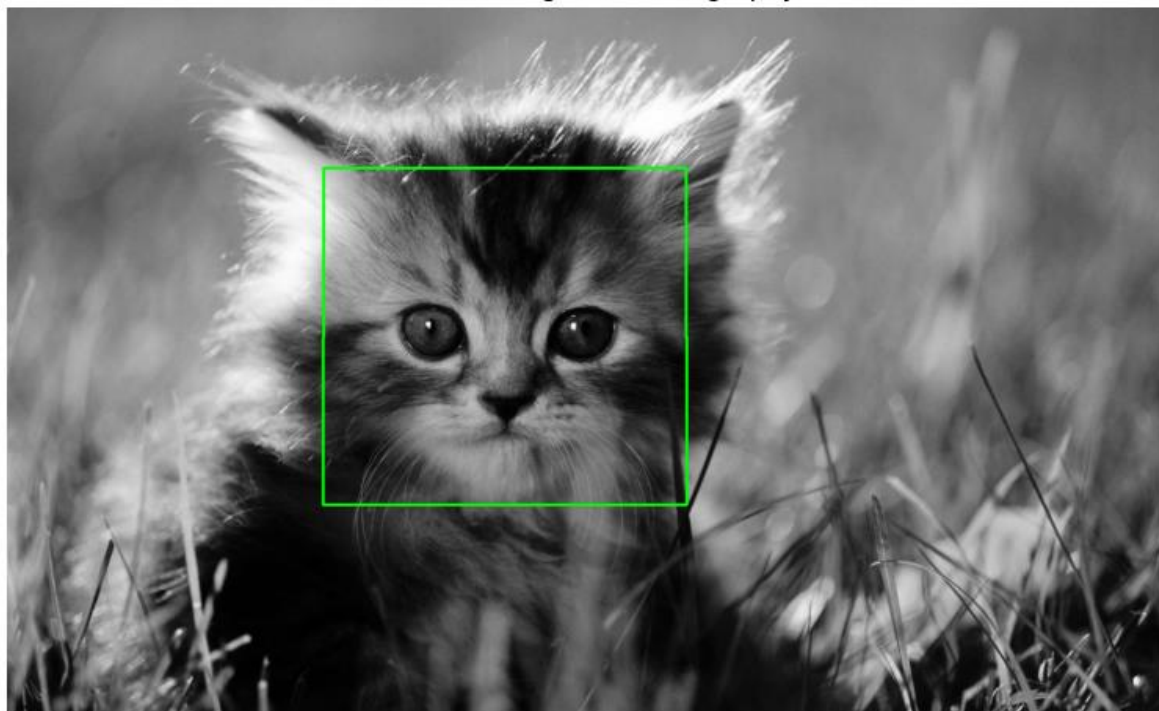
Template Matching Result



Filtered Keypoints Matching



SIFT Matching with Homography



4. Выводы

Template Matching удобен для простых задач, но ограничен при изменении угла, масштаба и освещенности.

SIFT более универсален, но требует больше вычислительных ресурсов.

5. Исходный код

Код опубликован на GitHub: <https://github.com/Nolasar/CV-semester-1>

6. Используемые источники

- OpenCV Documentation. <https://docs.opencv.org/>
- Lowe, D.G. (1999). *Object recognition from local scale-invariant features*.
- Визуализация результатов в Matplotlib: [Matplotlib.org](https://matplotlib.org/)