**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный университет**

**Факультет прикладной математики-процессов управления**

**Кафедра “фундаментальная информатика и информационные технологии”**

**отчет**

**по лабораторной работе №2**

**по дисциплине «Функциональное программирование»**

**на тему «Анализ астрономических изображений»**

**Вариант – 1**

|  |  |  |
| --- | --- | --- |
| Студент гр. 24Б16-пу |  | Бублик И.О. |
| Преподаватель |  | Киямов Ж. У. |

**Санкт-Петербург**

**2025 г**

Оглавление

[Цель работы 3](#_Toc179821939)

[Формализация задачи 3](#_Toc179821940)

[Основные шаги программы 6](#_Toc179821941)

[Описание программы 7](#_Toc179821942)

[Рекомендации по работе с программой 8](#_Toc179821943)

[Вывод 10](#_Toc179821944)

**Цель работы**

Целью данной работы является разработка программы для анализа больших объемов космических данных, полученных с телескопа, с использованием параллельных вычислений. Программа должна эффективно обрабатывать изображения, обнаруживать астрофизические объекты и собирать статистику по ним, оптимизируя процесс с помощью многопоточной обработки, что позволяет значительно сократить время выполнения анализа.

Эта цель включает:

* Применение параллельных вычислений для ускорения анализа.
* Обеспечение точного выделения и классификации астрофизических объектов.
* Сбор статистики по яркости, площади и координатам найденных объектов.

Формализация задачи

Задача анализа космических данных формализуется следующим образом:

1. **Входные данные**:
   * Набор изображений космических объектов, представленных в виде матриц пикселей. Каждое изображение содержит миллионы пикселей и представлено в формате, подходящем для обработки (например, градации серого).
2. **Цель**:
   * Разработка программы, которая:
     + Параллельно обрабатывает изображения, используя доступные ядра процессора.
     + Выделяет астрофизические объекты на каждом изображении.
     + Измеряет и классифицирует характеристики объектов.
     + Собирает статистические данные об объектах, включая яркость, координаты и площадь.
3. **Методы анализа**:
   * **Бинаризация**: Преобразование изображения в бинарное, где пиксели, соответствующие объектам, выделяются на фоне.
   * **Сегментация**: Использование методов выделения объектов (например, алгоритм связных компонент) для нахождения и характеристики объектов на бинаризованном изображении.
   * **Сбор статистики**: Рассчет статистических характеристик (яркость, площадь и координаты) для каждого выделенного объекта.
4. **Параллельная обработка**:
   * Использование ProcessPoolExecutor для распределения задач анализа изображений между несколькими процессами, что позволяет одновременно обрабатывать несколько изображений.
5. **Выходные данные**:
   * CSV файл, содержащий собранную статистику по каждому найденному астрофизическому объекту, включая его яркость, координаты и другие характеристики, а также сохраненные аннотированные изображения.
6. **Ограничения**:
   * Учитывать ограниченные вычислительные ресурсы и избегать гонок данных при параллельной обработке.
   * Обеспечить корректность обработки изображений, учитывая возможные ошибки при загрузке или чтении данных.

Теоретическая часть

1. **Обработка изображений**:
   * Обработка изображений представляет собой важный аспект анализа данных, получаемых с телескопов. Основные этапы включают предобработку, выделение объектов и анализ характеристик.
   * В данном проекте используется библиотека OpenCV для обработки изображений. Бинаризация изображений позволяет выделить объекты на фоне, что является первым шагом к их анализу.
2. **Бинаризация изображений**:
   * Бинаризация — это процесс преобразования изображения в черно-белый формат, где пиксели делятся на два класса: объект (белый) и фон (черный). Один из популярных методов бинаризации — пороговая фильтрация, при которой пиксели с яркостью выше заданного порога становятся белыми, а остальные — черными.
3. **Сегментация и выделение объектов**:
   * После бинаризации необходимо выделить астрофизические объекты. Это можно сделать с помощью алгоритмов сегментации, таких как алгоритм связных компонент.
   * В данном проекте применяется функция label() из библиотеки skimage, которая находит и маркирует отдельные объекты на бинаризованном изображении. Затем, используя regionprops(), извлекаются характеристики объектов, такие как площадь и координаты центра.
4. **Анализ характеристик объектов**:
   * После выделения объектов необходимо собрать статистику. Для каждого объекта вычисляются его характеристики:
     + **Яркость**: определяется как сумма значений пикселей, входящих в область объекта.
     + **Площадь**: вычисляется как количество пикселей, относящихся к объекту.
     + **Координаты центра**: определяются как средние значения координат пикселей, входящих в область объекта.
5. **Параллельные вычисления**:
   * Параллельные вычисления позволяют значительно ускорить процесс обработки большого объема данных. В этом проекте используется библиотека concurrent.futures с классом ProcessPoolExecutor, который позволяет распределять задачи между несколькими процессами.
   * Каждый процесс выполняет обработку отдельного изображения, что позволяет оптимально использовать ресурсы многоядерных процессоров и сократить общее время выполнения программы.
6. **Сбор и хранение результатов**:
   * Собранная статистика о найденных астрофизических объектах сохраняется в формате CSV, что позволяет удобно анализировать результаты и передавать их исследователям.
   * Аннотированные изображения, на которых выделены обнаруженные объекты, также сохраняются для визуального анализа.

Основные шаги программы

1. **Импорт необходимых библиотек**:
   * В начале программы импортируются библиотеки, необходимые для обработки изображений, работы с данными и выполнения параллельных вычислений. Ключевыми библиотеками являются cv2 для обработки изображений, numpy для работы с массивами, skimage для анализа изображений и pandas для обработки и сохранения статистики.
2. **Загрузка изображений с NASA API**:
   * Программа начинается с загрузки изображений космических объектов с помощью API NASA. Используя функцию download\_image(), программа отправляет запрос к API и получает набор изображений. Загруженные изображения сохраняются на локальном диске для последующей обработки.
3. **Анализ изображений**:
   * Для каждого загруженного изображения вызывается функция analyze(). Эта функция отвечает за:
     + Загрузку изображения в градациях серого.
     + Бинаризацию изображения для выделения объектов.
     + Сегментацию объектов с использованием алгоритма связных компонент.
     + Сбор статистики по каждому выделенному объекту, включая яркость, площадь и координаты.
4. **Параллельная обработка изображений**:
   * Используя ProcessPoolExecutor, программа осуществляет параллельную обработку изображений. Это позволяет одновременно обрабатывать несколько изображений, что значительно ускоряет выполнение анализа. Результаты анализа для каждого изображения собираются и объединяются в общий список.
5. **Сохранение статистики**:
   * После завершения обработки всех изображений статистика по найденным объектам конвертируется в формат DataFrame с использованием библиотеки pandas. Затем результаты сохраняются в CSV файл для удобного анализа и передачи исследователям.
6. **Сохранение аннотированных изображений**:
   * Для каждого изображения, на котором были обнаружены астрофизические объекты, программа сохраняет аннотированное изображение с выделенными объектами. Это позволяет визуально оценить результаты обработки.
7. **Вывод результатов**:
   * В конце работы программы выводятся сообщения о статусе выполнения, включая информацию о загруженных изображениях, сохранении аннотированных изображений и завершении сохранения статистики.

Описание программы

1. **Используемые библиотеки**:
   * **OpenCV (cv2)**: Библиотека для компьютерного зрения, использующаяся для обработки изображений, таких как чтение, бинаризация и аннотирование изображений.
   * **NumPy (numpy)**: Библиотека для работы с многомерными массивами и матрицами, а также для выполнения математических операций над ними. Используется для расчета яркости и обработки данных.
   * **scikit-image (skimage)**: Библиотека для обработки изображений, обеспечивающая инструменты для анализа изображений. В программе используются функции для выделения связных компонентов и измерения характеристик объектов.
   * **Pandas (pandas)**: Библиотека для обработки и анализа данных, используемая для создания и сохранения таблиц с собранной статистикой в формате CSV.
   * **Requests (requests)**: Библиотека для выполнения HTTP-запросов, используемая для загрузки изображений с API NASA.
   * **Concurrent Futures (concurrent.futures)**: Библиотека для параллельного выполнения задач, позволяющая использовать пул процессов для обработки изображений.
2. **Основные функции программы**:

|  |  |  |
| --- | --- | --- |
| **Название функции** | **Входная переменная** | **описание** |
| download\_image | count | Загружает изображения космических объектов с API NASA и сохраняет их на локальном диске. Возвращает список путей к загруженным изображениям. |
| analyze | path | Обрабатывает одно изображение, выполняя бинаризацию, выделение объектов и сбор статистики. Возвращает список статистических данных о найденных объектах и аннотированное изображение. |

1. **Переменные программы**:

|  |  |
| --- | --- |
| **Название переменной** | **Описание** |
| API\_KEY | Переменная для хранения API ключа для доступа к сервисам NASA. |
| BASE\_DIR | Путь к папке, где хранится проект |
| IMG\_DIR | Путь к папке, где будут хранится скачанные изображения |
| OUT\_DIR | Путь к папке, где будут хранится обработанные изображения |
| paths | Список, содержащий пути к загруженным изображениям |
| csv\_path | |  | | --- | | Итоговый файл с данными о каждом найденном обьекте | |

### 

### 

### Рекомендации пользователю

1. **Запуск программы:**
   * Для запуска программы используйте командную строку или терминал. Перейдите в каталог, где сохранен ваш код, и выполните:
   * **python main.py**
   * Убедитесь, что у вас есть права на запись в папки, в которые программа будет сохранять изображения и статистику.
2. **Обработка больших объемов данных:**
   * Если у вас есть большая коллекция изображений, вы можете модифицировать программу для обработки данных по частям или использовать более мощный компьютер с многоядерным процессором для повышения производительности.
   * Обратите внимание на потребление памяти, так как работа с большим количеством изображений может потребовать значительных ресурсов.
3. **Анализ результатов:**
   * Результаты анализа сохраняются в файле astro\_data\_stats.csv. Вы можете открыть этот файл в любом редакторе таблиц, таком как Microsoft Excel или Google Sheets, для удобного просмотра и анализа статистики.

Рекомендации программисту

1. **Настройка окружения:**
   * Перед запуском программы убедитесь, что у вас установлены все необходимые библиотеки. Вы можете использовать pip для установки зависимостей:
   * **pip install opencv-python numpy scikit-image pandas requests**
   * Убедитесь, что у вас установлена последняя 12 версия Python, так как некоторые функции могут не поддерживаться в более старых версиях.
2. **Получение API ключа:**
   * Чтобы загрузить изображения с NASA API, необходимо получить API ключ. Вы можете зарегистрироваться на сайте NASA и получить свой ключ. После этого замените значение переменной **API\_KEY** в коде на ваш собственный ключ.
3. **Модификация программы:**
   * Программа может быть доработана для добавления новых функций, таких как фильтрация объектов по различным критериям (яркость, площадь и т.д.) или более сложные методы анализа изображений (например, применение нейронных сетей для классификации объектов).
   * Вы можете добавить обработку ошибок и улучшить сообщения о статусе выполнения, чтобы улучшить пользовательский опыт.
4. **Обработка больших объемов данных:**
   * В случае обработки большого объема данных, рекомендуется улучшить программу с использованием многозадачности или многопроцессорности для повышения производительности, например, через модуль concurrent.futures или multiprocessing.
   * Обратите внимание на оптимизацию работы с памятью при анализе множества изображений.

# **Вывод**

В ходе выполнения работы была разработана программа для анализа космических данных, полученных с телескопа, с использованием параллельных вычислений. Программа эффективно справляется с задачей обработки большого объема изображений, применяя методы компьютерного зрения для выявления и анализа астрофизических объектов.

Основные результаты работы программы включают:

1. **Параллельная обработка**: Использование многопроцессорных вычислений значительно ускоряет процесс анализа, позволяя одновременно обрабатывать сотни и тысячи изображений. Это критически важно для эффективной работы с большими объемами данных.
2. **Анализ изображений**: Программа успешно выполняет бинаризацию изображений, выделение объектов и сбор статистических данных. Выявленные объекты анализируются по различным характеристикам, таким как яркость, площадь и координаты.
3. **Статистика**: Полученные результаты сохраняются в формате CSV, что упрощает дальнейший анализ и использование данных исследователями. Аннотированные изображения также сохраняются, что позволяет визуально оценивать результаты работы программы.
4. **Расширяемость**: Программа спроектирована с возможностью дальнейшего расширения и модификации. Пользователи могут адаптировать функциональность в соответствии с конкретными задачами и требованиями.
5. **Практическое применение**: Разработанная программа может быть использована в астрофизических исследованиях, в частности для поиска и анализа различных объектов в космосе, таких как звезды, галактики и другие астрономические явления.

Таким образом, достигнутые результаты подтверждают, что применение параллельных вычислений в анализе космических данных является эффективным подходом, который позволяет значительно сократить время обработки и улучшить качество анализа.

Исходный код:  
https://github.com/Ilya2914/system\_and\_functionalprogramy