**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Функциональное программирование»**

**на тему «Анализ космических данных с использованием параллельных вычислений»**

**Студент гр. 23Б15-пу**

**Абрахин Е.Д.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#_Toc179825326)
2. [Описание задачи (формализация задачи) 4](#_Toc179825327)
3. [Теоретическая часть 6](#_Toc179825328)
4. [Основные шаги программы 8](#_Toc179825329)
5. [Описание программы 10](#_Toc179825330)
6. [Рекомендации пользователя 11](#_Toc179825331)
7. [Рекомендации программиста 12](#_Toc179825332)
8. [Исходный код программы 12](#_Toc179825333)
9. [Контрольный пример 13](#_Toc179825334)
10. [Вывод 15](#_Toc179825335)
11. [Литература 16](#_Toc179825336)

# Цель работы

Целью данной работы является разработка программы, использующей параллельные вычисления для эффективного анализа больших объемов данных, полученных с космического телескопа. Программа должна обеспечить быструю обработку сотен тысяч изображений космических объектов с целью выявления интересующих астрофизических объектов и сбора статистических данных о них.

# Описание задачи (формализация задачи)

Необходимо разработать программное обеспечение для анализа данных с космического телескопа с использованием параллельных вычислений. Входные данные представляют собой наборы изображений, где каждое изображение — это матрица пикселей, содержащая информацию о космических объектах. Основная цель анализа заключается в автоматическом выявлении астрофизических объектов на изображениях и сборе их характеристик.

Задача делится на несколько подзадач:

1. Параллельная обработка данных:

Необходимо реализовать механизм параллельной обработки изображений, используя многопоточность или мультипроцессорные вычисления для ускорения обработки большого объема данных. Программа должна быть способна эффективно распределять задачи между ядрами процессора для обработки нескольких изображений одновременно.

1. Анализ изображений:

Для каждого изображения необходимо выполнить следующие операции:

* Выделение астрофизических объектов (например, звёзд, галактик) на изображении.
* Измерение характеристик каждого объекта: яркость, координаты, размер.
* Классификация объектов на основе их характеристик.

1. Сбор статистики:

По результатам анализа каждого изображения необходимо собрать статистические данные для каждого найденного объекта:

* Яркость.
* Координаты (положение объекта на изображении).
* Размер (площадь или другие характеристики формы).

1. Управление ресурсами:

Программа должна эффективно управлять вычислительными ресурсами (процессорами, памятью) для оптимальной производительности. Необходимо предусмотреть защиту от гонок данных, блокировок и других проблем, связанных с параллельными вычислениями.

1. Вывод результатов:

Программа должна генерировать отчет с собранной статистикой по всем обработанным изображениям, содержащий информацию обо всех найденных объектах и их характеристиках. Этот отчет должен быть представлен в удобном для анализа виде.

# Теоретическая часть

Астрономия и астрофизика требуют обработки и анализа огромных объемов данных, получаемых с различных телескопов и наблюдательных платформ. С ростом возможностей астрономических инструментов увеличиваются и размеры получаемых данных. Например, современные космические телескопы могут фиксировать изображения с разрешением в миллионы пикселей, что делает их анализ крайне сложным и трудоемким. Использование параллельных вычислений становится необходимым для эффективной обработки таких больших объемов данных.

Параллельные вычисления представляют собой метод обработки данных, который позволяет выполнять множество вычислений одновременно. Это достигается путем разделения задачи на более мелкие подзадачи, которые могут обрабатываться независимо друг от друга. Использование параллельных вычислений позволяет значительно сократить время обработки, особенно когда работа осуществляется на многоядерных процессорах.

Существует несколько подходов к реализации параллельных вычислений, включая многопоточность и многопроцессорные вычисления. В контексте анализа изображений использование многопроцессорной обработки позволяет запускать несколько процессов, каждый из которых обрабатывает свою часть данных.

Обработка изображений включает в себя ряд этапов, которые помогают выделить интересующие объекты и извлечь их характеристики. Важные шаги в этом процессе включают:

1. Преобразование цветового пространства: Для упрощения анализа изображения часто выполняется преобразование в градации серого. Это помогает уменьшить сложность обработки, так как из трех цветовых каналов (RGB) остается только один.
2. Сглаживание изображения: Применение фильтров (например, гауссовского размытия) позволяет уменьшить шум в изображении, что улучшает качество дальнейшего анализа.
3. Бинаризация: Преобразование изображения в бинарное (черно-белое) позволяет выделить объекты на фоне, используя пороговые значения.
4. Выделение контуров: Использование алгоритмов для нахождения контуров объектов на бинарном изображении позволяет извлечь формы, которые могут соответствовать астрофизическим объектам.
5. Измерение характеристик: После выделения контуров необходимо измерить различные характеристики объектов, такие как площадь, координаты, яркость и другие параметры, которые могут быть использованы для классификации.

Классификация астрофизических объектов на основе их характеристик — это важная задача, которая помогает астрономам лучше понять природу наблюдаемых объектов. В приведенном коде реализована простая система классификации, основанная на размере и яркости объектов:

* Звезды: маленькие объекты с высокой яркостью.
* Планеты: объекты среднего размера с умеренной яркостью.
* Галактики: крупные объекты с очень высокой яркостью.
* Неизвестные объекты: объекты, не подходящие под вышеуказанные критерии.

Эта классификация основана на эмпирических данных и может быть расширена с использованием машинного обучения для более точного определения типов объектов.

# Основные шаги программы

1. Запуск программы

Программа запускается в виде пользовательского интерфейса, где задаются пути к директории с изображениями и выходной директории для результатов и статистики. После этого вызывается функция для обработки всех изображений.

1. Обработка изображений

При вызове функции processAllImages программа:

* + Открывает директорию, указанную в inputDirectory.
  + Перебирает все файлы в этой директории и отбирает только те, которые имеют расширение изображений (например, PNG, JPG).
  + Для каждого подходящего изображения запускает функцию processImage, передавая путь к изображению.

1. Чтение и разбиение изображения

При вызове функции processImage:

* + Изображение загружается с помощью OpenCV (cv2.imread).
  + Определяется имя изображения для дальнейшего использования.
  + Изображение разбивается на части с помощью функции splitImage, каждая из которых будет обработана отдельно.

1. Параллельная обработка частей изображения

После разбиения изображения:

* + Для каждой части создаются аргументы, содержащие необходимые данные для обработки (часть изображения, индекс, имя изображения и смещения).
  + Создается пул процессов с помощью библиотеки multiprocessing, который будет обрабатывать части изображения параллельно с помощью функции analyzeImagePart.

1. Анализ частей изображения

В функции analyzeImagePart:

* + Каждая часть изображения преобразуется в градации серого.
  + Применяется гауссово размытие для уменьшения шума.
  + Изображение бинаризуется, чтобы выделить контуры.
  + Контуры анализируются для вычисления характеристик астрофизических объектов, таких как площадь и яркость.

1. Классификация объектов

Для каждого выделенного объекта в функции classifyObject:

* + Объекты классифицируются на основе их площади и яркости, присваивая им категории (звезды, планеты, галактики и т.д.).

1. Сбор результатов

Все результаты анализа собираются в общий список allObjectsData:

* + После завершения обработки всех частей для каждого изображения, данные о найденных объектах собираются в итоговый список.

1. Сохранение результатов

После обработки всех изображений:

* + Результаты сохраняются в виде кусков изображения с отмеченными объектами и в CSV-файл, где указываются все характеристики объектов, такие как имя изображения, координаты, яркость, площадь и тип.

1. Завершение работы программы
   * После завершения всех операций программа выводит сообщение о завершении анализа и сохранении результатов, уведомляя пользователя о готовности данных для дальнейшего использования.

# Описание программы

Программная реализация написана на языке Python 3.12.6 с использованием следующих библиотек: csv, cv2, numpy, tkinter, и multiprocessing. Программа организована в единый модуль multiprocessingSpaseImage.py, который акцентирует внимание на анализе изображений для обнаружения астрофизических объектов. В процессе разработки программы используется 8 функций, каждая из которых имеет четко определенное назначение, а также 1 основной модуль:

Таблица 1. DatasetAnonymization.py

|  |  |  |
| --- | --- | --- |
| Функция | Описание | Возвращаемое значение |
| analyzeImagePart | Анализирует часть изображения: преобразует в градации серого, размывает, бинаризует и находит контуры. Вычисляет характеристики объектов на основе контуров. | objectsData (список объектов) |
| classifyObject | Классифицирует объект на основе его площади и яркости. Определяет тип объекта (звезда, планета, галактика или неизвестный объект). | str |
| processAllImages | Обрабатывает все изображения в указанной директории: считывает изображения, разбивает их на части и передает части для анализа в пул процессов. | None |
| processImage | Читает изображение, разбивает его на части и организует параллельный анализ с использованием многопроцессорности. | allObjectsData (список объектов) |
| splitImage | Делит изображение на части заданного размера для параллельной обработки. | imageParts (список частей) |

# Рекомендации пользователя

Программа позволяет обрабатывать изображения и выявлять астрофизические объекты на основе их характеристик.

Перед началом работы запустите скрипт multiprocessingSpaseImage.py. Выберите папку, содержащую изображения формата PNG, JPG, JPEG, BMP или TIFF для анализа и папку для сохранения результатов. Для начала обработки нажмите на кнопку обработать. Программа будет обрабатывать все изображения в указанной папке, и результаты будут сохранены в CSV-файле под названием astroObjectsStatistics.csv. После запуска программа будет выводить информацию о процессе обработки изображений в текстовом поле.

# Рекомендации программиста

Для успешного запуска программы убедитесь, что у вас установлен Python и необходимые библиотеки, такие как csv, cv2, numpy, tkinter и multiprocessing. Все файлы программы должны находиться в одной директории для корректного выполнения.

Поддерживайте актуальные версии используемых библиотек и Python для обеспечения работоспособности программы на современных системах. Следуйте передовым практикам разработки, уделяя внимание четкому именованию переменных и функций, чтобы улучшить читаемость и понимание кода.

Регулярно проводите тестирование программы на различных входных данных, чтобы убедиться в ее надежности и корректности работы. Рассмотрите возможность добавления логирования для отслеживания процесса выполнения и отладки. Используйте комментарии для пояснения сложных участков кода и логики программы.

# Исходный код программы

<https://github.com/FasterXaos/Functional_programming>

# Контрольный пример

1. Запуск программы

Для запуска программы используйте файл **multiprocessingSpaseImage.py.** После выбора папок ввода и вывода программа запустит многопроцессорную обработку частей изображения и будет переходить к следующему по ее завершению (Рис. 1). Когда программа завершит свое выполнение она запишет результат в файл astroObjectsStatistics.csv (Рис. 3) в выбранной директории (Рис. 2).

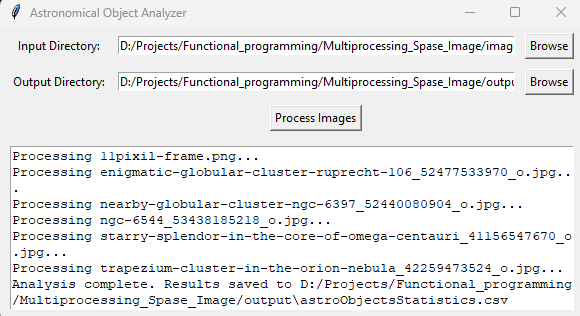


Рис 1. пример вывода программы во время выполнения

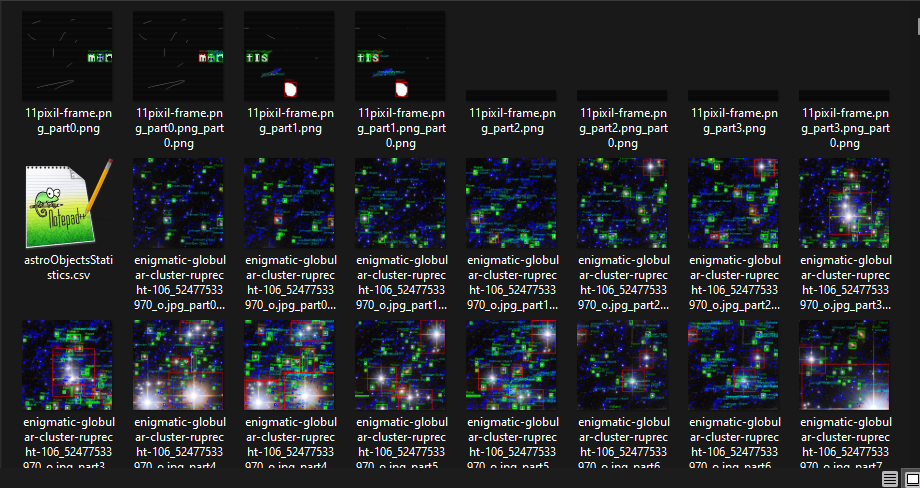


Рис 2. пример вывода программы

2. Просмотр результатов

После завершения выполнения программы результаты можно посмотреть в директории **multiprocessingSpaseImage.py в файле astroObjectsStatistics.csv** (Рис. 3).

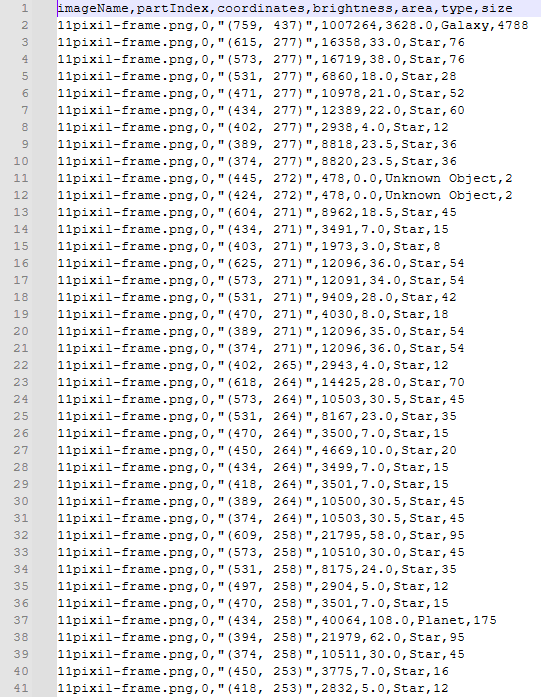


Рис 3. пример итоговых результатов

# Вывод

В рамках данной лабораторной работы была разработана система для анализа астрономических изображений с использованием методов компьютерного зрения и параллельной обработки данных. Основной целью работы было создание эффективного инструмента для выявления астрофизических объектов, таких как звезды, планеты и галактики, на основании анализа изображений.

Были рассмотрены такие характеристики, как площадь контуров, яркость объектов и их размеры, что позволило точно классифицировать объекты. Применялись методы обработки изображений, такие как преобразование в градации серого, размытие и бинаризация, для улучшения качества выявления контуров.

Программа была разработана на языке Python с использованием библиотек OpenCV и NumPy для обработки изображений и библиотек multiprocessing для параллельной обработки данных. Это позволило значительно ускорить анализ изображений, разделив их на части и обрабатывая каждый участок в отдельном процессе.

# Литература

1. csv (Работа с CSV-файлами)

Официальная документация Python: csv  
<https://docs.python.org/3/library/csv.html>

2. cv2 (OpenCV)

Официальная документация OpenCV: <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html>

3. numpy

Официальная документация NumPy  
<https://numpy.org/doc/stable/>

4. tkinter (Графический интерфейс на Python)

Официальная документация Python: tkinter  
<https://docs.python.org/3/library/tkinter.html>

5. multiprocessing (Параллельные вычисления на Python)

Официальная документация Python: multiprocessing  
<https://docs.python.org/3/library/multiprocessing.html>