**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный университет**

**Факультет прикладной математики-процессов управления**

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

**отчет**

**по лабораторной работе №1**

**по дисциплине «Функциональное программирование»**

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

**Вариант – 1**

|  |  |  |
| --- | --- | --- |
| Студент гр. 24Б15-пу |  | Шаймиев К. А. |
| Преподаватель |  | Киямов Ж. У. |

**Санкт-Петербург**

**2025 г**

Оглавление

[Цель работы 3](#_Toc210260005)

[Формализация задачи 3](#_Toc210260006)

[Основные шаги программы 5](#_Toc210260007)

[Блок-схема программы 7](#_Toc210260008)

[Описание программы 8](#_Toc210260009)

[Контрольный пример 12](#_Toc210260010)

[Рекомендации пользователя 14](#_Toc210260011)

[Рекомендации для программиста 14](#_Toc210260012)

[Вывод 15](#_Toc210260013)

[Список использованной литературы 16](#_Toc210260014)

# **Цель работы**

# Целью данной лабораторной работы является разработка программы для анализа космических изображений с использованием параллельных вычислений. Основная задача заключается в эффективной обработке астрономических снимков с космического телескопа, позволяющей автоматически находить и классифицировать астрономические объекты (звезды, галактики, туманности, кометы и др.), а также собирать статистические данные о них.

# **Формализация задачи**

Задача заключается в создании программы для анализа космических снимков с реализацией следующих функциональных требований:

1. **Загрузка и нормализация изображений**: Программа должна считывать изображения формата .tif и нормализировать их для корректной обработки яркости.
2. **Параллельная обработка**: Программа должна разбивать обрабатываемое изображение на блоки, которые должны параллельно обрабатываться в отдельных потоках.
3. **Выделение контуров объектов**: При обработке блока программа должна обводить контур и подписывать класс каждого найденного объекта, основываясь на яркости и площади объекта.
4. **Сохранение результатов**: Программа должна собирать статистику по проделанному анализу и сохранять файл с таблицей статистики в формате .xlsx.
5. **Графический интерфейс**: Необходимо реализовать графический интерфейс для взаимодействия с программой, который будет включать предпросмотр выбранного изображения, выбор количества блоков, просмотр результата и выбор директории для сохранения таблицы результатов.

**Теоретическая часть**

Анализ космических изображений — одна из ключевых задач современной астрофизики. С увеличением объёмов данных, получаемых с космических телескопов, возникает необходимость в автоматизации процесса обработки изображений и ускорении вычислений с помощью параллельных методов. Основные концепции, применяемые при разработке программы для анализа:

1. **Обработка изображений**

Обработка изображений представляет собой последовательность операций, направленных на выделение, анализ и классификацию объектов на цифровом изображении. Наиболее важные этапы включают:

* Предобработку (нормализация, фильтрация шумов, бинаризация);
* Выделение контуров и сегментацию объектов;
* Извлечение признаков (яркость, площадь, форма, координаты);
* Классификацию объектов по физическим характеристикам.

В работе применяется библиотека OpenCV, предоставляющая широкий набор инструментов для анализа изображений, включая пороговую фильтрацию, морфологические операции и поиск контуров.

1. **Бинаризация и выделение объектов**

Бинаризация позволяет отделить объекты от фона на изображении. Это достигается с помощью пороговых методов, где каждый пиксель изображения классифицируется как «объект» или «фон».

Для выделения контуров используется функция cv2.findContours(), которая возвращает границы связных областей. На основе контуров вычисляются площади и яркость объектов.

1. **Классификация астрономических объектов**

Классификация основана на физических характеристиках объектов — яркости и площади. В программе используются эмпирические пороги, позволяющие различать типы небесных тел:

* **Звёзды** — яркие компактные объекты;
* **Галактики** — крупные объекты средней яркости;
* **Туманности** — тусклые, обширные области;
* **Квазары** — яркие точечные источники;
* **Кометы и астероиды** — небольшие и тусклые объекты.

Для каждого найденного контура вычисляются:

* **Средняя и максимальная яркость**;
* **Площадь**;
* **Координаты центра**;
* **Класс объекта.**

1. **Параллельные вычисления**

Обработка изображений требует значительных вычислительных ресурсов. Для повышения производительности применяется многопоточность.  
Каждое изображение разбивается на блоки, которые анализируются параллельно с помощью потоков (threading.Thread). Это позволяет использовать ресурсы процессора более эффективно и сокращает общее время обработки.

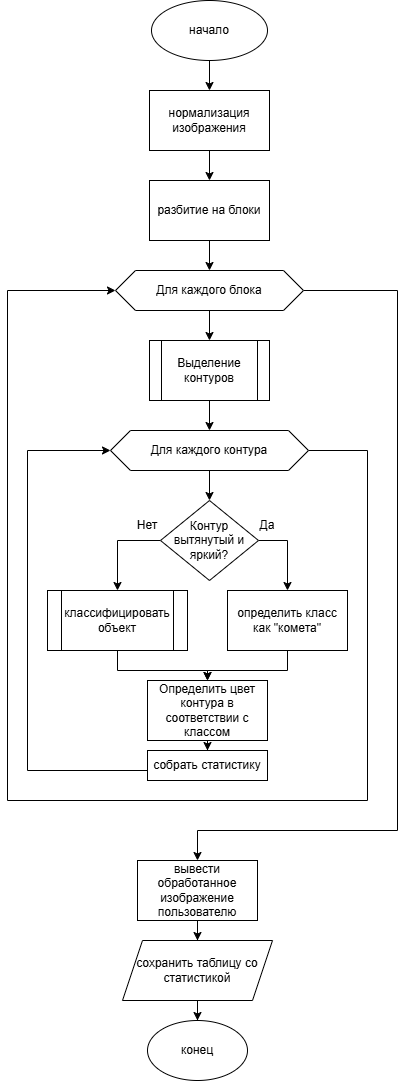
1. **Сбор статистики**

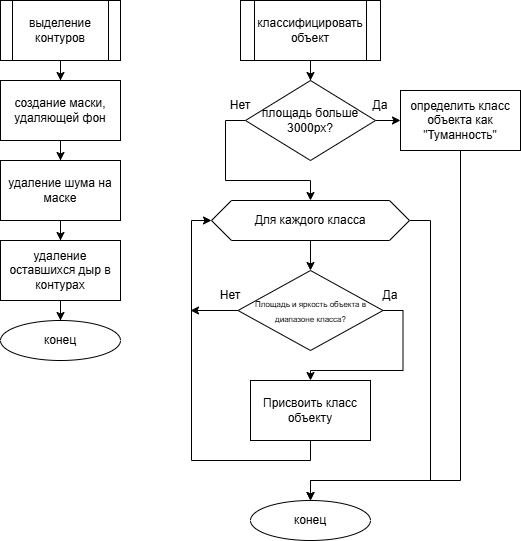
Результаты анализа сохраняются в виде Excel-таблицы, где для каждого объекта указаны его характеристики.  
Одновременно формируется аннотированное изображение, на котором все найденные объекты выделены цветом в зависимости от класса.

# **Основные шаги программы**

1. **Считывание изображения**. При запуске первым этапом является выбор и считывание .tif изображения. Для окна предварительного просмотра изображение конвертируется в формат .jpeg
2. **Нормализация яркости**. Для будущей обработки изображение нормализует яркость в диапазон [0,1] – это необходимо для классификации объектов по яркости и задания диапазона яркости для определяемых классов объектов.
3. **Разделение изображения на блоки**. Перед началом обработки изображение разбивается на блоки, количество которых указывает пользователь. Каждый блок выполняется параллельно в своем потоке.
4. **Поиск контуров объектов и расчет характеристик**. В каждом блоке программа выделяет контуры объектов и рассчитывает площадь, среднюю и максимальную яркость для каждого объекта.
5. **Классификация**. На основе рассчитанных характеристик каждый объект причисляется к тому или иному заранее определенному классу.

# **Блок-схема программы**





# **Описание программы**

Программа написана на языке Python и реализует анализ снимков космического пространства. Программа включает в себя пользовательский интерфейс, реализацию основного алгоритма анализа и составление статистической таблицы.

**Использованные библиотеки**

**os**: Библиотека для работы с файловой системой и выполнения операций над каталогами и файлами.

**threading**: Модуль для реализации многопоточности. Позволяет выполнять несколько операций параллельно для ускорения обработки данных или изображений.

**numpy**: Предоставляет инструменты для высокопроизводительных вычислений с многомерными массивами и матрицами, а также широкий набор математических функций.

**sys**: Применяется для управления процессом завершения программы.

**Pillow (PIL)**: Библиотека для работы с изображениями — открытия, преобразования, фильтрации и сохранения файлов различных форматов.

**cv2 (OpenCV)**: Мощная библиотека компьютерного зрения. Используется для обработки изображений, выделения контуров, фильтрации, преобразований и анализа.

**PyQt5**: Библиотека для создания графических интерфейсов (GUI). Позволяет объединять визуальные компоненты с логикой программы.

**Основные функции программы**

Таблица 1.1. Описание методов основной программы

|  |  |  |
| --- | --- | --- |
| **Название функции** | **Входные переменные** | **описание** |
| **normalize\_float01** | **arr** | |  | | --- | | Нормализует изображение в диапазон [0.0, 1.0]. Убирает фон (медианное значение), ограничивает верхнюю границу по 99.5 перцентилю. |  |  | | --- | |  | |
| **classify\_astronomical\_object** | **brightness, area** | |  | | --- | | Классифицирует объект (звезда, галактика, туманность и т.п.) по средней яркости и площади. Возвращает название класса и цвет для визуализации. |  |  | | --- | |  | |
| **process\_block** | **gray\_block, x\_off, y\_off, results, lock, block\_id, output\_dir, row\_idx, col\_idx, stats\_rows** | |  |  |  | | --- | --- | --- | | |  | | --- | | Выполняет обработку одного блока изображения. Находит контуры, выделяет кометы, звёзды и туманности, классифицирует их, сохраняет визуализацию и статистику. |  |  | | --- | |  | |  |  | | --- | |  | |
| |  | | --- | | **tiff\_to\_jpeg** |  |  | | --- | |  | | **input\_path, output\_path** | Преобразует изображение TIFF в формат JPEG с нормализацией яркости. Используется для предпросмотра исходных данных. |
| |  | | --- | | **find\_and\_draw\_contours\_multithread** |  |  | | --- | |  | | **gray\_float01, base\_color\_image, num\_rows, num\_cols, output\_dir** | |  | | --- | | Разбивает изображение на блоки, обрабатывает их в отдельных потоках, собирает результаты и сохраняет статистику в Excel-файл. |  |  | | --- | |  | |

**Основные переменные**

Таблица 2.1. Описание переменных тела демона

|  |  |
| --- | --- |
| **Имя переменной** | **Функция** |
| **ASTRO\_CLASSES** | |  | | --- | | Список правил классификации объектов по яркости и площади, с указанием названия и цвета класса. |  |  | | --- | |  | |
| **MIN\_CONTOUR\_AREA** | |  | | --- | | Минимально допустимая площадь контура для учёта объекта. |  |  | | --- | |  | |
| **gray\_block, gray\_float01** | Изображение (или его блок) в градациях серого, нормализованное в диапазоне [0,1]. |
| **results** | |  | | --- | | Список найденных контуров с параметрами (контур, цвет, тип объекта, площадь, яркость). |  |  | | --- | |  | |
| **stats\_rows** | |  | | --- | | Список статистических данных по объектам (строка, столбец, яркость, координаты и тип). |  |  | | --- | |  | |
| **output\_dir** | |  | | --- | | Папка для сохранения результатов — блоков изображений и таблицы Excel. |  |  | | --- | |  | |
| **tiff\_image, tiff\_path** | Исходное изображение TIFF и путь к нему. |

# **Контрольный пример**

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1. Начальное окно программы

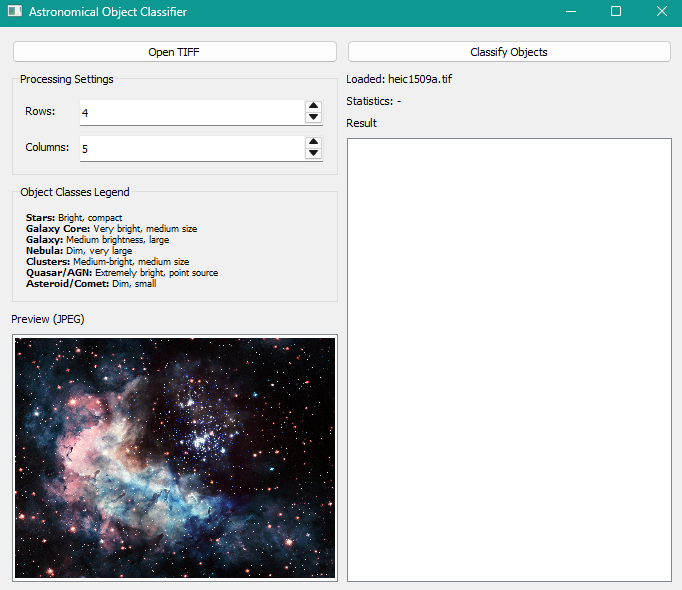


Рисунок 2. Настройка параметров и предпросмотр выбранного изображения

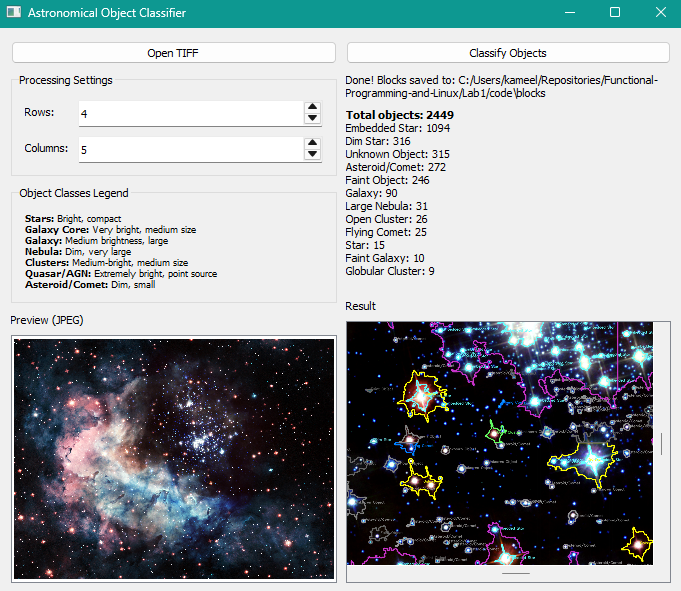


Рисунок 3. Просмотр обработанного файла и подсчета объектов

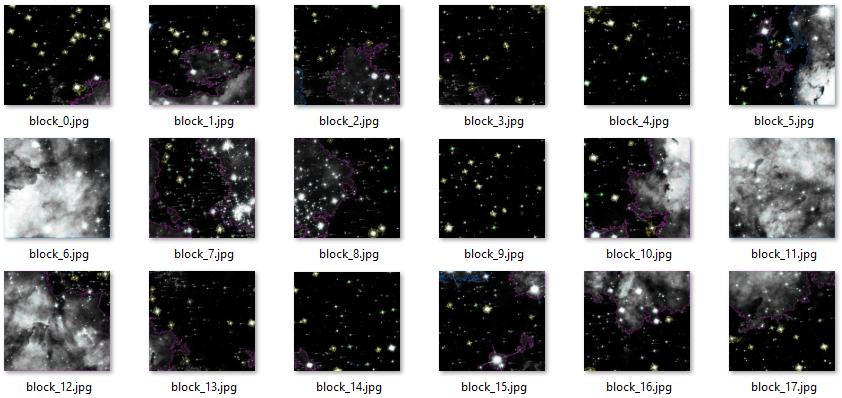


Рисунок 4. Отдельно сохранены блоки изображения

Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5. Таблица со статистикой

# **Рекомендации пользователя**

* + 1. **Запуск программы**
* Для запуска используйте командную строку или терминал. Перейдите в каталог с кодом и выполните:  
  python main\_ui.py
* Программа откроет графический интерфейс для обработки и анализа изображений.
  + 1. **Загрузка изображения**
* Нажмите Open TIFF и выберите файл изображения в формате .tif или .tiff.
* Программа автоматически создаст JPEG-превью и подготовит изображение к классификации.
* Для корректной работы убедитесь, что у вас есть права на запись в каталог с изображением.
  1. **Обработка данных**
* В разделе **Processing Settings** задайте количество строк и столбцов для разбиения изображения
* Нажмите **Classify Objects** для начала анализа
* Программа выделит объекты (звёзды, галактики, туманности, кометы и др.), классифицирует их и создаст результат с подписями.
  1. **Результаты анализа**
* Итоговое изображение сохраняется в файл classified\_result.jpg, а промежуточные блоки — в папку blocks/.
* Таблица со статистикой (object\_stats.xlsx) сохраняется рядом с исходным изображением и может быть открыта в Excel или Google Sheets для детального анализа.

# **Рекомендации для программиста**

* **Настройка окружения:**  
  Рекомендуется использовать Python версии **3.9 и выше**, так как более старые версии могут не поддерживать некоторые функции PyQt5 и OpenCV. Убедитесь, что установлены все зависимости:  
  pip install pyqt5 opencv-python pillow numpy openpyxl
  1. **Архитектура и структура программы:**  
     Программа построена на модульной архитектуре: логика обработки изображений отделена от пользовательского интерфейса.  
     Основной код интерфейса реализован в файле **main\_ui.py**, где создаётся главное окно приложения, обработчики событий и визуализация результатов.
  2. **Модификация программы:**  
     Для добавления новых классов астрономических объектов измените список **ASTRO\_CLASSES**, указав диапазоны яркости, площади и цвет отображения.  
     Можно внедрить более сложные алгоритмы классификации (например, с использованием нейронных сетей или обученных моделей), заменив вызов функции classify\_astronomical\_object.  
     При необходимости реализуйте настройку параметров через внешний .conf-файл, чтобы пользователь мог изменять пороги яркости и размеры блоков без правки кода.
  3. **Оптимизация и обработка больших данных:**  
     Для ускорения работы с крупными изображениями используется многопоточность (модуль threading). При необходимости можно перейти на многопроцессорный режим (multiprocessing) для повышения производительности.  
     При работе с TIFF-файлами большого размера рекомендуется использовать SSD и увеличить объём оперативной памяти.  
     Для анализа больших каталогов изображений можно создать цикл автоматической обработки с сохранением результатов в отдельные подпапки.

# **Вывод**

В ходе разработки была создана программа для автоматической обработки и классификации астрономических изображений с использованием библиотек **PyQt5**, **OpenCV**, **NumPy** и **Pillow**.  
Программа реализует графический интерфейс, позволяющий пользователю загружать изображения в формате **TIFF**, выполнять их нормализацию, разбиение на блоки и многопоточную обработку для выделения астрономических объектов.

В результате работы программы осуществляется выделение и классификация объектов по яркости и площади (звёзды, галактики, туманности, кометы и др.), а также сохранение статистики в формате **xlsx** и визуализация результатов на изображении.  
Использование многопоточности позволило существенно повысить производительность и обеспечить эффективную обработку больших изображений.

Разработанное приложение может быть расширено — возможно добавление новых классов объектов, улучшение алгоритмов анализа или внедрение машинного обучения для более точной классификации.  
Программа демонстрирует практическое применение методов компьютерного зрения и обработки изображений в астрономии и может использоваться как инструмент для исследовательских и учебных целей.

# Список использованной литературы

1. <https://docs.opencv.org/>
2. <https://habr.com/ru/companies/otus/articles/849136>
3. <https://python-scripts.com/pyqt5>