Санкт-Петербургский государственный университет

Факультет прикладной математики – процессов управления

**Отчет №5**

по дисциплине «Функциональное программирование»

**Параллельные вычисления**

**Вариант 1**

Автор работы: Дацык Р.В.

Группа: 22.Б15-пу

Преподаватель: Киямов Ж. У.

Санкт-Петербург, 2023

**Оглавление**

[**1.** **Цель работы** 3](#_Toc153300415)

[**2.** **Задачи** 3](#_Toc153300416)

[**3.** **Теоретическая часть** 4](#_Toc153300417)

[**4.** **Алгоритм метода** 4](#_Toc153300418)

[**6.** **Рекомендации пользователя** 8](#_Toc153300419)

[**7.** **Рекомендации программиста** 8](#_Toc153300420)

[**8.** **Контрольный пример** 8](#_Toc153300421)

[**9.** **Вывод** 9](#_Toc153300422)

[**10.** **Список использованной литературы** 10](#_Toc153300423)

# **Цель работы**

Создание программы для анализа изображений с использованием библиотек OpenCV и Tkinter с целью выявления объектов на изображении, их анализа и сохранения результатов в CSV-файле.

# **Задачи**

Для достижения цели поставлены следующие задачи:

1. **Разработка программы:**
   * Создание программы для анализа данных с космического телескопа.
   * Учет больших объемов данных и сложных астрофизических вычислений.
2. **Обработка данных:**
   * Разработка механизма обработки сотен тысяч изображений.
   * Эффективная работа с матрицами пикселей на изображениях.
3. **Параллельные вычисления:**
   * Реализация параллельной обработки изображений для ускорения анализа.
   * Оптимизация работы на нескольких ядрах процессора.
4. **Астрофизический анализ:**
   * Выделение и измерение характеристик астрофизических объектов.
   * Классификация объектов на основе сложных астрофизических вычислений.
5. **Сбор статистики:**
   * Разработка механизма сбора статистических данных для каждого объекта.
   * Сохранение статистики в удобном формате для последующего использования.
6. **Управление ресурсами:**
   * Обеспечение эффективного управления ресурсами при параллельной обработке.
   * Предотвращение гонок данных и конфликтов.
7. **Визуализация исследователям:**
   * Предоставление детальной статистики исследователям.
   * Возможность проведения дополнительного анализа и визуализации результатов.

# **Теоретическая часть**

В данной задаче решается проблема параллельной обработки изображений с использованием многозадачности. Ключевые концепции включают в себя многозадачность, обработку изображений и эффективное управление ресурсами. Для параллельной обработки данных используются потоки или процессы.

# **Алгоритм метода**

1. **Подготовка данных:**
   * Загрузка изображений космических объектов, представленных в виде матриц пикселей.
   * Инициализация параметров, таких как минимальная и максимальная площадь объекта.
2. **Параллельная обработка изображений:**
   * Разбиение изображений на части для параллельной обработки.
   * Создание и запуск нескольких потоков для обработки каждой части.
3. **Обработка каждой части:**
   * Применение преобразований к изображению: перевод в оттенки серого, размытие, выделение границ методом Canny.
   * Поиск контуров на изображении с использованием алгоритма findContours.
   * Для каждого контура:
     1. Вычисление описывающего круга (включая его радиус и центр).
     2. Выделение области, соответствующей объекту, на основе описывающего круга.
     3. Вычисление площади объекта и соотношения сторон его ограничивающего прямоугольника.
     4. Проверка условий на минимальную и максимальную площадь, а также соотношение сторон.
     5. Расчет среднего цвета и яркости внутри объекта.
4. **Сбор статистики:**
   * Запись результатов анализа для каждого объекта в файл CSV.
   * Визуализация результатов, если параметр отрисовки объектов установлен в True.
5. **Управление ресурсами:**
   * Использование очереди для распределения задач между потоками.
   * Параллельная обработка с учетом безопасности взаимодействия между потоками.
6. **Визуализация результатов:**
   * Сохранение анализированных изображений с выделенными объектами.
   * Создание файла CSV с детальной статистикой для каждого объекта.
7. **Окончание обработки:**
   * Завершение работы потоков и завершение программы.

Этот алгоритм позволяет проводить параллельный астрофизический анализ большого объема данных, выделять интересные объекты и собирать подробную статистику о каждом из них для последующего исследования.

1. **Описание программы**

Программа предназначена для анализа космических данных, представленных в виде изображений, полученных с космического телескопа. Она реализует параллельную обработку большого объема данных, выделяет астрофизические объекты на изображениях и собирает статистику по каждому объекту. Исходные изображения разбиваются на части, каждая из которых обрабатывается параллельно для ускорения процесса. Результаты анализа сохраняются в файле CSV, а также могут быть визуализированы с выделенными объектами.

Таблица 5.1. Описание методов

|  |  |
| --- | --- |
| Название функции | Назначение функции |
| **analyze\_and\_draw\_objects** | Осуществляет анализ и выделение объектов на изображении.  Принимает на вход изображение, путь для сохранения результатов, имя файла, параметры отрисовки и диапазоны площадей объектов.  Возвращает обработанное изображение и флаг, указывающий, были ли найдены объекты. |
| **process\_part** | Обрабатывает отдельную часть изображения, вызывая функцию analyze\_and\_draw\_objects.  Принимает изображение, путь для сохранения результатов и параметры отрисовки. |
| **process\_part\_threaded** | Аналогична функции process\_part, но предназначена для выполнения в отдельном потоке. |
| **process\_image\_worker** | Рабочий поток для параллельной обработки изображений.  Получает задачу из очереди, вызывает функцию process\_part\_threaded, помечает задачу как выполненную. |
| **split\_and\_analyze\_image** | Рабочий поток для параллельной обработки изображений.  Получает задачу из очереди, вызывает функцию process\_part\_threaded, помечает задачу как выполненную. |
| **get\_part\_size** | Открывает окно с вводом размера части изображения через графический интерфейс. |
| **browse\_image** | Открывает диалоговое окно для выбора изображения через графический интерфейс. |
| **image\_queue** | Очередь для хранения задач обработки изображений в нескольких потоках. |
| **main Tkinter GUI** | Создает графический интерфейс с кнопкой для выбора изображения, отображением выбранного файла и кнопкой для запуска анализа. |
| **Global Variables** | Глобальные переменные для хранения пути к изображению и папке для сохранения результатов. |

# **Рекомендации пользователя**

Для запуска программы пользователю нужно нажать кнопку «Browse image» и выбрать в диалоговом окне нужный файл. Нажать на кнопку «Analyse», а затем выбрать размер фрагментов, на которые будет делиться основное изображение. Выходные изображения и csv-файл будут сохранены в директории выбранной фотографии.

# **Рекомендации программиста**

Для запуска программы необходима 64-битная операционная система Windows, Linux или macOS. Для работы с кодом необходима среда разработки, совместимая с python 3.10. Также требуется установить следующие библиотеки: os, tkinter, threading, PIL, cv2, numpy, queue.

Исходный код программы доступен по ссылке:

<https://github.com/CapTopGrade/Functional_Programming/blob/main/%D0%A4%D0%9F/%D0%A4%D0%9F%20%D0%BF%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0/%D0%A4%D0%9F%20%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85/space%20image.py>

# **Контрольный пример**

В данном разделе представлен контрольный пример, демонстрирующий работу программы.

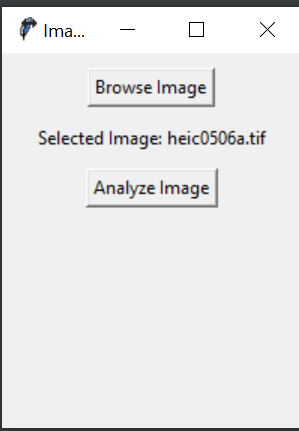


Рисунок 8.1, интерфейс программы

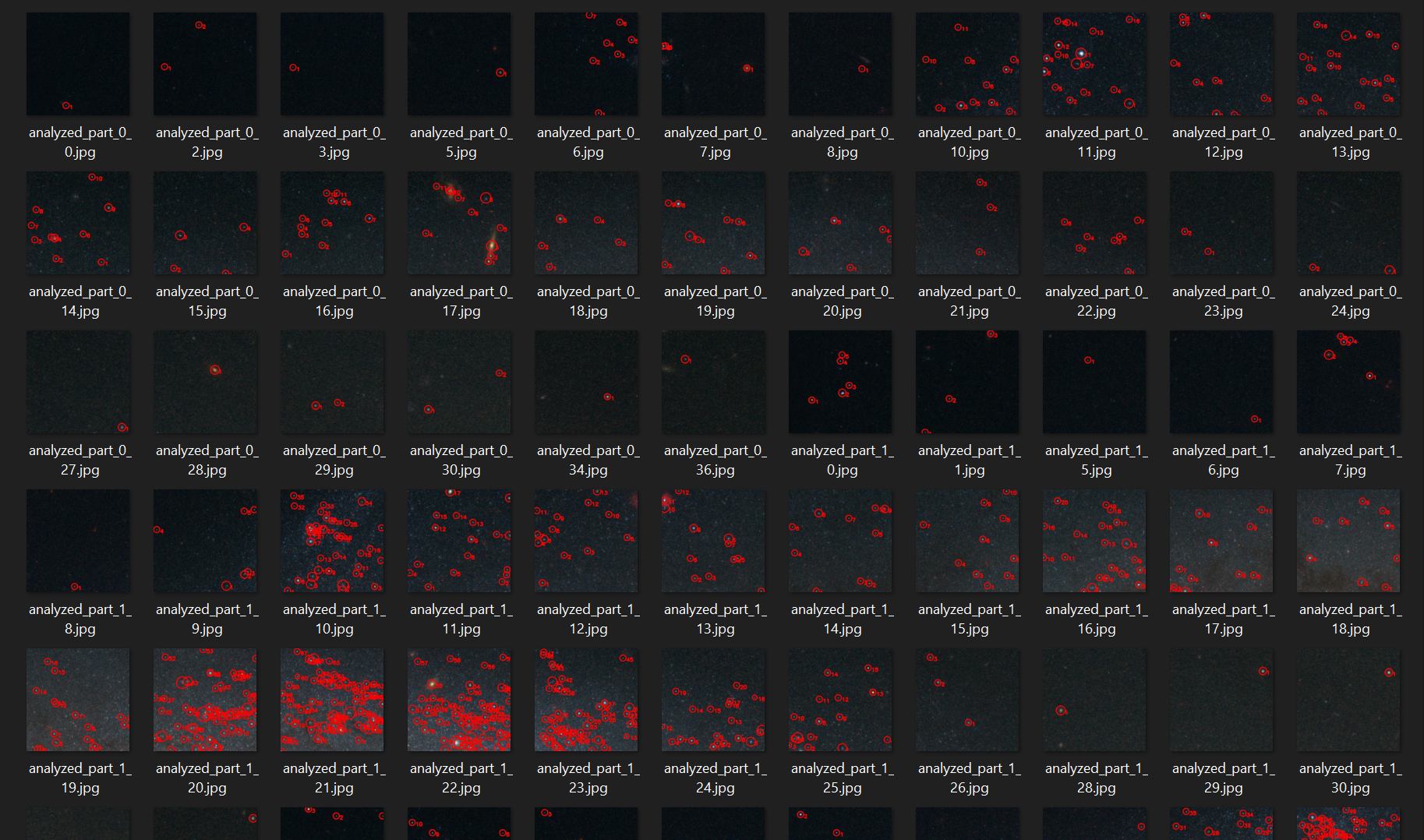


Рисунок 8.2, Пример результата работы программы

# **Вывод**

Разработана эффективная программа для параллельной обработки изображений с использованием многозадачности. Учтены особенности работы с разными фильтрами, обработка ошибок и распределение ресурсов для оптимальной производительности.

# **Список использованной литературы**

1. Документация OpenCV: <https://docs.opencv.org/>
2. Документация библиотеки NumPy: <https://numpy.org/doc/>
3. Документация Tkinter: <https://docs.python.org/3/library/tkinter.html>
4. Документация Python о многозадачности: <https://docs.python.org/3/library/threading.html>