МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Корпоративные информационные системы
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного модуля для сравнения алгоритмов сегментации изображений»

Студент А.И. Синицын
(И.О. Фамилия) (личная подпись)

Руководитель к.т.н., доцент, О.В. Аникина
(ученая степень, звание, И.О. Фамилия)

Консультант А.В. Москалюк
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы – «Разработка программного модуля для сравнения алгоритмов сегментации изображений».

Как показывает практика, наилучших результатов при решении конкретной задачи анализа изображения удается достичь при правильном выборе алгоритма его сегментации.

Объектом исследования бакалаврской работы является программный комплекс анализа изображений КИС.

Предметом исследования бакалаврской работы является программный модуль для сравнения алгоритмов сегментации изображений.

Цель выпускной квалификационной работы – разработка программного модуля для сравнения алгоритмов сегментации изображений, обеспечивающего повышение эффективности принятых управленческих решений.

На языке программирования Python реализован программный модуль для сравнения алгоритмов сегментации изображений. Функциональное тестирование подтвердило работоспособность модуля.

Результаты бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы для разработчиков программ, в которых для принятия управленческих решений используется анализ графической информации.

Выпускная квалификационная работа состоит из 40 страниц текста, 25 рисунков, 2 таблиц и 21 источника.

Abstract

The topic of the given graduation work is Development of a software module for comparing image segmentation algorithms.

As practice shows, the best results in solving a specific problem of image analysis can be achieved with the correct choice of the algorithm for its segmentation.

The objects of study of the graduation work is a software complex for the analysis of images of the Corporate Information System (CIS).

The subject of study of the graduation work is a software module for comparing image segmentation algorithms.

The aim of the graduation work is the development of a software module for comparing image segmentation algorithms, which improves the efficiency of management decisions.

Research methods: methods and technologies for designing information systems, image segmentation algorithms.

A software module for comparing image segmentation algorithms is implemented using Python programming language. Functional testing has confirmed the functionality of this module.

The results of the graduation work are of scientific and practical interest and can be recommended for developers of programs using the analysis of graphic information for management decisions making.

The graduation work consists of an explanatory note on 40 pages including 25 figures, 2 tables, the list of 21 references.

Оглавление

Введение

В настоящее время для поддержки принятия решения в различных областях управления социальными и экономическими системами используются различные виды аналитической информации, в том числе графическая.

Как показывает практика, для анализа изображений широко применяется метод их сегментация. Сегментация изображения – это процесс поиска и формирования групп пикселей, которые формируются с учетом контекста исходного изображения. Так, в медицине сегментация используется для выделения различных патологий, определения объемов тканей и при решении задач хирургии с использованием компьютера. Сегментация также применяется при выделении объектов на спутниковых снимках, при распознавании лиц в системах контроля и управления дорожным движением и т.д.

Как показывает практика, наилучших результатов при решении конкретной задачи анализа изображения удается достичь при правильном выборе алгоритма его сегментации. Это существенно повышает эффективность принятого на основе результатов анализа управленческого решения.

Следует также отметить, что правильность такого выбора обусловлена применением соответствующего программного обеспечения, которое должно входить в состав подсистемы анализа информации корпоративной информационной системы предприятия (КИС) в виде программного отдельного модуля.

Разработка такого модуля представляет актуальность и научно-практический интерес.

Объектом исследования бакалаврской работы является программный комплекс анализа изображений КИС.

Предметом исследования бакалаврской работы является программный модуль для сравнения алгоритмов сегментации изображений.

Цель выпускной квалификационной работы – разработка программного модуля для сравнения алгоритмов сегментации изображений, обеспечивающего повышение эффективности принятых управленческих решений.

Для достижения данной цели необходимо выполнить следующие задачи:

– произвести анализ предметной области автоматизации;

– выбрать технологию и спроектировать программный модуль для сравнения алгоритмов сегментации изображений;

– реализовать и протестировать программный модуль для сравнения алгоритмов сегментации изображений.

Методы исследования – методы и технологии проектирования информационных систем, алгоритмы сегментации изображений.

Практическая значимость бакалаврской работы заключается в разработке программного модуля, обеспечивающего правильность выбора алгоритма сегментации изображений и повышения эффективности принятых управленческих решений.

Данная работа состоит из введения, трех глав, заключения, списка используемой литературы и приложений.

Первая глава посвящена анализу предметной области автоматизации.

Вторая глава посвящена выбору технологии и проектированию программного модуля для сравнения алгоритмов сегментации изображений.

В третьей главе описан процесс реализации и тестирования программного модуля для сравнения алгоритмов сегментации изображений.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Приложения содержат фрагменты программного кода приложения.

Бакалаврская работа состоит из 40 страниц текста с приложением, 23 рисунков, 2 таблиц и 21 источника.

Глава 1 Анализ предметной области автоматизации

Для поддержки принятия управленческих решений по результатам анализа графической информации на предприятии используется система, функциональная архитектура которой представлена на рисунке 1.

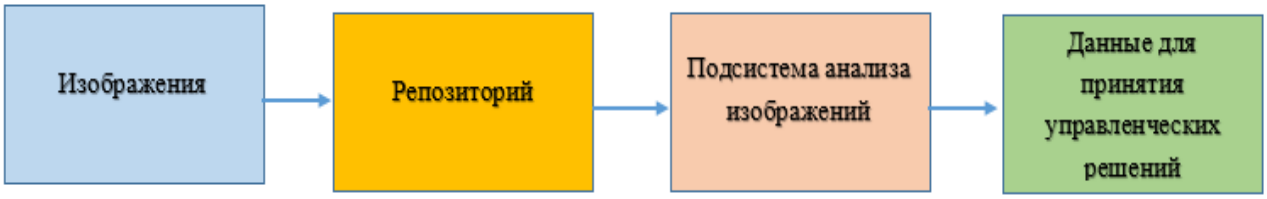

Рисунок 1 – Функциональная архитектура система поддержки управленческих решений

Рассмотрим существующий бизнес-процесс анализа изображений.

1.1 Моделирование бизнес-процесса анализа изображений

Бизнес-процесс анализа изображений относится к вспомогательным бизнес-процессам поддержки принятия управленческих решений на предприятии.

Существующий бизнес-процесс анализа изображений организован следующим образом:

- Бизнес-аналитик загружает из репозитория изображение в подсистему анализа изображений;
- Подсистема анализа изображений анализирует изображения с помощью встроенного алгоритма сегментации и передает результаты анализа бизнес-аналитику для формирования аналитического отчета.

Для реинжиниринга бизнес-процесса необходимо выполнить его моделирование [6].

Для моделирования бизнес-процесса используем нотацию BPMN и

бесплатный онлайн-сервис BPMN Studio [12].

На рисунке 2 изображена BPMN-диаграмма бизнес-процесса анализа изображений «КАК ЕСТЬ».

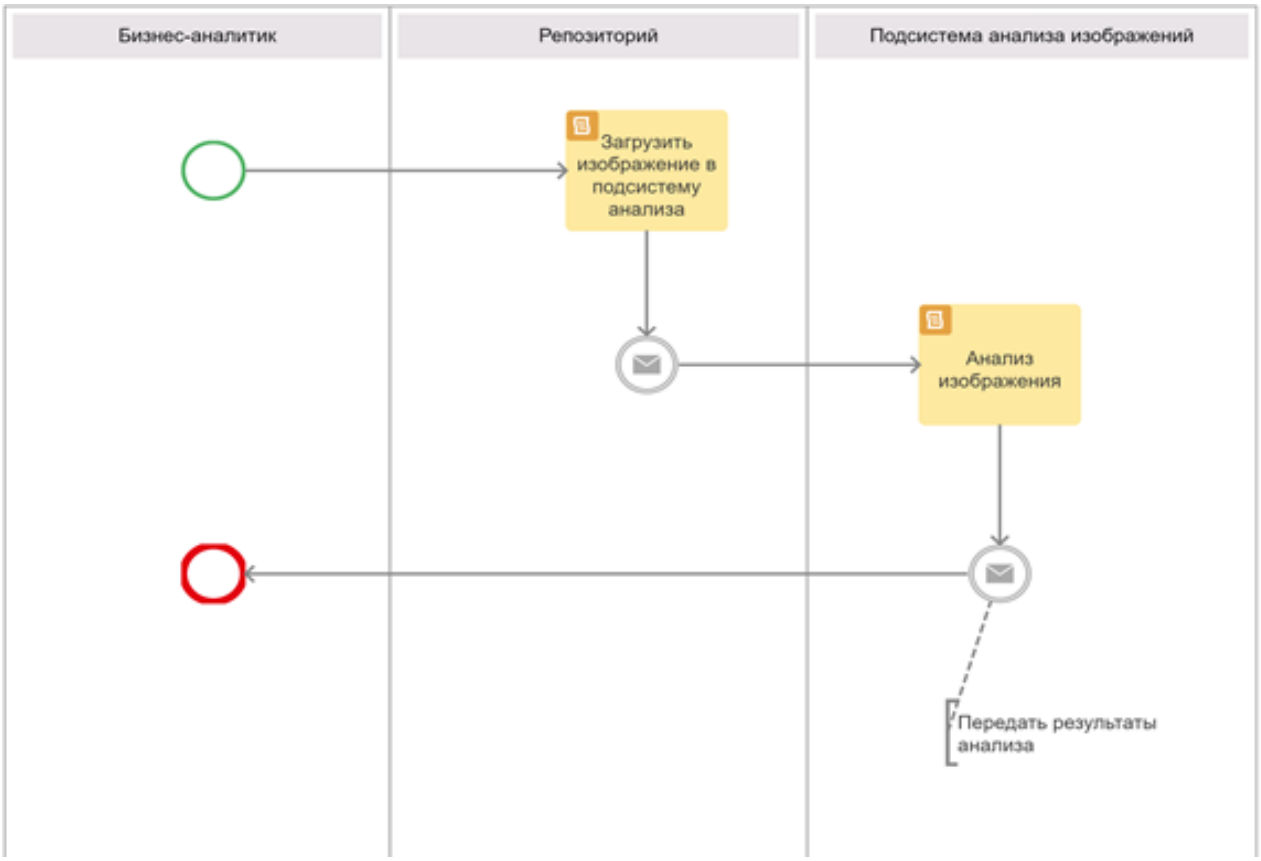

Рисунок 2 – BPMN-диаграмма бизнес-процесса анализа изображений «КАК ЕСТЬ»

Анализ модели бизнес-процесса «КАК ЕСТЬ» позволил выявить следующий недостаток: в существующей подсистеме анализа изображений отсутствует функция выбора оптимального алгоритма сегментации, что не обеспечивает необходимую точность результатов анализа, а, следовательно, эффективность принятых управленческих решений.

Для улучшения бизнес-процесса необходимо разработать и внедрить программный модуль, который позволяет выбрать для сегментации конкретного изображения оптимальный алгоритм.

На рисунке 3 изображена BPMN-диаграмма бизнес-процесса анализа

изображений «КАК ДОЛЖНО БЫТЬ».

Данная модель изображает автоматизированный процесс анализа изображений.

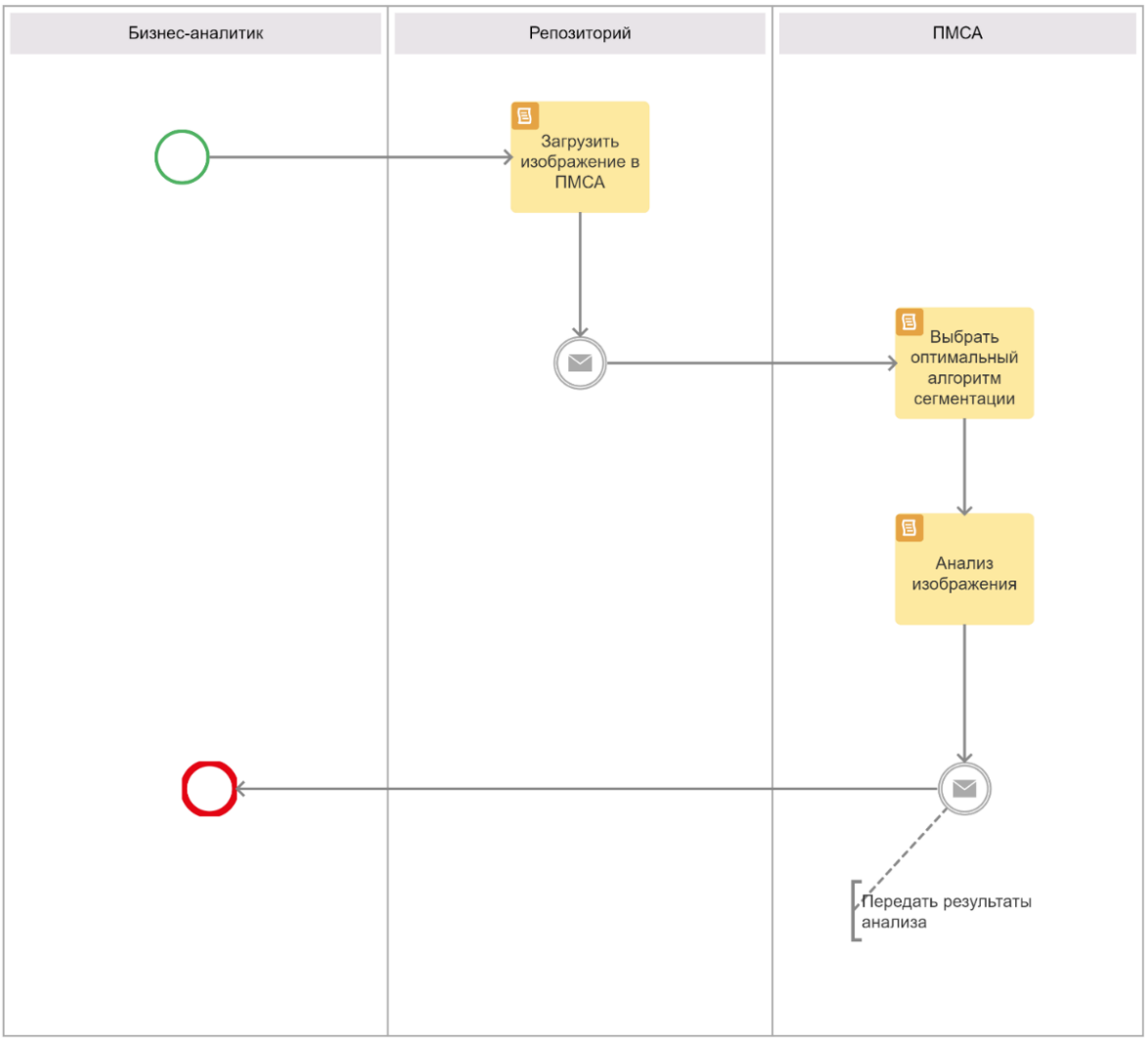

Рисунок 3 – BPMN-диаграмма бизнес-процесса анализа изображений «КАК ДОЛЖНО БЫТЬ»

В модели бизнес-процесса подсистема анализа изображений заменена на программный модуль сравнения алгоритмов (ПМСА) сегментации изображений.

Бизнес-модель «КАК ДОЛЖНО БЫТЬ» является концептуальной моделью ПМСА.

1.2 Разработка требований к программному модулю сегментации изображений

Для разработки требований к ПМСА используем методологию FURPS+ [8].

FURPS+ - это акроним, описывающий расширенную модель для классификации атрибутов качества программного обеспечения.

FURPS+ довольно широко используется в индустрии программного обеспечения.

Опишем основные требования к ПМСА в форме таблицы 1.

Таблица 1 – Требования к ПМСА сегментации изображений

№	Требование	Статус	Полезность	Риск	Стабильность
Functionality — Функциональные требования					
1.	Реализация алгоритмов сегментации изображений: Чана-Везе, ACWE, GAC и SLIC	Одобренное	Критическая	Средний	Низкая
2.	Анализ изображений по методу сегментации	Одобренное	Критическая	Средний	Низкая
Usability— Требования к удобству использования					
3.	Интуитивно понятный интерфейс	Одобренное	Критическое	Средний	Низкая
4.	Отсутствие функциональной избыточности	Одобренное	Критическое	Средний	Низкая
Reliability— Требования к надежности					
5.	Допустимая частота/периодичность сбоев: 1 раз в 300 часов	Одобренное	Важная	Средний	Средняя
6.	Среднее время сбоев: 1 раб. день	Одобренное	Важная	Средний	Средняя
7.	Возможность восстановления системы после сбоев: 1 раб. день	Одобренное	Важная	Средний	Средняя
8.	Режим работы: 7/24/365	Одобренное	Важная	Средний	Средняя

Продолжение таблицы 1

Performance — Требования к производительности					
9.	Допустимое количество одновременно работающих пользователей: 2	Предложенное	Важная	Средний	Средняя
10.	Время реакции на возникновение аварийной ситуации: 1 мин.	Предложенное	Важная	Средний	Средняя
Supportability — Требования к поддержке					
11.	Время устранения критических проблем: в течение рабочего дня	Предложенное	Важная	Средний	Средняя
Проектные ограничения					
12.	Низкая стоимость владения	Предложенное	Критическое	Средний	Низкая
13.	Использование интегральной среде разработки	Предложенное	Критическое	Средний	Низкая

Разработанный набор требований является основой для реализации ПМСА.

Рассмотрим алгоритмы, которые должен сравнивать ПМСА.

Проанализируем принципы действия данных алгоритмов и сформируем принципы их сравнения.

1.2.1 Алгоритм Чана-Везе

Модель Чан-Весе для активных контуров - мощный и гибкий метод, который может сегментировать многие типы изображений, в том числе такие, которые довольно сложно сегментировать с помощью «классической» сегментации, т. е. с использованием пороговых значений или методов на основе градиента [13].

Эта модель основана на функционале Мамфорда-Шаха для сегментации и широко используется в области медицинской визуализации, особенно для сегментации мозга, сердца и трахеи.

Блок-схема алгоритма Чана-Везе представлена на рисунке 4.

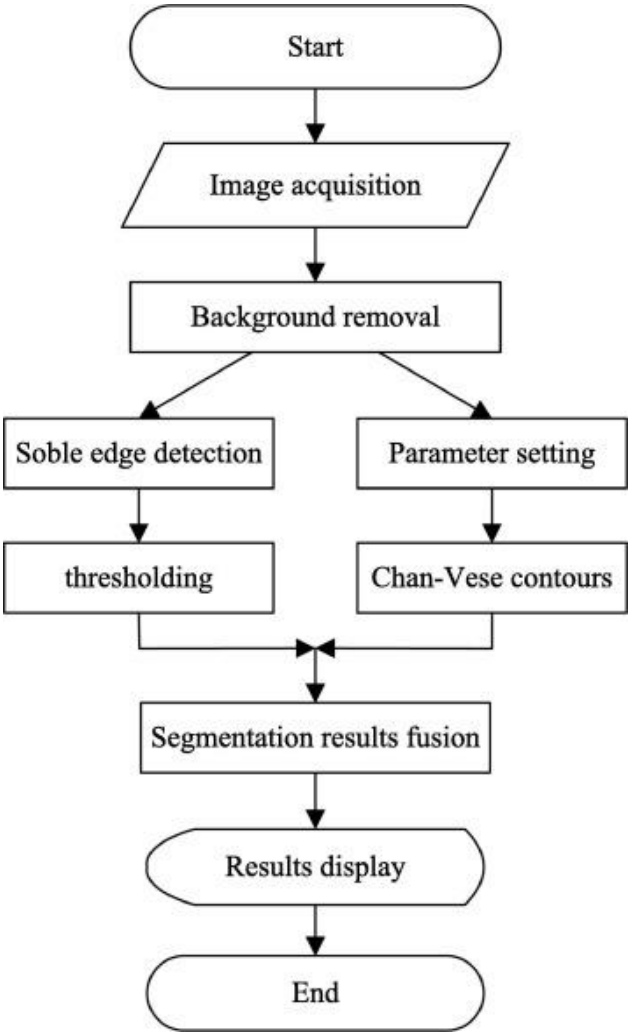

Рисунок 4 – Блок-схема алгоритма Чана-Везе

Модель основана на задаче минимизации энергии, которую можно переформулировать в формулировке набора уровней, что приведет к более простому способу решения проблемы.

1.2.2 Алгоритмы морфологических активных контуров

Метод активных контуров, также известный как метод змей, основан на изменении формы контура вокруг объекта под действием внутренних и внешних сил. Внешние и внутренние силы устроены таким образом, что под их

воздействием контур сожмётся до границ объекта.

«Внутренние силы придают контуру жёсткость, ограничивая таким образом его поведение и форму. К преимуществам метода по сравнению с другими подходами относится способность адаптироваться к объектам различной формы и находить состояния с минимальной энергией, в которых внутренние и внешние силы уравновешены.

Метод активных контуров может использоваться для отслеживания объектов как в пространстве, так и во времени» [16].

Морфологические змеи - это семейство связанных методов для управляемой изображениями эволюции кривых и поверхностей, представленных как набор уровня функции вложения. Они применяются в нескольких областях компьютерного зрения, таких как отслеживание и сегментация изображений.

Для этих методов разработаны два алгоритма: алгоритм Geodesic Active Contours (GAC) - геодезические активные контуры и алгоритм Morphological Active Contours Without Edges (ACWE) - морфологические активные контуры без краев.

Методы работают путем решения уравнений в частных производных для функции вложения, которая имеет контур в качестве нулевого уровня.

Для метода GAC используется уравнение вида:

$$\frac{\partial u}{\partial t} = g(I)\,|\nabla \mathrm{u}|\,\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) + g(\mathrm{I})\,|\nabla \mathrm{u}|\,v + \nabla g(I)\,|\nabla \mathrm{u}| \qquad (1)$$

Для метода ACWE используется уравнение вида:

$$\frac{\partial u}{\partial t} = |\nabla \mathrm{u}|\left(\mu\,\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) - v - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2\right) \qquad (2)$$

Морфологические операторы определяются как операторы трансляции и инвариантные по контрасту операторы.

Морфологический оператор преобразует входную функцию в зависимости только от формы ее наборов уровней. Например, два хорошо известных морфологических оператора - это дилатация и эрозия.

1.2.3 Алгоритм SLIC

Алгоритм Simple Linear Iterative Clustering (SLIC) или простая линейная итеративная кластеризация предназначен для генерации суперпикселей.

«Суперпиксель можно определить как группу пикселей, которые имеют общие характеристики (например, интенсивность пикселей). Суперпиксели становятся полезными во многих алгоритмах компьютерного зрения и обработки изображений, таких как сегментация изображения, семантическая маркировка, обнаружение и отслеживание объектов и т. д.» [15].

Алгоритм SLIC генерирует суперпиксели путем кластеризации пикселей на основе их цветового сходства и близости в плоскости изображения.

Псевдокод алгоритма SLIC представлен на рисунке 5.

1: Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S.
2: Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient position.
3: **repeat**
4: **for** each cluster center C_k **do**
5: Assign the best matching pixels from a $2S \times 2S$ square neighborhood around the cluster center according to the distance measure (Eq. 1).
6: **end for**
7: Compute new cluster centers and residual error E {$L1$ distance between previous centers and recomputed centers}
8: **until** $E \leq$ threshold
9: Enforce connectivity.

Рисунок 5 - Псевдокод алгоритма SLIC

«Процесс связывания пикселей с ближайшим центром кластера и пересчета центра кластера повторяется до сходимости. В конце этого процесса может остаться несколько случайных меток, то есть несколько пикселей в непосредственной близости от большего сегмента, имеющего такую же метку, но не связанных с ним» [15].

Связность может быть обеспечена на последнем этапе алгоритма путем перемаркировки непересекающихся сегментов метками самого большого соседнего кластера.

Рассмотрим методику сравнения алгоритмов сегментации.

Поскольку ручная сегментация сложных изображений, например, трехмерных медицинских снимков, очень сложна и связана с существенными затратами, необходимы программы, автоматизирующие процесс сегментации изображений. Для поиска лучших алгоритмов необходимо оценить несколько алгоритмов на наборе экземпляров тестовых изображений.

В настоящее время наиболее эффективной считается методика, основанная на визуальном методе сравнения алгоритмов сегментации.

Метод основан на загрузке тестового изображения в программу сегментации, реализующей конкретный алгоритм, и сохранении полученного результата в специальном репозитории.

Указанная процедура выполняется для одного и того же изображения и для всех сравниваемых алгоритмов. Далее эксперт проводит визуальное сравнение результатов сегментации и определяет наиболее эффективный алгоритм для его сегментации. Несмотря на некоторую субъективность данная методика широко применяется на практике.

Выводы к главе 1

Первая глава посвящена анализу предметной области автоматизации.

Результаты проделанной работы позволили сделать следующие выводы:

- Для улучшения бизнес-процесса анализа изображений необходимо разработать и внедрить программный модуль, который позволяет выбрать для сегментации конкретного изображения оптимальный алгоритм.
- В настоящее время наиболее эффективной считается методика, основанная на визуальном методе сравнения алгоритмов сегментации.

Глава 2 Проектирование программного модуля для сравнения алгоритмов сегментации изображений

2.1 Логическое проектирование программного модуля

Логическое проектирование приложения представляет собой процесс разработки его объектной модели и программной архитектуры.

Объектная модель приложения представляет собой комплекс базовых диаграмм языка UML, отражающих различные аспекты приложения: диаграммы вариантов использования, диаграммы классов и диаграммы последовательности.

«Для отражения функционального аспекта ПМСА применим диаграмму вариантов использования UML.

Назначение диаграммы вариантов использования - дать графический обзор функций, предоставляемых системой, с точки зрения субъектов-акторов, их целей (представленных в виде вариантов использования) и любых зависимостей между этими вариантами использования.

Диаграмма вариантов использования описывает использование системы.

Связи между акторами и вариантами использования представляют собой коммуникации, которые происходят между акторами и субъектами для выполнения функций, связанных с вариантами использования.

Суть варианта использования может быть представлена через границу системы.

Акторы могут напрямую или косвенно взаимодействовать с системой. Они часто являются специализированными, чтобы представлять таксономию типов пользователей или внешних систем.

Акторы связаны с вариантами использования посредством каналов связи, каждый из которых представлен отношениями» [7].

Для упрощения процесса построения диаграммы вариантов использования используем методологию проектирования RUP (Rational Unified Process) [18].

Единственным актором в процессе оптимизации веб-клиента является

Бизнес-аналитик.

Варианты использования (прецеденты) представлены в таблицах 2-5.

Таблица 2 – Загрузка изображения

Прецедент: Загрузка изображения
ID: 1
Краткое описание: загрузка изображения для анализа
Главный актор: Бизнес-аналитик
Второстепенные акторы: нет
Предусловие: нет
Постусловие: нет
Основной поток: Бизнес-аналитик загружает из репозитория в программу изображение
Альтернативные потоки: нет

Таблица 3 – Выбор алгоритма сегментации

Прецедент: Выбор алгоритма сегментации
ID: 2
Краткое описание: Выбор алгоритма сегментации изображения
Главный актор: Бизнес-аналитик
Второстепенный актор: нет
Предусловие: подготовка программы к сегментации
Основной поток: Бизнес-аналитик выбирает из списка алгоритм сегментации
Постусловие: нет
Альтернативные потоки: нет

Таблица 4 – Сегментация изображения

Прецедент: Сегментация изображения
ID: 3
Главный актор: Бизнес-аналитик
Второстепенный актор: нет
Предусловие: загрузка изображения и выбор алгоритма сегментации
Основной поток: Бизнес-аналитик запускает процедуру сегментации
Постусловие: нет
Альтернативные потоки: нет

Таблица 5 – Формирование отчета

Прецедент: Формирование отчета
ID: 4
Краткое описание: Формирование отчета по результатам анализа
Главный актор: Бизнес-аналитик
Второстепенный актор: нет
Предусловие: нет
Основной поток: Бизнес-аналитик запускает процедуру формирование отчета
Постусловие: отчет для сравнительного анализа алгоритмов
Альтернативные потоки: нет

На рисунке 6 изображена диаграмма вариантов использования ПМСА сегментации изображения, построенная на основе описанных таблиц.

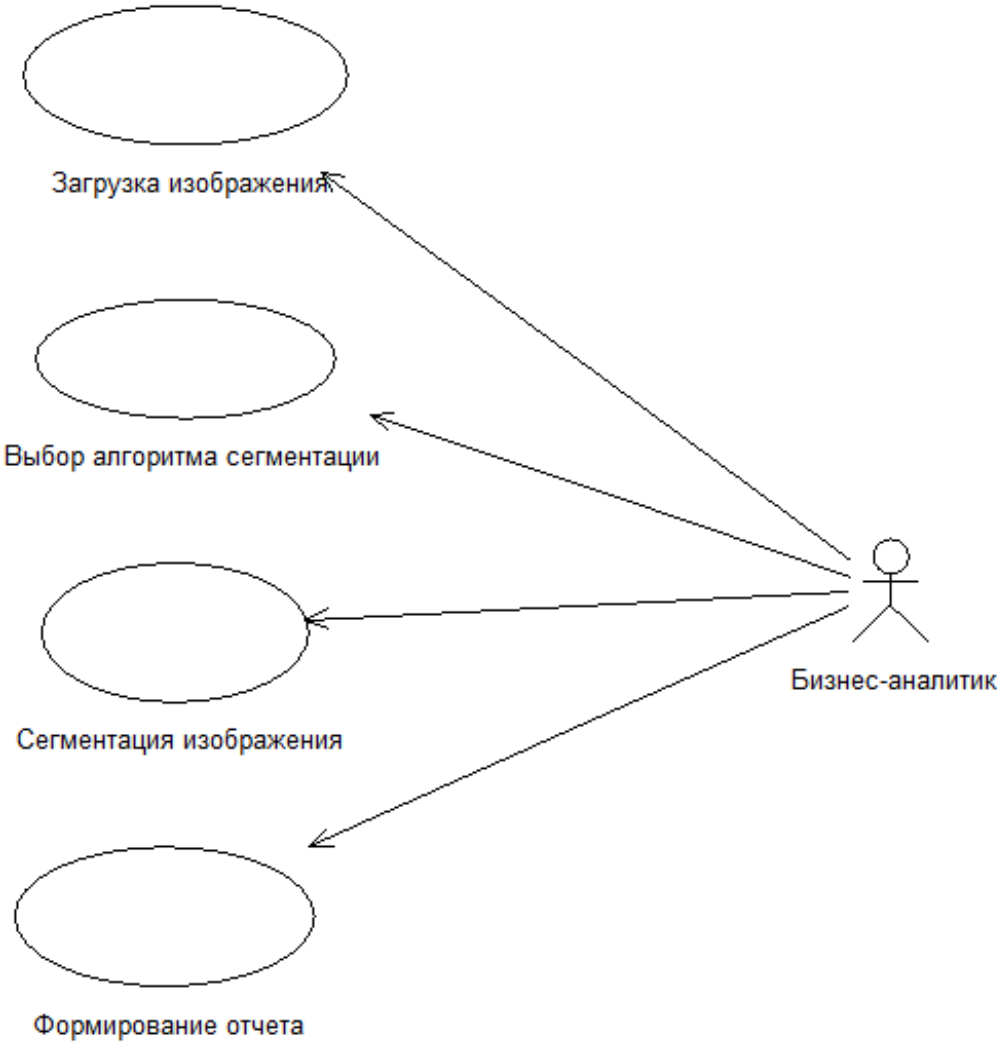

Рисунок 6 – Диаграмма вариантов использования ПМСА

Представленная диаграмма отражает функциональный аспект ПМСА.

Для представления классов программного модуля и связей между ними разработана диаграмма классов UML.

Диаграммы классов показывают статическую структуру модели, в частности, существующие вещи, такие как классы, их внутреннюю структуру и их отношения с другими классами. Диаграммы классов не отображают временную информацию.

Диаграмма классов представлена как набор (статических) декларативных элементов модели, таких как классы, пакеты и их отношения, связанных в виде графа друг с другом и с их содержимым. Диаграммы классов могут быть организованы в пакеты (и принадлежать им), показывая только то, что имеет отношение к конкретному пакету.

Диаграмма классов ПМСА представлена на рисунке 7.

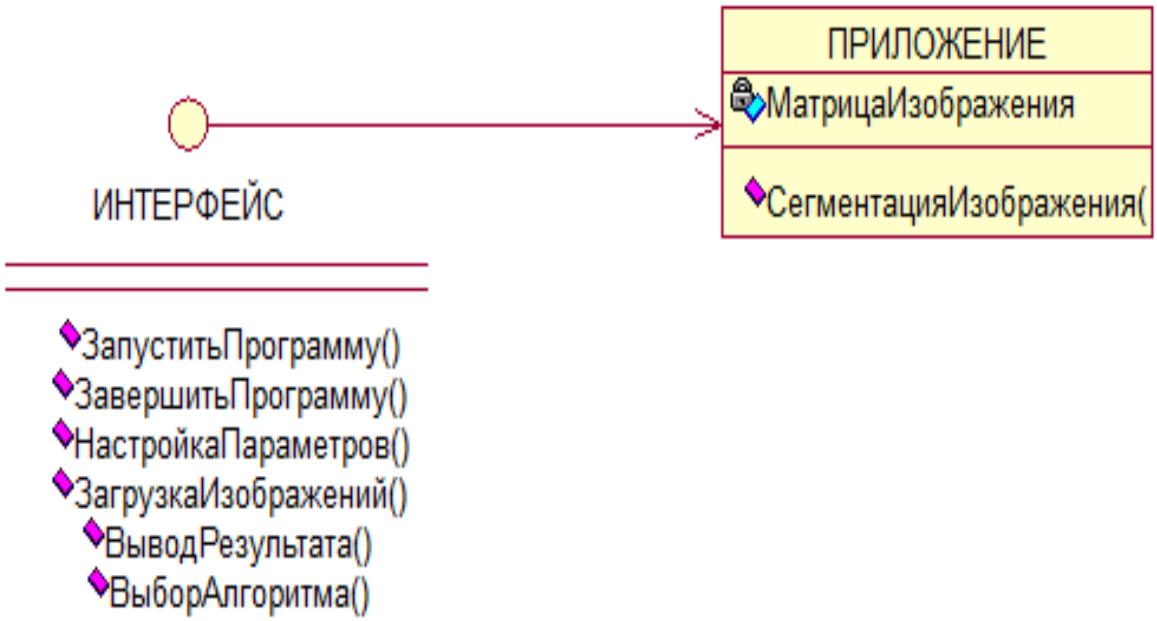

Рисунок 7 – Диаграмма классов ПМСА

Спецификация классов ПМСА представлена в таблице 6

Таблица 6 – Спецификация классов ПМСА

Класс	Описание
Интерфейс	Класс объектов, моделирующих на логическом уровне интерфейс программы
Приложение	Класс объектов, моделирующих на логическом уровне приложение программы

Для представления сценария выполнения программных функций используем диаграмму последовательности UML.

Диаграммы последовательности подчеркивают хронологический ход обмена информацией.

Диаграммы последовательностей легче понять разработчикам и аналитикам.

Диаграмма последовательности иллюстрирует различные сценарии бизнес-варианта использования.

На рисунке 8 изображена диаграмма последовательности сценария сегментации изображений в ПМСА.

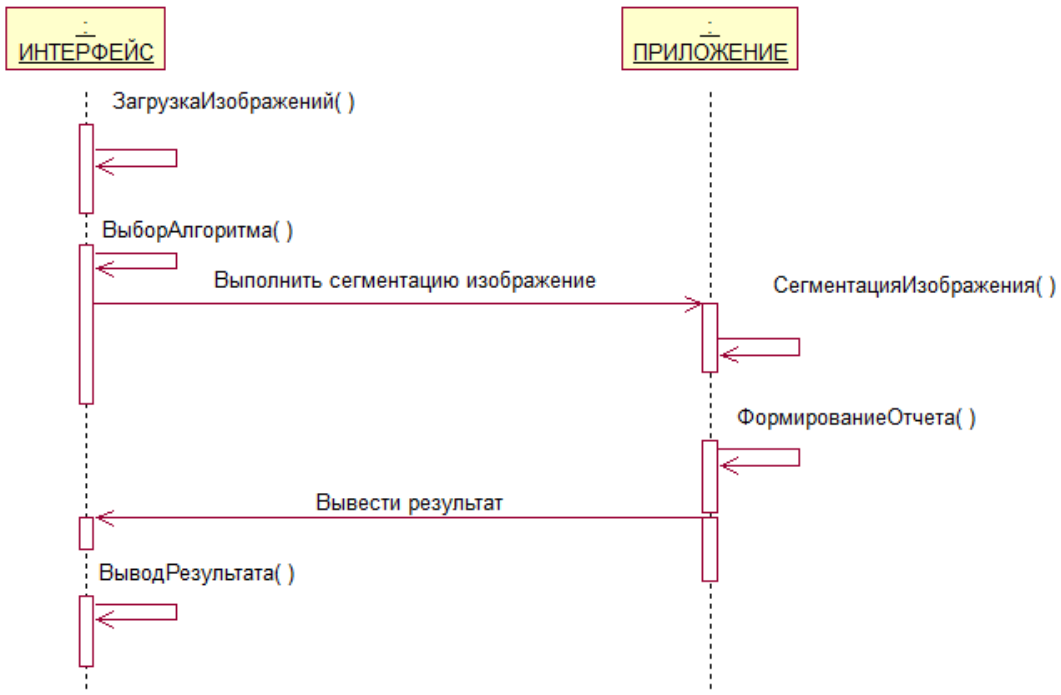

Рисунок 8 – Диаграмма последовательности сценария сегментации изображений в ПМСА

Сценарий организован следующим образом:

В случайный момент времени объект Интерфейс по команде пользователя активизирует функцию загрузки изображения.

Далее объект Интерфейс по команде пользователя активизирует функцию выбора алгоритма сегментации.

Объект Приложения выполняет процедуру сегментации изображения по выбранному алгоритму и обращается к объекту Интерфейс с запросом на вывод результат.

Объект Интерфейс выводит результат сегментации на экран и/или на печать.

Процесс сегментации завершается.

Как и диаграммы деятельности, диаграммы последовательности могут быть смоделированы, охватывая несколько вариантов использования, а также использоваться для уточнения бизнес-вариантов использования.

Диаграмма последовательности отображает динамический аспект ПМСА.

2.2 Выбор средств реализации программного модуля

В соответствии с проектными ограничениями для реализации программного модуля ПМСА сегментации изображений используем технологию Integrated Development Environment (IDE).

«IDE – интегрированная среда разработки представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

Интегрированная среда разработки позволяет программистам объединять различные задачи написания компьютерной программы.

Интегрированные среды разработки повышают продуктивность программиста за счет объединения общих действий по написанию программного обеспечения в одном приложении: редактирование исходного кода, создание исполняемых файлов и отладка» [2].

В состав типовой интегрированной среды разработки входят:

- текстовый редактор;
- транслятор – компилятор или интерпретатор;
- средства автоматизации сборки;
- отладчик.

Рассмотрим функциональные и архитектурные особенности популярных интегрированных сред разработки.

2.2.1 Интегрированная среда разработки Visual Studio

«Интегрированная среда разработки или IDE Visual Studio– это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений.

Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки (рисунок 9)» [3].

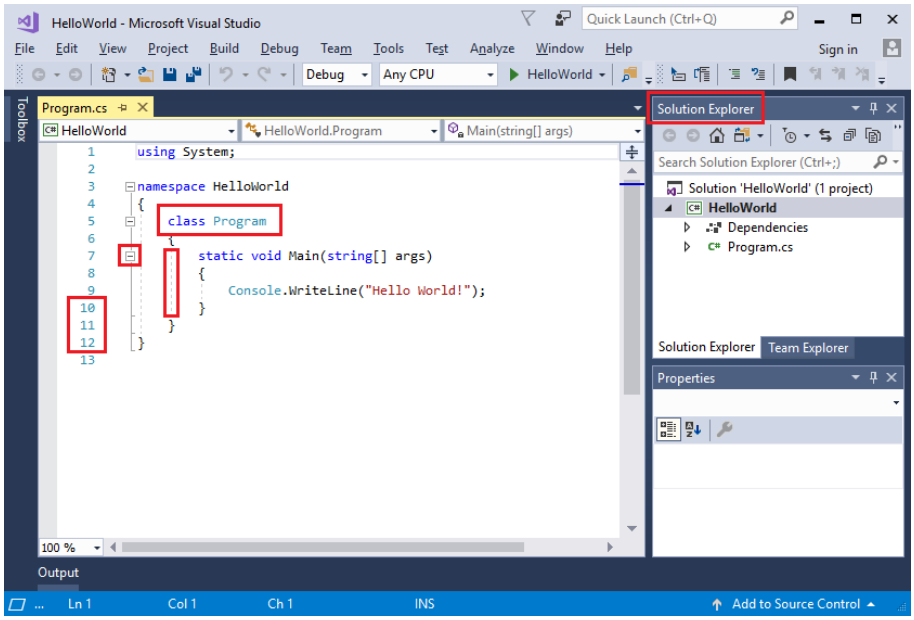

Рисунок 9 – Экран обозревателя IDE Visual Studio

«Основными инструментами разработчика является следующие:

- обозреватель решений, который позволяет просматривать файлы кода, перемещаться по ним и управлять ими. Обозреватель решений позволяет упорядочить код путем объединения файлов в решения и проекты;
- редактор, отображающий содержимое файла. Здесь можно редактировать код или разрабатывать пользовательский интерфейс, например, окно с кнопками или текстовые поля;
- командный обозревать, который позволяет отслеживать рабочие элементы и использовать код совместно с другими пользователями с помощью технологий управления версиями, таких как Git и система управления версиями Team Foundation.

Среда Visual Studio доступна для операционных систем Windows и Mac.

Возможности Visual Studio оптимизированы для разработки кроссплатформенных и мобильных приложений.

Существует три выпуска Visual Studio: Community, Professional и Enterprise» [3].

2.2.2 Интегрированная среда разработки NetBeans

NetBeans – это бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения (рисунок 10).

«Среда NetBeans предоставляет все средства, необходимые для создания профессиональных десктоп-приложений, корпоративных, мобильных и веб-приложений на платформе Java, а также C++, PHP и других языках» [11] .

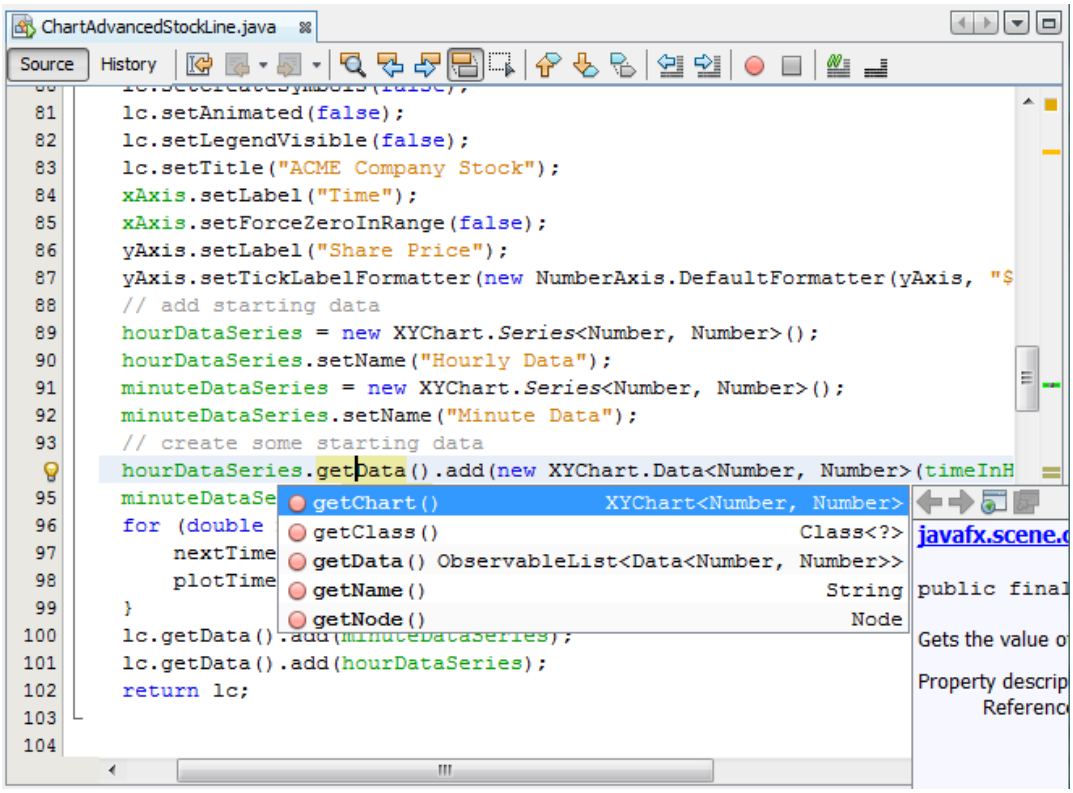

Рисунок 10 – Экран IDE NetBeans

«Основные характеристики среды NetBeans следующие:

- рабочая область среды является полностью настраиваемой – существует возможность пользовательской настройки действий, выполняемых с помощью панели, назначения «горячих» клавиш и т. д.;
- среда имеет в своем составе расширенный многоязыковой редактор для различных языков программирования, прежде всего для Java. Существует возможность расширения функций редактора с целью поддержки любого другого языка;
- Редактор среды делает отступы строк, проверяет соответствие скобок и слов, подсвечивает синтаксис исходного кода.
- Производятся проверка ошибок во время ввода, отображение вариантов для автозавершения кода и фрагментов документации по требуемому языку программирования» [11].
- Редактор может генерировать и вставлять в исходный код стандартные

фрагменты кода на Java или других языках.

- Браузер классов позволяет просматривать иерархию и структуру любого класса Java и другие.

«Среда NetBeans позволяет разрабатывать Java десктоп-приложения с профессиональными графическими интерфейсами пользователя, а также создавать веб- и корпоративные приложения в соответствии с современными стандартами» [11].

2.2.3 Интегрированная среда разработки Eclipse + PyDEv

Интегрированная среда разработки Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation [14].

Среда написана на языке программирования Java.

Основной целью её создания является повышение продуктивности процесса разработки программного обеспечения.

Фрагмент среды представлен на рисунке 11.

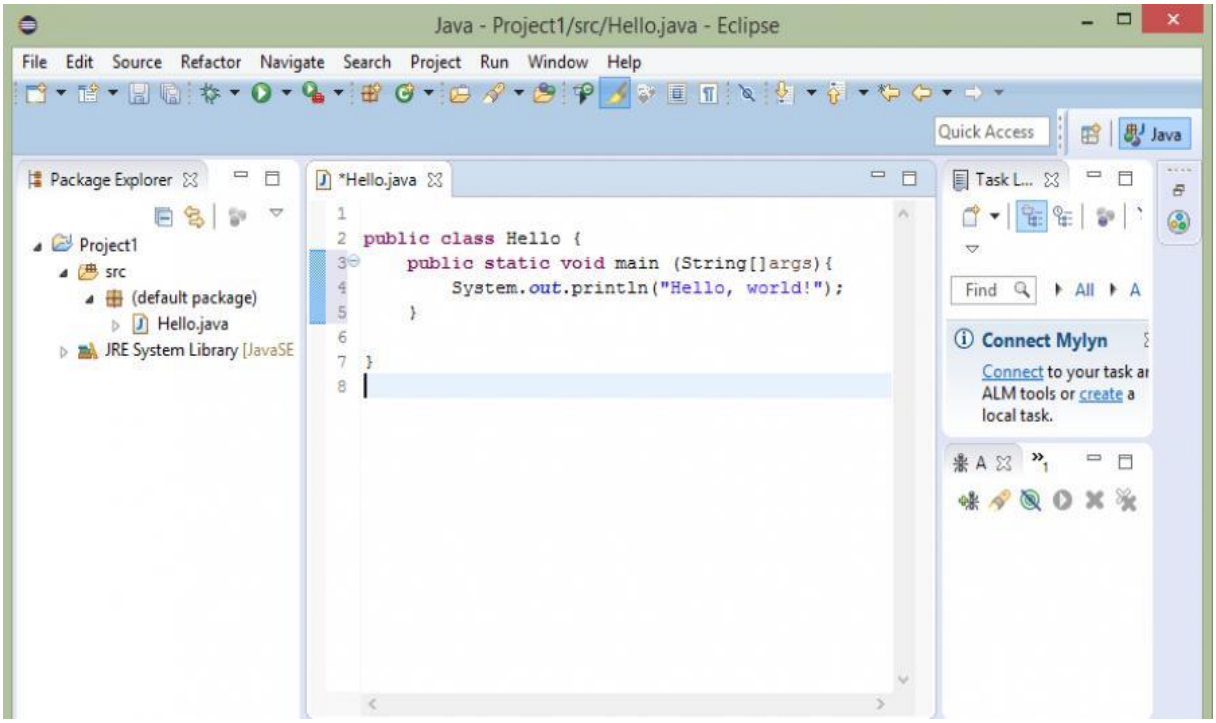

Рисунок 11 – Экран IDE Eclipse

«Основным компонентом является исполняемая среда – Eclipse Runtime, в которой выполняются коды расширений и модулей.

Она обеспечивает всю базовую функциональность – управление расширениями и обновлениями, взаимодействие с операционной системой, обеспечение работы системы помощи.

Следующим ключевым компонентом является собственно интегрированная среда разработки – она отвечает за управление элементами программы, управление проектами, отладку и сборку проектов, поиск по файлам и командную программу» [14].

Eclipse претендует на статус наиболее популярной Java IDE и является единственным конкурентом такой мощной платформы, как NetBeans.

Но в отличие от последней, использующей для создания элементов пользовательского интерфейса платформонезависимую библиотеку Swing, Eclipse использует платформозависимую библиотеку SWT.

«Интегрированные среды разработки, созданные на платформе Eclipse, широко применяются для создания программного обеспечения на различных языках программирования.

Это обусловлено универсальностью Eclipse, работающей по принципу «Плагины для Eclipse разрабатываются в самой Eclipse.

Следует учесть, что в рамках проекта Eclipse Foundation предлагается несколько платформ для создания подключаемых модулей для настольных инструментов, распределенных сервисов и интерфейсов браузера.

Одним из таких расширений является PyDev, предоставляющий интерактивную консоль Python и возможности для отладки и автодополнения кода» [5].

Установить его просто: запустите Eclipse, выберите Help → Eclipse Marketplace, затем найдите PyDev. Нажмите «Install» и при необходимости перезапустите Eclipse (рисунок 12).

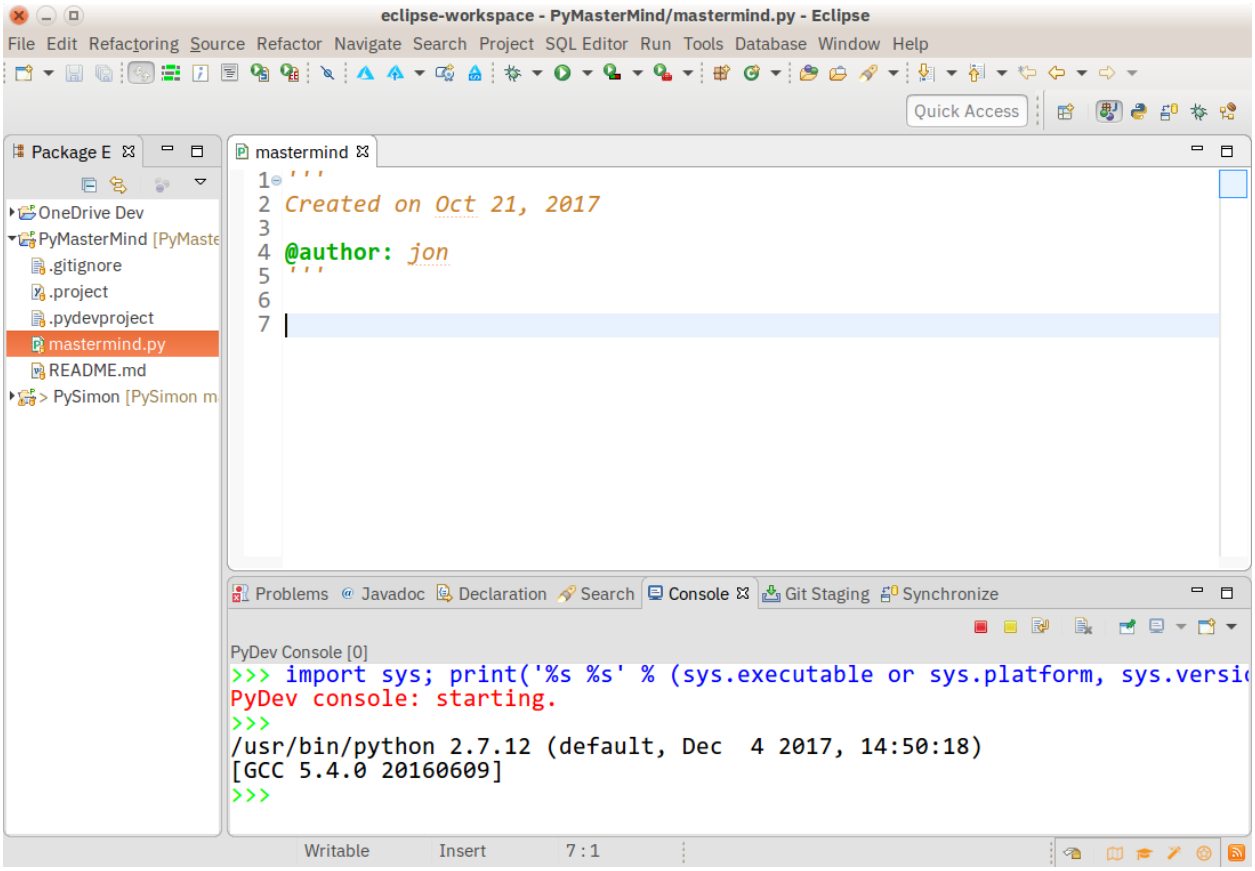

Рисунок 12 – Экран редактора языка Python

Установка PyDev пройдёт быстро и гладко.

У опытного пользователя Eclipse не возникнет проблем с изучением этого расширения.

Для выбора интегрированной среды разработки используем таблицу 2, составленную на основе анализа блогов по данной тематике.

Критерии оценивания:

0 – полное несоответствие требованиям;

1 – значительное несоответствие требованиям;

2 – незначительное несоответствие требованиям;

3 – полное соответствие требованиям.

Таблица 2 – Сравнительный анализ интегрированных сред разработки

Характеристика/балл	Visual Studio	NetBeans	Eclipse+PyDEv
стоимость	0	0	3
поддержка языка Python	0	3	3
юзабилити	3	2	2
предпочтение разработчика	1	3	2
Итого	4	8	10

Таким образом, наилучшими характеристиками обладает IDE Eclipse+PyDEv.

Выбираем IDE Eclipse+PyDEv в качестве среды для разработки программного модуля для сравнения алгоритмов сегментации.

Выводы к главе 2

Первая глава посвящена проектированию ПМСА.

Результаты проделанной работы позволили сделать следующие выводы:

- Логическое проектирование ПМСА представляет собой процесс разработки его объектной модели и программной архитектуры.
- Объектная модель приложения представляет собой комплекс базовых диаграмм языка UML, отражающих различные аспекты приложения: диаграммы вариантов использования, диаграммы классов и диаграммы последовательности.
- Наилучшими характеристиками в качестве среды для реализации ПМСА обладает IDE Eclipse+PyDEv.

Глава 3 Реализация и тестирование программного модуля для сравнения алгоритмов сегментации изображений

Для представления программной архитектуры ПМСА используем диаграмму компонентов UML.

Диаграмма компонентов UML показывает компоненты, предоставленные и требуемые интерфейсы, порты и отношения между ними. Диаграммы компонентов предлагают аналитикам естественный формат для начала моделирования решения и позволяют проверить, требуются ли для системы дополнительные функциональные возможности. Диаграмма компонентов имеет более высокий уровень абстракции, чем диаграмма классов UML.

Разработчики находят диаграмму компонентов полезной, поскольку она предоставляет им высокоуровневое архитектурное представление системы, которую они будут строить, в то время как системные администраторы находят диаграммы компонентов полезными, поскольку они получают раннее представление о логических компонентах программного обеспечения.

На рисунке 13 изображена диаграмма компонентов ПМСА.

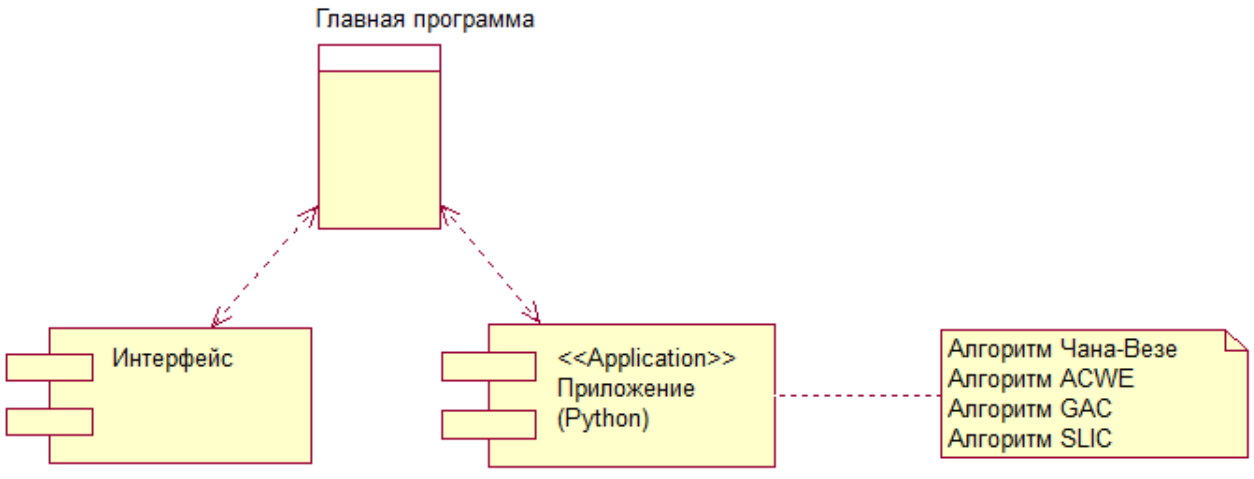

Рисунок 13 – Диаграмма компонентов ПМСА

ПМСА реализован на языке Python в среде Eclipse+PyDEv [10].

Разработаны программные реализации алгоритмов сегментации, листинги которых представлены ниже.

Листинг программного кода алгоритма Чана-Везе показана на рисунке 14 [13].

```
import matplotlib.pyplot as plt
from skimage import data, img_as_float
from skimage.segmentation import chan_vese

image = img_as_float(data.camera())
# Feel free to play around with the parameters to see how they impact the result
cv = chan_vese(image, mu=0.25, lambda1=1, lambda2=1, tol=1e-3, max_iter=200,
               dt=0.5, init_level_set="checkerboard", extended_output=True)

fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.flatten()

ax[0].imshow(image, cmap="gray")
ax[0].set_axis_off()
ax[0].set_title("Original Image", fontsize=12)

ax[1].imshow(cv[0], cmap="gray")
ax[1].set_axis_off()
title = "Chan-Vese segmentation - {} iterations".format(len(cv[2]))
ax[1].set_title(title, fontsize=12)

ax[2].imshow(cv[1], cmap="gray")
ax[2].set_axis_off()
ax[2].set_title("Final Level Set", fontsize=12)

ax[3].plot(cv[2])
ax[3].set_title("Evolution of energy over iterations", fontsize=12)

fig.tight_layout()
plt.show()
```

Рисунок 14 - Листинг программного кода алгоритма Чана-Везе

Листинг программного кода алгоритма SLIC показан на рисунке 15 [19].

```
# import the necessary packages
from skimage.segmentation import slic
from skimage.segmentation import mark_boundaries
from skimage.util import img_as_float
from skimage import io
import matplotlib.pyplot as plt
import argparse

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required = True, help = "Path to the image")
args = vars(ap.parse_args())

# load the image and convert it to a floating point data type
image = img_as_float(io.imread(args["image"]))

# loop over the number of segments
for numSegments in (100, 200, 300):
    # apply SLIC and extract (approximately) the supplied number
    # of segments
    segments = slic(image, n_segments = numSegments, sigma = 5)

    # show the output of SLIC
    fig = plt.figure("Superpixels -- %d segments" % (numSegments))
    ax = fig.add_subplot(1, 1, 1)
    ax.imshow(mark_boundaries(image, segments))
    plt.axis("off")

# show the plots
plt.show()
```

Рисунок 15 - Листинг программного кода алгоритма SLIC

Листинг программного кода алгоритма ACWE показан на рисунке 16 [17].

```
# Morphological ACWE
image = img_as_float(data.camera())

# Initial level set
init_ls = checkerboard_level_set(image.shape, 6)
# List with intermediate results for plotting the evolution
evolution = []
callback = store_evolution_in(evolution)
ls = morphological_chan_vese(image, 35, init_level_set=init_ls, smoothing=3,
                             iter_callback=callback)

fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.flatten()

ax[0].imshow(image, cmap="gray")
ax[0].set_axis_off()
ax[0].contour(ls, [0.5], colors='r')
ax[0].set_title("Morphological ACWE segmentation", fontsize=12)

ax[1].imshow(ls, cmap="gray")
ax[1].set_axis_off()
contour = ax[1].contour(evolution[2], [0.5], colors='g')
contour.collections[0].set_label("Iteration 2")
contour = ax[1].contour(evolution[7], [0.5], colors='y')
contour.collections[0].set_label("Iteration 7")
contour = ax[1].contour(evolution[-1], [0.5], colors='r')
contour.collections[0].set_label("Iteration 35")
ax[1].legend(loc="upper right")
title = "Morphological ACWE evolution"
ax[1].set_title(title, fontsize=12)
```

Рисунок 16 - Листинг программного кода алгоритма ACWE

Листинг программного кода алгоритма GAC показан на рисунке 17.

```
# Morphological GAC
image = img_as_float(data.coins())
gimage = inverse_gaussian_gradient(image)

# Initial level set
init_ls = np.zeros(image.shape, dtype=np.int8)
init_ls[10:-10, 10:-10] = 1
# List with intermediate results for plotting the evolution
evolution = []
callback = store_evolution_in(evolution)
ls = morphological_geodesic_active_contour(gimage, 230, init_ls,
                                           smoothing=1, balloon=-1,
                                           threshold=0.69,
                                           iter_callback=callback)

ax[2].imshow(image, cmap="gray")
ax[2].set_axis_off()
ax[2].contour(ls, [0.5], colors='r')
ax[2].set_title("Morphological GAC segmentation", fontsize=12)

ax[3].imshow(ls, cmap="gray")
ax[3].set_axis_off()
contour = ax[3].contour(evolution[0], [0.5], colors='g')
contour.collections[0].set_label("Iteration 0")
contour = ax[3].contour(evolution[100], [0.5], colors='y')
contour.collections[0].set_label("Iteration 100")
contour = ax[3].contour(evolution[-1], [0.5], colors='r')
contour.collections[0].set_label("Iteration 230")
ax[3].legend(loc="upper right")
title = "Morphological GAC evolution"
ax[3].set_title(title, fontsize=12)

fig.tight_layout()
plt.show()
```

Рисунок 17 - Листинг программного кода алгоритма GAC

Для проверки работоспособности ПМСА использован метод функционального тестирования.

Функциональное тестирование - это тип тестирования программного обеспечения, при котором проверяется соответствие системы программного

обеспечения функциональным требованиям / спецификациям.

«Целью функциональных тестов является тестирование каждой функции программного приложения путем предоставления соответствующих входных данных и проверки выходных данных на соответствие функциональным требованиям.

Функциональное тестирование в основном включает в себя тестирование черного ящика и не касается исходного кода приложения. Это тестирование проверяет пользовательский интерфейс, API, базу данных, безопасность, связь клиент / сервер и другие функции тестируемого приложения. Тестирование можно проводить вручную или с помощью автоматизации» [4].

Для выгрузки файла изображений из репозитория используется стандартный диалог, представленный на рисунке 18.

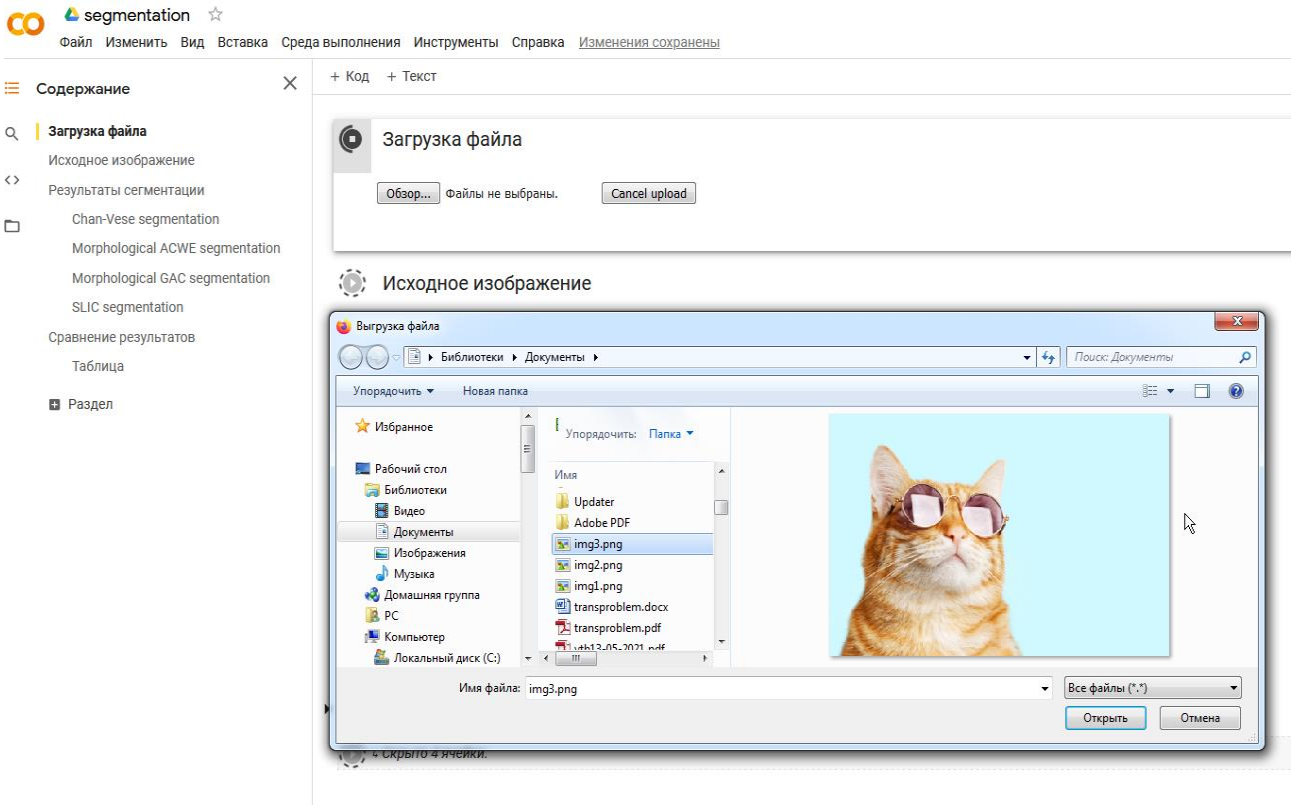

Рисунок 18 – Окно выгрузки файла изображений из репозитория

Далее выполняется загрузка файла изображения в ПСА, выбор алгоритма

сегментации из списка и запуск функции сегментации (рисунок 19).

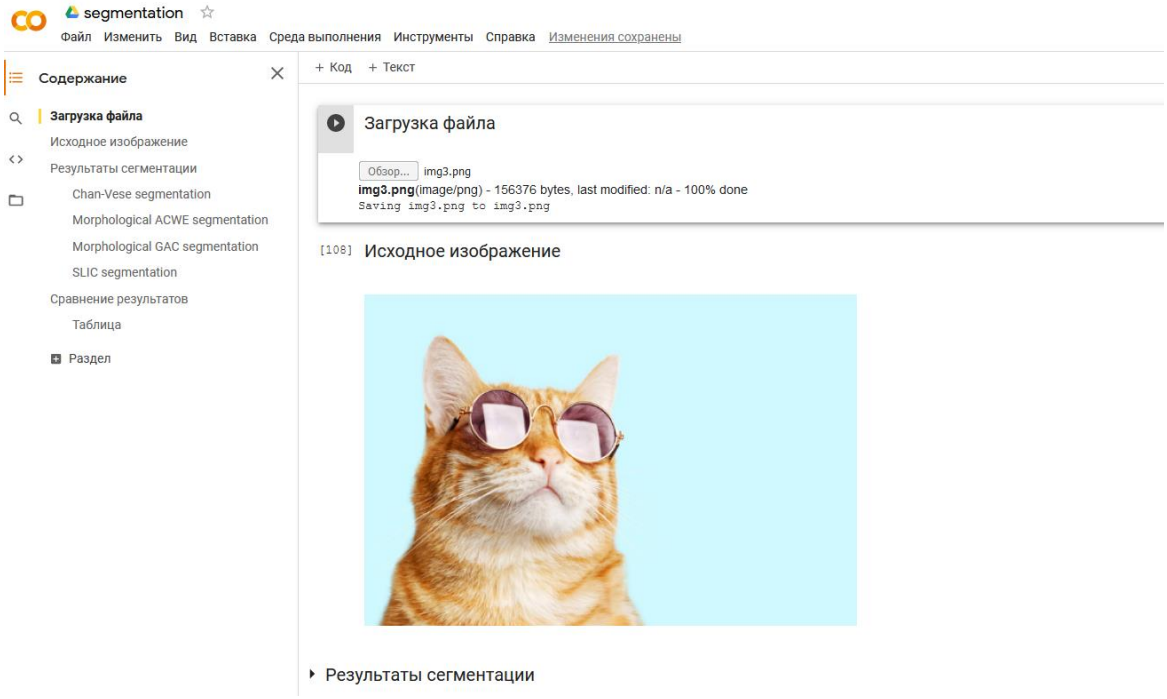

Рисунок 19 – Окно запуска сегментации изображения «Кошка»

Результаты сегментации для разных алгоритмов представлены на рисунке 20.

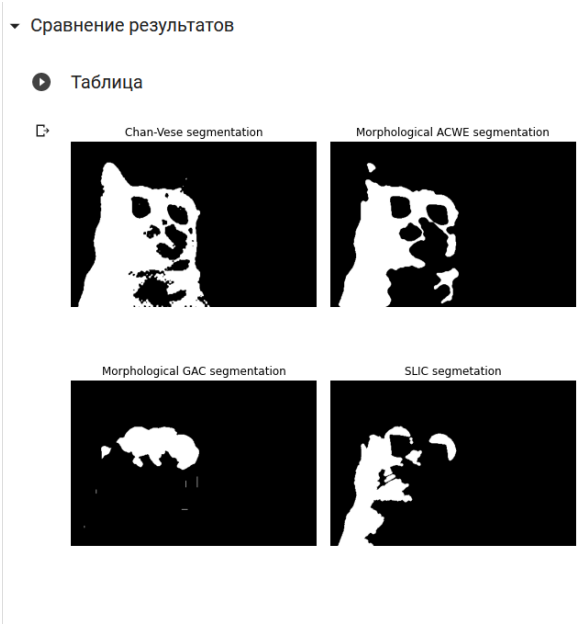

Рисунок 20 – Результате сегментации изображения на рисунке 19

Аналогичным образом проведены сравнения для других тестовых изображений (рисунки 21-24).

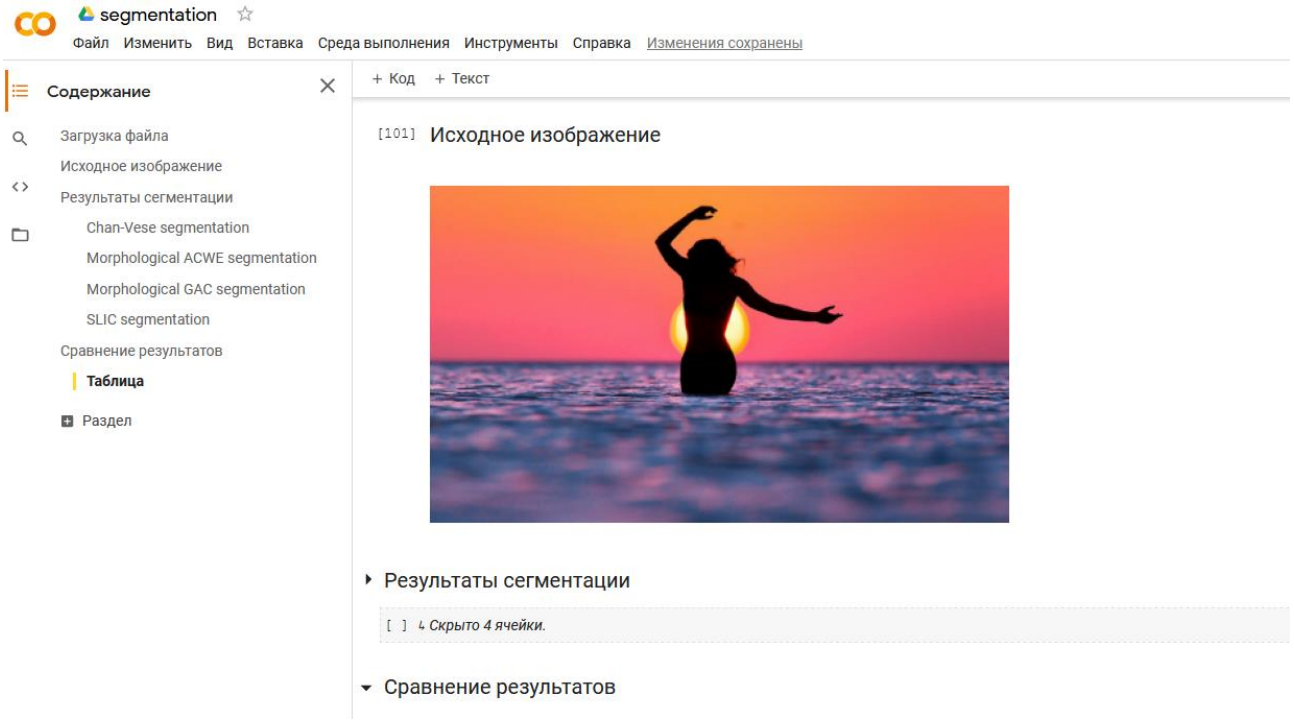

Рисунок 21 – Сегментация изображения «Закат»

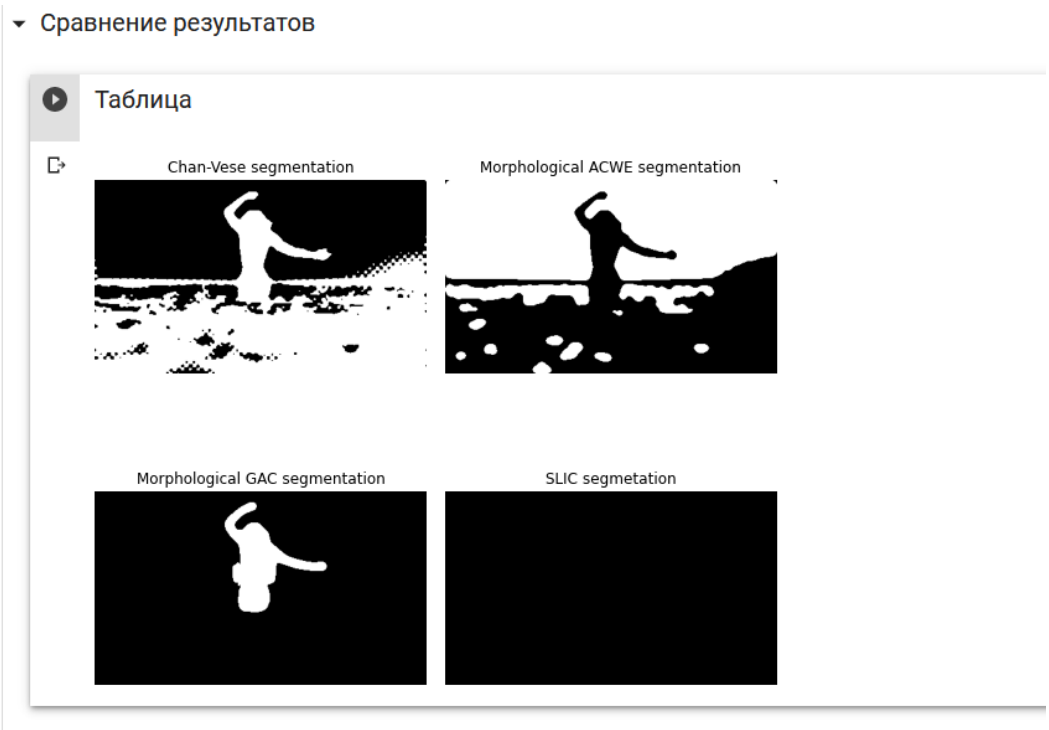

Рисунок 22 – Результате сегментации изображения на рисунке 19

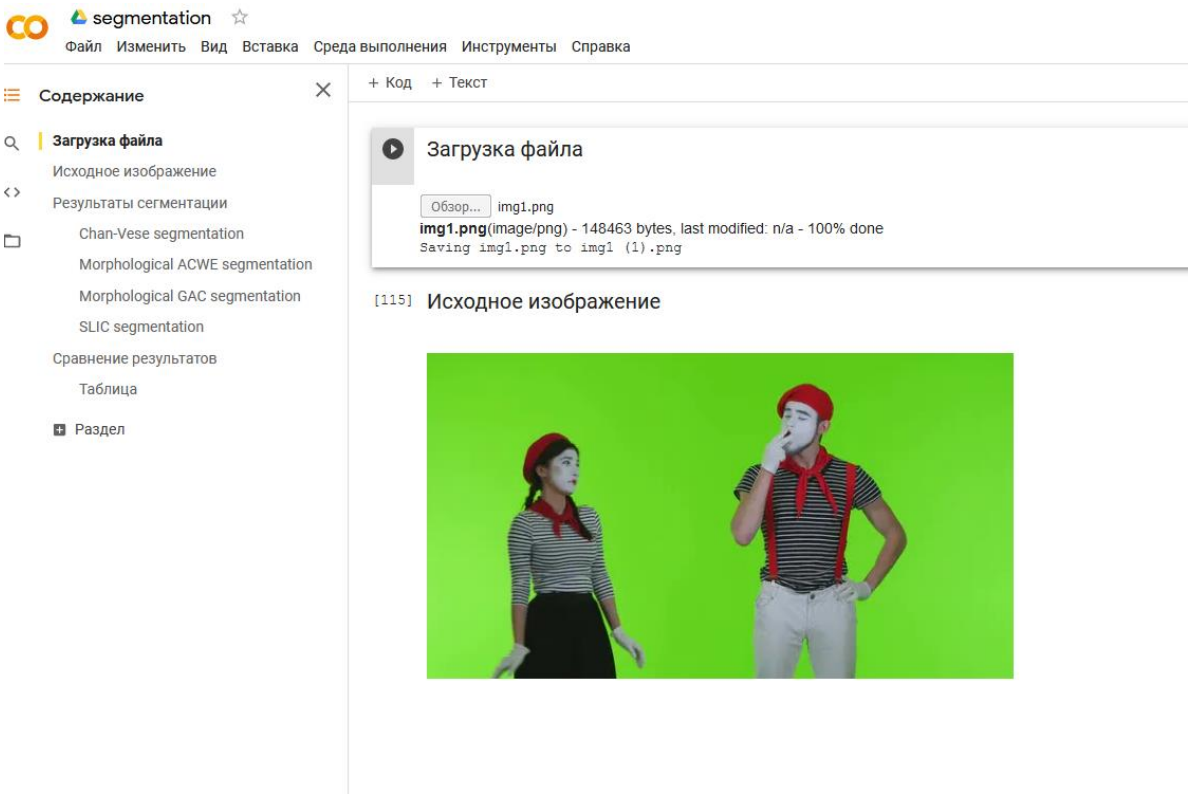

Рисунок 23 – Сегментация изображения «Клоуны»

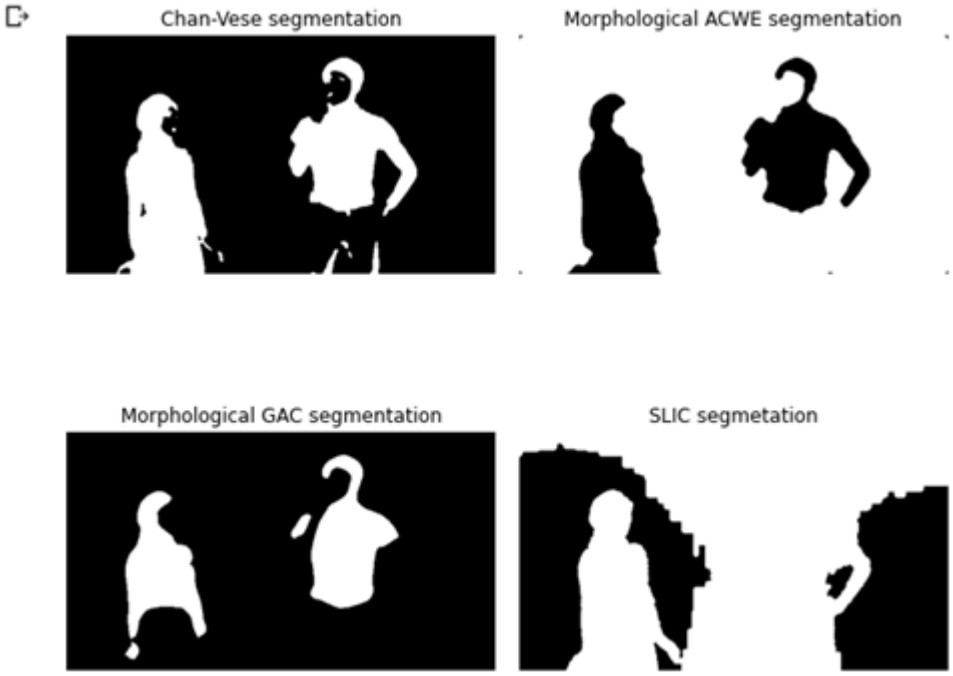

Рисунок 24 – Результате сегментации изображения на рисунке 23

После внедрения ПМСА функциональная архитектура системы поддержки управленческих решений будет иметь вид, представленный на рисунке 25.

Рисунок 25 – Функциональная архитектура системы поддержки управленческих решений

Таким образом, функциональное тестирование подтвердило работоспособность ПМСА и возможность проведения с его помощью сравнительного анализа алгоритмов сегментации, что в конечном итоге должно обеспечить повышения качество аналитической информации и, как следствие, эффективности принятых управленческих решений.

Выводы к главе 3

Третья глава посвящена реализации ПМСА.

Результаты проделанной работы позволили сделать следующие выводы:

- Диаграмма компонентов UML дает наглядное представление программной архитектуры ПМСА.
- Функциональное тестирование подтвердило работоспособность ПМСА и возможность проведения с его помощью сравнительного анализа алгоритмов сегментации, что в конечном итоге должно обеспечить повышения качество аналитической информации и, как следствие, эффективности принятых управленческих решений.

Заключение

Выпускная квалификационная работа посвящена проблеме разработки программного модуля для сравнения алгоритмов сегментации изображений, обеспечивающего повышение эффективности принятых управленческих решений.

Для анализа изображений широко применяется метод их сегментация.

Как показывает практика, наилучших результатов при решении конкретной задачи анализа изображения удается достичь при правильном выборе алгоритма его сегментации. Это существенно повышает эффективность принятого на основе результатов анализа управленческого решения.

Правильность такого выбора обусловлена применением соответствующего программного модуля для сравнения алгоритмов сегментации изображений.

Разработка такого модуля представляет актуальность и научно-практический интерес.

В процессе выполнения ВКР были решены следующие задачи:

− произведен анализ предметной области автоматизации. Как показал анализ, для улучшения бизнес-процесса анализа изображений необходимо разработать и внедрить программный модуль, который позволяет выбрать для сегментации конкретного изображения оптимальный алгоритм;

− выбрана технология и спроектирован программный модуль для сравнения алгоритмов сегментации изображений. Отмечено, что в настоящее время наиболее эффективной считается методика, основанная на визуальном методе сравнения алгоритмов сегментации;

− разработана объектная модель ПМСА, которая представляет собой комплекс базовых диаграмм языка UML, отражающих различные аспекты приложения: диаграммы вариантов использования, диаграммы классов и диаграммы последовательности. В качестве среды для реализации ПМСА выбрана IDE Eclipse+PyDEv;

− на языке Python реализован программный модуль для сравнения алгоритмов сегментации изображений. Для проверки работоспособности ПМСА

использован метод функционального тестирования;

− разработана функциональная архитектура системы поддержки управленческих решений после внедрения ПМСА;

− в процессе тестирования в программу загружались тестовые изображения и визуально сравнивались результаты их сегментации с помощью различных алгоритмов.

Функциональное тестирование подтвердило работоспособность ПМСА и возможность проведения с его помощью сравнительного анализа алгоритмов сегментации, что в конечном итоге должно обеспечить повышения качество аналитической информации и, как следствие, эффективности принятых управленческих решений.

Результаты бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы для разработчиков программ, в которых поддержка принятия управленческих решений осуществляется на основе результатов анализа графической информации.

Список используемой литературы и используемых источников

1. Ильясова Н.Ю. Методы цифрового анализа сосудистой системы человека. Обзор литературы [Электронный ресурс]. URL: http://www.computeroptics.ru/KO/PDF/KO37-4/370416.pdf (дата обращения: 25.05.2021).

2. Интегрированные среды разработки программ [Электронный ресурс]. URL: http://bourabai.ru/einf/ide.htm (дата обращения: 15.05.2021).

3. Краткое руководство. Знакомство с интегрированной средой разработки Visual Studio [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/visualstudio/ide/quickstart-ide-orientation?view=vs-2019 (дата обращения: 15.05.2021).

4. Липаев В. В. Тестирование компонентов и комплексов программ : учебник. Москва : СИНТЕГ, 2010. 393 с.

5. Лучшие IDE и редакторы кода для Python [Электронный ресурс]. URL: https://tproger.ru/translations/python-ide/ (дата обращения: 25.05.2021).

6. Молоткова Н. В., Д.Л. Хазанова. Реинжиниринг бизнес-процессов : учебное пособие. Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2019. 81 с.

7. Носова Л. С. Case-технологии и язык UML : учебно-методическое пособие. Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. 67 с.

8. Подходы к управлению требованиями в IBM OpenUP и FURPS+ [Электронный ресурс]. URL: https://analytics.infozone.pro/requirements-in-ibm-openup-furps/ (дата обращения: 25.05.2021).

9. Хачумов М.В. Расстояния, метрики и кластерный анализ // Искусственный интеллект и принятие решений. №1.2012. С.81-89.

10. Язык программирования Python [Электронный ресурс]. URL: https://intuit.ru/studies/courses/49/49/info (дата обращения: 25.05.2021).

11. Apache NetBeans [Электронный ресурс]. URL: https://netbeans.apache.org/ (дата обращения: 15.05.2021).

12. BPMN Studio. Официальный сайт [Электронный ресурс]. URL: https://bpmn.studio/ru (дата обращения: 25.05.2021).

13. Chan-Vese Segmentation [Электронный ресурс]. URL: https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_chan_vese.html (дата обращения: 25.05.2021).

14. Eclipse IDE [Электронный ресурс]. URL: https://www.eclipse.org/eclipseide/ (дата обращения: 15.05.2021).

15. Geurts A. et al. Visual Comparison of 3D Medical Image Segmentation Algorithms Based on Statistical Shape Models, Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: Ergonomics and Health, 2015.

16. Morphological Snakes [Электронный ресурс]. URL: https://github.com/Borda/morph-snakes (дата обращения: 25.05.2021).

17. Morphological Snakes Algorithms Python [Электронный ресурс]. URL: https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_morphsnakes.html (дата обращения: 25.05.2021).

18. Rational Unified Process [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Rational_Unified_Process (дата обращения: 25.05.2021).

19. Segmentation: A SLIC Superpixel Tutorial using Python [Электронный ресурс]. URL: https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/ (дата обращения: 25.05.2021).

20. Superpixels and SLIC [Электронный ресурс]. URL: https://darshita1405.medium.com/superpixels-and-slic-6b2d8a6e4f08 (дата обращения: 25.05.2021).

21. The Chan-Vese Algorithm [Электронный ресурс]. URL: https://arxiv.org/abs/1107.2782 (дата обращения: 25.05.2021).