

**«Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И.Ульянова (Ленина)» (СПбГЭТУ «ЛЭТИ»)**

|                    |                         |
|--------------------|-------------------------|
| <b>Направление</b> | 11.03.01 – Радиотехника |
| <b>Профиль</b>     | Аудиовизуальная техника |
| <b>Факультет</b>   | РТ                      |
| <b>Кафедра</b>     | ТВ                      |

*К защите допустить*

Зав. кафедрой

Обухова Н.А.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
БАКАЛАВРА**

**Тема: Алгоритм сегментации легких на КТ-снимках**

|              |   |                |                 |
|--------------|---|----------------|-----------------|
| Студентка    |   | <hr/>          | Завражнова Ю.Д. |
|              |   | <i>Подпись</i> |                 |
| Руководитель | к.т.н., доцент<br>(Уч. степень, уч. звание) | <hr/>          | Мотыко А.А.     |
|              |   | <i>Подпись</i> |                 |
| Консультант  | к.э.н., доцент<br>(Уч. степень, уч. звание) | <hr/>          | Косухина М.А.   |
|              |   | <i>Подпись</i> |                 |

Санкт-Петербург

2023

## РЕФЕРАТ

55 с., 36 рис., 4 табл.

КОМПЬЮТЕРНАЯ ТОМОГРАФИЯ, СЕГМЕНТАЦИЯ, АЛГОРИТМ,  
ВОДОРАЗДЕЛ, МЕРА ДАЙСА.

Цель работы создание алгоритма, способного сегментировать легкие на снимках компьютерной томографии с высокой точностью.

В данной выпускной квалификационной работе был произведен анализ существующих методов сегментации. В рамках исследования был определен метод, более остальных подходящий под задачу данной работы, и разработана структурная схема алгоритма. Был разработан алгоритм, позволяющий с высокой точностью сегментировать легкие на снимках компьютерной томографии, и проведена оценка его точности с помощью меры Дайса.

## ANNOTATION

The purpose of writing this final qualifying work was to create an algorithm that segments the lungs in computed tomography images with high accuracy.

To achieve this goal, an analytical review of existing solutions in the subject area was carried out, a choice of software tools for further modeling was implemented, and a simulation of the target algorithm and an assessment of its accuracy was performed.

## ОГЛАВЛЕНИЕ

|  |    |
|--|----|
| ВВЕДЕНИЕ .....   | 5  |
| 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....   | 7  |
| 1.1 Методы диагностики легочных заболеваний в медицинских<br>видеосистемах .....                     | 7  |
| 1.2 Классификация методов сегментации .....  | 13 |
| 1.3 Сравнительный анализ методов сегментации.....  | 22 |
| 1.4 Выводы по главе .....  | 23 |
| 2. ПОДГОТОВКА ДАННЫХ И РАЗРАБОТКА АЛГОРИТМА .....  | 24 |
| 2.1 Подготовка исходных данных .....   | 24 |
| 2.2 Среда разработки и программные средства моделирования.....                                       | 25 |
| 2.3 Описание алгоритма .....   | 25 |
| 2.4 Выводы по главе .....  | 33 |
| 3. АНАЛИЗ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА.....  | 34 |
| 4. РАЗРАБОТКА И СТАНДАРТИЗАЦИЯ ПРОГРАММНЫХ<br>СРЕДСТВ.....   | 41 |
| 4.1 Планирование работ проекта.....  | 41 |
| 4.2 Расчет затрат на выполнение проекта, расчет цены проекта и<br>цены предполагаемого продукта..... | 44 |
| 4.3 Обеспечение качества программного продукта .....   | 47 |
| 4.4 Определение кода программного средства .....   | 48 |
| 4.5 Определение списка стандартов для ВКР .....  | 48 |
| ЗАКЛЮЧЕНИЕ .....   | 49 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....  | 50 |
| ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ.....   | 53 |

## ВВЕДЕНИЕ

С развитием новых технологий, компьютерного зрения, искусственного интеллекта увеличивается их вклад в современную медицину. С каждым годом разрабатывается все больше алгоритмов, способных безошибочно обрабатывать огромное количество медицинских данных и изображений, что значительно упрощает вынесение врачом верного диагноза.

В настоящее время одним из самых активно развивающихся направлений является цифровая обработка медицинских изображений. Одной из основных задач этого направления является распознавание отдельных органов и их патологий на биомедицинском изображении – сегментация. С использованием сегментации изображений можно проводить количественный анализ формы, размеров и объема органов, создавать обучающие выборки для искусственного интеллекта, позволяющего эффективно обрабатывать огромное количество медицинских данных.

На сегодняшний день повсеместно распространены заболевания дыхательной системы. Согласно данным статистического сборника здравоохранения в России от 2021 года [1], самым большим сегментом общей заболеваемости населения были и остаются болезни органов дыхания. Более того, с 2005 года по 2021 год процент этих заболеваний увеличился на 2,6% (с 24,2% до 26,8% от числа всех возможных заболеваний).

Согласно официальной статистике Всемирной организации здравоохранения (ВОЗ) [2], заболевания легких занимают одно из первых мест среди причин смерти в мире. Например, в 2019 году более 1,8 миллионов людей умерли от заболеваний легких. Среди них наиболее распространены хроническая обструктивная болезнь легких (ХОБЛ), рак легких и пневмония. Величина смертности от этих заболеваний может зависеть от различных факторов, таких как возраст, образ жизни, наличие других заболеваний, доступность качественной медицинской помощи и, конечно же, своевременность выявления патологии, постановка верного диагноза и незамедлительность начала необходимого лечения.

Для диагностики состояния легких зачастую используются изображения, сделанные с помощью компьютерной томографии. Плюсы и минусы данного метода будут приведены ниже в рамках данной выпускной квалификационной работы. Ключевым достоинством данного вида обследо-

ния грудной клетки, прежде всего, является высокая информативность об ее тканях и его способность выявлять заболевание на начальных стадиях.

В связи со всем вышеперечисленным, целью данной выпускной квалификационной работы является создание алгоритма, способного сегментировать легкие на снимках компьютерной томографии с высокой точностью. Данная разработка впоследствии позволит эффективно создавать обучающую выборку для нейронной сети, что будет крайне актуально в нынешних реалиях высокого процента заболеваний легких.

## 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Целями анализа предметной области является поиск эффективных методов обследования пациентов и наиболее точного метода сегментации изображений. Для достижения поставленных целей в данной главе были рассмотрены теоретические аспекты диагностики заболеваний легких, различные методы обработки цифровых изображений.

### 1.1 Методы диагностики легочных заболеваний в медицинских видеосистемах

В медицине в настоящее время используются различные методы для исследования легких, их можно поделить на три вида: лучевая, инструментальная и лабораторная диагностики [3]. Принцип работы первых двух видов диагностики построен на визуализации исследуемых органов, в то время как последний – отслеживает наличие заболевания, основываясь непосредственно на анализе образца биоматериала пациента.

В связи с тем, что целью нашей работы является разработка алгоритма, который будет помогать специалистам в постановке верного диагноза на начальных этапах заболевания – мы остановимся на методах лучевой диагностики. Они подразумевают под собой «просвечивание» грудной клетки, в то время как остальные виды диагностики используют инвазивные методы и применяются на более поздних сроках протекания болезни, что не совсем актуально для нашего исследования.

Рассмотрим три основных метода лучевой диагностики легких.

1. Рентгенография грудной клетки – быстрый и неинвазивный метод диагностики, который используется для обнаружения различных заболеваний и патологий, использующий электромагнитные лучи для создания изображения внутренних органов и тканей грудной клетки человека [4]. Существует всего два типа рентгенографии: аналоговая и цифровая.

При аналоговой рентгенографии рентгеновские лучи проходят через ткани и органы области исследования и попадают на рентгеночувствительную пленку, которая находится по другую сторону области исследования. В результате прохождения через образования разной плотности и состава пучок излучения рассеивается и тормозится, лучи попадают на пленку и воздействуют на фоточувствительный

слой внутри пленки, который содержит серебряные соли. Это приводит к тому, что на пленке образуются частицы серебра, формирующие изображение, разной интенсивности в зависимости от того, насколько сильно рассеялся пучок излучения при прохождении через разные ткани. Доза облучения при аналоговой рентгенографии зависит от многих факторов, таких как тип и мощность рентгеновской трубки, качество пленки, расстояние между объектом и источником излучения и т.д., но в среднем она составляет 0,3 мЗв.

В цифровой же рентгенографии используются датчики для преобразования рентгеновских лучей в цифровой сигнал. Датчики могут быть различных типов, например, фосфорных или полупроводниковых. Фосфорные датчики представляют собой слой фосфора, который преобразует рентгеновские лучи в световые вспышки, которые, в свою очередь, преобразуются в цифровой сигнал. Полученный сигнал затем передается на компьютер, где и формируется изображение. Доза облучения при цифровой рентгенографии обычно гораздо ниже, чем при аналоговой – примерно 0,03 мЗв, - так как данные системы могут использовать меньшую мощность рентгеновской трубки и короче время экспозиции. Однако данный метод требует наличия дорогостоящего оборудования и определенного уровня квалификации персонала, поэтому применяется не повсеместно.

Примеры аналоговой и цифровой рентгенографии приведены на рисунке 1 а) и б).



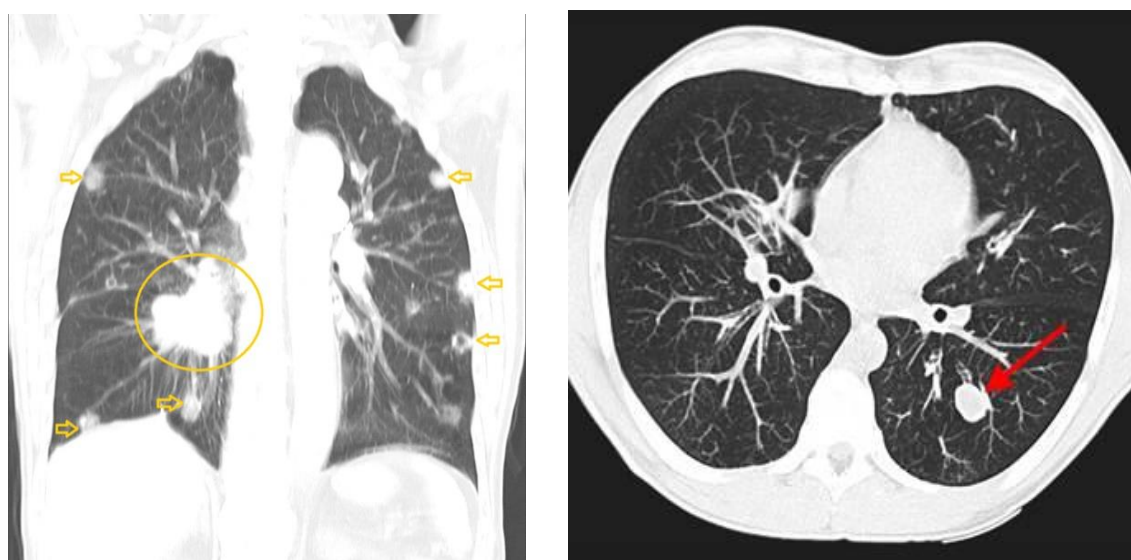
а) Снимок, полученный с помощью аналоговой рентгенографии; б) С помощью цифровой рентгенографии;

Рисунок 1 – Снимки грудной клетки



Рентгенография грудной клетки помогает выявлять в легких очаги воспаления, характерные для пневмонии и туберкулеза, наличие опухолевых образований в легких, пневмоторакса (механического разрыва легочной ткани). Но в некоторых случаях данный вид диагностики не информативен и демонстрирует ложно-отрицательный результат, что может быть обусловлено проекционным наложением тени патологического очага на тень нормальной анатомической структуры, малой интенсивностью очага воспаления (например, начальная стадия заболевания) и некорректной проекцией исследования.

2. Компьютерная томография (КТ) – диагностический метод, который использует рентгеновские лучи и компьютерную обработку изображений для создания трехмерной картины органов и тканей. КТ позволяет получить более точную и подробную картину, чем обычная рентгенография. КТ широко используется для обнаружения доброкачественных, злокачественных новообразований, метастазов, пневмонии, абсцессов, эмфизем легких, аномалий их развития и ряда других патологий [5]. На рисунке 2 приведены два примера отображения онкологического процесса на КТ-снимках.



а) Опухоль в правом легком и метастазы по краям; б) Опухоль в правом легком;

Рисунок 2 – Онкологический процесс на КТ-снимках

Принцип работы КТ [6] основан на использовании рентгеновских лучей, проходящих через тело пациента и регистрирующихся детекторами, расположенными на противоположной стороне от источника излучения. Пациент лежит на столе, который перемещается внутри кольцевой конструкции, содержащей высокочастотную рентгеновскую трубку и большое количество детекторов. Высокочастотная рентгеновская трубка посылает узкий пучок лучей через тело пациента, а детекторы регистрируют количество прошедших лучей. Каждый детектор измеряет различные уровни прохождения лучей через тело пациента, в зависимости от плотности тканей, которые они проходят. Эта информация используется компьютером для создания серии тонких срезов.

Каждый срез представляет собой изображение тонких поперечных слоев тела пациента, толщина каждого слоя составляет от 0,5 до 1 мм, что позволяет получить очень детальное изображение исследуемых тканей и органов. Затем полученные данные подвергаются компьютерной обработке с помощью специальных программ, которые строят трехмерную модель внутренней структуры легких (применительно к нашему случаю). Данные могут быть сохранены на жесткий диск или на другие носители информации и использоваться для последующей обработки и анализа. На рисунке 3 приведен пример проведения данной процедуры.



Рисунок 3 – Принцип работы компьютерной томографии

Для улучшения качества изображения и диагностической точности при КТ-сканировании легких может использоваться введение в организм пациента контрастирующих веществ, позволяющих более четко выделить опухоли, кисты и другие изменения в тканях легких и получить более понятную картину этих изменений.

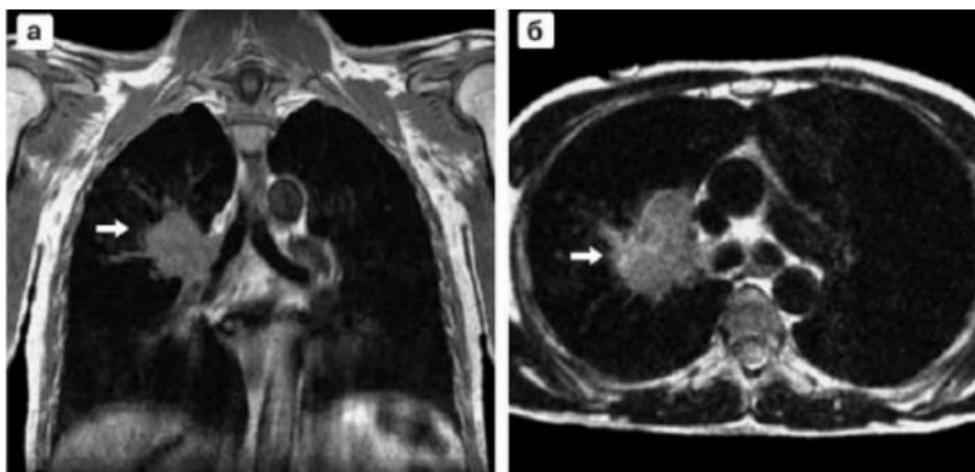
КТ-сканирование легких является безопасным и очень эффективным методом диагностики патологий легких, который позволяет получить очень детальное изображение внутренней структуры легких и обнаружить самые мелкие изменения в тканях и структурах. Однако, как и в любой медицинской процедуре, КТ-сканирование может иметь риски, включающие в себя высокую дозу облучения и редкую аллергическую реакцию на контрастирующие вещества. Поэтому перед проведением КТ-сканирования легких необходимо проконсультироваться с врачом.

Также стоит учитывать дороговизну данного вида обследования.

3. Магнитно-резонансная томография (МРТ) [7]. Принцип работы МРТ основан на использовании магнитного поля и радиоволн для создания детальных изображений внутренних структур тела. В основе метода лежит явление, называемое ядерным магнитным резонансом.

Внутри каждого атома есть ядро, содержащее протоны и нейтроны. Протоны, в свою очередь, имеют электрический заряд и вращаются вокруг своей оси. В магнитном поле протоны начинают вращаться в определенном направлении, а когда на них подается радиоволновой сигнал, протоны меняют свое движение и выходят из состояния равновесия. После прекращения подачи радиоволнового сигнала протоны возвращаются в свое исходное состояние и при этом испускают электромагнитные волны. Так, ядра атомов водорода в организме человека, если на них воздействуют радиочастотные импульсы в магнитном поле, дают особые эхо-сигналы, которые тут же регистрируются датчиками МРТ и используются для формирования изображения.

На Рисунке 4 а) и б) приведены изображения онкологического процесса на МРТ в разных проекциях.



а) фронтальная проекция; б) аксиальный срез; (стрелка указывает на новообразование в средней части легкого)

Рисунок 4 – Онкологический процесс на МРТ

Во многом, результаты, полученные с помощью КТ и МРТ, очень похожи. Оба метода хороши для распознавания опухолей, кист или инфекций, оба используются для оценки размера и формы легких. Но есть и различия.

Так, компьютерная томография обычно считается более точным и эффективным методом для обнаружения изменений в легочной ткани [8]. Это связано с возможностью КТ создавать более детальные изображения ткани, может использоваться для оценки ее плотности и обнаруживать в ней присутствие сторонних жидкостей, а также ярче распознавать наличие раковых опухолей (см. на рисунках 3 и 4).

МРТ же, в свою очередь лучше распознает наличие фиброза. Абсолютные противопоказания в случае компьютерной томографии достаточно малы: беременность или подозрение на нее, возраст до 18 лет (без наличия клинических показаний), масса тела пациента больше 130-140кг (в зависимости от аппарата).

Также магнитно-резонансная терапия, безусловно, наносит меньший вред организму, так как данный метод не использует облучение пациента радиационной дозой (как в КТ), а использует магнитные поля и волны. Однако данный вид диагностики также имеет свои противопоказания: болевой синдром у пациента, не позволяющий человеку находиться длительное время в неподвижном состоянии, наличие в теле металлических имплантатов, кардиостимулятора, вес паци-

ента более 120кг (для закрытых томографов), беременность (1 триместр) и др.

## 1.2 Классификация методов сегментации

На сегодняшний день существует множество самых разнообразных методов сегментации [9], которые объединяет одна цель – разбиение изображения на заданные области. Каждый из них предлагает свой подход к решению этой проблемы, но все их можно разделить по принципу работы на группы, представленные на рисунке 5.

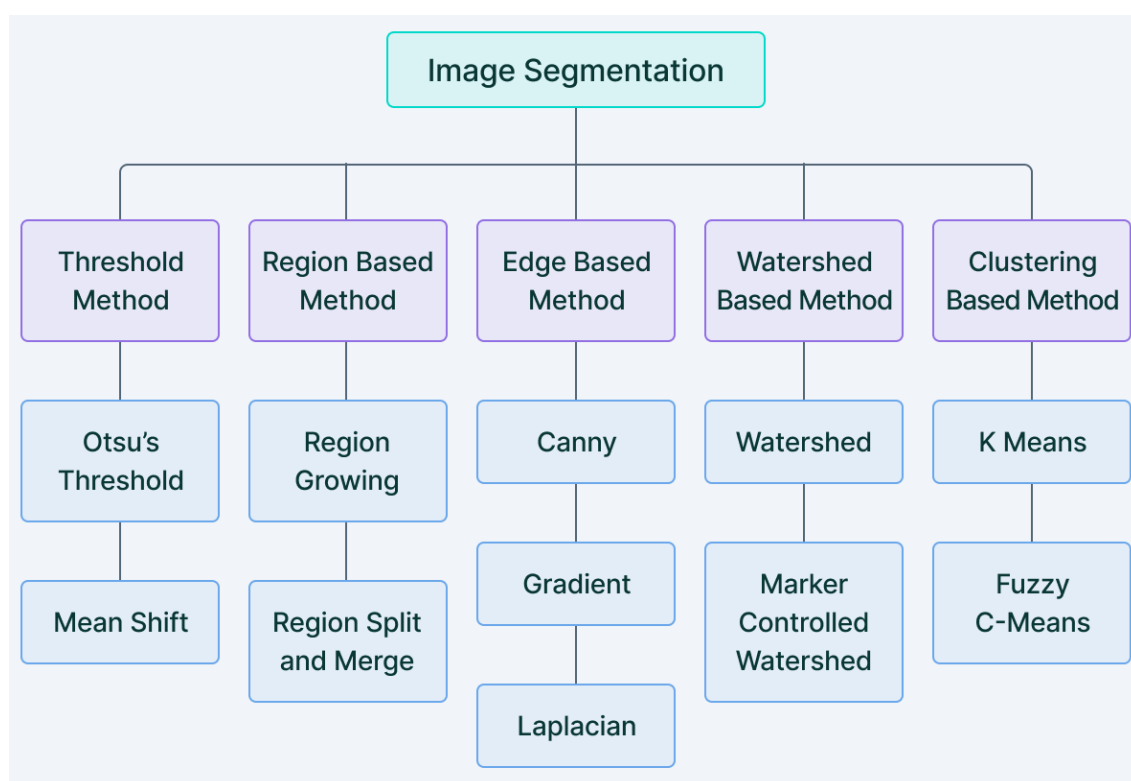


Рисунок 5 – Классификация методов сегментации изображений

Threshold Method (пороговые методы).

Пороговые методы [10] отличаются своей простотой и высокой вычислительной эффективностью в реализации, благодаря чему находят широкое применение в решении самых разнообразных задач. Пороговые критерии используются, так или иначе, во всех методах сегментации по яркости, контрастности или цветности.

Так, предположим, что гистограмма, изображенная на рисунке 6, соответствует некоторому изображению  $f(x, y)$ , которое содержит светлые объекты на темном фоне, из чего следует, что яркости пиксе-

лей фона и объекта сосредоточены близи двух преобладающих значений. Очевидно, что для сегментации объекта необходимо выбрать значение порога  $T$ , разграничивающего моды распределения яркостей.

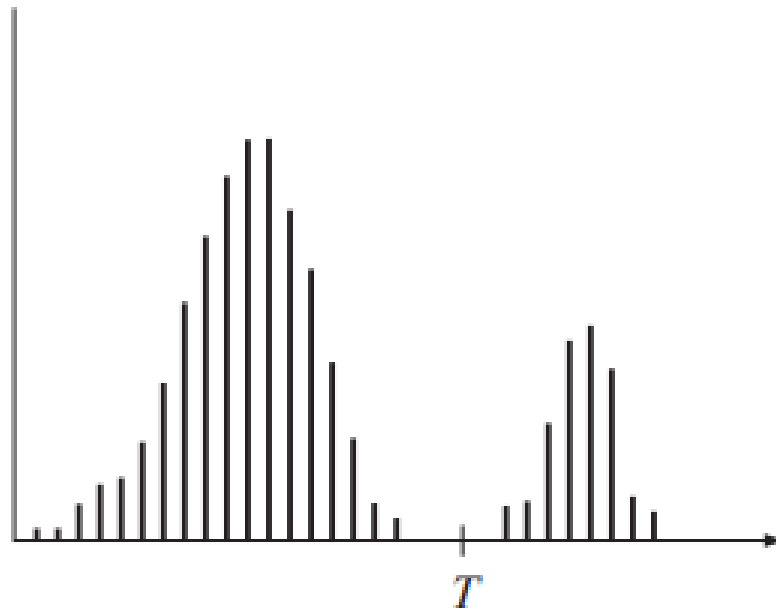


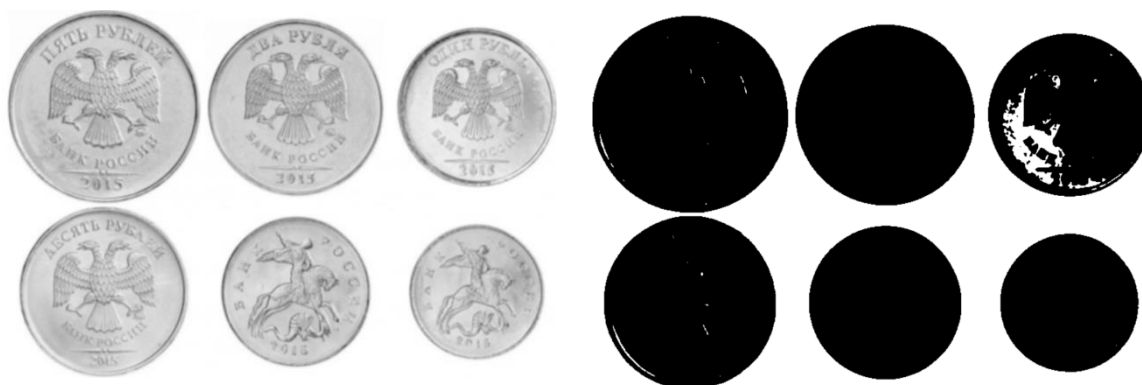
Рисунок 6 – Гистограмма яркости, допускающая разделение с помощью одного порога( $T$ )

Любая точка  $(x, y)$ , в которой  $f(x, y) > T$  – точка искомого объекта, в противном случае – точка фона. Таким образом, сегментированное изображение  $g(x, y)$  задается простейшим соотношением:

$$g(x, y) = \begin{cases} a, & f(x, y) > T; \\ b, & f(x, y) \leq T, \end{cases}$$

где  $f(x, y)$  - исходное изображение,  $T$  – пороговое значение,  $a \neq b$ .

Пример применения пороговых методов показан на рисунке 7.



а) исходное изображение; б) сегментированное изображение;

Рисунок 7 – Сегментация изображения пороговым методом,  $T = 230$ ,  $a = 255$ ,  $b = 0$

Region Based Method (сегментация на отдельные области).

Как говорилось выше, основная цель сегментации – разбиение изображения на области, и если в пороговом методе эта цель достигается с помощью пороговых преобразований, то этот основан на непосредственном поиске самих областей.

Так, например, метод Region Growing (выращивания областей), заключается в группировании пикселей или подобластей в более крупные области по заданным заранее критериям. Принцип метода состоит в том, что изначально берется множество точек, представляющих собой «центры кристаллизации», на которые в дальнейшем «наращиваются» соседние пиксели, которые по своим свойствам близки к центру кристаллизации (например, имеют яркость или цвет в определенном диапазоне). Процесс происходит до тех пор, пока не останется подобных пикселей.

Допустим, пусть есть начальный пиксель 2 в данном изображении и пороговое значение 3. Если значение пикселя меньше 3, то он будет рассматриваться внутри области начального пикселя, иначе – будет принадлежать другому региону. Иллюстрация данного примера приведена на рисунке 8.

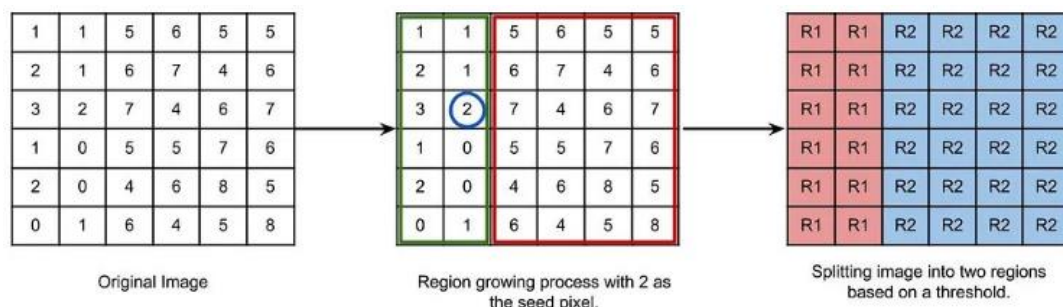


Рисунок 8 – Процесс «выращивания» области

Edge Based Segmentation (сегментация изображения на основе границ).

Основа метода заключается в поиске границ объекта [11], которые в свою очередь представляют собой резкие перепады в яркости или цвете на изображения. Для вычисления расположения границ используются операторы градиента. Так, например, для обнаружения яркостных перепадов в диагональных направлениях применяют пере-крестный градиентный оператор Робертса, вычисляющий разности пикселей по диагонали с помощью двумерных масок, изображенных на рисунке 9.

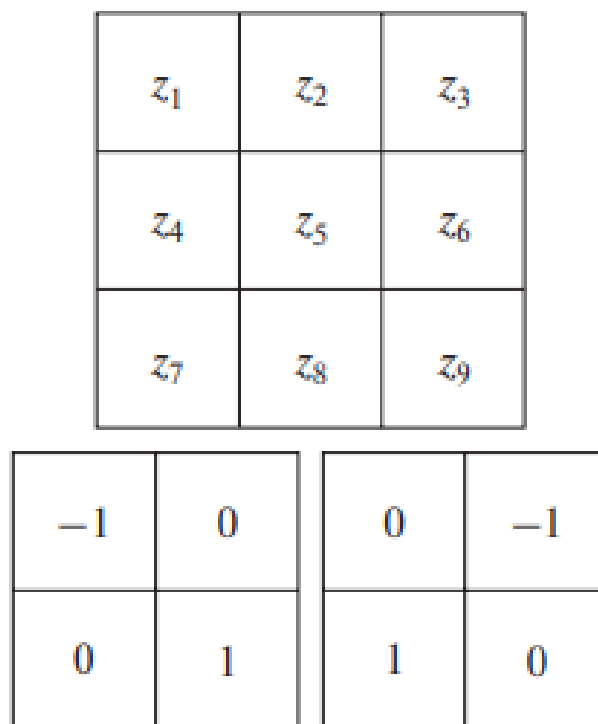


Рисунок 9 – Окрестность 3x3 внутри изображения и маски опе- ратора Робертса (переменные  $z_i$  – значения яркости)



Важно учитывать то, что перед поиском самих границ объектов необходимо выполнить несколько шагов обработки изображения. Это обосновывается тем, что каждый перепад можно представить в виде импульса, примеры представлены на рисунке 10:

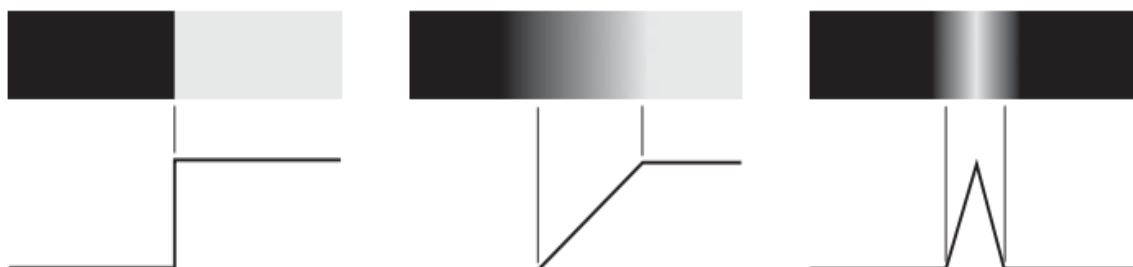


Рисунок 10 – Слева направо модели: идеального, наклонного перепадов, треугольного импульса и соответствующие им профили яркости

На рисунке представлены идеальные перепады, которые редко встречаются в реальной жизни. На практике изображения размыты и зашумлены. Это отражается на следующих этапах, когда возникает необходимость в вычислении первой и второй производных профиля яркости перепада исследуемого изображения (рисунок 11). Смысл в их вычислении прост – значение первой производной позволяет судить о наличии перепада, а знак второй – определить, на светлой или темной его стороне располагается пиксель. При проведении двух таких операций, шумы и резкость границ исходного изображения значительно увеличиваются, что продемонстрировано на рисунке 12:

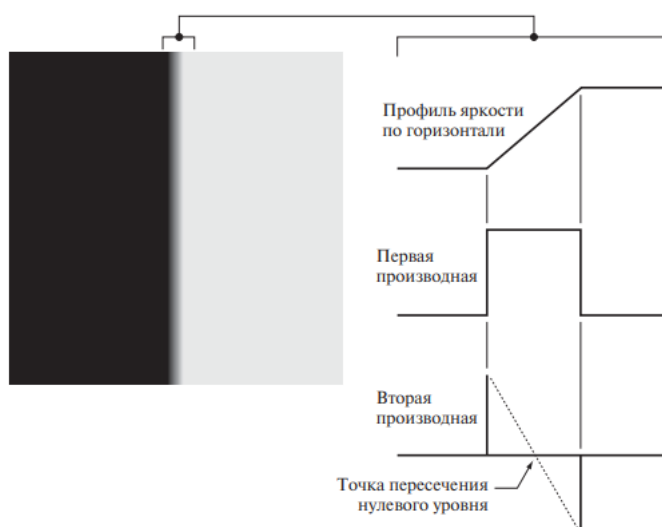
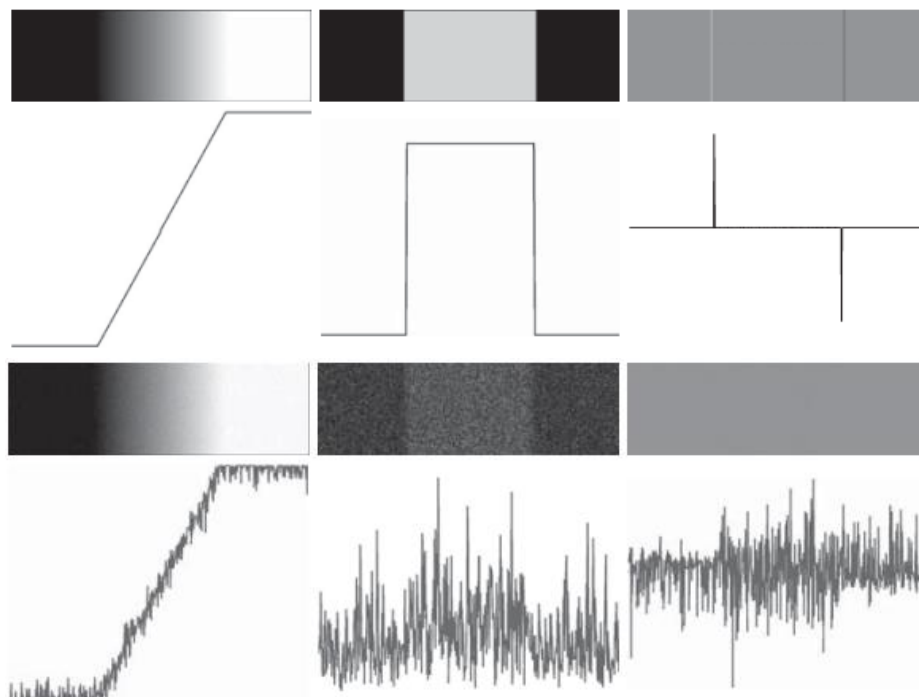


Рисунок 11 – Нахождение двух производных перепада



Первый столбец: изображения и профили яркости перепада, искаженного гауссовым шумом со средним 0 и  $\sigma=10$ . Второй столбец: изображения первой производной и их профили яркости. Третий столбец: изображения второй производной и их профили яркости

Рисунок 12 – Изображения и перепады

В связи с этим, перед поиском границ с помощью градиентных операторов необходимо сгладить изображение, чтобы уменьшить шумы, а после – выбрать нужные, соединить, чтобы получить замкнутые контуры вокруг объектов. После проведения этих процедур сегментированные области могут быть обработаны с учетом целей задачи: залиты цветом или отмечены метками, чтобы их было легче различать и анализировать.

#### Watershed Based Method (метод водораздела)

Метод водораздела – алгоритм обработки изображений, который применяется для сегментации изображений и выделения на них объектов.

Основной принцип работы метода заключается в использовании аналогии с водосборных площадей, где каждая ячейка изображения рассматривается как точка, принадлежащая определенному бассейну. В этом случае, ячейки изображения с высокими яркостями и низкими интенсивностями рассматриваются как вершины холмов, а ячейки с

низкими значениями яркости и высокими интенсивностями – как области долин.

Процесс сегментации изображения начинается с разбиения изображения на набор мелких регионов. Затем каждый такой регион рассматривается как точка, которая принадлежит определенному бассейну. Для этого строится граф, где каждая такая точка представлен в виде вершины, а ребра – расстояния между соседними вершинами.

Далее применяется алгоритм поиска локальных минимумов на графе, которые определяют точки начала долин. Эти точки являются местами, где «вода» собирается при «затоплении». Далее, из локальных минимумов начинается распространение воды во все направления до тех пор, пока она не достигнет границы бассейна, при этом каждая точка должна принадлежать только одному бассейну.

Чтобы вода в бассейне не смешивалась, устанавливаются так называемые «перегородки», которые регулируют, чтобы в каждом бассейне был лишь один локальный минимум. Наглядное изображение процесса сегментации с помощью данного метода приведено на рисунках 13 и 14.

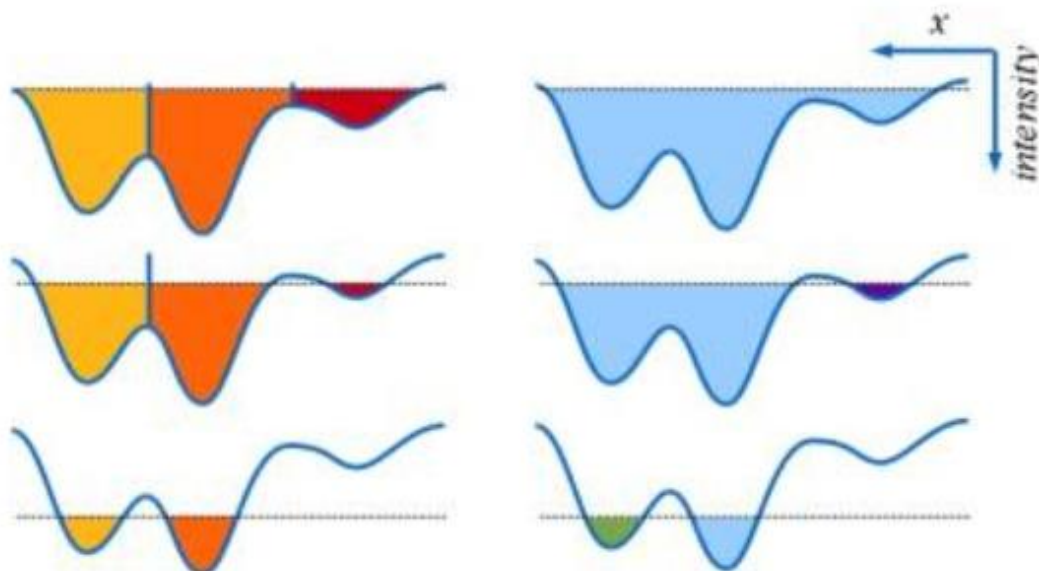


Рисунок 13 – Принцип работы метода Watershed

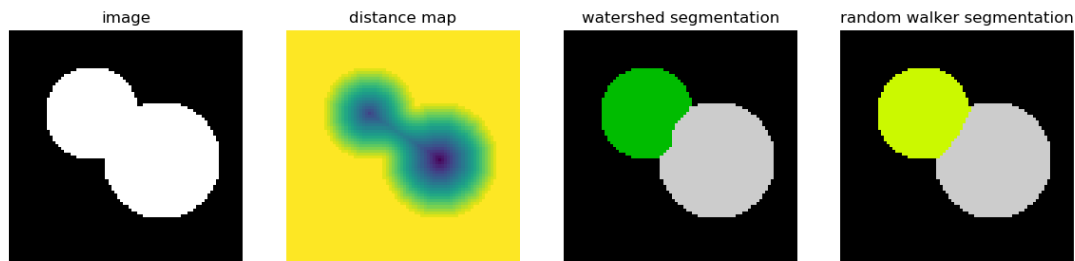


Рисунок 14 – Этапы реализации метода водоразделов

Результат работы алгоритма представляет собой маски найденных сегментов.

#### Clustering Based Method (сегментация на основе кластеризации)

Кластеризация – тип алгоритма неконтролируемого машинного обучения. Он часто используется для сегментации изображений. Одним из наиболее популярных алгоритмов на основе кластеризации, используемых для сегментации, является кластеризация К-средних (K-means clustering), на примере которой в данной работе и будет рассмотрен принцип работы данной группы методов.

Для лучшей визуализации был выбран двумерный набор данных. В наборе данных случайным образом инициализируются центроиды, после чего вычисляется расстояние от всех точек до всех кластеров и точка причисляется к ближайшему кластеру (до которого расстояние меньше). Затем центроиды всех кластеров пересчитываются, принимая среднее значение этого кластера в качестве центроида. Процедура повторяется. Процесс продолжается до тех пор, пока алгоритм не придет к оптимальному решению (рисунок 15).

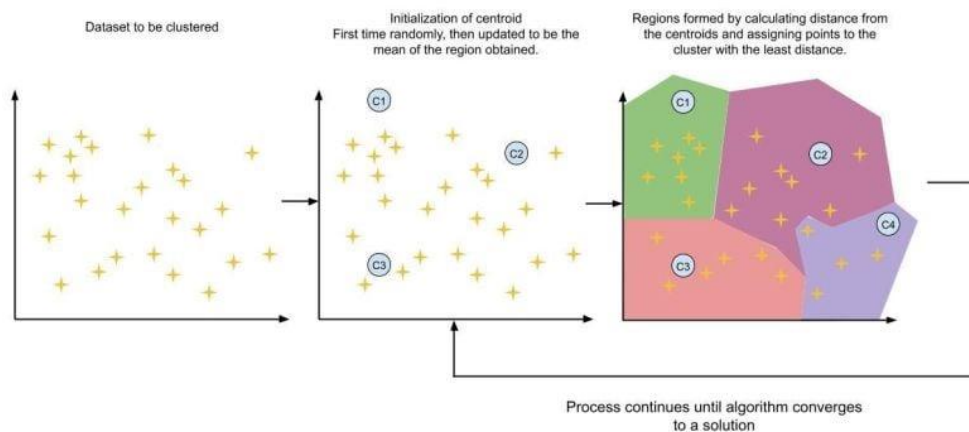


Рисунок 15 – Процесс кластеризации при реализации метода К-средних

В зависимости от количества кластеров получим разные результаты, представленные на рисунке 16.

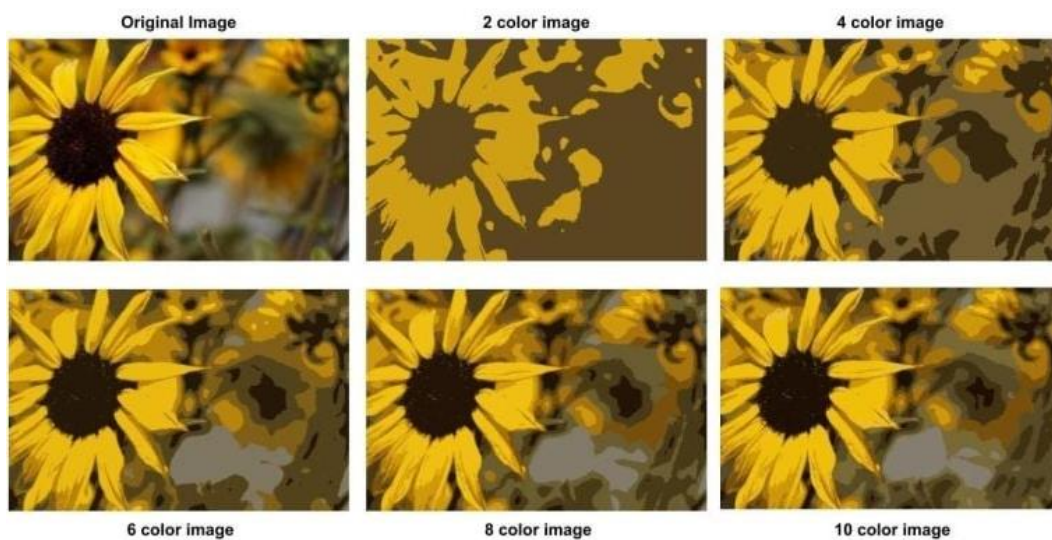


Рисунок 16 – Полученные результаты

### 1.3 Сравнительный анализ методов сегментации

Каждая из рассмотренных ранее групп сегментации, безусловно, подходит идеально для решения своей области задач, исходя из своих достоинств и недостатков.

Все преимущества и недостатки описанных методов были собраны и представлены в виде таблицы 1.

Сравнительная таблица методов сегментации Таблица 1

| Метод           | Преимущества  | Недостатки   |
|-----------------|---|--|
| 1. Threshold    | <ul style="list-style-type: none"><li>• Простота реализации;</li><li>• Эффективен для изображений с простой структурой и без шума;</li></ul>  | <ul style="list-style-type: none"><li>• Неэффективен для изображений с шумом или сложной структурой;</li><li>• Требуется выбора правильного порога, что может быть сложным для изображений с разной освещенностью;</li></ul> |
| 2. Region based | <ul style="list-style-type: none"><li>• Эффективен для изображений со сложными структурами;</li><li>• Позволяет получить группы пикселей с одинаковыми свойствами, что облегчает дальнейший анализ;</li></ul>   | <ul style="list-style-type: none"><li>• Требуется больших вычислительных ресурсов;</li><li>• Неэффективен для изображений с малым количеством объектов или простой структурой;</li></ul>                                     |
| 3. Watershed    | <ul style="list-style-type: none"><li>• Позволяет сегментировать изображение на основе естественных границ между объектами;</li><li>• Позволяет сохранить естественные границы изображения;</li><li>• Эффективен для сегментации изображений со сложной структурой;</li><li>• Позволяет сегментировать соприкасающиеся области;</li></ul> | <ul style="list-style-type: none"><li>• Требователен к вычислительным ресурсам;</li></ul>  |

Таблица 1 (продолжение)

| Метод               | Преимущества   | Недостатки  |
|---------------------|--|---|
| 4. Edge based       | <ul style="list-style-type: none"> <li>• Эффективен для сегментации изображений с четкими границами объектов;</li> <li>• Точность сегментации;</li> <li>• Прост в реализации;</li> </ul> | <ul style="list-style-type: none"> <li>• Может быть чувствителен к шуму;</li> </ul>   |
| 5. Clustering Based | <ul style="list-style-type: none"> <li>• Прост в реализации;</li> <li>• Эффективен для сегментации изображений со сложной структурой;</li> </ul>   | <ul style="list-style-type: none"> <li>• Неэффективен для изображений с нечеткими границами;</li> <li>• Количество кластеров должно быть определено заранее, что может быть сложным для большой выборки изображений;</li> </ul> |

#### 1.4 Выводы по главе

Поиск алгоритма, способного эффективно и точно сегментировать легкие на снимках компьютерной томографии является актуальной задачей в современной науке.

В данной выпускной квалификационной работе будет предложено одно из возможных решений данной проблемы и проведена оценка полученных результатов.

## 2. ПОДГОТОВКА ДАННЫХ И РАЗРАБОТКА АЛГОРИТМА

В данном разделе была изучена и выполнена предварительная подготовка данных, которые будут использоваться для тестирования программы, было проведено исследование особенностей работы со снимками компьютерной томографии. На основе выбранного метода водораздела также реализован и описан алгоритм, позволяющий сегментировать легкие.

### 2.1 Подготовка исходных данных

Тренировочные данные представлены наборами кт-снимков, загруженными с сервиса Kaggle [12] и представленными в формате DICOM.

Формат DICOM (Digital Imaging and Communications in Medicine) – стандарт, который используется в медицинской диагностике для хранения, передачи и обработки медицинских изображений и сопутствующей информации, такой как текстовые описания, результаты измерений и т.д. Данный формат имеет множество преимуществ, включая стандартизацию медицинской информации, возможность обработки и передачи медицинских изображений между различными устройствами и системами, конфиденциальность и безопасность данных пациента, а также обеспечение точности и надежности медицинской информации.

Файлы DICOM содержат двоичные данные изображения, а также метаданные, которые описывают пациента, тип и характеристики изображения, условия его получения. Метаданные хранятся в виде тэгов, каждый из которых имеет уникальный идентификатор и значение (рисунок 17).

```
Dataset.file_meta -----
(0002, 0000) File Meta Information Group Length  UL: 196
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID        UI: CT Image Storage
(0002, 0003) Media Storage SOP Instance UID     UI: 1.3.6.1.4.1.14519.5.2.1.7085.2626.214140401149739061368142942055
(0002, 0010) Transfer Syntax UID               UI: Explicit VR Little Endian
(0002, 0012) Implementation Class UID          UI: 1.2.40.0.13.1.1.1
(0002, 0013) Implementation Version Name       SH: 'dcm4che-1.4.35'
-----
(0008, 0005) Specific Character Set             CS: 'ISO_IR 100'
(0008, 0008) Image Type                        CS: ['ORIGINAL', 'PRIMARY', 'AXIAL', 'CT_SOM5 SPI']
(0008, 0016) SOP Class UID                     UI: CT Image Storage
(0008, 0018) SOP Instance UID                   UI: 1.3.6.1.4.1.14519.5.2.1.7085.2626.214140401149739061368142942055
(0008, 0020) Study Date                        DA: '20100227'
(0008, 0021) Series Date                       DA: '20100227'
(0008, 0022) Acquisition Date                  DA: '20100227'
(0008, 0023) Content Date                     DA: '20100227'
(0008, 0030) Study Time                        TM: '161937.171'
(0008, 0031) Series Time                       TM: '162536.14'
(0008, 0032) Acquisition Time                  TM: '162203.028699'
(0008, 0033) Content Time                     TM: '162203.028699'
(0008, 0050) Accession Number                  SH: '1598252606449858'
(0008, 0060) Modality                          CS: 'CT'
(0008, 0070) Manufacturer                     LO: 'SIEMENS'
(0008, 0090) Referring Physician's Name       PN: ''
(0008, 1030) Study Description                 LO: 'CT CHEST W IV CONTRAST'
(0008, 103e) Series Description                LO: 'LUNG 3.0 870f'
(0008, 1090) Manufacturer's Model Name        LO: 'Sensation 16'
(0008, 1110) Referenced Study Sequence 1 item(s) ----
(0008, 1150) Referenced SOP Class UID         UI: Detached Study Management SOP Class
(0008, 1155) Referenced SOP Instance UID      UI: 1.3.6.1.4.1.14519.5.2.1.7085.2626.369429505691935722415265978954
-----
(0008, 1140) Referenced Image Sequence 1 item(s) ----
(0008, 1150) Referenced SOP Class UID         UI: CT Image Storage
```

Рисунок 17 – Представление метаданных формата DICOM



Для работы с данными формата DICOM используется инструмент `pydicom` [13], который позволяет считывать файлы и работать с ними в естественных структурах языка Python. Изображения были считаны и представлены в виде матриц чисел при помощи этого инструмента.

## 2.2 Среда разработки и программные средства моделирования

Для реализации алгоритма данной работы в качестве среды разработки был выбран Visual Studio Code. Данный текстовый редактор – мощный и удобный инструмент для написания программ, обладающий приятным интерфейсом, большим количеством расширений и функций, что позволяет значительно упростить и ускорить процесс разработки.

В качестве средства разработки был выбран язык Python. Данный язык программирования имеет обширную экосистему библиотек, которые используются для решения задач компьютерного зрения. Еще одним преимуществом Python является простота и читабельность кода – простой и лаконичный синтаксис делает ревью кода значительно проще. Данный язык программирования распространен и доступен, имеет обширную поддержку и качественную документацию.

В данной работе, инструменты Python обеспечат анализ и подготовку исходных данных, обработку большой выборки изображений, вывод полученных результатов на экран.

## 2.3 Описание алгоритма

Предполагается разбиение алгоритма на несколько основных этапов: осуществление загрузки изображения среза из указанной директории пациента, распаковка изображения формата DICOM, предобработка изображения, реализация алгоритма сегментации для поиска легких с помощью метода водораздела и сохранение полученного изображения в новую директорию.

На основе перечисленных этапов была разработана структурная схема алгоритма, представленная на рисунке 18.



Рисунок 18 – Структурная схема алгоритма

## Загрузка изображений срезов

Прежде всего, необходимо обеспечить загрузку исходных изображений в программу для последующей обработки. Для этого используется модуль `os` [14], предоставляющий множество функций для взаимодействия с операционной системой и позволяющий создавать, удалять, перемещать и переименовывать файлы и директории, а также манипулировать переменными окружения, работать с процессами и т.д.

В данной работе используется функция `.listdir()`, возвращающая список файлов и директорий в указанной директории. С помощью этой функции мы считываем и создаем список директорий, в которых хранятся срезы каждого пациента, после чего производится их сортировка для удобства дальнейшей работы.

## Чтение файла DICOM

Как говорилось ранее, файлы формата DICOM хранят не только само изображение, но и большое количество метаданных. При задействовании `pydicom.read_file()` из специализированного модуля `pydicom` загруженный файл представляется в виде словаря (рисунок 19) [15].

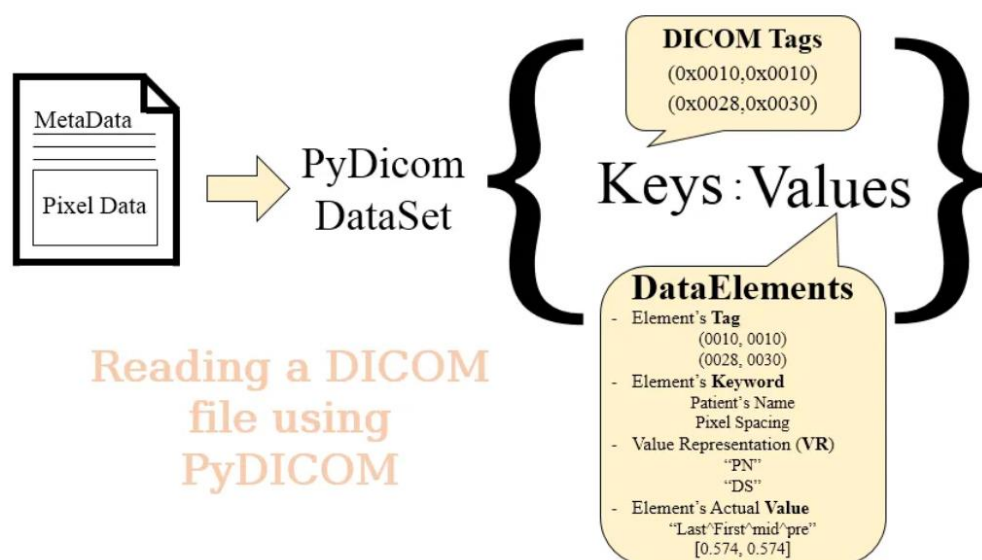


Рисунок 19 – Чтение файла с помощью модуля `pydicom`

Ключами словаря являются теги атрибутов DICOM, указанных в читаемом файле. Примеры таких ключей: `(0x0010, 0x0010)` атрибут – PatientName или `(0x7fe0, 0x0010)` атрибут – PixelData. Сами теги состоят из

двух шестнадцатеричных чисел: первое указывает на группу, а второе относится к конкретному элементу.

Значения словаря обычно содержат самую разнообразную информацию: ключевое слово, указание на конкретный элемент или то, какой тип данных и формат значения имеет атрибут.

Самым важным атрибутом является `.pixel_array`, который извлекает массив значений пикселей изображения и предоставляет возможность работы с изображением, например, вывод на экран (рисунок 20). Важно отметить, что значения этого массива имеют тип `uint16`, т.е. содержат целые числа от 0 до 65 535.



Рисунок 20 – Открытое исходное изображение формата DICOM

#### Пересчет значений пикселей в единицы Хаунсфилда

Числовые значения пикселей, из которых состоит изображение в данном формате, представляют собой плотности ткани в конкретной точке, что далее необходимо для визуализации и анализа полученного результата. При КТ-сканировании значения точек удобно преобразовывать считанные значения плотностей в единицы Хаунсфилда[16].

Единицы Хаунсфилда (HU) – шкала, используемая в медицинской диагностике для измерения плотности тканей тела человека на основе значений пикселей в медицинских изображениях (рисунок 21). Она была создана британским инженером Годри Хаунсфилдом. Данная шкала основана на сравнении плотности человеческих тканей с плотностью дистиллированной воды. Значение 0 HU соответствует плотности воды, отрицательные значения соответствуют тканям, имеющим меньшую плотность (воздух), а положительные – значениям выше (костная ткань).

| <b>Material</b> | <b>Hounsfield Unit</b> |
|-----------------|------------------------|
| Air             | -1000                  |
| Lung            | -500 to -200           |
| Fat             | -200 to -50            |
| Water           | 0                      |
| Blood           | 25                     |
| Muscle          | 25 to 40               |
| Bone            | 200 to 1000            |

Рисунок 21 – Таблица соответствия

Для получения коэффициентов Хаунсфилда необходимо выполнить следующее преобразование:

$$HU = mP + b,$$

где  $P$  – массив значений пикселей,  $m$  – масштабный коэффициент (Rescale Slope),  $b$  – смещение (Rescale Intercept).

Rescale Slope и Rescale Intercept – атрибуты DICOM-изображений, которые используются для коррекции и преобразования значения пикселей, чтобы они соответствовали значениям HU, представляющим собой меру плотностей тканей. Данные значения извлекаются с помощью модуля `pydicom` и соответствующих методов.

## Нахождение маркеров водораздела

Прежде всего, были найдены «внутренние» маркеры, представляющие собой бинарные маски и соответствующие областям легких, с помощью значений шкалы Хаунсфилда – определившим полости органов дыхания. Также были задействованы модули `segmentation` и `measure` из библиотеки `skimage` [17] для уменьшения шумов создаваемой маски и для создания меток связанных областей соответственно.

Создание «внешних» маркеров (не относящихся к областям интереса) основывалось на принципе дилатации контуров внутренних маркеров с помощью методов библиотеки `scipy.ndimage`.

Результаты создания маркеров представлены на рисунке 22.



Рисунок 22 – Найденные маркеры

## Реализация водораздела

Сначала выполняется предобработка изображения с помощью фильтра Собела – оператора, который используется для нахождения градиента яркости в изображении. Он состоит из двух ядер, которые применяются к изображению для вычисления производных в направлениях  $x$  и  $y$ .

Градиент Собела относится к методам сегментации изображения на основе границ. Он позволяет выделить контуры объектов на изображении, что поможет точнее отобразить границы сегментируемых легких на результирующем изображении.

Для вычисления производной яркости в направлении  $x$  (горизонтальное) на изображении используется модуль `scipy` [18] и метод `.ndimage.sobel`, который применяет оператор Собела к переданному ей

изображению. Далее с помощью тех же инструментов реализуется вычисление производной яркости в направлении  $y$ .

Затем вычисляется общая интенсивность градиента. Для этого находим квадраты производных по  $x$  и  $y$ , затем складываем их и вычисляем квадратный корень. В этом помогает функция `np.hypot()` из библиотеки NumPy [19], которая позволяет реализовать данные операции для двух массивов и вернуть новый массив, содержащий общую интенсивность градиента яркости.

После производится нормализация значений пикселей так, чтобы они были в диапазоне от 0 до 255. Для этого каждый пиксель делится на максимальное значение в новом массиве и умножается на 255. Данный шаг позволяет привести значения пикселей к диапазону, удобному для дальнейшей обработки и визуализации. На выходе имеется бинарное изображение, в котором только границы объектов остаются белыми, а остальное изображение становится черным фоном. Результат представлен на рисунке 23.



Рисунок 23 – Результат применения фильтра Собела

Конечным шагом на этом этапе является вызов метода `.watershed()` модуля `skimage`. Данный метод «заливает водой» заготовленную форму – размеченное маркерами изображение. В результате имеем бинарную маску с выделенными легкими. Результат представлен на рисунке 24.

Watershed

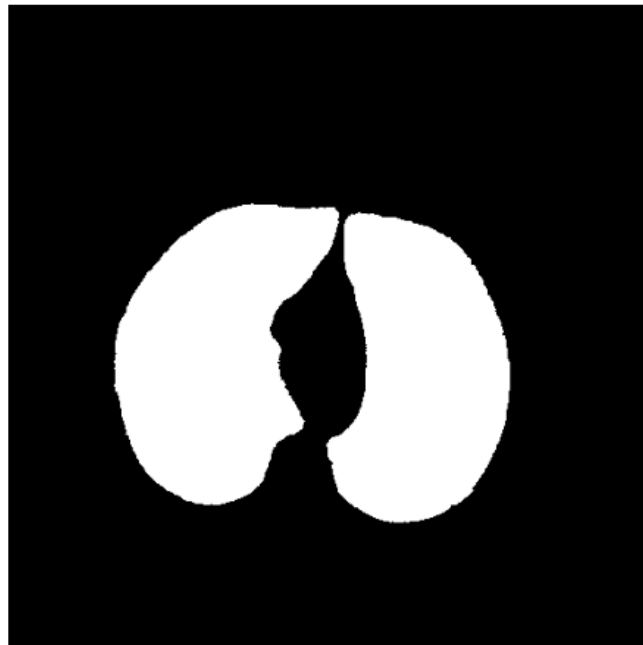


Рисунок 24 – Сегментированные легкие

Для большей наглядности представим полученный результат в ином виде (рисунок 25):

Result 1

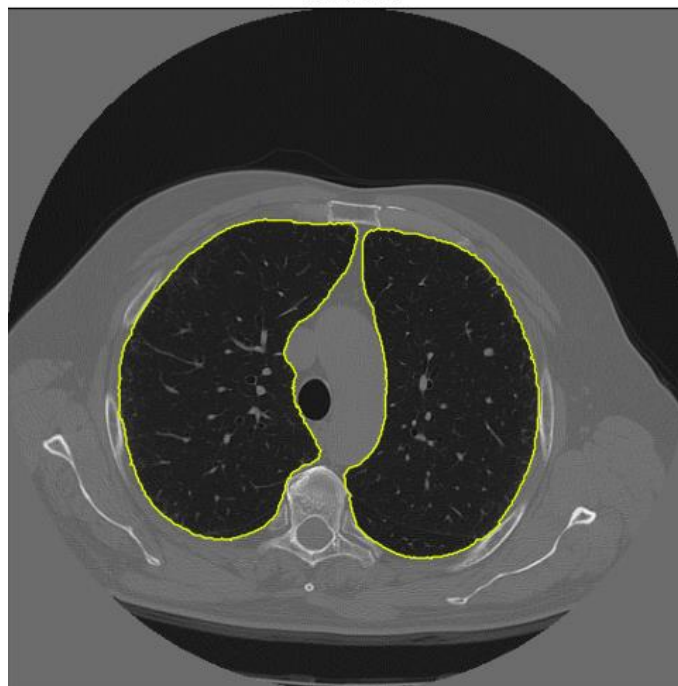


Рисунок 25 – Результат выполнения алгоритма водораздела



## Сохранение результата

На последнем этапе сохраняем полученное в результате выполнения алгоритма изображение в новую папку для дальнейшего анализа результатов. Для этого используется библиотека `cv2` и метод `.imwrite()`.

## 2.4 Выводы по главе

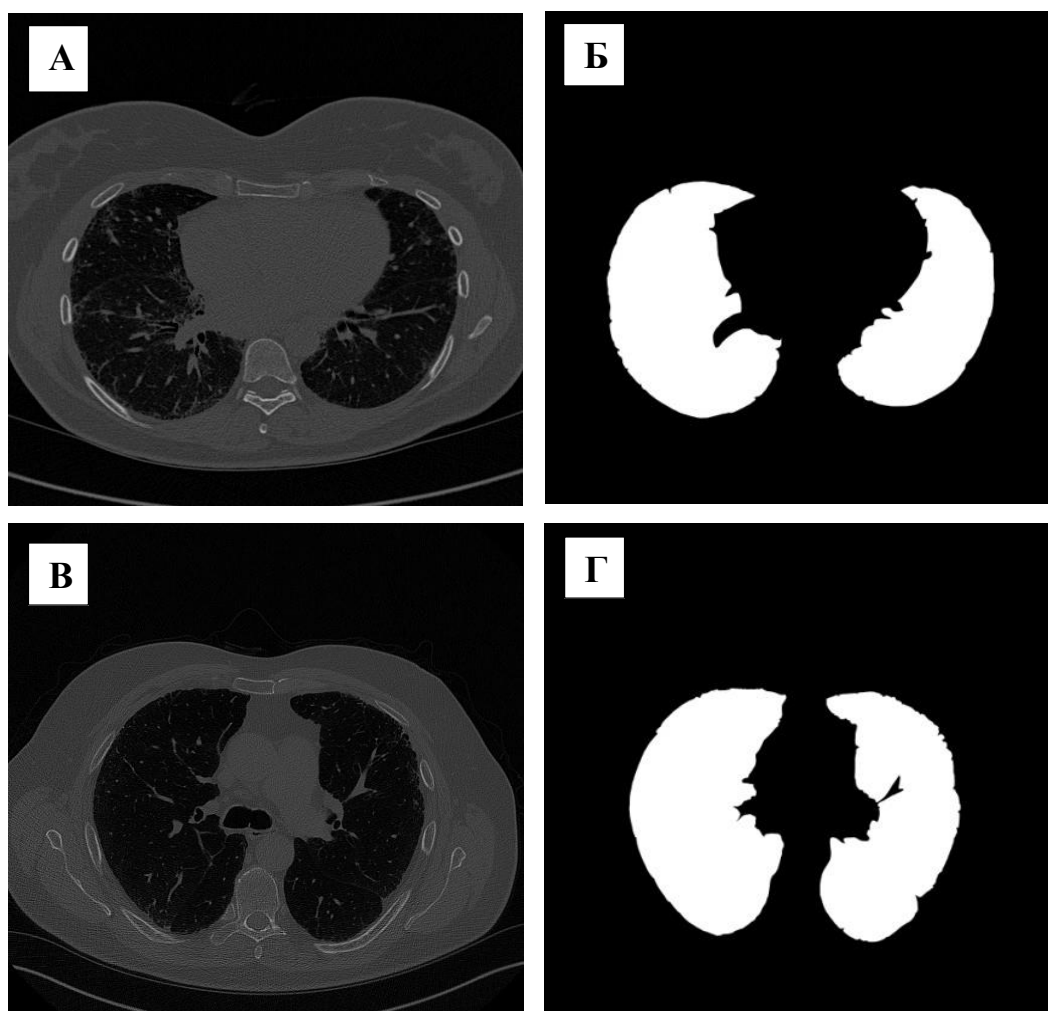
На базе языка программирования Python была разработан алгоритм, сегментирующий легкие на снимках компьютерной томографии на основе метода водораздела, способного работать с изображениями со сложной структурой и при этом сохранять естественные границы сегментированной области.

Работа проводилась с изображениями формата DICOM, получаемого непосредственно с КТ-сканера, чтобы избежать ненужных операций и ускорить процесс сегментирования

### 3. АНАЛИЗ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА

#### Наборы данных

Для моделирования были использованы два набора данных, найденных на сервисе Kaggle. Один из них состоит из 100 снимков компьютерной томографии двух пациентов, представленных в формате DICOM(см.2.1). Второй набор данных включает в себя 100 размеченных врачами «эталонных» изображений легких формата JPEG, полученных из первого набора данных и представленных в виде бинарных масок. Каждое изображение имеет фиксированное разрешение 512x512 пикселей. Примеры элементов двух датасетов представлены на рисунке 26.



а) Снимок КТ первого пациента; б) Бинарная маска легких со снимка а); в) Снимок КТ второго пациента; г) Бинарная маска легких со снимка в).

Рисунок 26 – Снимки компьютерной томографии и их бинарные маски

## Оценка точности разработанного алгоритма

Для оценки точности сегментации легких с помощью алгоритма, разработанного в рамках данной выпускной квалификационной работы, использовалась мера Дайса [21].

Коэффициент Дайса, также известный как мера Дайса (Dice coefficient) используется для оценки сходства между двумя множествами. В медицинской практике его широко применяют для оценки точности алгоритмов сегментации изображений.

Коэффициент Дайса определяется как отношение удвоенного пересечения множества к их сумме. Данная формула выглядит следующим образом:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|},$$

где  $X$  и  $Y$  – два множества, которые сравниваются (в данном случае, два сегментированных изображения),  $|X|$  и  $|Y|$  – количество элементов в каждом из множеств,  $|X \cap Y|$  – количество элементов, принадлежащих обоим множествам одновременно.

Значение коэффициента Дайса может изменяться в пределах от 0 до 1. Если коэффициент равен 1, то это означает, что два множества полностью совпадают, а если коэффициент равен 0 – множества не имеют общих элементов. Чем ближе коэффициент Дайса к 1, тем более точным считается алгоритм сегментации.

Данная мера имеет ряд преимуществ перед другими критериями сходства, так как она более чувствительна к небольшим различиям между множествами, что особенно важно для задач сегментации изображений. Кроме того, она не зависит от размера множеств, что позволяет сравнивать между собой данные различных размеров.

Для демонстрации точности работы алгоритма были взяты первые 20 снимков из наборов первого и второго пациентов.

На рисунке 27 представлены результаты оценки точности для набора снимков компьютерной томографии пациента №1. На рисунке 28 – для набора снимков пациента №2.

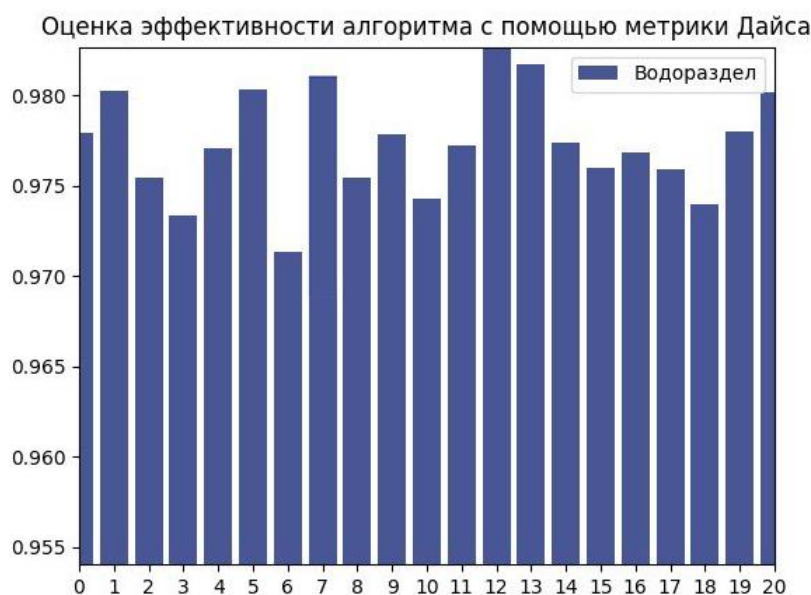


Рисунок 27 – Диаграмма зависимости коэффициентов Дайса от номера снимка для набора пациента №1

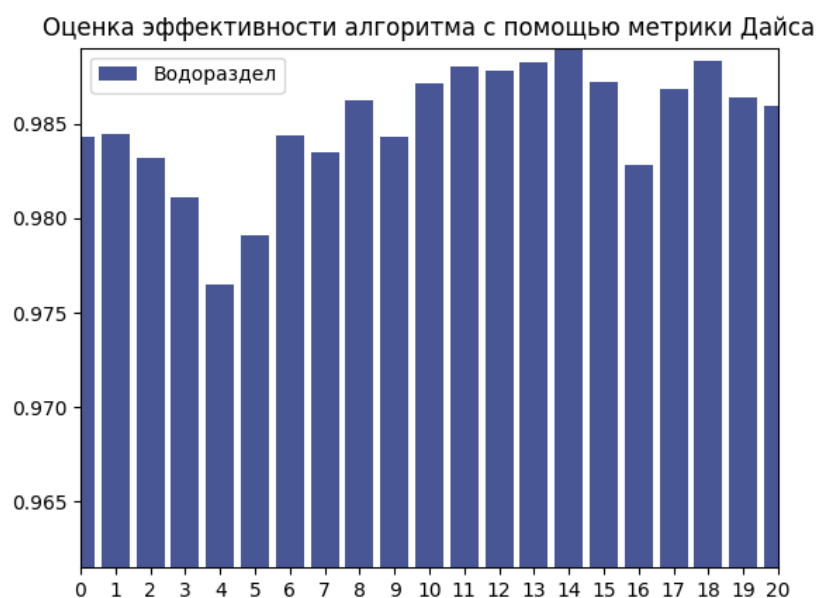
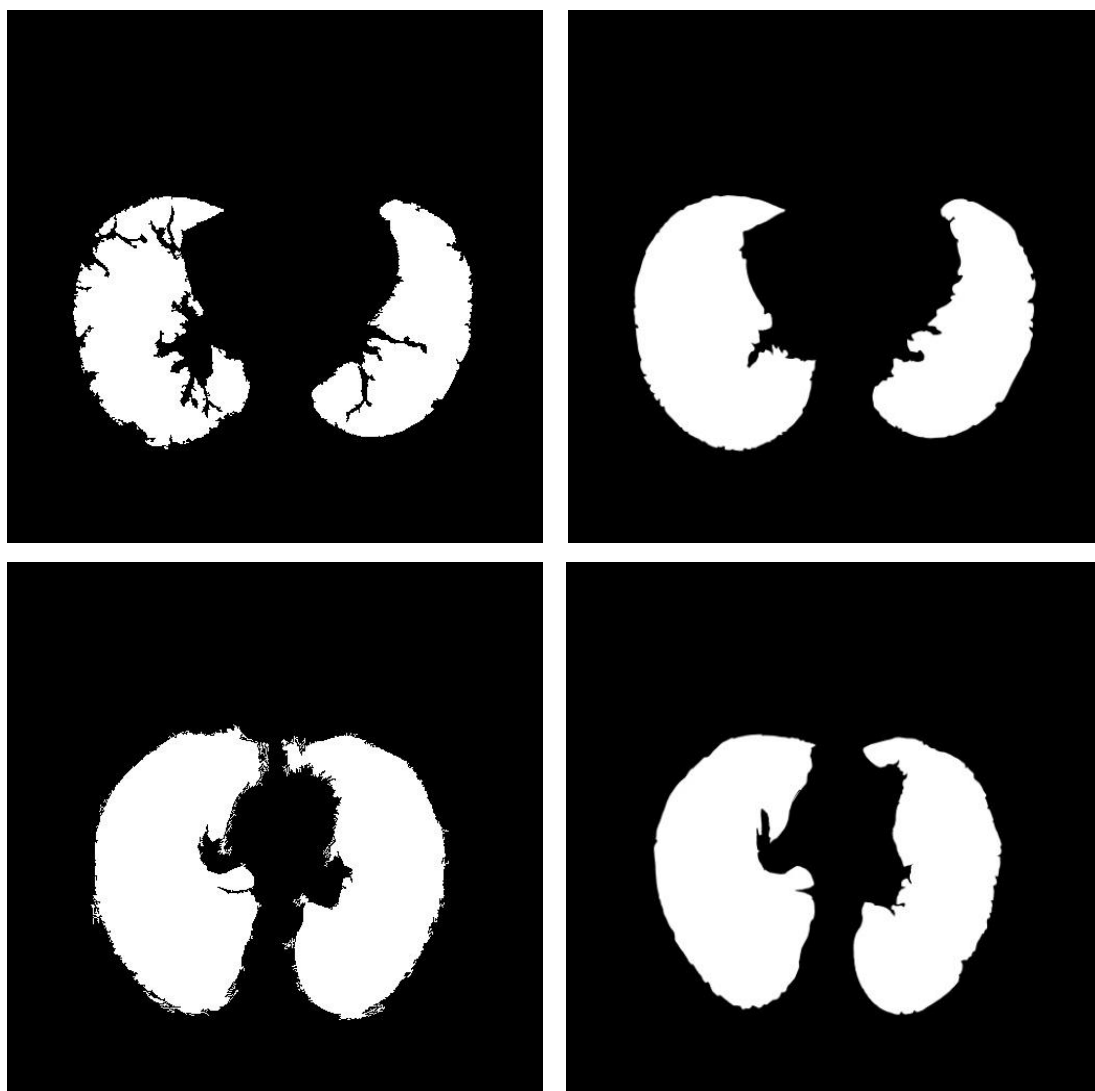


Рисунок 28 – Диаграмма зависимости коэффициентов Дайса от номера снимка для набора пациента №2

Более высокая точность сегментации на втором наборе данных объясняется четкостью сосудистой системы на изображениях, что помогает алгоритму лучше считывать контуры самих легких. В среднем, точность сегментации по первому набору для пятидесяти снимков составляет 97,9%, а по второму – 98,6%.

### Сравнение с распространенным аналогом

Для оценки эффективности разработанного алгоритма было выполнено сравнение результатов его работы с работой одного из его распространенных аналогов на базе метода edge based segmentation (сегментации на основе границ). Выделение границ на исходном изображении в данном случае осуществлялось с помощью пороговой бинаризации и выделения контуров по связности [22]. Полученные результаты и сравнение с «эталонными» изображениями представлены на рисунке 29.

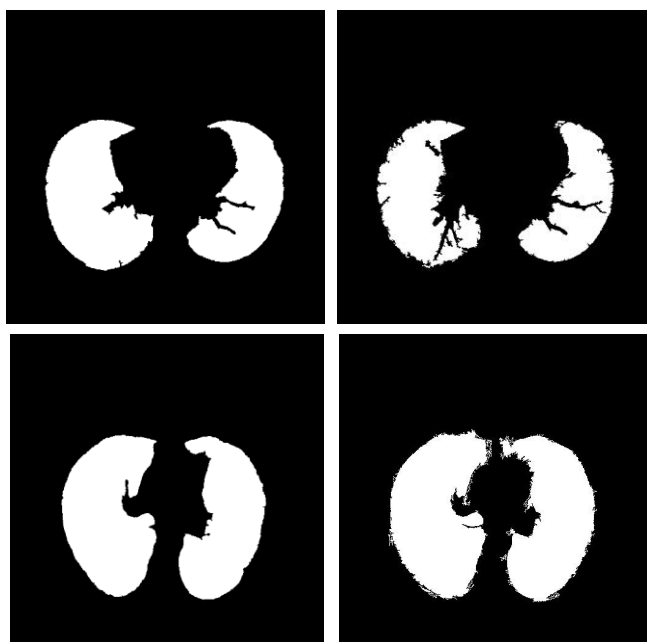


Слева: маски легких, полученные с помощью метода поиска контуров;

Справа: эталонные изображения

Рисунок 29 – Маски сегментированных легких

Также приведем изображения результатов двух алгоритмов для сравнения. Наиболее удачные примеры для двух пациентов представлены на рисунке 30, а неудачные – на рисунке 31.



Слева: результаты работы нашего алгоритма на основе водораздела;

Справа: Результаты работы аналога;

Рисунок 30 – Результаты работы двух алгоритмов для наборов двух пациентов



Слева: результаты работы нашего алгоритма на основе водораздела;

Справа: Результаты работы аналога;

Рисунок 31 – Результаты работы двух алгоритмов для наборов двух пациентов

Оценим точность сегментации с помощью данного метода, воспользовавшись метрикой Дайса. Для первого набора данных (50 изображений) точность составляет 92%, а для второго – 94,3%.

Проиллюстрируем разницу в оценке точности алгоритмов с помощью столбчатых диаграмм, представленных на рисунке 32 и рисунке 33.

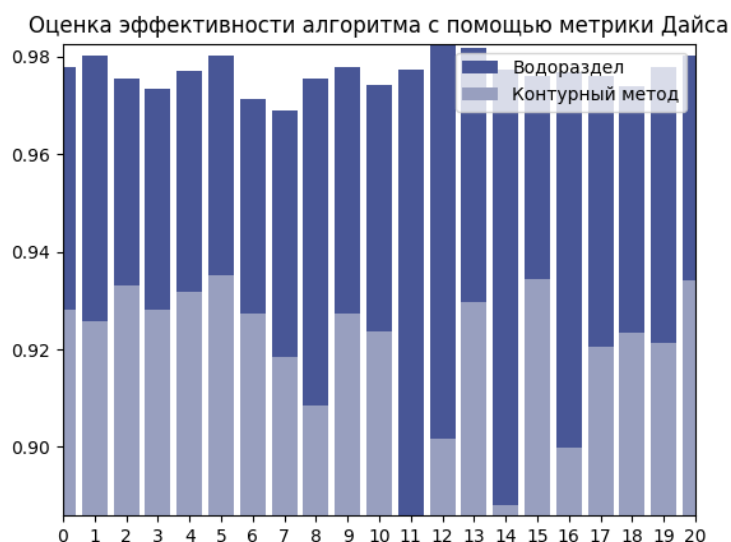


Рисунок 32 – Сравнение коэффициентов Дайса двух методов для набора снимков пациента №1

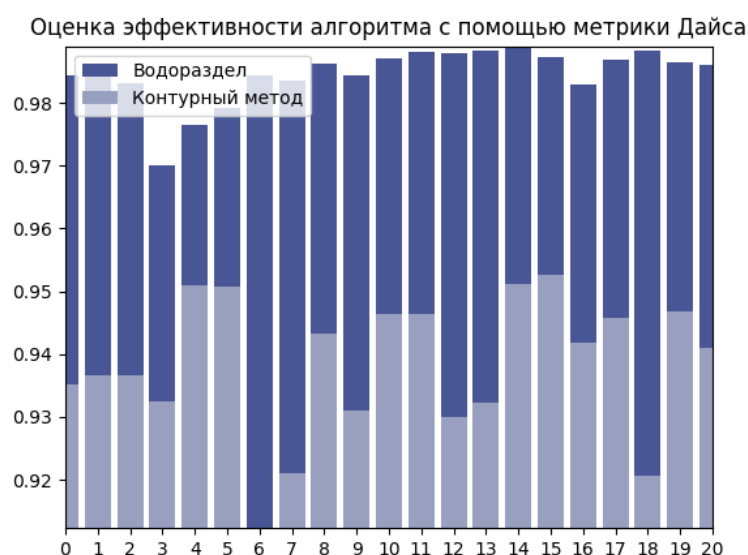


Рисунок 33 – Сравнение коэффициентов Дайса двух методов для набора снимков пациента №1

Таким образом, очевидна разница работы двух алгоритмов. Алгоритм на базе метода водораздела показал лучшие результаты, по сравнению с алгоритмом на основе поиска контуров.

### Вывод

Была решена задача оценки качества сегментации разработанного в рамках данной дипломной работы алгоритма с помощью меры Дайса. В ходе исследования производилась обработка 200 изображений форматов DICOM и JPEG. Точность алгоритма составила 98,25%, что, безусловно, является крайне высоким показателем.

Распространенный аналог, осуществляющий свою работу на базе метода сегментирования на основе границ, показал результат много ниже – в среднем 93,15% на 100 снимков.

Скорость работы разработанного алгоритма на пользовательском компьютере с процессором Intel Core i5-1135G7, 2.4ГГц составила 340 секунд, что составляет в среднем по 3 секунды на одно изображение.

Цель выпускной квалификационной работы достигнута.



## 4. РАЗРАБОТКА И СТАНДАРТИЗАЦИЯ ПРОГРАММНЫХ СРЕДСТВ

### 4.1 Планирование работ проекта

В выпускной классификационной работе был написан программный код для различных исследований. Для планирования проекта было принято решение формализовать представление совокупности планируемых работ, которое позволит контролировать и редактировать организацию рабочего процесса. Для этого была использована диаграмма Ганта (ленточный график), которая подходит для небольших проектов, как ВКР. Данная диаграмма позволяет наглядно представить работы в виде протяженных во времени отрезков, что позволяет контролировать график выполнения работ. Однако, в данном графике нет явной корреляции между отдельной работой и объемами ресурсов для ее выполнения, что может создать препятствие при изменении графика работ. Следовательно, ленточный график не особо подходит для управления ходом разработки сложного проекта, но при этом дает наглядное представление о графиках работы. На рисунке 34 представлена диаграмма Ганта [23] для данного проекта.

Оперграмма представляет собой график организации выполнения работ, состыкованных по видам работ между исполнителями и соисполнителями с учетом сроков работ. В данном случае работу выполняет один человек — исполнитель, им является студент, который пишет программный код. Соисполнителями являются руководитель и консультанты, они участвуют в написании дополнительного раздела и оценивают работу студента, указывая на недочеты работы. В данной работе построение оперграммы не целесообразно, так как соисполнители не влияют на критический путь, т.е. выполняют свой блок работ после того, как исполнитель выполнил свой.

Сетевой график – графическое отображение работ и связей между ними. Он представляет собой более сложный вид оперграммы, который подходит для более сложных проектов. В данном графике оперируют понятиями работа и событие. Работа – это деятельность, которую необходимо выполнить для достижения определенного результата. Событие – это результат достижения промежуточной или конечной цели проекта, которое не имеет протяженности во времени. Для наглядности был построен сетевой график, рисунок 35.

| Структура задач | Наименование задач                                   | Исполнитель                          | Дата начала | Дата окончания | Длительность (дни) | Предшественник |
|-----------------|--|--------------------------------------|-------------|----------------|--------------------|----------------|
| 1               | Выбор темы   | Юлия Завражнова,                     | 01.03.2023  | 04.03.2023     | 4                  |                |
| 2               | Изучение материала                                   | Юлия Завражнова                      | 04.03.2023  | 21.03.2023     | 16                 | 1              |
| 3               | Поиск датасетов                                      | Юлия Завражнова                      | 26.03.2023  | 06.04.2023     | 11                 | 2              |
| 4               | Написание теоретического разде                       | Юлия Завражнова                      | 06.04.2023  | 13.04.2023     | 7                  | 2              |
| 5               | Подготовка датасета                                  | Юлия Завражнова                      | 13.04.2023  | 20.04.2023     | 7                  | 3              |
| 6               | Написание 1 варианта программы                       | Юлия Завражнова                      | 20.04.2023  | 23.04.2023     | 3                  | 5              |
| 7               | Отладка программы                                    | Юлия Завражнова                      | 23.04.2023  | 24.04.2023     | 1                  | 6              |
| 8               | Анализ результатов                                   | Юлия Завражнова,<br>Александр Мотыко | 24.04.2023  | 25.04.2023     | 1                  | 6, 7           |
| 9               | Изучение библиотек для распаковки формата кт-снимков | Юлия Завражнова                      | 25.04.2023  | 28.04.2023     | 3                  | 8              |
| 10              | Написание 2-ого варианта                             | Юлия Завражнова                      | 28.04.2023  | 05.05.2023     | 7                  | 9              |
| 11              | Отладка программы                                    | Юлия Завражнова                      | 05.05.2023  | 06.05.2023     | 1                  | 10             |
| 12              | Анализ результатов                                   | Юлия Завражнова,<br>Александр Мотыко | 06.05.2023  | 07.05.2023     | 1                  | 10, 11         |
| 13              | Написание 2-ой главы                                 | Юлия Завражнова,                     | 08.05.2023  | 12.05.2023     | 4                  | 4, 10          |
| 14              | Написание дополнительного раздела                    | Юлия Завражнова, Ко-<br>сухина Мария | 12.05.2023  | 15.05.2023     | 3                  | 13             |
| 15              | Написание заключительного разд                       | Юлия Завражнова                      | 15.05.2023  | 17.05.2023     | 2                  | 14             |
| 16              | Подготовка к защите практики                         | Юлия Завражнова                      | 17.05.2023  | 21.05.2023     | 4                  | 15             |
| 17              | Подготовка дипломной работы к сдаче                  | Юлия Завражнова                      | 21.05.2023  | 24.05.2023     | 3                  | 16             |

Перечень работ

Таблица 2

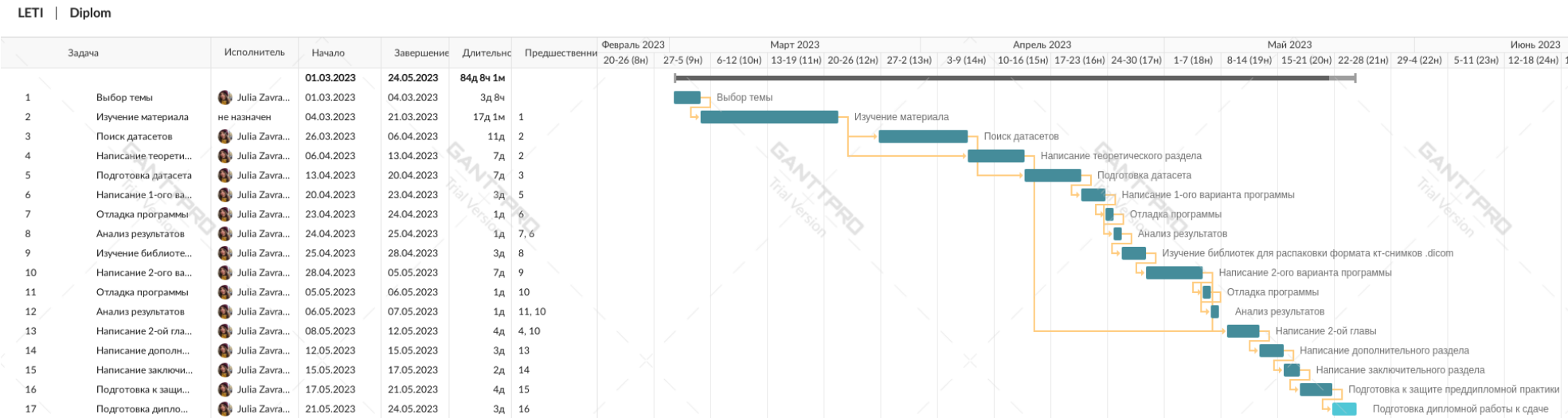


Рисунок 34 — диаграмма Ганта

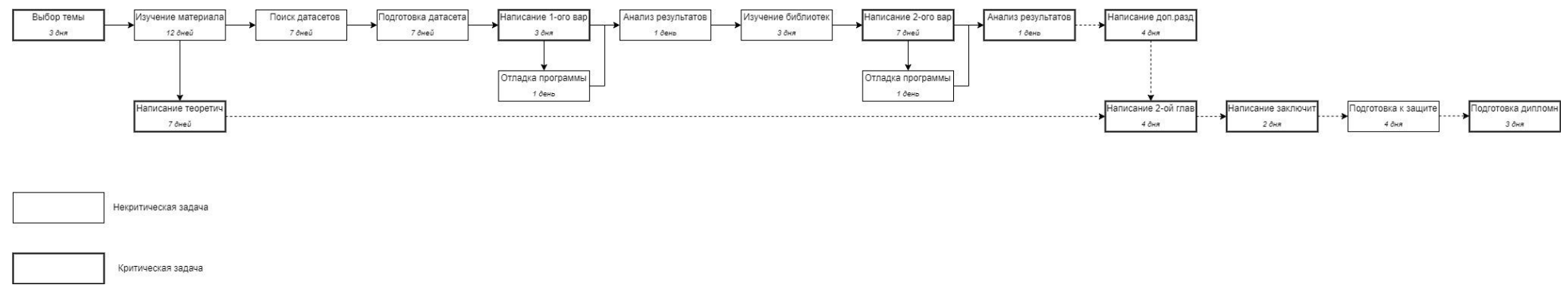


Рисунок 35 — Сетевой график

#### 4.2 Расчет затрат на выполнение проекта, расчет цены проекта и цены предполагаемого продукта

Для расчета затрат на выполнение ВКР принимаются условия:

- Коэффициент загрузки исполнителя ВКР (студента) равен 1 ( $K_{\text{загр.испол.}} = 1$ );
- Коэффициент загрузки соисполнителя (руководителя) равен 0.05 ( $K_{\text{загр.рук.}} = 0.05$ );
- Коэффициент загрузки консультанта по дополнительному разделу равен 0.04 ( $K_{\text{загр.конс.}} = 0.04$ );

Таким образом, цена проекта может быть рассчитана по формуле:

$$C_{\text{пр}} = \sum_{i=1}^n C_{\text{полн } i} T_i K_{\text{загр } i},$$

где  $C_{\text{полн.}i}$  - полная дневная стоимость работы  $i$ -го специалиста [руб./день],  $T_i$  - время участия  $i$ -го специалиста в работе над проектом [дней],  $K_{\text{загр.}i}$  - коэффициент загрузки  $i$ -го специалиста работами в проекте,  $n$  - число специалистов, занятых в проекте.

Для расчета полной стоимости проекта необходимо рассчитать полную дневную стоимость работы. Для удобства все расчеты будут сведены в таблицу для каждого участника.

В таблице 3 представлены расчеты для студента.

Полные затраты в день для студента

Таблица 3

| Наименование статей   | ед. изм.   | нормативы/затраты | Примечание  |
|---|------------|-------------------|---|
| Величина среднемесячной начисленной заработной платы специалиста (Нс1) [24] | руб./месяц | 51 857,50         | Оклад сотрудника Зп                                     |
| Среднемесячное количество рабочих дней (Тср) [25]                           | дней./мес  | 19,67             | Среднее за I-II кварталы 2023г.                         |
| Расчет ставки специалиста в день:   |            |                   |   |
| Тарифная ставка дневная (Нс),   | руб./день  | 2636,38           | $Z_d = Z_{п}/T_{ср}$                                    |
| Страховые взносы 30,2% от суммы зарплаты работников,                        | руб./день  | 790,91            | $C_{сд} = Z_d \times \Phi$                              |
| Оплата основных работников с со страховыми взносами (Zдс),                  | руб./день  | 3 427,29          | $Z_{дс} = Z_d + C_{д} = Z_d \times (1 + \Phi)$          |
| Накладные расходы (Снр )  | руб./день  | 785,40            | $C_{нр} = Z_d \times H$                                 |
| Себестоимость одного человек/дня (Сч/д),                                    | руб./день  | 4 212,69          | $C_{ч/д} = Z_{дс} + C_{нр} = Z_d \times (1 + \Phi + H)$ |
| Дневная прибыль (Спрд),   | руб./день  | 524,55            | $C_{прд} = C_{ч/д} \times \Pi$                          |
| Ставка специалиста без учета НДС (Сдсс),                                    | руб./день  | 4 737,24          | $C_{дсс} = C_{ч/д} + C_{прд}$                           |
| Дневная сумма НДС (Сндс),   | руб./день  | 947,45            | $C_{ндс} = C_{дсс} \times H_{ндс}$                      |
| Ставка специалиста в день с учётом НДС (Сполн),                             | руб./день  | 5 684,7           | Сполн   |

В таблице 4 представлены расчеты для руководителя и консультанта по дополнительному разделу, т.к. заработная плата у данных соисполнителей одинаковая.

Полные затраты в день для руководителя и консультанта

по дополнительному разделу

Таблица 4

| Наименование статей   | ед. изм.    | нормативы/затраты | Примечание  |
|---|-------------|-------------------|---|
| Величина среднемесячной начисленной заработной платы специалиста (Нс1) [26] | руб./месяц  | 98 304,00         | Оклад сотрудника Зсп                                    |
| Среднемесячное количество рабочих дней (Тср)                                | дней./месяц | 19,67             | Среднее за I-II кварталы 2023г.                         |
| Расчет ставки специалиста в день:   |             |                   |   |
| Тарифная ставка дневная (Нс),   | руб./день   | 4 997,67          | $Z_d = Z_{сп}/T_{ср}$                                   |
| Страховые взносы 30,2% от суммы зарплаты работников,                        | руб./день   | 1 509,3           | $C_{сд} = Z_d \times \Phi$                              |
| Оплата основных работников с со страховыми взносами (Zдс),                  | руб./день   | 6 506,97          | $Z_{дс} = Z_d + C_d = Z_d \times (1 + \Phi)$            |
| Накладные расходы (Снр )  | руб./день   | 3 769,94          | $C_{нр} = Z_d \times H$                                 |
| Себестоимость одного человек/дня (Сч/д),                                    | руб./день   | 10 276,91         | $C_{ч/д} = Z_{дс} + C_{нр} = Z_d \times (1 + \Phi + H)$ |
| Дневная прибыль (Спрд),   | руб./день   | 1 797,74          | $C_{прд} = C_{ч/д} \times \Pi$                          |
| Ставка специалиста без учета НДС (Сдсс),                                    | руб./день   | 12 074,65         | $C_{дсс} = C_{ч/д} + C_{прд}$                           |
| Дневная сумма НДС (Сндс),   | руб./день   | 2 414,93          | $C_{ндс} = C_{дсс} \times H_{ндс}$                      |
| Ставка специалиста в день с учётом НДС (Сполн),                             | руб./день   | 14 489,58         | Сполн   |

Исходя из данных в таблицах 3 и 4, можно рассчитать полную цену проекта по формуле, приведенной выше.

$$C_{\text{пр}} = (5\,684,7 \text{руб/день} \cdot 90 \text{дней} \cdot 1) + (14\,489,58 \text{руб/день} \cdot 6 \text{дней} \cdot 0,05) + (14\,489,58 \text{руб/день} \cdot 4 \text{дня} \cdot 0,04) = 535\,390 \text{руб.}$$

Данный проект пока что не рассчитан на вывод на рынок, поэтому делать расчет цены программы не имеет смысла.

#### 4.3 Обеспечение качества программного продукта

При разработке программного продукта должны быть предусмотрены мероприятия по обеспечению его качества, в данном случае таким мероприятием является отладка программы. В данном процессе происходит нахождение ошибок в программе и проверка правильной работы кода. Наглядно процесс обеспечения качества продукта представлен на рисунке 36.

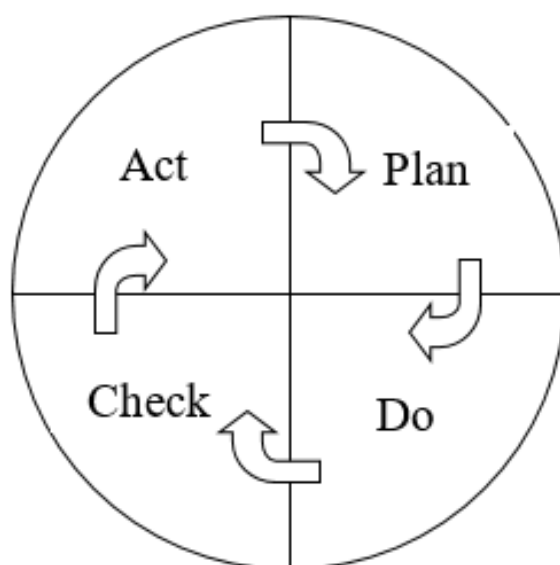


Рисунок 36 — цикл Деминга

В данной работе целесообразно использовать цикл Деминга, т.к. работа будет циклично повторяться. В рамках цикла Демминга предусмотрен следующий порядок действий:

Р – (Plan) – установить конечную цель и определить ресурсы для достижения, в ВКР целью будет написание небольшого блока программного кода, который в дальнейшем будет модернизироваться;

D – (Do) – внедрение процессов, на данном этапе будет происходить написание блока в соответствии с целями;

C – (Check) – контроль (мониторинг) кода, здесь происходит проверка кода на правильность работы и сравнения результатов с эталоном;

A – (Act) – ликвидация ошибок или отладка программного кода.

#### 4.4 Определение кода программного средства

Для определения кода программного средства необходимо выяснить какую функцию выполняет программный код. В данном случае был написан код для прикладных задач, т.е. по классификации ОКПД2 ВКР подпадает под код 62.01.11 — услуги по проектированию, разработке информационных технологий для прикладных задач и тестированию программного обеспечения.

#### 4.5 Определение списка стандартов для ВКР

При написании данной выпускной квалификационной работы не ставилось задачи выпуска программной продукции на рынок, поэтому при постановке задачи и написания кода программы не было привязки к каким-либо стандартам, но при оформлении ВКР стоит обратить внимание на следующие ГОСТы: ГОСТ 19.102-77, ГОСТ 19.401-78, - а также стоит обратиться к ГОСТам, регламентирующим оформление проектной документации.



## ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы были рассмотрены методы диагностики заболеваний в современных видеосистемах и произведен анализ существующих методов сегментации и цифровой обработки изображений.

Так же было предложено решение проблемы сегментации легких на снимках компьютерной томографии с помощью метода сегментации watershed (водораздела). Был разработан алгоритм на базе языка программирования Python. В качестве среды разработки был выбран редактор исходного кода Visual Studio Code (VS Code).

Оценка точности сегментирования легких с помощью разработанного алгоритма осуществлялась с использованием меры Дайса и составила 98,25%. Также было приведено сравнение результатов, полученных с помощью метода водораздела и метода сегментирования на основе границ. Согласно проведенному исследованию, разработанный нами алгоритм обеспечил точность на 5,1% лучше, чем его аналог.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Здравоохранение в России 2021. Стат. сб. М.: Росстат, 2021. 171 с. URL: <https://rosstat.gov.ru/storage/mediabank/Zdravoohran-2021.pdf> (дата обращения 21.04.2023).
2. 10 ведущих причин смерти в мире / ВОЗ. URL: <https://www.who.int/ru/news-room/fact-sheets/detail/the-top-10-causes-of-death> (дата обращения 21.04.2023).
3. Виды обследования легких / Ами. URL: <https://kt-spb.ru/medical-articles/vidy-obsledovaniya-legkikh/> (дата обращения 21.04.2023).
4. Рентгенография грудной клетки / Википедия. URL: [https://ru.wikipedia.org/wiki/Рентгенография\\_грудной\\_клетки](https://ru.wikipedia.org/wiki/Рентгенография_грудной_клетки) (дата обращения 21.04.2023).
5. Компьютерная томография / Википедия. URL: [https://ru.wikipedia.org/wiki/Компьютерная\\_томография](https://ru.wikipedia.org/wiki/Компьютерная_томография) (дата обращения 21.04.2023).
6. Что такое компьютерная томография? Преимущества томографов / Диагност КТ. URL: <https://diagnost-kt.org.ua/chto-takoe-kompyuternaya-tomografiya/> (дата обращения 21.04.2023).
7. Магнитно-резонансная томография / Википедия. URL: [https://ru.wikipedia.org/wiki/Магнитно-резонансная\\_томография](https://ru.wikipedia.org/wiki/Магнитно-резонансная_томография) (дата обращения 21.04.2023).
8. КТ и МРТ: в чем разница? / Ами. URL: <https://kt-spb.ru/medical-articles/kt-i-mrt-v-chem-raznica/> (дата обращения 21.04.2023).
9. An Introduction to Image Segmentation: Deep Learning vs. Traditional / V7. URL: <https://www.v7labs.com/blog/image-segmentation-guide> (дата обращения 21.04.2023).
10. Image Segmentation Explained / Built In. URL: <https://builtin.com/machine-learning/image-segmentation> (дата обращения 21.04.2023).
11. Гонсалес Р. С., Вудс Р. Е. Цифровая обработка изображений. 3-е изд. М.: Техносфера, 2012. 1105 с.
12. Chest CT-Scan images Dataset / Kaggle. URL: <https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images?resource=download> (дата обращения 21.04.2023).
13. Pydicom. URL: <https://pydicom.github.io/> (дата обращения 21.04.2023).
14. Miscellaneous operating system interfaces / OS. URL: <https://docs.>

- python.org/3/library/os.html (дата обращения 21.04.2023).
15. Alkousa O. Introducing PyDicom, Its Classes, Methods, and Attributes <https://towardsdatascience.com/introducing-pydicom-its-classes-methods-and-attributes-518c1d71162> (дата обращения 21.04.2023).
16. Шкала Хаунсфилда / Википедия. URL: [https://ru.wikipedia.org/wiki/Шкала\\_Хаунсфилда](https://ru.wikipedia.org/wiki/Шкала_Хаунсфилда) (дата обращения 21.04.2023).
17. Scikit-image. URL: <https://scikit-image.org/> (дата обращения 21.04.2023).
18. SciPy. URL: <https://scipy.org/> (дата обращения 21.04.2023).
19. NumPy documentation / NumPy. URL: <https://numpy.org/doc/stable/> (дата обращения 21.04.2023).
20. OpenCV. URL: <https://docs.opencv.org/4.7.0/> (дата обращения 21.04.2023).
21. Sørensen–Dice coefficient / Wikipedia. URL: [https://en.wikipedia.org/wiki/Sørensen–Dice\\_coefficient](https://en.wikipedia.org/wiki/Sørensen–Dice_coefficient) (дата обращения 21.04.2023).
22. Обнаружение объекта на изображении методом цветовой сегментации (Python). URL: <https://waksoft.susu.ru/2019/05/26/obnaruzhenie-obekta-na-izobrazhenii-opirajas-na-cvetovuju-segmentacii-python/> (дата обращения 21.04.2023).
23. Онлайн-диаграмма Ганта для управления проектами. Построение диаграммы Ганта / Gantt Rro. URL: <https://ganttpro.com/> (дата обращения 21.04.2023).
24. Рынок труда, занятость и заработная плата / Росстат. URL: [https://rosstat.gov.ru/labor\\_market\\_employment\\_salaries](https://rosstat.gov.ru/labor_market_employment_salaries) (дата обращения 21.04.2023).
25. Статья112 ТК РФ. Нерабочие праздничные дни Гарант.Ру. URL: <https://base.garant.ru/12125268/ea54c1918750348cf1860e01a0121200/> (дата обращения 21.04.2023).
26. Постановление Правительства РФ от 29.08.2022 № 1505 «О переносе выходных дней в 2023 году» / Гарант.Ру. URL: <https://www.garant.ru/products/ipo/prime/doc/405112215/> (дата обращения 21.04.2023).
27. Средняя заработная плата научных сотрудников государственной и муниципальной форм собственности по субъектам Российской Федерации 2021 год / Росстат. URL: [https://docs.yandex.ru/docs/view?url=yabrowser://4dt1uxeprrjrxlufoewruffsr2wjvfqmczdbex4pnlsj5z6wnpghr4-c462vfoidzmroqxu2\\_muqysn7syvyrcnhjipuwwfogm7cy2wu3\\_dnua7hcm-u0\\_i](https://docs.yandex.ru/docs/view?url=yabrowser://4dt1uxeprrjrxlufoewruffsr2wjvfqmczdbex4pnlsj5z6wnpghr4-c462vfoidzmroqxu2_muqysn7syvyrcnhjipuwwfogm7cy2wu3_dnua7hcm-u0_i)

\_uhgywuxc3rpmm8n68ey5-yeyzyda==?sign=nezm5crmqln9en0-hxg6pmcjhvzi  
3jnlks1pv9tm8uu=&name=Научные сотрудники (160).xlsx&nosw=1 (дата  
обращения 21.04.2023).

## ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
from skimage import measure, segmentation
import os
import pydicom
import scipy.ndimage as ndimage
from datetime import datetime

start = datetime.now()
input_path = "/Diplom/train/"
list_patients = os.listdir(input_path)
list_patients.sort()

def load_scan(path):
    images_slices = [pydicom.read_file(path + '/' + s) for s in os.listdir(path)]
    images_slices.sort(key = lambda x: int(x.InstanceNumber))
    return images_slices

def coeff_hu(many_pictures):
    image = np.stack([scan.pixel_array for scan in many_pictures])
    image = image.astype(np.int16)
    image[image == -2000] = 0
    # HU = m*P + b
    b = many_pictures[0].RescaleIntercept
    m = many_pictures[0].RescaleSlope
    if m != 1:
        image = m * image.astype(np.float64)
        image = image.astype(np.int16)
    image += np.int16(b)
    return np.array(image, dtype=np.int16)

def all_markers(image):
    internal_marker = image < -350
```

```

internal_marker = segmentation.clear_border(internal_marker)
internal_labels = measure.label(internal_marker)
areas = [r.area for r in measure.regionprops(internal_labels)]
areas.sort()
if len(areas) > 2:
    for region in measure.regionprops(internal_labels):
        if region.area < areas[-2]:
            for coordinates in region.coords:
                internal_labels[coordinates[0], coordinates[1]] = 0
internal_marker = internal_labels > 0
external_1 = ndimage.binary_dilation(internal_marker, iterations=10)
external_2 = ndimage.binary_dilation(internal_marker, iterations=55)
external_marker = external_2 ^ external_1
watershed_markers = np.zeros((512, 512), dtype=np.int64)
watershed_markers += internal_marker * 255
watershed_markers += external_marker * 128
return internal_marker, external_marker, watershed_markers

def segmentation_lungs(image):
    marker_watershed = all_markers(image)
    sobel_dx = ndimage.sobel(image, 1)
    sobel_dy = ndimage.sobel(image, 0)
    sobel_filter = np.hypot(sobel_dx, sobel_dy)
    sobel_filter *= 255.0 / np.max(sobel_filter)
    watershed = segmentation.watershed(sobel_filter, marker_watershed)
    return sobel_filter, watershed

def apply_a_contour(image, labels):
    contours = measure.find_contours(labels)
    for contour in contours:
        plt.plot(contour[:,1], contour[:,0], linewidth=2, color = '#681cff')
    plt.imshow(image, cmap = 'gray')
    plt.title('Result')
    plt.show()

test_patient_scans = load_scan(input_path + list_patients[24])

```

```

test_patient_images = coeff_hu(test_patient_scans)
original_image = test_patient_images[78]
plt.imshow(original_image, cmap='gray')
plt.title("Original Slice")
plt.show()

test_patient_internal, test_patient_external, test_patient_watershed =
all_markers(original_image)
f, (ax1, ax2, ax3) = plt.subplots(1, 3, sharey=True, figsize=(10, 10))
ax1.imshow(test_patient_internal, cmap='gray')
ax1.set_title("Internal Marker")
ax1.axis('off')
ax2.imshow(test_patient_external, cmap='gray')
ax2.set_title("External Marker")
ax2.axis('off')
ax3.imshow(test_patient_watershed, cmap='gray')
ax3.set_title("Watershed Marker")
ax3.axis('off')
plt.show()

test_sobel_gradient, test_watershed = segmentation_lungs(original_image)
f, (ax1, ax2) = plt.subplots(1, 2, sharey=True, figsize=(10,10))
ax1.imshow(test_sobel_gradient, cmap='gray')
ax1.set_title("Sobel")
ax1.axis('off')
ax2.imshow(test_watershed, cmap='gray')
ax2.set_title("Watershed")
ax2.axis('off')
plt.show()
print(test_watershed.dtype)
print(test_watershed.shape)
saving_watershed = cv2.imwrite(r'C:\Diplom\Result\Patient_2\3.jpg',
test_watershed)
result = apply_a_contour(test_patient_images[78], test_watershed)
end = (datetime.now() - start)
print(end)

```