

Содержание

Введение.....	4
1. Теоретическая часть.....	5
2. Программная реализация	11
3. Тестирование	14
Заключение	23
Список использованных источников	24

					МИВУ 09.03.02-00.005 ПЗ						
Изм.	Лист	№ докум.	Подпись	Дата							
Разраб.		Минеев Р.Р.			Исследование работы методов фрактальной сегментации на маммографических снимках			Лит.	Лист	Листов	
Провер.		Садыков С.С.								3	24
Конс		Подгорнова Ю.А.						МИ ВлГУ ИС-117			
Н.Контр.											
Утв.											

Введение

Рак молочной железы — злокачественное образование молочной железы, наиболее частая форма рака у женщин (21,1% в структуре женской онкозаболеваемости). В 2017 году в РФ было выявлено 70 569 случаев РМЖ, почти на 20 000 случаев больше, чем в 2007 году.

Заболеваемость раком молочной железы растет не только в России, но и во всем мире. РМЖ научились лучше выявлять, методы диагностики постоянно улучшаются, внедряются скрининговые программы. Вместе с тем за счет увеличения продолжительности жизни населения увеличивается число женщин пожилого возраста, которым ставят такой диагноз.

Цель скрининга — раннее активное выявление и лечение бессимптомного рака. Необходимо четко отличать скрининг от ранней диагностики, т.е. выявления заболевания у больных, которые сами обратились за медицинской помощью и, чаще всего, имеют те или иные жалобы и симптомы. Скрининг не всегда эффективен, что противоречит укоренившемуся среди врачей мнению, будто любой скрининг должен дать положительный результат. Скрининг должен быть направлен на выявление образований, которые прогрессируют и дают метастазы и, соответственно, могут привести к смерти.

Целью данной работы является: реализация программы скрининга, основанного на методах фрактальной сегментации поражённых областей на маммографических снимках.

Задачи данной научно–исследовательской работы:

- изучить работу алгоритмов фрактальной сегментации;
- выбрать алгоритм для реализации;
- реализовать выбранный алгоритм сегментации на маммограммах;
- провести тестирование реализованного алгоритма на маммографических снимках.

1. Теоретическая часть

Одним из фундаментальных этапов анализа графической информации является сегментация изображения, означающая разбиение поступающего изображения на множество областей, однородных в соответствии с теми или иными критериями, и тем самым ассоциируемых с объектами наблюдаемой сцены или их частями. Результатом сегментации является получение карты областей (сегментов) изображения.

Одним из важнейших аспектов при решении задачи сегментации изображения как выделения однородных областей, является вопрос «похожести» элементов и/или кластеров изображения, т.е. близости их признаков. Поскольку как элементы, так и кластеры могут быть отображены в одно и то же пространство признаков, то вопрос измерения близости может быть переформулирован как введение метрики в пространстве признаков.

Задачей стадии первичной сегментации является преобразование исходного векторного изображения в первоначальный набор кластеров небольшого размера, в котором каждый сформированный кластер характеризуется как собственными параметрами, так и параметрами связи с соседними кластерами. Эта задачу удобно решать с помощью алгоритмов фрактальной сегментации.

Примерами таких алгоритмов могут служить алгоритм оценивания размера фрактала по длине контура, метод броуновской функции и метод пирамиды.

Алгоритм оценивания размера фрактала по длине контура выглядит следующим образом. Производится разбиение динамического диапазона яркостей изображения на равные интервалы [3]. Для полученного набора пороговых значений яркости строятся бинарные изображения. При этом элементам изображения, яркость которых меньше порога, присваивается значение «0», а элементам, яркость которых выше или равна порогу, присваивается значение «1». Производится оценка длин контуров единичной

					МИВУ 09.03.02-00.005	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

области для серии шагов s_i . Увеличение шага эквивалентно представлению анализируемого изображения с разрешением меньшим, чем разрешение полученного изображения. Длина контура аппроксимируется числом переходов уровней яркости бинарного изображения из 0 в 1 и из 1 в 0 для каждого значения шага. По полученным значениям оценивается регрессия логарифма длины контура от логарифма шага. Для каждого из бинарных изображений вычисляется оценка размера фрактала D_i контуров единичных областей. Оценка размера фрактала D исходного изображения получается усреднением оценок D_i .

Данный метод можно использовать для оценки размера фрактала D при условии ее НЧ-фильтрации [4]. Исключение некоторых интервалов из рассмотрения при оценке размера фрактала приводит к неразличимости фракталов, имеющих разный размер. Для изотропных текстур большей эффективностью обладает алгоритм оценки размера фрактала по строкам и по столбцам.

Еще один метод - метод броуновской функции [3]. Математической моделью таких фракталов является фрактальная (обобщенная) броуновская функция $L(x)$ — вещественная случайная функция, такая что для всех x и Δx существует значение h , при котором выполняется

$$P \left\{ \left[\frac{L(x+\Delta x) - L(x)}{\|\Delta x\|^h} \right] < t \right\} = F(t), \quad (1)$$

где x — точка в n -мерном евклидовом пространстве R^n ,

Δx — расстояние между элементами изображения,

$F(t)$ — функция распределения случайной величины t .

Функция $F(t)$ описывает нормальное распределение с нулевым математическим ожиданием и дисперсией σ^2 . Обобщение броуновской функции состоит в том, что вместо $1/2$ вводится действительный параметр h — некоторая постоянная, диапазон изменения которой $[0,1]$.

Оценка D определяется выражением (2) при $K=3$:

$$D = K - \hat{\theta}_{1,.} \quad (2)$$

					МИВУ 09.03.02-00.005	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

где $\hat{\theta}_1$ — тангенс угла наклона линии выборочной регрессии.

Изображение рассматривается как двумерная функция яркости $L(x)$, определенная для $x=(x,y) \in R^2$, формирующая трехмерную поверхность. Для вычисления h необходимо получить оценку линейной выборочной регрессии логарифма математического ожидания модуля разностей яркостей всех пар элементов, находящихся на определенном расстоянии Δx , от логарифма этого расстояния. Расстояния объединяются в кластеры, и математические ожидания вычисляются для каждого кластера

Эффективность метода броуновской функции может быть повышена за счет НЧ-фильтрации оценки размера фрактала [4]. Оценку размера фрактала можно производить, учитывая изменения яркости только по строкам или только по столбцам, что позволит повысить эффективность сегментации анизотропных текстур.

В рамках данной работы наиболее подходящим будет являться пирамидальный алгоритм сегментации изображений.

Рассматриваемый алгоритм содержит два уровня (стадии). На первом изображение разбивается на множество мелких кластеров с помощью пирамидального алгоритма, а на втором происходит их окончательное слияние [3].

Пирамидальный алгоритм обработки изображения требует задания способа вычисления уменьшенного изображения уровня $n+1$ на основе имеющегося изображения уровня n , и применения этой процедуры рекурсивно до предельного уменьшения размера изображения. Часто для этого выбирают элементарный домен некоторой формы, позволяющей плотно покрыть всю площадь изображения, и задают способ нахождения значения элемента следующего уровня по значениям элементов домена предыдущего уровня. Рекурсивное применение данной процедуры позволяет построить дерево (граф), в котором каждый элемент изображения некоторого уровня (кроме самого нижнего) является узлом, связанным с элементами домена

предыдущего уровня, а число нисходящих связей определяется формой и размерами выбранного домена. В случае, когда в качестве домена выбран квадрат размерами 2×2 элемента, получается так называемое квадродерево (рис.1, а).

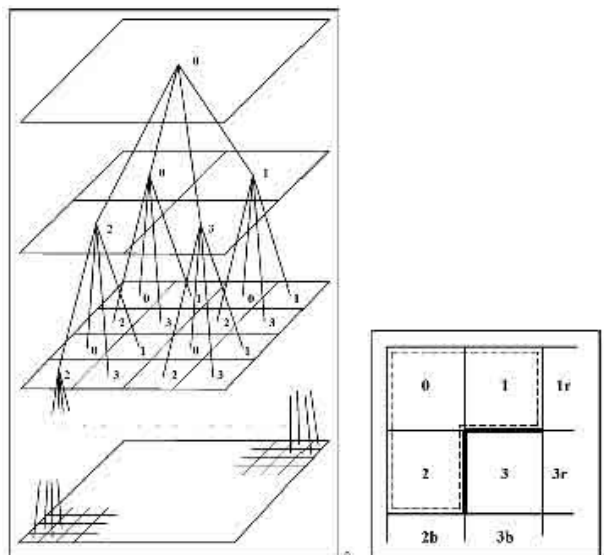


Рисунок 1 - Пирамидальная структура представления данных — квадродерево (а); домен из четырех элементов и его соседние элементы справа и снизу (б)

При прямом проходе вверх по квадродереву происходит рекурсивный анализ всех уровней пирамиды, начиная с самого нижнего (исходного изображения) и заканчивая верхним уровнем, состоящим из одного узла; одновременно с этим строится само квадродерево. На каждом шаге на основе анализа четырех нижних узлов уровня n создается узел уровня $n+1$ (рис.1, а), а в соответствующей структуре нового узла запоминается информация об узлах предыдущего уровня, соединенных с данным узлом, их средней яркости, наличии контуров. Таким образом, каждый узел является вершиной некоторого квадродерева, охватывающего расположенные под ним элементы изображения, и содержит информацию о поддеревьях предыдущего уровня [3].

Основной задачей пирамидального этапа сегментации является объединение соседних элементов, имеющих близкие признаки и не разделенных контуром. Эта процедура требует прослеживания контурных линий на всех уровнях пирамиды.

Анализ и объединение элементов домена размерами 2×2 (рис.1, б) состоит из следующих шагов.

Шаг 1. Анализ наличия контурных перепадов между парами соседних элементов: 0-1, 2-3, 0-2 и 1-3. Пары элементов (узлов графа), не разделенные контуром, считаются принадлежащими одному и тому же кластеру. На рисунке 1,б показан контур, проходящий между элементами 1 и 3, а также между элементами 2 и 3. Тем самым, элементы 0, 1, и 2 соответствуют некоторому одному кластеру, а элемент 3 — другому кластеру.

Шаг 2. Выбор наиболее представительного набора элементов, принадлежащих одному кластеру нижнего уровня (в данном случае в такой набор входят элементы 0, 1, и 2), и передача усредненной информации соответствующему узлу следующего уровня. Оставшийся элемент (номер 3 на рисунке) считается “подвешенным”, информация о нем наверх не передается и сохраняется только в самом описании текущего узла.

Шаг 3. Создание карты контуров для следующего уровня. Это осуществляется с помощью анализа внутренних элементов самой четверки, плюс четверки соседних с ней элементов в последовательной развертке, т.е. справа и снизу; согласно примеру на рисунке 1,б будет проверяться наличие контурных перепадов между элементами 1-1г и 2-2б.

После окончания прямого прохода по всему квадродереву, каждый узел любого уровня (кроме корневого узла) будет либо принадлежать поддереву, начинающемуся на одном из верхних уровней, либо сам являться корнем некоторого поддерева. При обратном проходе вниз по квадродереву происходит синтез первичных кластеров. На каждом из уровней вершинам новых поддеревьев (т.е. тем, которые оказались «подвешены» на шаге 2

прямого прохода) присваиваются новые номера кластеров; данные номера распространяются вниз на все элементы, связанные с данным поддеревом. Эта процедура рекурсивно распространяется на все дерево с верхнего уровня (вершины дерева) до нижнего (самого изображения); элементы изображения, получившие при этом одинаковые номера, образуют первичные кластеры (первичные сегменты) обрабатываемого изображения.

Процедура синтеза на каждом из узлов дерева состоит в следующем:

а) если текущий узел не имеет присвоенного номера, он получает следующий свободный номер кластера;

б) номер кластера, присвоенный текущему узлу, присваивается всем узлам следующего, более низкого, уровня, связанным с текущим узлом при прямом проходе.

Результатом данной процедуры является первичная карта кластеров сегментируемого изображения [3].

Согласно структуре алгоритма сегментации, разбиение образовавшихся кластеров на более мелкие части в дальнейшем невозможно. Следовательно, результаты первой стадии должны удовлетворять двум следующим требованиям (качественному и количественному). Качественное требование состоит в том, что первичный набор кластеров должен быть как можно более детализированным, чтобы обеспечить разделение всех областей изображения, которые должны быть отдельными на финальной карте сегментов. Количественное требование заключается в том, что число кластеров первичного набора не должно быть излишне большим, чтобы предотвратить перегруженность процедуры анализа на второй стадии [4].

2. Программная реализация

Листинг программы:

```
import os
import cv2
import pathlib
import numpy as np
from sklearn.cluster import KMeans

level = 7
size = (600, 600)
sizeShow = (300, 300)

def GetPyrImg(image):
    G = image.copy()
    gpImg = [G]
    for i in range(level):
        G = cv2.pyrDown(G)
        gpImg.append(G)

    # generate Laplacian Pyramid
    lpImg = [gpImg[level - 1]]
    for i in range(level - 1, 0, -1):
        GE = cv2.pyrUp(gpImg[i])
        L = cv2.subtract(gpImg[i - 1], GE)
        lpImg.append(L)

    ls_ = lpImg[0]
    for i in range(1, level):
        ls_ = cv2.pyrUp(ls_)
        ls_ = cv2.add(ls_, lpImg[i])

    img = cv2.resize(ls_, size)
    return img

def GetKMeans(image):
    img_r = (image / 255.0).reshape(-1, 3)
    k_colors = KMeans(n_clusters=4).fit(img_r)
    img1 = k_colors.cluster_centers_[k_colors.labels_]
    img1 = np.reshape(img1, (image.shape))
    return (img1 * 255).astype(np.uint8)

path = 'Pics/'

for imageName in pathlib.Path(path).iterdir():
    if imageName.is_file():
        # image = cv2.imread( str(imageName.resolve()).replace('\\', '/'), 0)
        image = cv2.imread( str(imageName), 0)
        os.chdir('Pics/Mod/')
        PyrImage = GetPyrImg(image)
        KImage = GetKMeans(PyrImage)
        image = cv2.resize(image, sizeShow)
        k_image = cv2.resize(KImage, sizeShow)
        saveImage = np.concatenate([image, k_image], axis=1)
        cv2.imwrite(str(imageName).split('\\')[-1], saveImage)
        os.chdir('../..')
```

Более наглядно алгоритм работы программы представлен на блок-схеме на рисунке 2.

					МИВУ 09.03.02-00.005	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

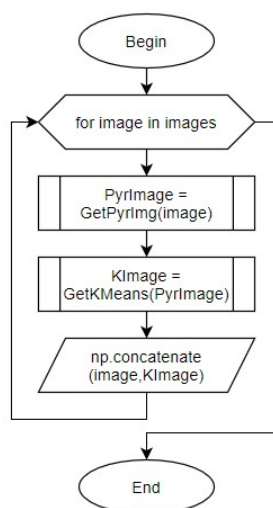


Рисунок 2 – общая блок-схема работы программы

Исходя из данной блок-схемы, видно, что алгоритм содержит в себе две процедуры: алгоритм пирамидальной сегментации и метод k -средних. Блок-схемы этих процедур представлены на рисунках 3 и 4.

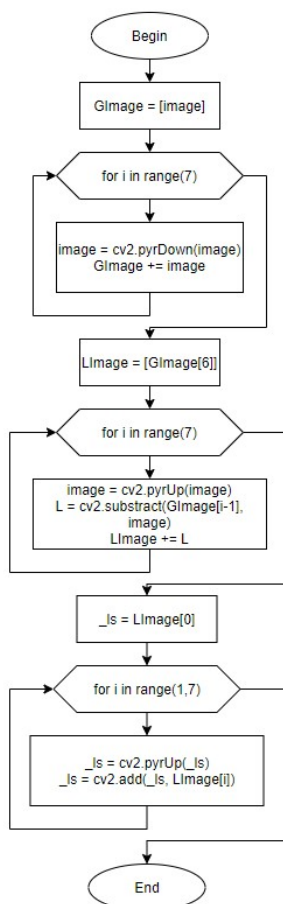


Рисунок 3 – блок-схема процедуры метода пирамидальной сегментации

В алгоритме пирамидальной сегментации происходит спуск на 7 уровней с последующим получением пирамид Лапласа и удалением «излишней» информации. Происходит сжатие изображения. После выполняется подъём обратно на 7 уровней для получения исходного изображения с убранными мелкими деталями.

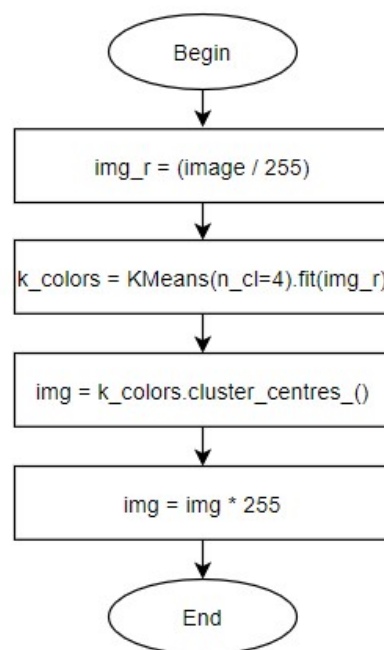
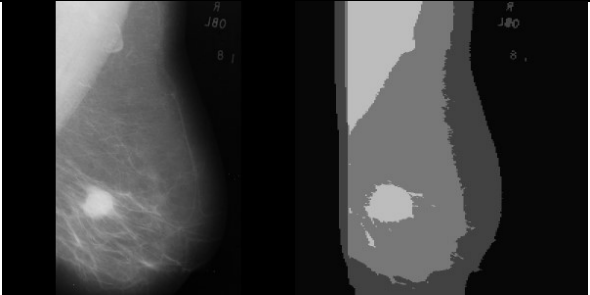
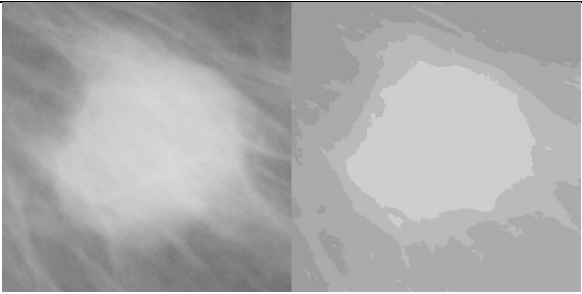
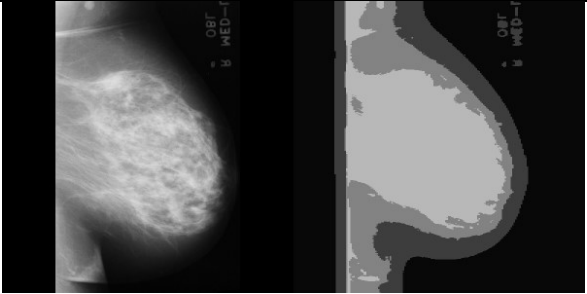
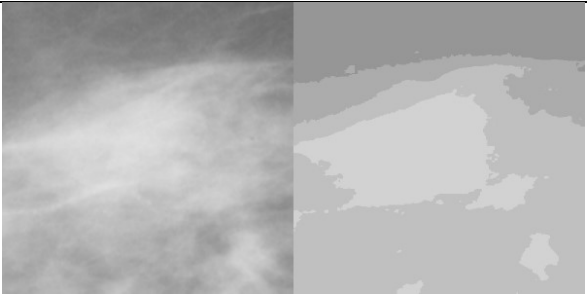
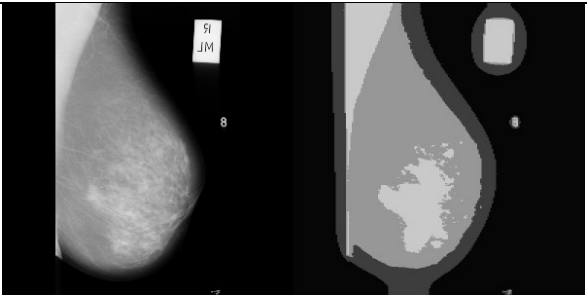


Рисунок 4 - блок-схема процедуры метода k-средних

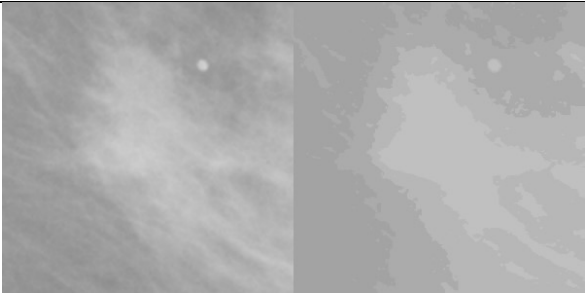
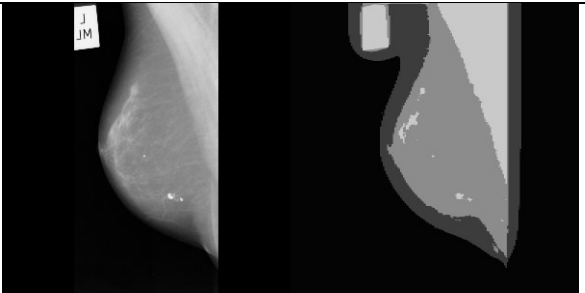
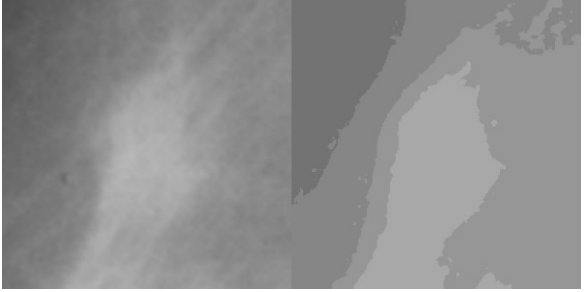
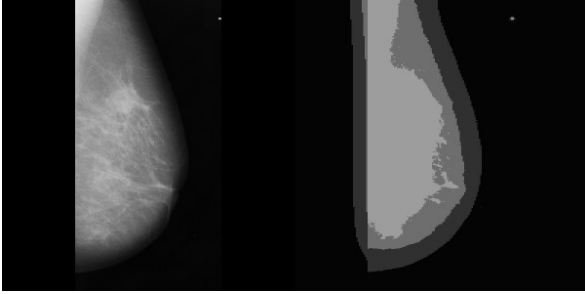
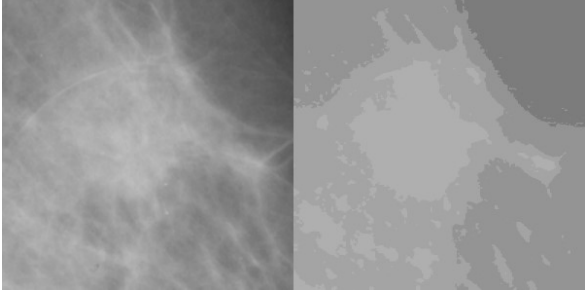
В данной работе используется функция KMeans из библиотеки SciPy. В ней происходит выделение 4 кластеров получившегося изображения, где самым ярким кластером, самым ярким «пятном» будет искомая опухоль, которую необходимо было найти.

3. Тестирование

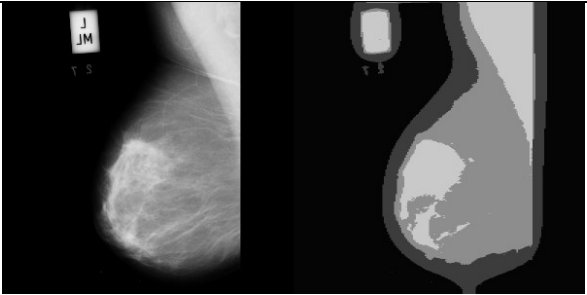
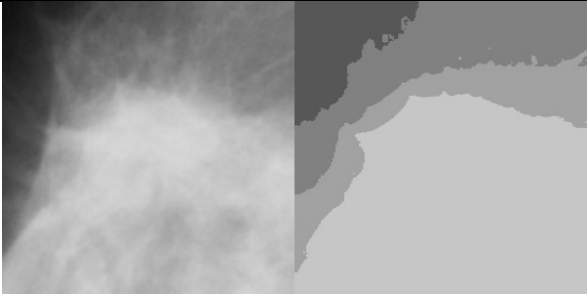
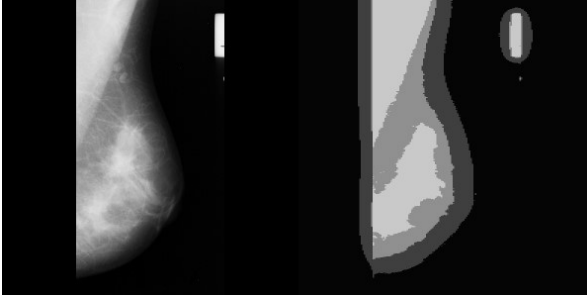
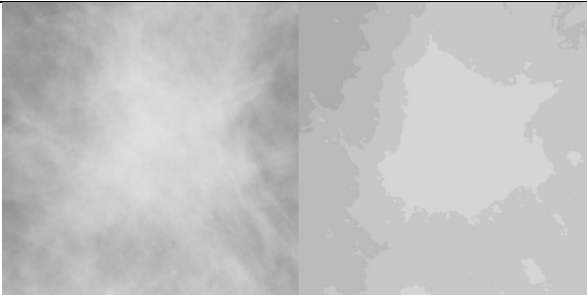
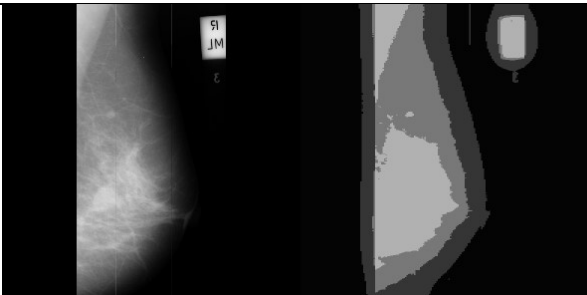
Таблица – Результаты обработки изображений

Название изображения	До и после обработки
1	2
Mdb028	
Mdb028_1	
Mdb032	
Mdb032_1	
Mdb058	


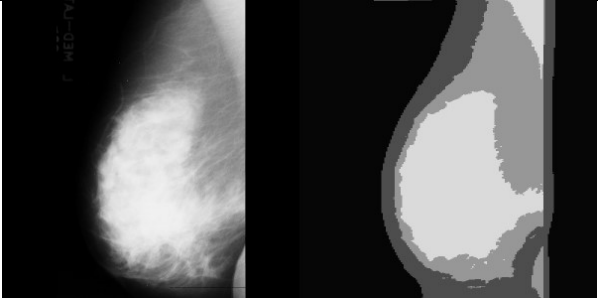

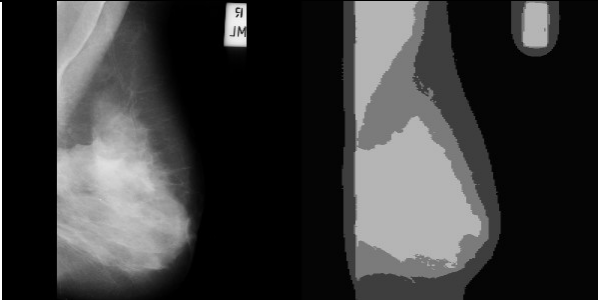

Продолжение таблицы - Результаты обработки изображений

1	2
Mdb058_1	
Mdb075	
Mdb075_1	
Mdb092	
Mdb092_1	

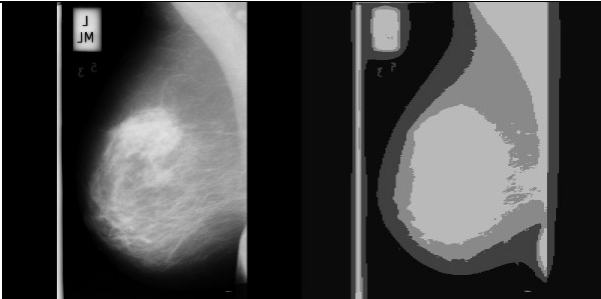
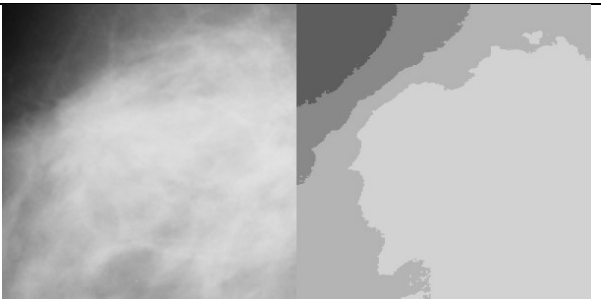
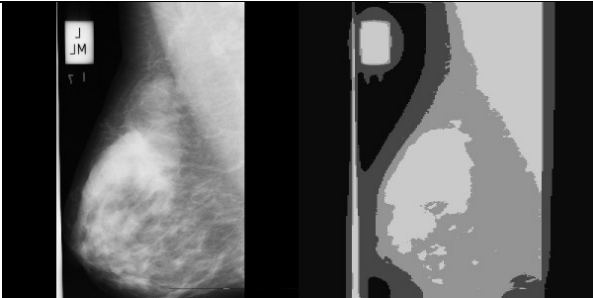
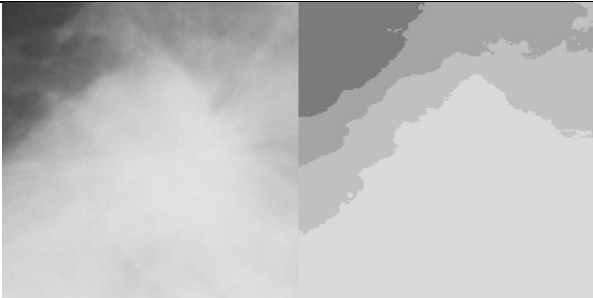
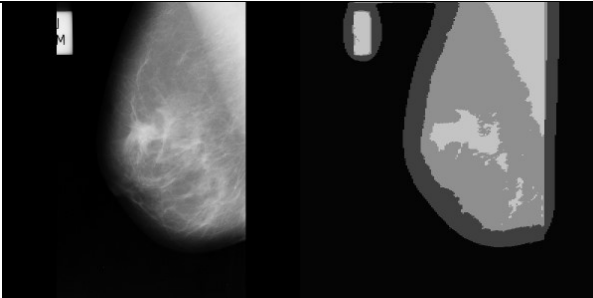
Продолжение таблицы - Результаты обработки изображений

1	2
Mdb095	
Mdb095_1	
Mdb102	
Mdb102_1	
Mdb104	

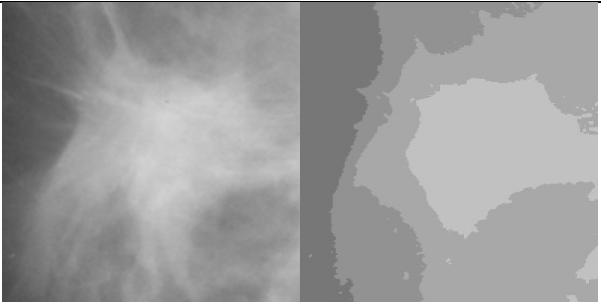
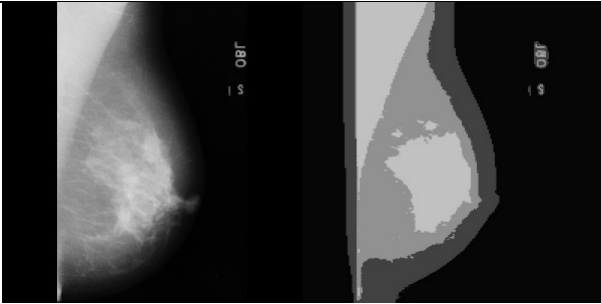

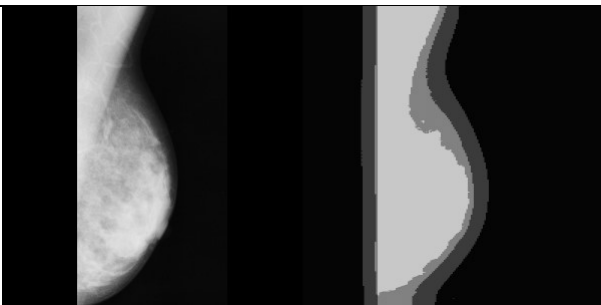

Продолжение таблицы - Результаты обработки изображений

1	2
Mdb104_1	
Mdb105	
Mdb105_1	
Mdb110	
Mdb110_1	

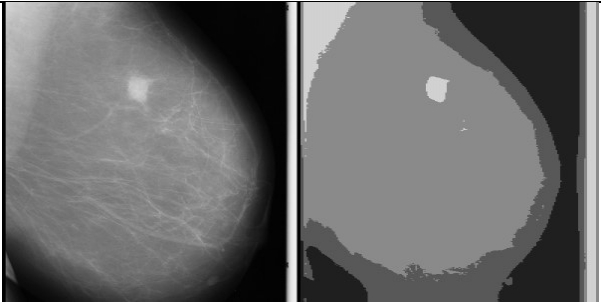
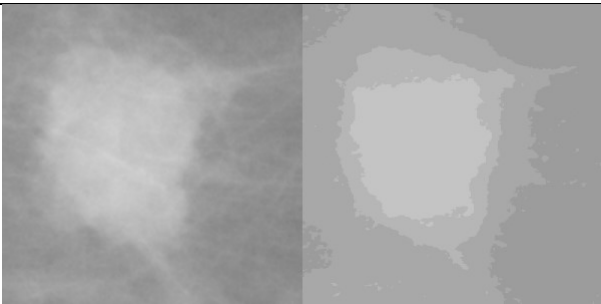
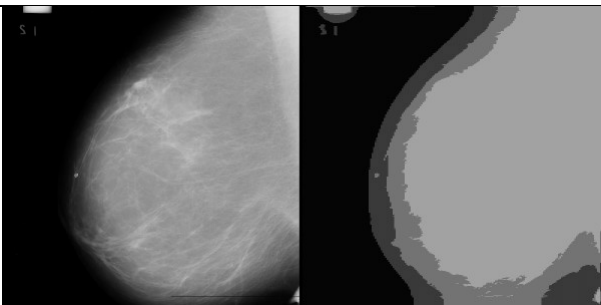
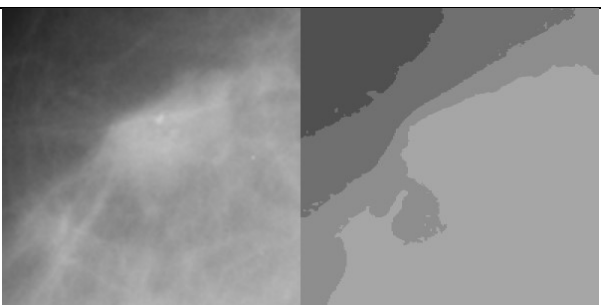
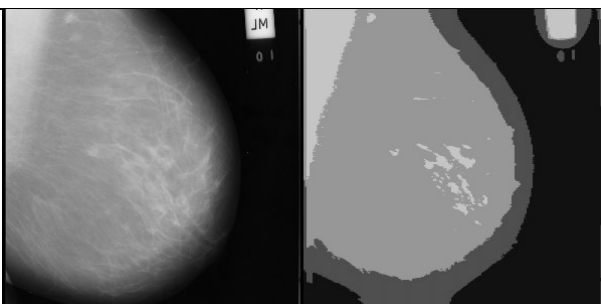
Продолжение таблицы - Результаты обработки изображений

1	2
Mdb111	
Mdb111_1	
Mdb115	
Mdb115_1	
Mdb117	

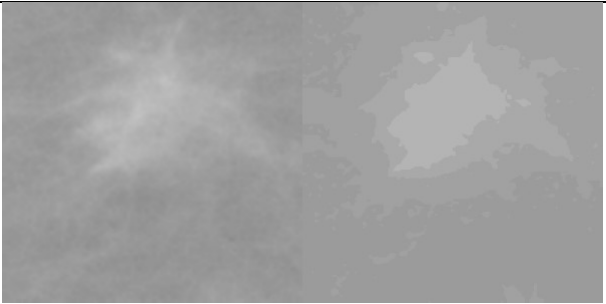
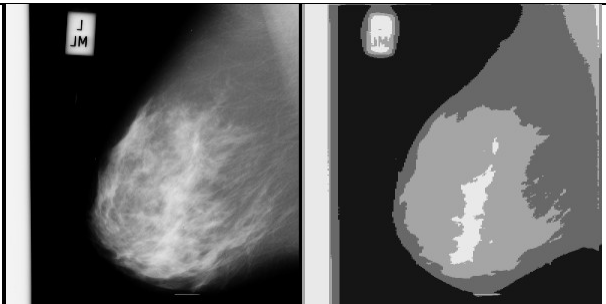
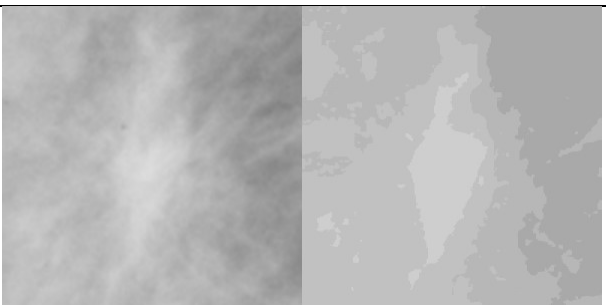
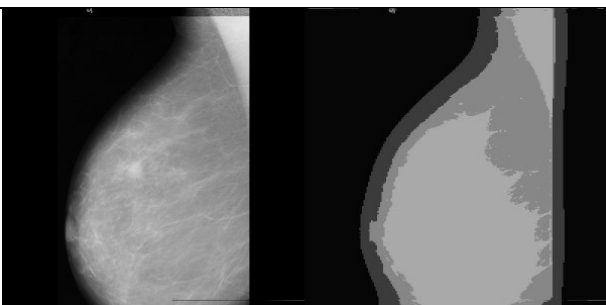
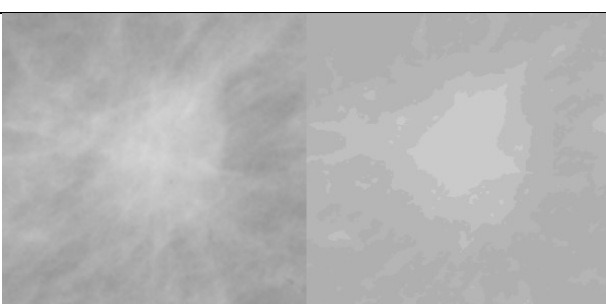
Продолжение таблицы - Результаты обработки изображений

1	2
Mdb117_1	
Mdb120	
Mdb120_1	
Mdb124	
Mdb124_1	

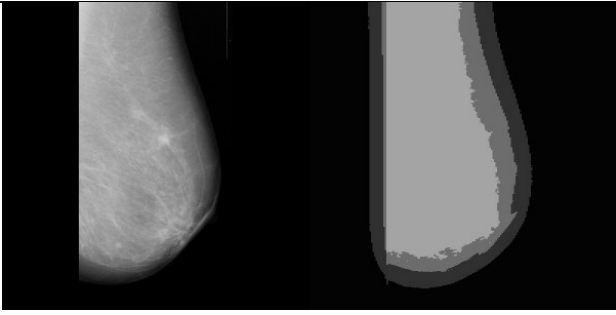
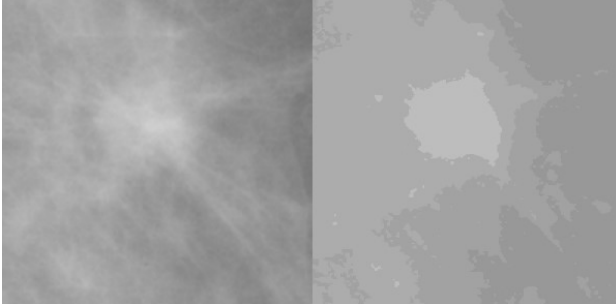
Продолжение таблицы - Результаты обработки изображений

1	2
Mdb134	
Mdb134_1	
Mdb141	
Mdb141_1	
Mdb144	

Продолжение таблицы - Результаты обработки изображений

1	2
Mdb144_1	
Mdb145	
Mdb145_1	
Mdb155	
Mdb155_1	

Окончание таблицы - Результаты обработки изображений

1	2
Mdb158	
Mdb158_1	

Представленным методом фрактальной сегментации изображений, можно сегментировать опухоль на маммологических снимках. Кластеризация методом k-средних удачно проходит на изображениях, обработанных пирамидальной сегментацией в силу того, что после неё происходит фильтрация возможных шумов на изображении и становится возможным более четко определить границы контуров опухоли. Данный метод сегментации имеет большую эффективность, чем метод оценки размера фрактала по длине контура, а также он многих случаях позволяет решить проблему разделения областей, имеющих границу с разрывами (в отличие от метода броуновской функции).

Заключение

Используя метод фрактальной сегментации, можно сгладить или выделить некоторые детали изображения, которые впоследствии можно будет сегментировать от исходного изображения.

В ходе выполнения научно-исследовательской работы были достигнуты следующие цели:

- произведен обзор алгоритмов фрактальной сегментации по итогам которого для реализации был выбран пирамидальный алгоритм;
- представленный алгоритм сегментации реализован на маммограммах;
- тестирование реализованного алгоритма проведено на маммографических снимках.

Тестирование показало, что данный алгоритм хорошо работает на предоставленных тестовых изображениях.

В результате тестирования был сделан вывод о том, что кластеризация методом k -средних удачно проходит на изображениях, обработанных пирамидальной сегментацией в силу того, что после неё происходит фильтрация возможных шумов на изображении и становится возможным более четко определить границы контуров опухоли.

Таким образом, можно сделать вывод о том, что цель работы выполнена, все поставленные задачи реализованы в полном виде.

Список использованных источников

1. Manisha Bhagwat1. Simplified Watershed Transformation [электронный ресурс]. Режим доступа: <http://www.csjournals.com/IJCSC/PDF1-1/34.pdf> (дата обращения: 12.03.21).
2. Feature Detection and Description [электронный ресурс]. Режим доступа: <https://opencv-python-tutroals.readthedocs.io> (дата обращения: 16.02.21).
3. В.Т. Фисенко, Т.Ю. Фисенко. Фрактальные методы сегментации текстурных изображений. - ИЗВ. ВУЗОВ. ПРИБОРОСТРОЕНИЕ. 2013. Т. 56, № 5. – с.63-70.
4. Сегментация изображения с OpenCV и Python [электронный ресурс]. Режим доступа: <https://pythonru.com/biblioteki/segmentacija-izobrazhenija-s-opencv-i-python.html> (дата обращения: 6.03.21).