

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

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

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

01.04.02 Прикладная математика и информатика
(код и наименование направления подготовки)

Математическое моделирование
(направленность (профиль))

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему «Выявление потенциальной патологии на легочном (сосудистом) рисунке»

Студент	<u>Д.И. Краснов</u> (И.О. Фамилия)	_____ (личная подпись)
Научный руководитель	<u>А.В. Очеповский</u> (И.О. Фамилия)	_____ (личная подпись)

Руководитель программы д.ф.-м.н., профессор С.В. Талалов _____
(ученая степень, звание, И.О. Фамилия) (личная подпись)
« _____ » _____ 2019 г.

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский _____
(ученая степень, звание, И.О. Фамилия) (личная подпись)
« _____ » _____ 2019 г.

Тольятти 2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 АНАЛИЗ СОСТОЯНИЯ ТЕКУЩЕГО АЛГОРИТМА ВЫДЕЛЕНИЯ ГРАНИЦ И ПОИСКА ПАТОЛОГИИ НА ИЗОБРАЖЕНИИ	8
1.1 Обзор ранее разработанного программного обеспечения для выделения границ легочного (сосудистого) рисунка	8
1.2 Анализ языка технических расчетов MATLAB	20
1.3 Формализация требований и постановка задач к разрабатываемому программному обеспечению	21
2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ПОТЕНЦИАЛЬНОЙ ПАТОЛОГИИ НА ЛЕГОЧНОМ (СОСУДИСТОМ) РИСУНКЕ	23
2.1 Реализация алгоритмов управления базой данных	23
2.2 Реализация алгоритма разбиения изображения на секторы	26
2.3 Реализация алгоритма поиска ядра патологии.....	28
2.4 Реализация алгоритма установки границ поиска патологии	33
3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	36
3.1 Соответствие разработанного программного обеспечения формализованным требованиям	36
3.2 Тестирование работы алгоритмов программного обеспечения	37
3.3 Сравнение скорости работы старых и новых алгоритмов программного обеспечения	43
3.4 Обновление интерфейса программного обеспечения	45
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	58
ПРИЛОЖЕНИЕ А	62
ПРИЛОЖЕНИЕ Б	63
ПРИЛОЖЕНИЕ В	64
ПРИЛОЖЕНИЕ Г	65
ПРИЛОЖЕНИЕ Д	66
ПРИЛОЖЕНИЕ Е	67

ВВЕДЕНИЕ

Актуальной проблемой в современном мире является туберкулез, который является одной из главных причин смерти. В статистике за 2015 год говорится о том, что более 10 миллионов человек по всему миру были больны туберкулезом, а около 2 миллионов умерли от него в тот же год. Для людей с ВИЧ-инфекцией туберкулез также является одной из главных причин смертности: на состояние 2015 года туберкулез стал причиной смертельного исхода для более чем 35% случаев среди людей с вирусом иммунодефицита.

Основываясь на той же статистике за 2015 год, приблизительно у 480 000 человек было замечено развитие МЛУ-ТБ (туберкулез с множественной лекарственной устойчивостью) на всех континентах. В дополнение к этой статистике у более чем ста тысяч людей была приобретена устойчивость к рифампицину, который является самым эффективным препаратом первой линии. Этим людям также требовался курс лечения от МЛУ-ТБ. Для Российской Федерации, Республики Индии и Китайской Народной Республики проблема стоит особенно остро, только для этих трех стран в сумме случается около половины всех случаев заболевания МЛУ-ТБ.

Начиная с нулевых годов текущего века заболеваемость населения туберкулезом по общим показателям снижалась приблизительно на полтора процента в год. Стратегия по ликвидации туберкулеза заключается в ускорении снижения темпа заболевания до 4-5% в год, чтобы к 2020 году достичь контрольных показателей. Статистика за пятнадцатилетний период, начиная с 2000 года, говорит о спасении 49 миллионов человек благодаря своевременной диагностике болезни и ее лечению. Поэтому одной из главных задач в области здравоохранения в рамках недавно принятых Целей в области устойчивого развития является абсолютная ликвидация эпидемии туберкулеза к 2030 году [21].

В Российской Федерации множеству людей требуется проведение всевозможных обследований с использованием рентгенографии.

Рентгенологическое исследование – это применение рентгеновского излучения в медицине для изучения строения и функций различных органов и систем и распознавания заболеваний [22].

На данный момент в рентгенологии имеются такие методы исследования как специальные, вспомогательные и общие. К общим методам можно отнести компьютерную томографию (КТ), магнитно-резонансную томографию (МРТ), флюорографию и некоторые другие.

Специфическое свойство поглощения рентгеновского излучения, которое зависит от плотности определенных тканей организма человека, заложено в основе компьютерной томографии. Во время томографического рентгеновского исследования на изучаемую специалистами область воздействует пучок рентгеновских лучей, при его прохождении через ткани человека с различной плотностью происходит поглощение лучей. При их поглощении появляются послойные изображения среза тела пациента. Высокотехнологичные компьютеры производят обработку полученных в результате воздействия лучей данных, а затем преобразуют их в трехмерные изображения, с помощью которых выявляются особенности исследуемого у пациента участка тела или органа.

Несмотря на эффективность метода, компьютерная томография производит сильное лучевое воздействие на организм человека, в результате чего многократное ее применение недопустимо.

При использовании магнитно-резонансной томографии получение данных происходит при применении сильного магнитного поля, который называется ядерно-магнитным резонансом. В результате подобного резонанса происходит изменение положения атомов водорода в организме пациента. Аппарат направляет электромагнитные импульсы, затем происходит преобразование в трехмерные изображения эффекта, улавливаемого специальным оборудованием в организме человека.

При проведении подобного обследования происходит воздействие рентгеновских лучей на ткани и органы человека до десяти секунд, но для

проведения полного МРТ-обследования требуется от 10 до 20 минут времени, также со стороны пациента требуется сохранять тело в неподвижном состоянии, а при проведении МРТ-обследования у детей и вовсе применяется наркоз.

Так как магнитно-резонансная томография является для человека совершенно безвредным методом обследования, в современном мире она нашла широкое применение в области выявления всевозможных проблем с органами и тканями, а также МРТ считается наиболее точным методом в диагностике обширного спектра патологий. Но учитывая все достоинства проведения обследования с помощью МРТ, у него имеются такие недостатки, как не полностью достоверная визуализация изображений полых органов (а к ним в том числе относятся и легочные доли), в результате чего часто для подтверждения найденного на МРТ-снимке усиления специалистами назначается прохождение флюорографии.

Флюорография является рентгенологическим исследованием, которое основано на фотографировании изображения на флуоресцентном экране. Подобное изображение образуется при прохождении рентгеновских лучей сквозь пациента, в результате чего происходит неравномерное их поглощение тканями и органами. Самым широко применяемым методом обследования, который основан на принципе флюорографии, является фотографирование органов грудной клетки пациента. В первую очередь флюорография используется для обнаружения туберкулеза и новообразований легких, также метод флюорографии позволяет обнаружить усиления, симптомы которых не проявляются.

Основные преимущества флюорографии перед другими методами рентгенологического исследования – большая пропускная способность при малой затрате времени на каждое исследование, возможность приблизить обследование к населению, экономичность, относительно небольшая профессиональная вредность [23].

Для проведения рентгенологического обследования разработаны стационарные и мобильные флюорографические аппараты, но в комплект программного обеспечения таких аппаратов, приобретаемых медицинскими учреждениями в Российской Федерации, не входят программы для анализа снимков и выявления на них усилений.

Исходя из статистики существенного увеличения количества болеющих туберкулезом людей по всей России, которым требуется проведение всевозможных обследований с использованием методов рентгенографии, актуальной задачей является повышение скорости проведения подобных обследований путем частичной автоматизации процесса анализа снимков.

Исходя из актуальности рассматриваемой темы определены объект и предмет исследования, а также поставлена цель диссертационного исследования.

Объект диссертационного исследования: процесс выявления потенциальной патологии на рисунке с использованием языка технических расчетов MATLAB.

Предмет диссертационного исследования: автоматизация процесса выявления потенциальной патологии легочного (сосудистого) рисунка.

Цель диссертационного исследования: выявление потенциальной патологии на легочном (сосудистом) рисунке.

Рабочей гипотезой диссертационного исследования является предположение, что повышение скорости обработки флюорографических снимков достигается путем автоматизации и оптимизации процесса выявления потенциальной патологии.

Научная новизна исследования состоит в том, что в работе впервые применен поиск ядра патологии для выявления потенциального усиления на рисунке. Практическая значимость работы заключается в использовании алгоритмов выявления потенциальной патологии в городских больницах и поликлиниках.

Диссертационное исследование включает несколько основных этапов:

- выявление проблемы и обоснование ее актуальности;
- реализация алгоритмов в соответствии с сформулированной гипотезой и поставленными задачами;
- формирование выводов и оценка полученных результатов.

В ходе выполнения диссертационного исследования должно быть разработано программное обеспечение для выявления потенциальной патологии на легочном (сосудистом) рисунке.

Диссертационное исследование состоит из введения, трех глав и заключения.

Во введении описывается актуальность рассматриваемой темы, определяются объект и предмет диссертационного исследования, ставится цель и выявляются задачи, а также формулируется гипотеза исследования.

В первой главе происходит анализ состояния вопроса, а именно обзор существующего программного обеспечения, формализация требований к разрабатываемым алгоритмам и описание использованного языка программирования для разработки программного обеспечения.

Во второй главе описывается разработка алгоритмов программного обеспечения для выявления потенциальной патологии на легочном (сосудистом) рисунке.

В третьей главе происходит тестирование разработанного программного обеспечения в соответствии с формализованными требованиями к программному продукту.

В заключении подводятся итоги разработки программного обеспечения, формируются окончательные выводы по рассматриваемой теме.

На защиту выносятся результаты исследования, которые заключаются в выявлении потенциального усиления на основе применения поиска ядра патологии.

1 АНАЛИЗ СОСТОЯНИЯ ТЕКУЩЕГО АЛГОРИТМА ВЫДЕЛЕНИЯ ГРАНИЦ И ПОИСКА ПАТОЛОГИИ НА ИЗОБРАЖЕНИИ

1.1 Обзор ранее разработанного программного обеспечения для выделения границ легочного (сосудистого) рисунка

Ранее в Тольяттинском Государственном Университете при выполнении бакалаврской работы был разработан алгоритм выделения границ легочного (сосудистого) рисунка с помощью языка технических расчетов MATLAB. Разработанное программное обеспечение состоит из нескольких основных алгоритмов:

- выделение границ легочного рисунка;
- поиск патологий на флюорографических снимках;
- проверка масок.

Данное программное обеспечение имеет алгоритм загрузки в тело программы набора цветов восьмибитного изображения – цифровой снимок флюорографии. Затем для текущего изображения происходит запуск алгоритма выделения границ области легких, который использует метод бинаризации изображения. Для проведения проверки оригинального снимка на наличие на нем образований загружаются маски с усилениями, содержащие наборы цветов полутонового изображения, эквивалентно алгоритму загрузки основных изображений, после чего происходит запуск алгоритма поиска патологий на выделенной ранее области цифрового снимка, использующий метод поиска массива внутри массива.

Обнаружение патологии происходит с указанной в теле программы погрешностью – различием оттенков цвета между маской с патологией и набором пикселей на загруженном цифровом снимке. Кроме того, в теле алгоритма устанавливается допустимая погрешность не совпадающих с шаблоном патологии элементов.

Если будет обнаружено совпадение между шаблоном и набором пикселей загруженного цифрового снимка, алгоритм программы произведет проверку на

соответствие найденной маски эталону патологии. В случае прохождения проверки маской будет создан новый шаблон усиления на основе элементов, которые совпали с маской при поиске на снимке, а также он сохраняется к уже имеющимся шаблонам в указанную директорию. Обнаруженные границы патологии на флюорографическом снимке выделяются, после чего снимок с выделенными границами также сохраняется в соответствующую директорию.

Применение алгоритма поиска патологий происходит в цикле до тех пор, пока для загруженного цифрового снимка не будет произведена проверка по всем имеющимся маскам с усилениями. Оригинальное изображение считается нормальным при условии, что алгоритмы программы не обнаружат образований при использовании всех эталонных шаблонов.

Алгоритм выделения границ легочного рисунка производит выделение области на снимке, которая содержит легочные доли. Элементы изображения за пределами области легких окрашиваются в белый цвет, в том числе и фон флюорографического снимка.

Блок-схема алгоритма выделения границ легочного рисунка представлена на рисунке 1.1.

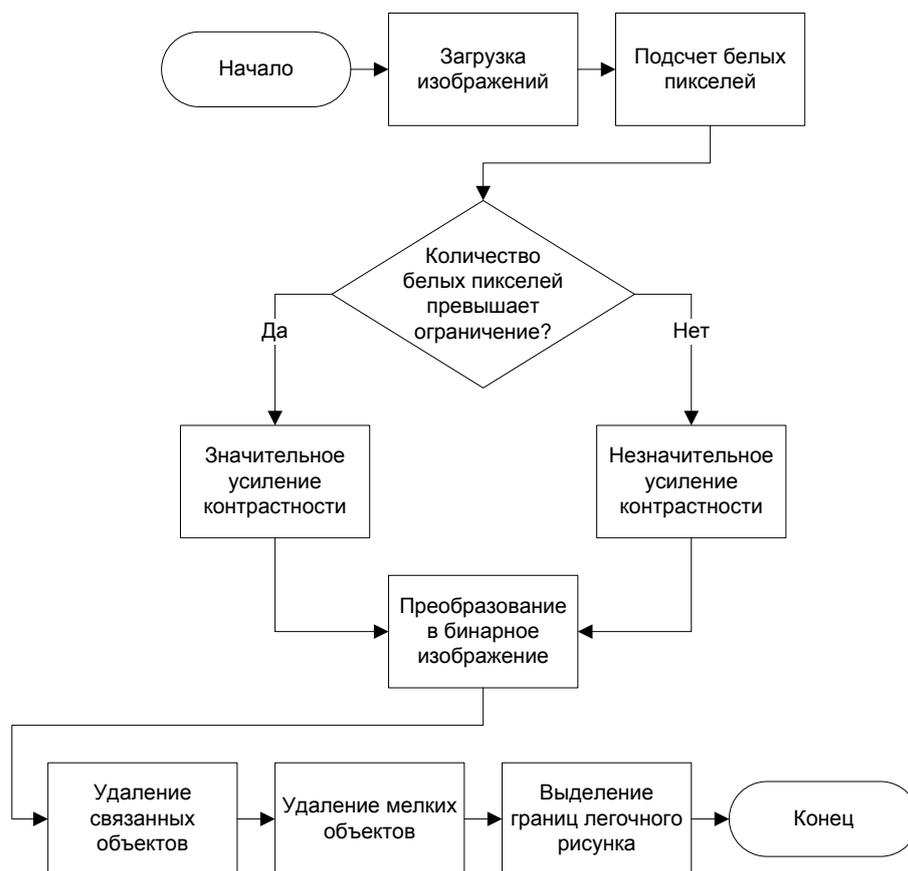


Рисунок 1.1 – Блок-схема алгоритма выделения границ легочного рисунка

Для корректного выделения границ легочных долей на снимке флюорографии изначально увеличивается контрастность оригинального полутонового изображения, степень контрастности зависит от количества оттенков белого цвета.

Существует два типа снимков рентгеновского аппарата: вид грудной клетки спереди, а также вид сбоку. Изображения легочной доли для вида сбоку имеют затемнение на внутренней области рисунка легких, а также на подобных снимках содержится множество пикселей белого цвета вне области грудной клетки, чего не имеется на флюорографических снимках для вида спереди. Исходя из этого для проведения выделения границ легочных долей на изображении необходимо провести увеличение контрастности снимка в зависимости от его вида, после чего возможно корректное преобразование в бинарный рисунок.

Далее происходит преобразование цифрового снимка в бинарное изображение, состоящее только из двух цветов – черного и белого, используется метод отсечения по порогу яркости. Также на изображении удаляются все лишние объекты за пределами легочных долей, а также происходит очистка области легких от мелких объектов, таких как точки и пятна.

Затем на оригинальном изображении происходит окраска белым цветом области, соответствующей черному цвету на бинарном изображении, и в результате выводится исходное изображение с выделенными границами легочных долей.

После окончания работы алгоритма выделения границ на цифровом снимке производится сохранение результата выделения в директорию «Borders». В начале имени оригинального изображения приписываются символы «S_» (англ. selected – выбранный, выделенный), а все последующие снимки после прохождения через этап выделения легочных долей также будут отмечены подобными символами.

На рисунках 1.2 и 1.3 представлено сравнение оригинальных снимков флюорографии и полученных при выполнении работы программы изображений, где произведено выделение границ легких как для вида сбоку, так и для вида спереди. В левой части представлены оригинальные снимки, а в правой части полученные изображения с выделением границ.

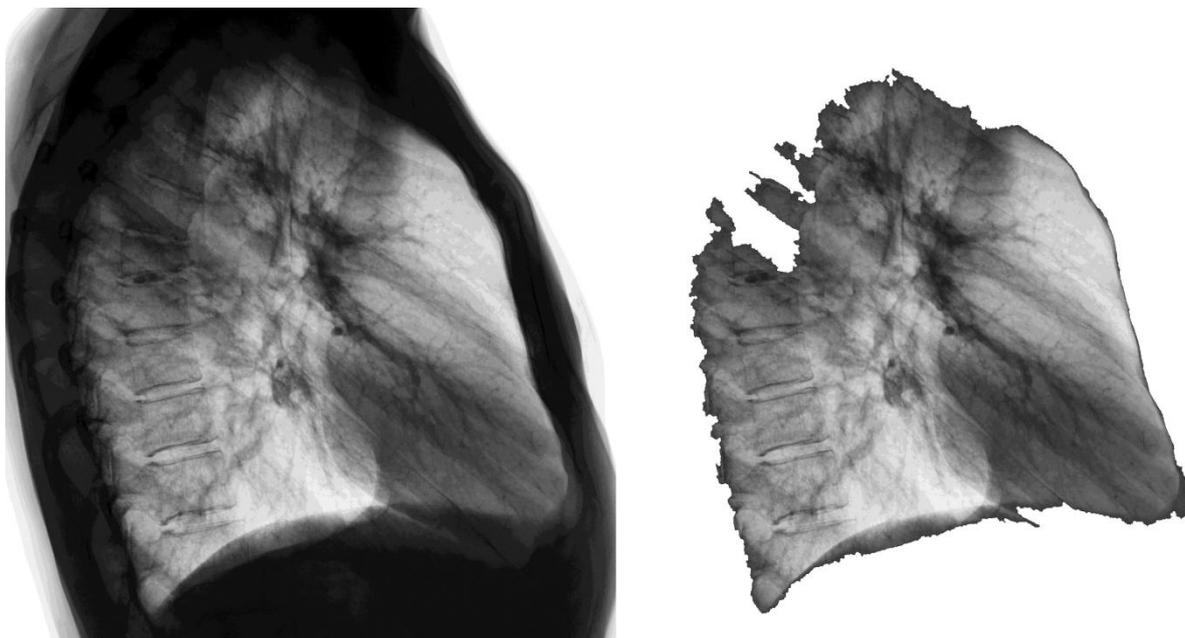


Рисунок 1.2 – Сравнение исходного флюорографического снимка и изображения с выделенными границами легочного рисунка (вид сбоку)

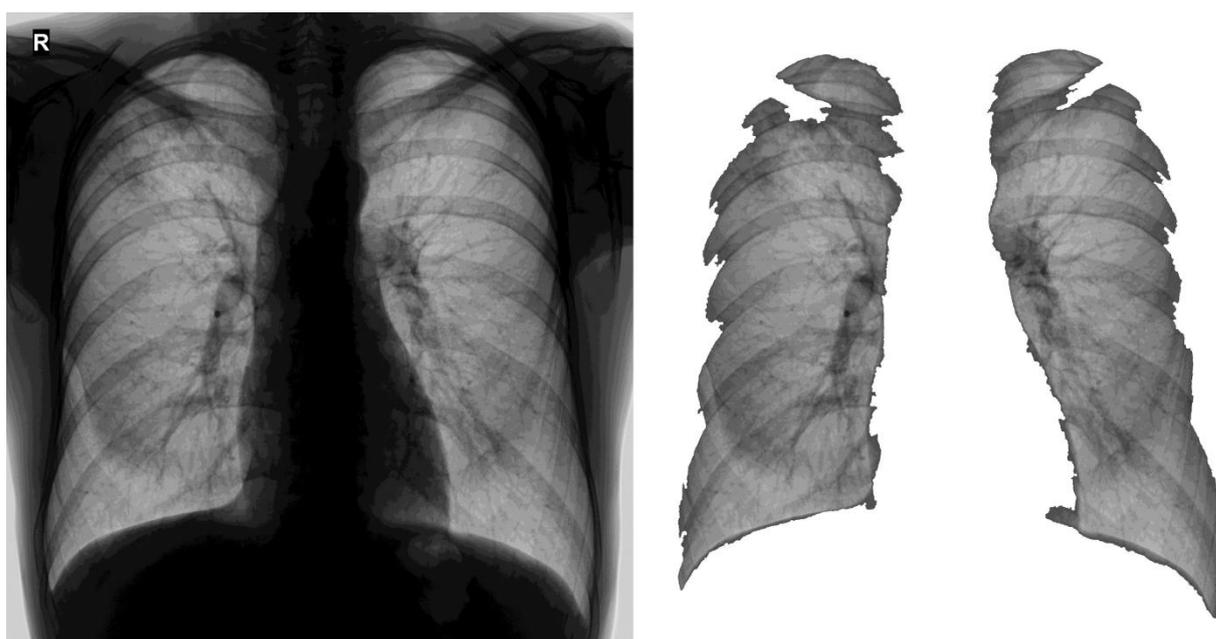


Рисунок 1.3 – Сравнение исходного флюорографического снимка и изображения с выделенными границами легочного рисунка (вид спереди)

Проведение поиска усилений на изображении было реализовано с помощью метода поиска маски с патологией на флюорографическом снимке. Учитывая факт, что каждое изображение – это набор цветов и оттенков, находящийся в массиве, а снимки рентгеновского аппарата также возможно

относить к изображениям, допустимо сделать вывод, что проведение поиска усиления на основе масок эквивалентно поиску массива внутри массива.

Работа алгоритма поиска усилений построена по следующим принципам:

- происходит загрузка массива цветов полутонового (черно-белого) изображения – флюорографический снимок;
- алгоритмы производят выделение области легочных долей на загруженном изображении;
- происходит последовательная загрузка массивов цветов шаблонов с усилениями (маски) для последующей проверки оригинального снимка на наличие патологий;
- Алгоритмы осуществляют поиск усиления внутри флюорографического снимка.

Блок-схема алгоритма поиска патологий на флюорографических снимках представлена на рисунке 1.4.

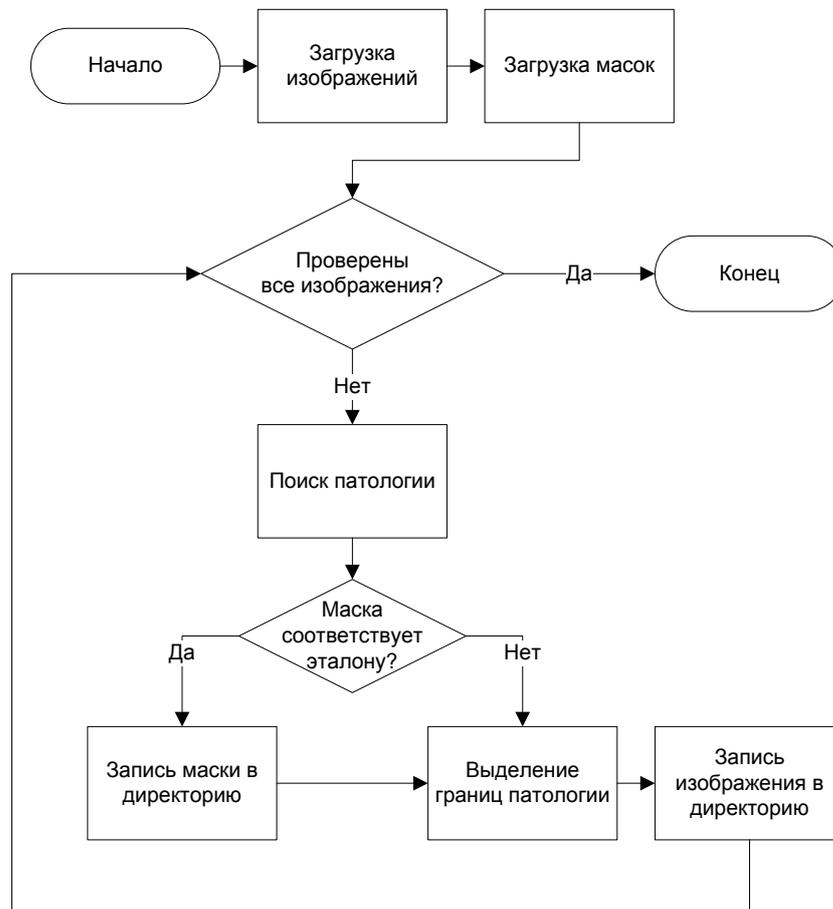


Рисунок 1.4 – Блок-схема алгоритма поиска патологий

Поиск усиления на изображении происходит с указанной внутри алгоритма погрешностью оттенков между шаблоном маски и набором цветов на оригинальном снимке. Кроме того, в алгоритме указывается допустимая разница не совпадающих с шаблоном маски оттенков цвета в процентах.

После произведения загрузки шаблонов с патологиями в программное обеспечение в цикле запускается поиск усиления внутри снимка рентгенологического аппарата. Произведение поиска происходит с указанным в теле программы отличием оттенков цвета между текущей маской и наборами пикселей на проверяемом изображении. Поиск происходит только на области, соответствующей легочным долям, так как при поиске исключаются оттенки белого цвета (255 оттенков), соответствующие фону изображения. Внутри алгоритма при использовании специальной переменной задается допустимое различие оттенков цвета при проведении поиска, таким образом происходит поиск с допустимой погрешностью и исключается четкое соответствие проверяемой маске при сохранении ее эталонных свойств.

После выявления совпадения маски с усилением и набором оттенков на оригинальном снимке запускается алгоритм создания нового шаблона патологии, который учитывает совпадающие с текущей маской оттенки, а затем сохраняет новый шаблон. Затем алгоритмы программы производят выделение области обнаруженной патологии, после чего создается новое изображение с уже выделенным усилением.

При обнаружении совпадения на снимке флюорографии по одной из масок происходит выделение границ области найденного совпадения. Такой границей будет считаться квадрат с белым оттенком (255 оттенков для полутонового изображения), имеющий толщину в два пикселя. Область указанного квадрата на десять пикселей по каждой стороне превышает найденную область совпадения по маске. После обрисовки границы образования происходит сохранение изображения с уже выделенной областью в директорию «Pathologies», а в начале имени файла будет приписан символ

решетки «#». Этот символ добавляется к каждому снимку с обнаруженной на нем патологией.

В случае, если усиление не будет обнаружено при использовании первого шаблона с патологией, будет произведен поиск по следующей маске, а оригинальный снимок является нормой до тех пор, пока алгоритмы не выявят усиление при использовании всех существующих масок с патологиями.

Существует также эталон маски с усилением, который используется алгоритмом проверки масок, и перед запуском алгоритма для поиска усилений на изображении необходимо провести проверку шаблонов с патологиями в директории «Masks» на их соответствие эталону.

Шаблон маски с усилением является эталоном для поиска патологий при условии, что он является изображением, содержащим глубину цвета 8 бит (256 оттенков), что соответствует полутоновому изображению. Внутри алгоритма имеется функция, записывающая информацию о загружаемом изображении в переменную, с помощью которой в последствии происходит проверка глубины цвета. Если маска соответствует указанным параметрам, то она может быть использована для последующего поиска усилений, а при несоответствии глубины цвета с помощью специальной функции производится преобразование шаблона в восьмибитное изображение.

Происходит преобразование RGB-составляющих цветов изображения в соответствующие им оттенки цветов полутонового изображения, а затем новый корректный шаблон сохраняется в директорию «Masks» без изменения имени, и теперь может использоваться для проверки снимков рентгеновского аппарата на наличие образований.

Блок-схема алгоритма проверки масок представлена на рисунке 1.5.

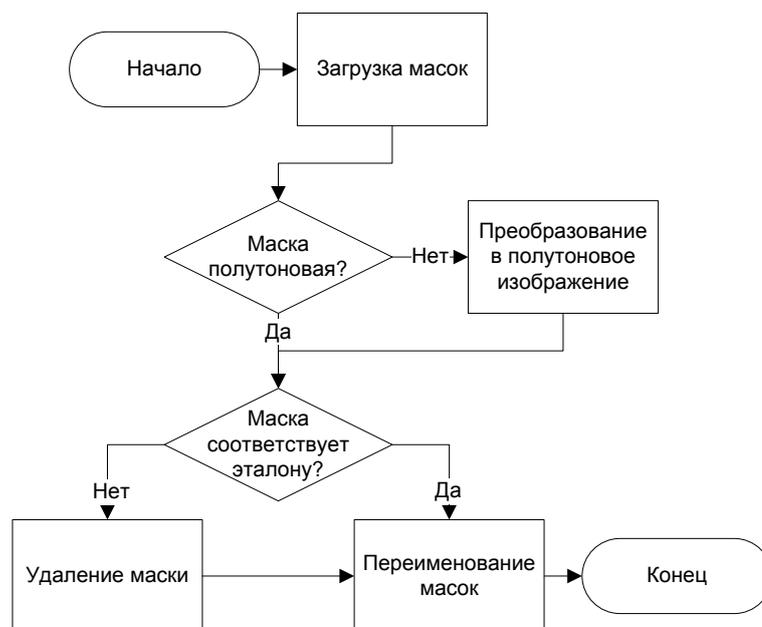


Рисунок 1.5 – Блок-схема алгоритма проверки масок

Аналогичный алгоритм используется при сохранении новой маски с патологией при ее выявлении.

Маска с изображением усиления будет считаться эталоном, если значения оттенков цвета не ниже 70 занимают половину размера шаблона. В том случае, если после проведения проверки шаблон с усилением не будет соответствовать установленным эталоном требованиям, происходит удаление проверяемой маски.

Для произведения расчета количества пикселей, содержащих оттенок цвета менее 70, в алгоритм добавлена соответствующая переменная. При условии, что количество указанных пикселей в шаблоне с усилением больше половины размерности, новая маска сохраняется к остальным шаблонам в директорию «Masks». Также существует переменная, содержащая значение количества шаблонов в директории, при этом любая из масок с усилением содержит в своем имени определенный номер в порядке нумерации. При сохранении нового шаблона происходит приписка к имени файла следующего номера, после чего возможно проведение поиска усиления при учете нового добавленного шаблона.

Для исключения ошибок в алгоритмах программного обеспечения при работе с шаблонами масок из директории «Masks» происходит задание имени масок с усилениями в порядке нумерации. Для проведения нумерации масок реализован специальный алгоритм, который задает значения цифр в имени маски в требуемом порядке, а для исключения ошибок при обычной приписке цифры в порядке нумерации к имени маски производится двойное изменение имени файлов.

Изначально файлам задаются имена по типу «Rename_%d.jpg», где числу с требуемым порядковым номером соответствует %d. После этого происходит повторный алгоритм переименования масок, где происходит задание имен для файлов по типу «Mask_%d.jpg», где числу с необходимым порядковым номером также соответствует %d.

Для использования разработанных алгоритмов и их тестирования создан графический интерфейс, который включает в себя окно для вывода информации и следующие кнопки: «проверка масок», «выделение легочного рисунка», «поиск патологий», «поиск патологий на области».

Запуск алгоритма проверки масок с усилениями происходит при нажатии кнопки «проверка масок». Алгоритм проверяет соответствие находящихся в директории «Masks» шаблонов с усилениями на соответствие эталону.

Запуск алгоритма выделения границ легочных долей происходит при нажатии кнопки «выделение легочного рисунка», в процессе работы которого из директории «Images» происходит загрузка снимков. После загрузки изображений к каждому из них применяется указанный алгоритм для выделения области легких с использованием бинаризации снимков. Затем происходит сохранение в директорию «Borders» всех полученных при выполнении алгоритма изображений.

Запуск алгоритма поиска патологий на оригинальных снимках, загружаемых из директории «Images», происходит при нажатии кнопки «поиск патологий». Изначально происходит загрузка масок с усилениями из директории «Masks» для проверки оригинальных снимков на возможные

патологии, после чего запускается алгоритм поиска усилений на изображениях. В конце работы алгоритма происходит сохранение полученных изображений с припиской к их имени символа решетки «#» в директорию «Pathologies».

Запуск алгоритма выделения границ легочных долей на снимках флюорографии, который основан на методе бинаризации изображения, происходит из директории «Images» при нажатии кнопки «поиск патологий на области». Для использования алгоритма поиска патологий изначально загружаются из директории «Masks» шаблоны с усилениями, после чего выполняется проверка снимков с выделенными границами легочных долей на наличие образований (усилений). При завершении работы алгоритма происходит сохранение изображений с припиской к их имени символа «S#» в директорию «Pathologies».

В качестве примера на рисунке 1.6 представлен графический интерфейс программного обеспечения, где в окне вывода информации указаны этапы выполнения алгоритма поиска патологий на исходных флюорографических снимках.

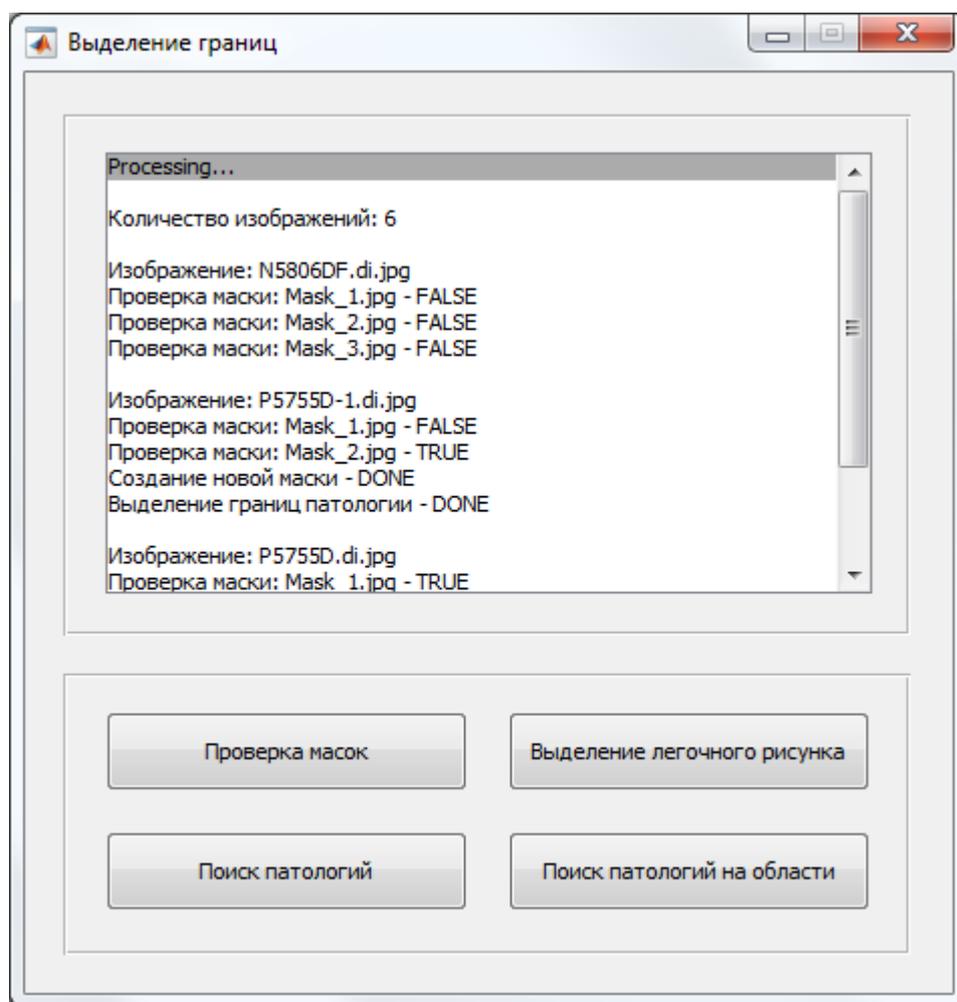


Рисунок 1.6 – Графический интерфейс программного обеспечения

В окне интерфейса пользователя имеется информация о количестве изображений, которые будут использованы в работе алгоритмов, указывается текущий обрабатываемый снимок, представляются результаты проверки каждой маски на наличие патологии, а также все действия алгоритмов при сохранении изображений или масок.

Разработанное в результате выполнения бакалаврской работы программное обеспечение является самообучаемым, но в числе недостатков имеется низкая скорость работы алгоритма поиска патологий при наличии множества шаблонов с усилениями. Также на производительности сказывается малая оптимизации областей поиска, поэтому данный программный продукт имеет перспективу улучшения и развития.

1.2 Анализ языка технических расчетов MATLAB

MATLAB является языком высокого уровня, а также интерактивной средой программирования для проведения численных расчетов и визуализации их результатов. При использовании MATLAB возможно провести анализ данных, разработать новые алгоритмы или создать необходимые модели и приложения.

Язык MATLAB, используемые инструменты и широкий набор математических функций позволяют проводить исследования различных подходов, а также получать решения задач быстрее, чем при аналогичном использовании языков программирования C/C++ или Java.

Упрощенная разработка новых алгоритмов, а также сокращенное время выполнения типовых задач выделяет MATLAB среди традиционных языков программирования.

Представляя основу всего семейства продуктов компании MathWorks, MATLAB используется в качестве главного инструмента при решении научных и прикладных задач. Широкое применение MATLAB находит в областях моделирования объектов и разработке систем управления, а также часто используется при обработке сигналов и изображений.

Упрощенную работу с матрицами реальных, комплексных или аналитических данных, а также с их структурами и таблицами позволяет устройство ядра MATLAB.

Большим плюсом MATLAB является наличие функций линейной алгебры и быстрого преобразования Фурье. Также имеются функции для работы с полиномами, базовыми статистиками и численными решениями дифференциальных уравнений.

Все встроенные функции ядра MATLAB разработаны и оптимизированы специалистами и работают быстрее их эквивалентов на C/C++ или Java [25], поэтому использование MATLAB для эффективной реализации алгоритмов выявления потенциальной патологии на легочном (сосудистом) рисунке является отличным решением.

1.3 Формализация требований и постановка задач к разрабатываемому программному обеспечению

Программное обеспечение для выделения границ легочного (сосудистого) рисунка должно содержать следующие новые алгоритмы:

- управление базой данных;
- разбиение изображения на секторы;
- поиск ядра патологии;
- установка границ поиска патологии.

Алгоритм управления базой данных должен включать в себя такие алгоритмы, как создание новой базы данных, а также добавление и удаление записей. После обнаружения патологии на флюорографическом снимке должен быть создан новый шаблон маски с патологией и добавлен в базу данных для последующего использования при поиске новых патологий.

Алгоритм разбиения изображения на секторы должен загружать флюорографический снимок из указанной директории и производить его разделение на прямоугольные области, в которых будет производиться поиск патологии.

Алгоритм поиска ядра патологии должен производить первичный поиск потенциальной патологии на загруженном флюорографическом снимке, а также сохранять координаты найденного ядра для последующего поиска усиления по шаблонам с масками патологий внутри соответствующей координатам области (сектора).

Алгоритм установки границ поиска патологии должен определять по сохраненным координатам ядра область для последующего поиска патологии с использованием шаблонов с масками патологий из базы данных. Границы области (сектора) устанавливаются по аналогии с алгоритмом разбиения изображения на секторы.

Исходя из объекта и предмета исследования и учитывая особенности существующего программного обеспечения, на основе формализованных

требований к разрабатываемым алгоритмам для достижения поставленной цели были определены следующие задачи:

1) изучить методы управления базой данных в среде технических расчетов MATLAB, а также способы разбиения изображения на области (секторы);

2) разработать архитектуру алгоритмов для управления базой данных и разбиения изображения на области (секторы);

3) разработать программный код алгоритмов для управления базой данных и разбиения изображения на области (секторы), а также поиска ядра патологии на флюорографическом снимке с использованием языка технических расчетов MATLAB;

4) провести тестирование работы новых алгоритмов выявления потенциальной патологии на легочном (сосудистом) рисунке.

В ходе выполнения магистерской диссертации должны быть разработаны новые алгоритмы для выявления потенциальной патологии на легочном (сосудистом) рисунке.

Произведены анализы существующего программного обеспечения и языка программирования новых алгоритмов для выявления потенциальной патологии на легочном (сосудистом) рисунке. Формализованы требования к алгоритмам и определены задачи для достижения поставленной цели.

2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ПОТЕНЦИАЛЬНОЙ ПАТОЛОГИИ НА ЛЕГОЧНОМ (СОСУДИСТОМ) РИСУНКЕ

2.1 Реализация алгоритмов управления базой данных

Для ускорения и оптимизации работы программного обеспечения автоматизированного выделения границ легочного (сосудистого) рисунка создана база данных, содержащая маски с патологиями, которые используются для поиска подобных патологий на флюорографических снимках. База данных состоит из множества записей. Каждая запись является массивом, который содержит набор оттенков цвета маски с патологией.

Пример созданной базы данных, содержащей 6 элементов, приведен на рисунке 2.1.

1x6 cell

	1	2	3	4	5	6	7
1	32x32 uint8	34x34 uint8	40x40 uint8	34x34 uint8	40x40 uint8	32x32 uint8	
2							
3							
4							
5							
6							
7							

Рисунок 2.1 – Пример созданной базы данных

При выборе одного из элементов базы данных можно увидеть набор оттенков цветов, на основе которых производится поиск потенциальной патологии. Также отображены размеры шаблонов масок с патологиями и их глубина цвета (8 бит).

Набор оттенков цвета для элемента в первой ячейке базы данных приведен на рисунке 2.2.

Matrix{1, 1}								
	1	2	3	4	5	6	7	8
1	114	122	114	116	112	104	110	102
2	124	130	119	119	114	104	109	101
3	131	134	121	121	114	101	104	95
4	129	131	119	122	115	98	96	86
5	126	128	118	126	120	98	93	81
6	128	128	119	130	125	102	95	84
7	135	131	120	130	126	103	97	88
8	140	133	118	127	123	101	97	89
9	142	131	137	137	120	113	100	70
10	153	138	138	134	115	110	102	75
11	155	141	141	135	115	108	99	72
12	147	138	145	142	120	109	95	65
13	146	138	147	145	123	111	97	68
14	152	142	147	142	119	111	103	79
15	150	141	147	141	117	109	101	76
16	141	136	146	144	119	106	92	65
17	154	144	139	129	109	98	93	82
18	145	134	131	125	111	103	95	81
19	137	124	120	118	110	104	92	74
20	138	121	113	111	104	99	86	66

Рисунок 2.2 – Набор оттенков цвета для элемента в первой ячейке базы данных

С помощью созданных алгоритмов осуществляется управление элементами в базе данных. Схема алгоритмов управления базой данных представлена на рисунке 2.3.

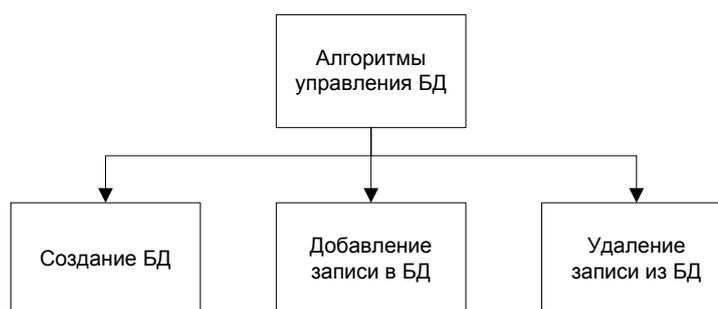


Рисунок 2.3 – Схема алгоритмов управления базой данных.

Массив «Matrix» необходим для последующих записей масок с патологиями в базу данных. Массив является динамическим, так как количество записей в базе данных может изменяться. С помощью команды

«save» динамический массив «Matrix» сохраняется в новую базу данных с именем «DataBase» и с расширением «mat».

Код алгоритма создания базы данных приведен в приложении А.

Управление элементами базы данных состоит из двух основных модулей: добавление новой и удаление существующей записи.

Добавление записи в базу данных происходит в несколько действий. Сначала загружается существующая база данных с помощью команды «load» и проверяется количество записей в ней. Затем в переменную «Mask» помещается изображение маски с патологией, после чего оно записывается в конец базы данных.

Блок-схема алгоритма добавления новой записи в базу данных представлена на рисунке 2.4.

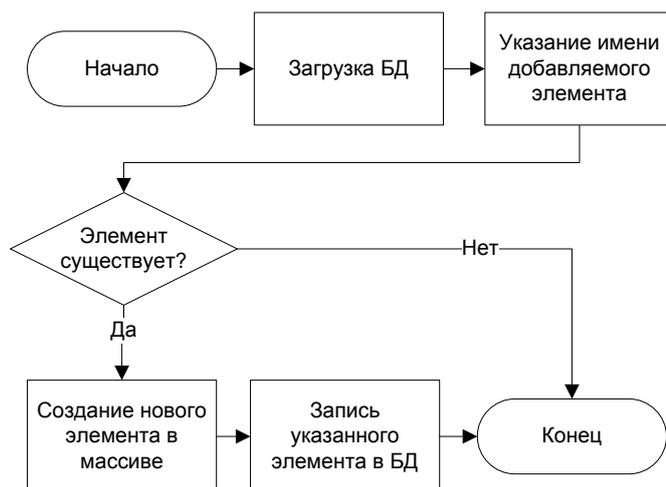


Рисунок 2.4 – Блок-схема алгоритма добавления новой записи в базу данных

Код для алгоритма добавления записи в базу данных приведен в приложении Б.

Удаление записи из базы данных также происходит в несколько действий. Сначала загружается существующая база данных с помощью команды «load» и проверяется количество записей в ней. Затем переменной «del» присваивается номер удаляемой записи. Если такая запись существует, она удаляется, а все записи после нее сдвигаются на одну позицию вниз.

Блок-схема алгоритма удаления записи из базы данных представлена на рисунке 2.5.

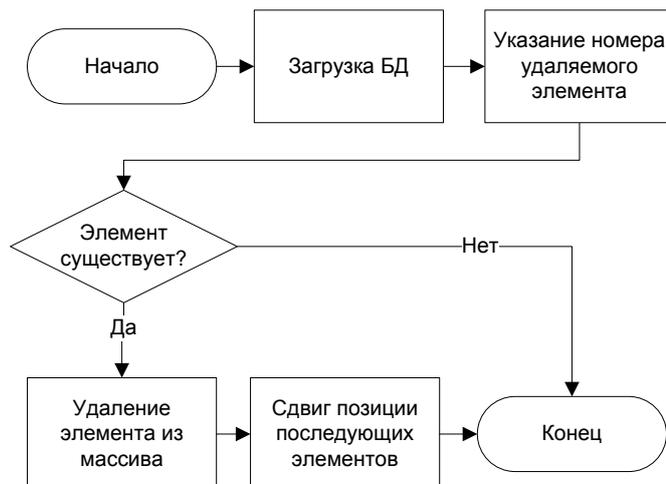


Рисунок 2.5 – Блок-схема алгоритма удаления записи из базы данных

Код для алгоритма удаления записи из базы данных приведен в приложении В.

2.2 Реализация алгоритма разбиения изображения на сектора

Для ускорения и оптимизации поиска патологий на флюорографических снимках разработан алгоритм разбиения изображения на сектора.

Поиск патологии внутри сектора производится быстрее, чем при поиске на всем изображении. Это существенно повышает эффективность работы программы. Чем на большее количество секторов будет разбито изображение, тем быстрее будет проводиться поиск патологии, однако время поиска уменьшается по экспоненте.

Сначала загружается исходное изображение и определяются его размеры в переменных «row» и «column». Затем определяется количество разбиений по строкам и столбцам в переменной «nSector». С помощью этой переменной в двух циклах происходит разбиение изображения на сектора по строкам и столбцам соответственно.

Блок-схема алгоритма разбиения изображения на сектора представлена на рисунке 2.6.

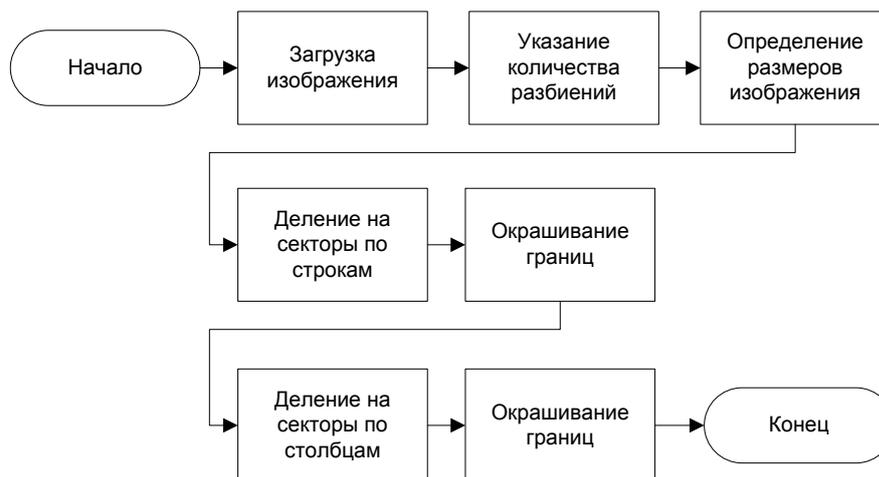


Рисунок 2.6 – Блок-схема алгоритма разбиения изображения на секторы

Для наглядности каждый полученный в результате работы программы сектор был обведен белой рамкой. Пример разбиения оригинального изображения на секторы представлен на рисунке 2.7. Для текущего примера было указано количество разбиений по строкам и столбцам в переменной «nSector» равное 10.

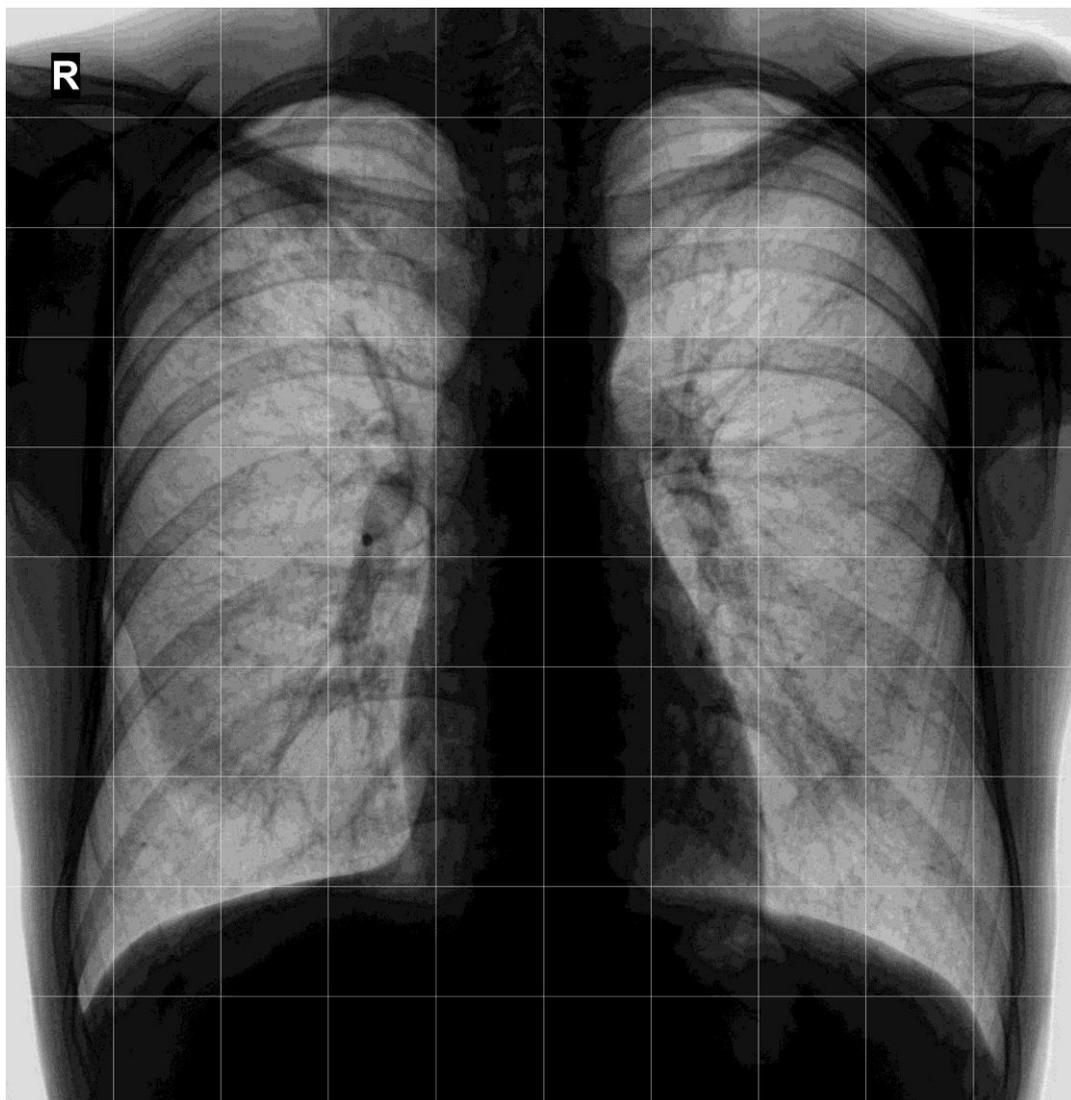


Рисунок 2.7 – Пример разбиения изображения на секторы

Код алгоритма разбиения изображения на секторы приведен в приложении Г.

2.3 Реализация алгоритма поиска ядра патологии

Значения оттенков цвета патологии увеличиваются от ее центра к границам. Самые низкие значения цвета в центре патологий находятся в диапазоне от 25 до 40 оттенка, а самые высокие значения цвета у границ патологий находятся в диапазоне от 60 до 70 оттенка. Следовательно, ядром патологии является набор пикселей с оттенками цвета ниже 40, а минимальный радиус ядра патологии равен 5 пикселям.

Сначала загружается исходное изображение и определяются его размеры в переменных «rowImg» и «columnImg». Радиус патологии присваивается переменной «R», а порог оттенков цвета ядра указывается в переменной «threshold».

Так как патология не может находиться у границ изображения, устанавливается ограничение на область поиска. Для этого в переменной «borderPercent» устанавливается отступ от границ изображения в 10% от числа пикселей в ширину и высоту, а в переменных «rowImgBorder» и «columnImgBorder» вычисляется это количество пикселей для последующего использования в цикле поиска ядра патологии.

После вычисления значений переменных в цикле происходит сравнение оттенков цвета. Если их значение меньше указанного порога, а радиус достигает минимального значения, указанного в переменной «R», ядро считается найденным, его координаты записываются в переменные «iKernel» и «jKernel» для последующей установки границ поиска патологий, а логической переменной «flag» присваивается значение «true», что позволяет в дальнейшем запустить алгоритм поиска патологий.

Блок-схема алгоритма поиска ядра патологии представлена на рисунке 2.8.

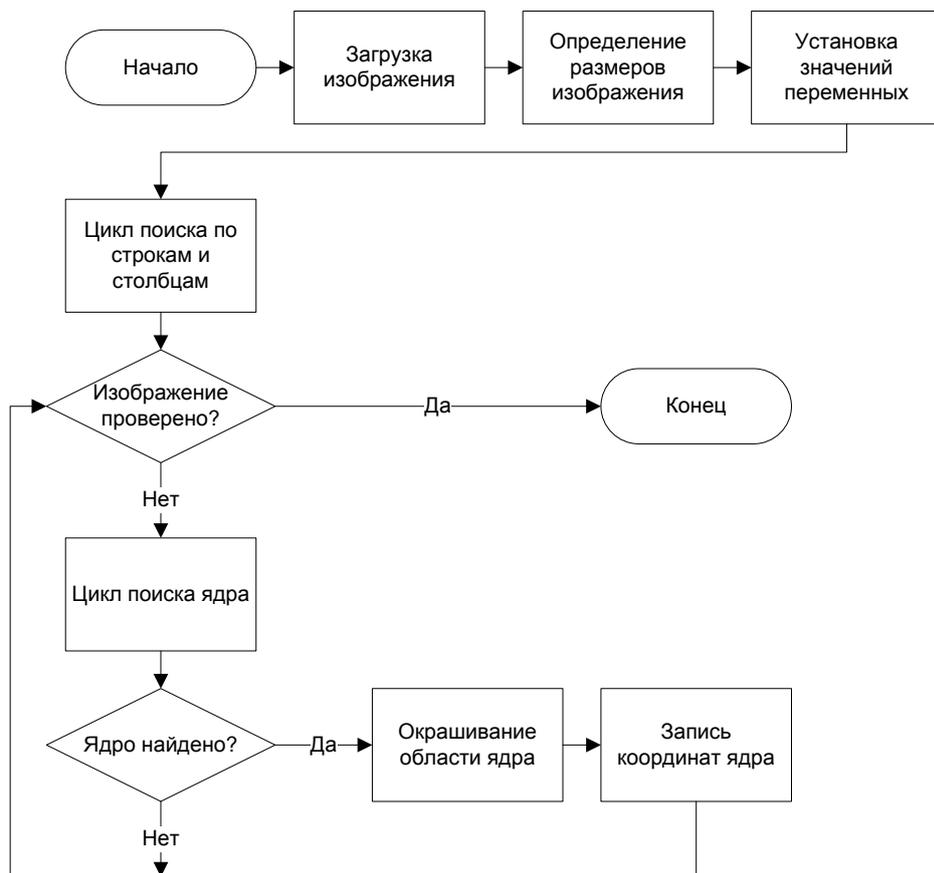


Рисунок 2.8 – Блок-схема алгоритма поиска ядра патологии

Для наглядности границы полученного в результате работы программы ядра патологии выделяются белыми точками. Пример выделения границ ядра патологии представлен на рисунке 2.9.

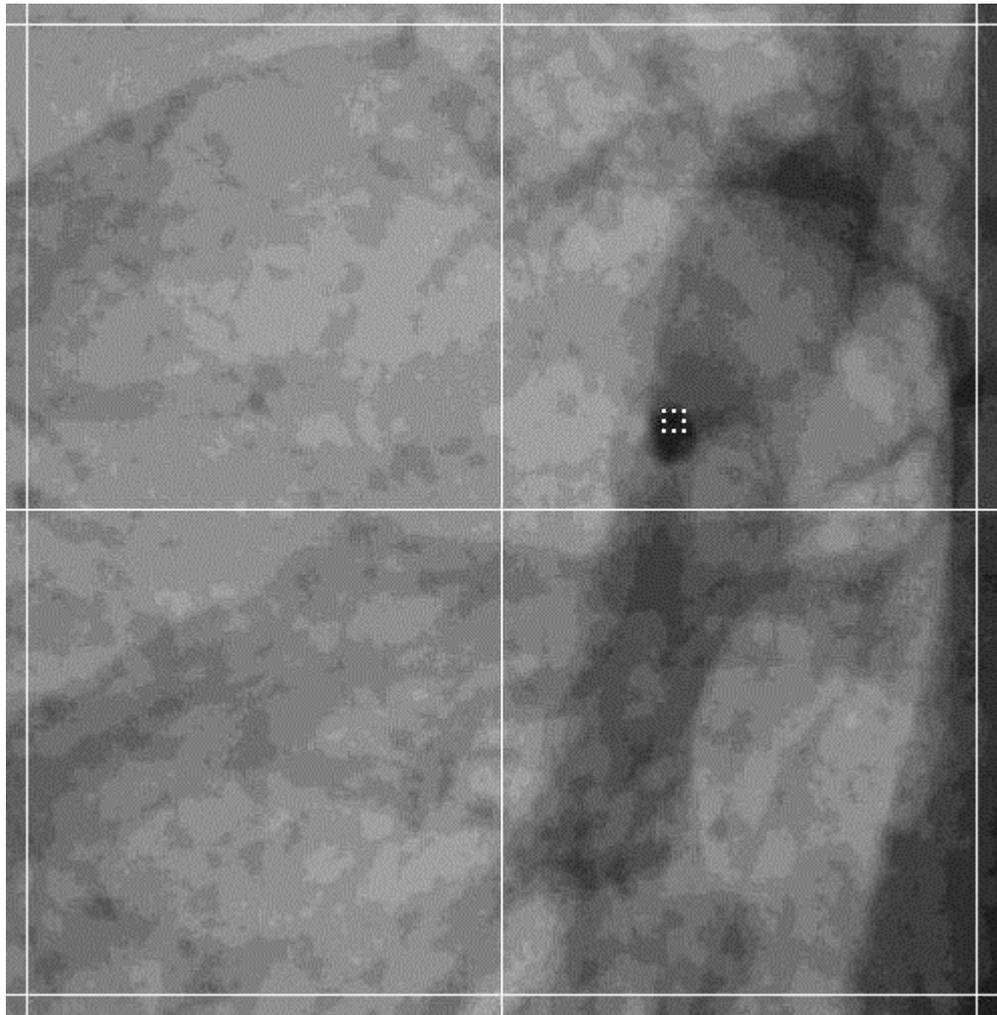


Рисунок 2.9 – Выделение границ ядра патологии

Кроме записи координат текущего ядра во временные переменные «iKernel» и «jKernel» также происходит запись координат в динамические массивы «iFoundKernel» и «jFoundKernel» для постоянного хранения значения. Это требуется для исключения повторного поиска ядра патологии в уже проверенной области с найденным ядром.

Пример сохраненных в динамическом массиве «iFoundKernel» координат по оси Y для найденных на флюорографическом снимке пяти ядер представлен на рисунке 2.10.

1x5 cell

	1	2	3	4	5	6	7
1	526	670	904	958	1147		
2							
3							
4							
5							

Рисунок 2.10 – Пример сохраненных в динамическом массиве «iFoundKernel» координат по оси Y

Пример сохраненных в динамическом массиве «jFoundKernel» координат по оси X для найденных на флюорографическом снимке пяти ядер представлен на рисунке 2.11.

1x5 cell

	1	2	3	4	5	6	7
1	744	902	1078	505	1142		
2							
3							
4							
5							

Рисунок 2.11 – Пример сохраненных в динамическом массиве «jFoundKernel» координат по оси X

Таким образом, полные координаты ядер для указанного флюорографического снимка будут следующими: 526,744; 670,902; 904,1078; 958,505; 1147,1142. Следовательно, повторный поиск ядра и патологии в секторах, содержащих указанные координаты, проведен не будет, что позволяет оптимизировать работу программы.

Код алгоритма поиска ядра патологии приведен в приложении Д.

2.4 Реализация алгоритма установки границ поиска патологии

Если в результате работы алгоритма поиска ядра патологии логической переменной «flag» присваивается значение «true», запускается алгоритм установки границ поиска патологий внутри сектора с найденным ядром.

Сектор, содержащий ядро патологии, вычисляется с помощью созданных в предыдущем алгоритме переменных «iKernel» и «jKernel», которые содержат координаты центра ядра патологии. Эти координаты используются в формулах для вычисления границы сектора, в котором будет проводиться поиск патологии с использованием базы данных.

Верхняя граница сектора с ядром патологии вычисляется по следующей формуле:

$$i = 1 + \text{floor} \frac{iKernel}{\text{round} \frac{row}{nSector}} * \text{round} \frac{row}{nSector}, \quad (2.1)$$

где i – верхняя граница сектора;

$iKernel$ – номер строки ядра патологии;

row – ширина изображения, пикс;

$nSector$ – количество секторов;

floor – округление числа до ближайшего целого в меньшую сторону;

round – округление числа до ближайшего целого.

Нижняя граница сектора с ядром патологии вычисляется по следующей формуле:

$$i = \left(1 + \text{floor} \frac{iKernel}{\text{round} \frac{row}{nSector}} \right) * \text{round} \frac{row}{nSector}, \quad (2.2)$$

где i – нижняя граница сектора;

$iKernel$ – номер строки ядра патологии;

row – ширина изображения, пикс;

$nSector$ – количество секторов;

floor – округление числа до ближайшего целого в меньшую сторону;

round – округление числа до ближайшего целого.

Левая и правая границы сектора вычисляются по аналогичным формулам с использованием переменной «jKernel», которой обозначается номер столбца ядра патологии.

С помощью этих формул устанавливается цикл, внутри которого происходит поиск патологий на флюорографическом снимке в указанном секторе. Блок-схема алгоритма задания цикла поиска патологии внутри сектора представлена на рисунке 2.12.



Рисунок 2.12 – Блок-схема алгоритма задания цикла поиска патологии внутри сектора

Подобный алгоритм установки границ также используется для исключения повторного поиска ядра патологии в секторе, где ядро уже было найдено ранее.

Код алгоритма установки границ поиска патологий приведен в приложении Е.

Реализованы алгоритмы управления базой данных (создание базы данных, добавление новой записи, удаление существующей записи), разбиения изображения на секторы, поиска ядра патологии и установки границ поиска патологии, а также описана их работа, и приведены блок-схемы выполнения алгоритмов.

3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Соответствие разработанного программного обеспечения формализованным требованиям

Разработанные алгоритмы программного обеспечения выявления потенциальной патологии на легочном (сосудистом) рисунке содержат алгоритмы управления базой данных, разбиения изображения на секторы, поиска ядра патологии и установки границ для поиска патологии на основе шаблонов с масками из базы данных.

Алгоритм управления базой данных включает в себя такие алгоритмы, как создание новой базы данных, а также добавление и удаление записей. После обнаружения патологии на флюорографическом снимке создается новый шаблон маски с патологией и добавляется в базу данных для последующего использования при поиске новых патологий.

Алгоритм разбиения изображения на секторы загружает флюорографический снимок из указанной директории и производит его разделение на прямоугольные области, в которых производится поиск патологии.

Алгоритм поиска ядра патологии производит первичный поиск потенциальной патологии на загруженном флюорографическом снимке, а также сохраняет координаты найденного ядра для последующего поиска усиления по шаблонам с масками патологий внутри соответствующей координатам области (сектора).

Алгоритм установки границ поиска патологии определяет по сохраненным координатам ядра область для последующего поиска патологии с использованием шаблонов с масками патологий из базы данных. Границы области (сектора) устанавливаются по аналогии с алгоритмом разбиения изображения на секторы.

3.2 Тестирование работы алгоритмов программного обеспечения

При запуске алгоритма создания базы данных в директории с алгоритмом создается пустая база данных, в которую в последствии будут добавляться в ручном или автоматическом режиме все новые шаблоны масок с патологиями. При успешном создании базы данных в командной строке алгоритма можно увидеть соответствующее сообщение.

Действия алгоритма создания новой базы данных представлены на рисунке 3.1.

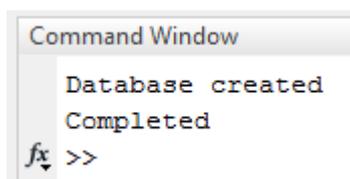


Рисунок 3.1 – Действия алгоритма создания новой базы данных

При открытии созданной базы данных внутри среды технических расчетов MATLAB можно увидеть необходимую информацию о содержимом базы. Так как это новая база, записи шаблонов масок с патологиями в ней отсутствуют, и в дальнейшем их необходимо добавить.

Информация о содержимом новой базы данных внутри среды технических расчетов MATLAB представлена на рисунке 3.2.

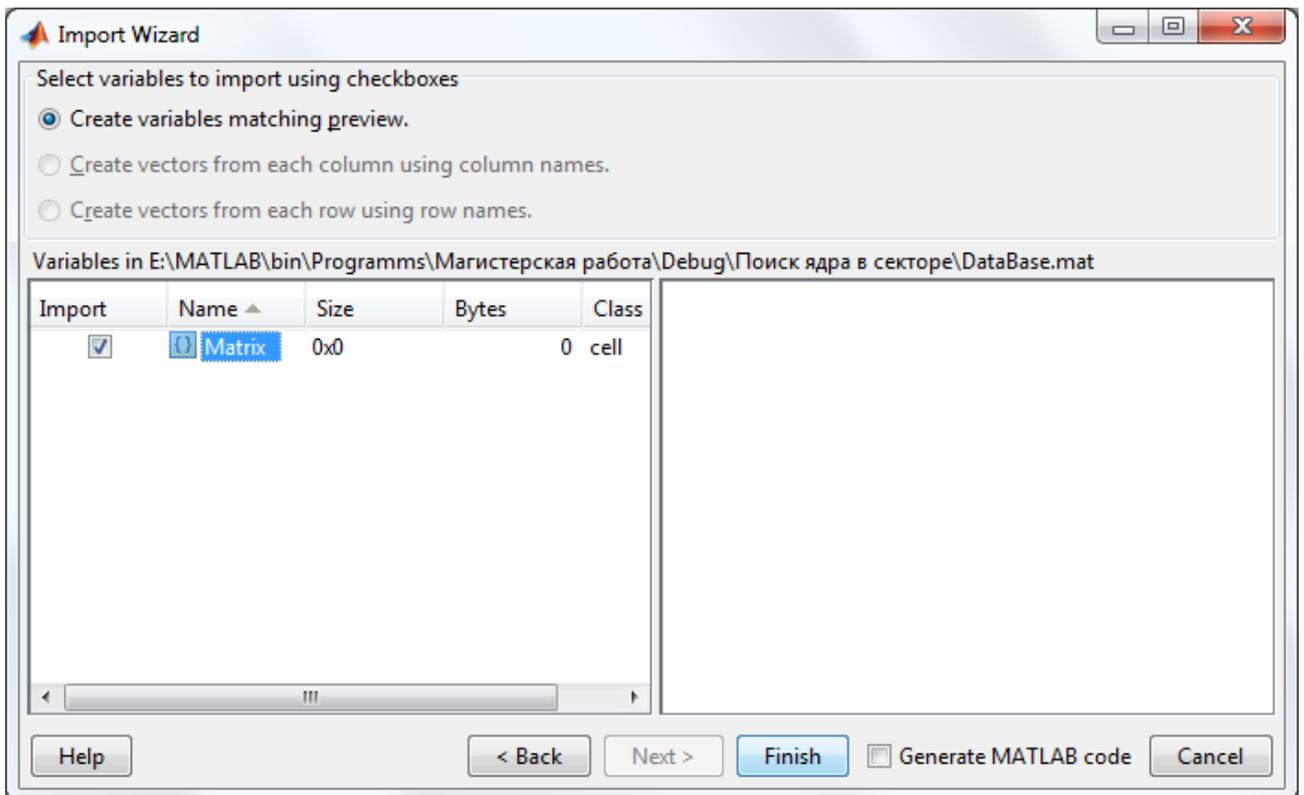


Рисунок 3.2 – Информация о содержимом новой базы данных внутри среды технических расчетов MATLAB

В новую базу данных необходимо добавить шаблоны масок с патологиями, чтобы в последующем алгоритм поиска патологий на флюорографических снимках мог использовать шаблоны для поиска усилений, а также создавать новые маски и добавлять их в базу данных. При успешном добавлении новой записи в базу данных в командной строке алгоритма можно увидеть соответствующее сообщение, а также указывается текущее количество записей в базе.

Действия алгоритма добавления новой маски в базу данных представлены на рисунке 3.3.

```

Command Window
Количество записей: 2
Mask_3.jpg добавлена в базу данных.
Количество записей: 3
Completed
fx >>

```

Рисунок 3.3 – Действия алгоритма создания новой базы данных

При открытии базы данных внутри среды технических расчетов MATLAB можно убедиться, что новые записи были добавлены в базу. В поле «Size» указан текущий размер массива базы данных, где правое значение указывает на количество записей. В поле «Bytes» указан текущий размер базы данных в байтах. В правой части окна указана информация о размере в пикселях шаблонов масок с патологиями, а также их глубина цвета в 8 бит, что соответствует черно-белому изображению.

Информация о добавленных в базу данных записях внутри среды технических расчетов MATLAB представлена на рисунке 3.4.

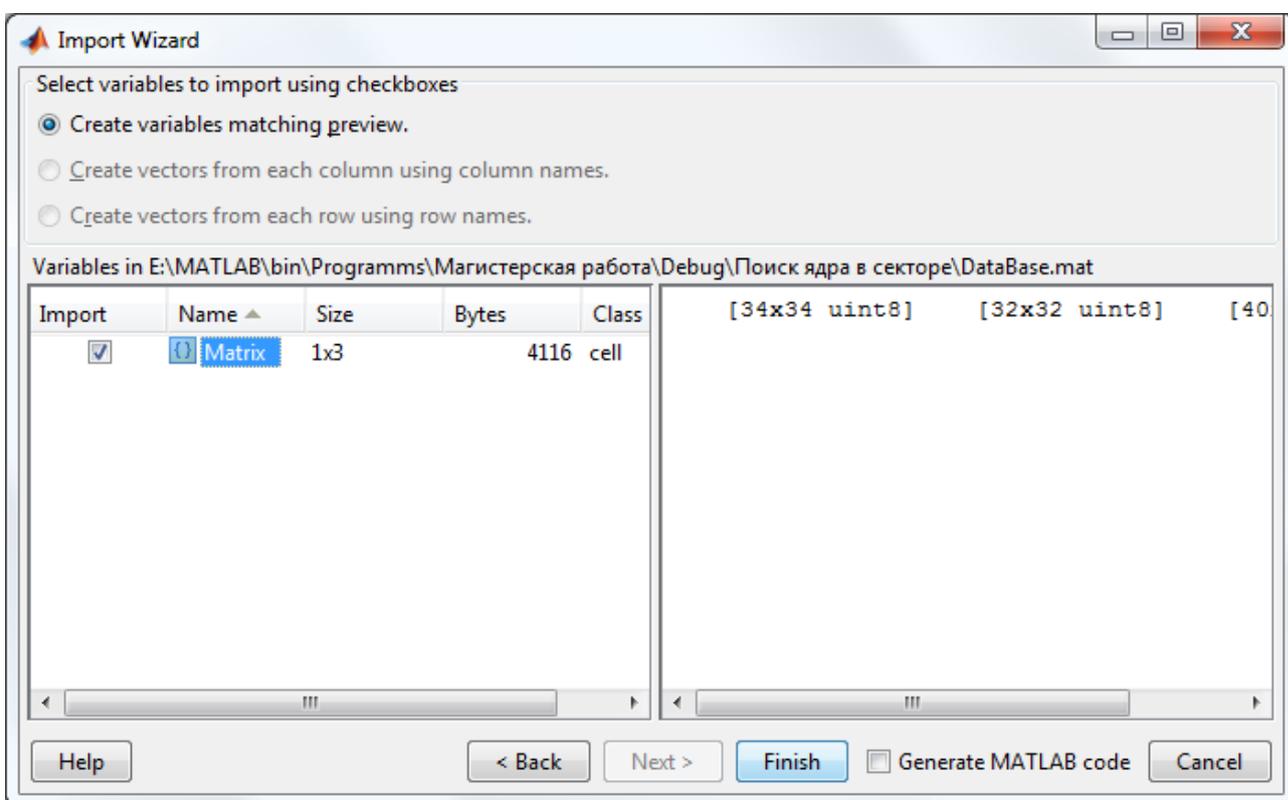
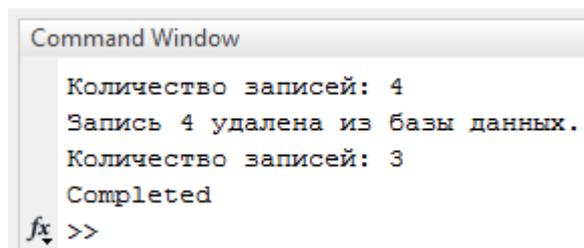


Рисунок 3.4 – Информация о добавленных в базу данных записях внутри среды технических расчетов MATLAB

С помощью алгоритма удаления записи из базы данных возможно удаление шаблонов масок с патологиями. В алгоритме требуется указать номер удаляемой записи, после чего происходит ее удаление из базы, а все последующие записи сдвигаются на одну позицию вниз во избежание пустых ячеек в базе.

Действия алгоритма удаления записи из базы данных представлены на рисунке 3.5.



```
Command Window
Количество записей: 4
Запись 4 удалена из базы данных.
Количество записей: 3
Completed
fx >>
```

Рисунок 3.5 – Действия алгоритма удаления записи из базы данных

После завершения разработки новых модулей они были интегрированы в основное программное обеспечение для оптимизации его работы.

Новый алгоритм производит поиск патологий после разбиения изначального изображения на секторы и предварительного поиска ядра патологии на исходных флюорографических снимках, которые загружаются из директории «Images». Для проверки исходных изображений на наличие патологий сначала загружаются шаблоны масок с патологиями из базы данных «DataBase», затем производится разбиение изображения на секторы, после чего применяется алгоритм поиска патологий на флюорографическом снимке в том секторе, где ранее было обнаружено ядро. Полученные в результате работы алгоритма изображения записываются в директорию «Pathologies» с добавлением к имени файла символа решетки «#».

Действия алгоритма поиска патологий на исходных флюорографических снимках представлены на рисунке 3.6.

```
Command Window
>> Начало выполнения программы
Количество изображений: 3

Изображение: N5806DF.di.jpg
Результат: ядро патологии НЕ найдено

Изображение: N5836DM-1.di.jpg
Результат: найдено ядро патологии
Координаты ядра: [996,946]
Количество масок в базе данных: 3
Проверка маски: ячейка [1] - FALSE
Проверка маски: ячейка [2] - FALSE
Проверка маски: ячейка [3] - FALSE

Изображение: P5755D.di.jpg
Результат: найдено ядро патологии
Координаты ядра: [1171,794]
Количество масок в базе данных: 3
Проверка маски: ячейка [1] - FALSE
Проверка маски: ячейка [2] - TRUE
>> Создание новой маски
Маска №4 добавлена в базу данных.
Количество записей: 4
>> Выделение границ патологии

Создано новых масок: 1
>> Конец выполнения программы
Elapsed time is 3.537062 seconds.
fx >>
```

Рисунок 3.6 – Действия алгоритма поиска патологий на исходных флюорографических снимках

В результате работы алгоритма поиска патологий на исходных флюорографических снимках также был создан новый шаблон маски с патологией и добавлен к другим шаблонам в базу данных «DataBase». На рисунке 3.6 имеются соответствующие строки, подтверждающие добавление новой маски, а также обновление количества записей в базе данных. Дальнейший поиск патологий будет осуществляться уже с учетом новой маски в базе данных.

Изображения, записанные в директорию «Pathologies», имеют выделенные границы легочного (сосудистого) рисунка, разделены на секторы в

количестве, указанном в программном коде, а также имеют выделенные границы ядра патологии и найденного по шаблонам из базы данных усиления.

Пример полученного в результате работы алгоритма изображения с патологией представлен на рисунке 3.7.

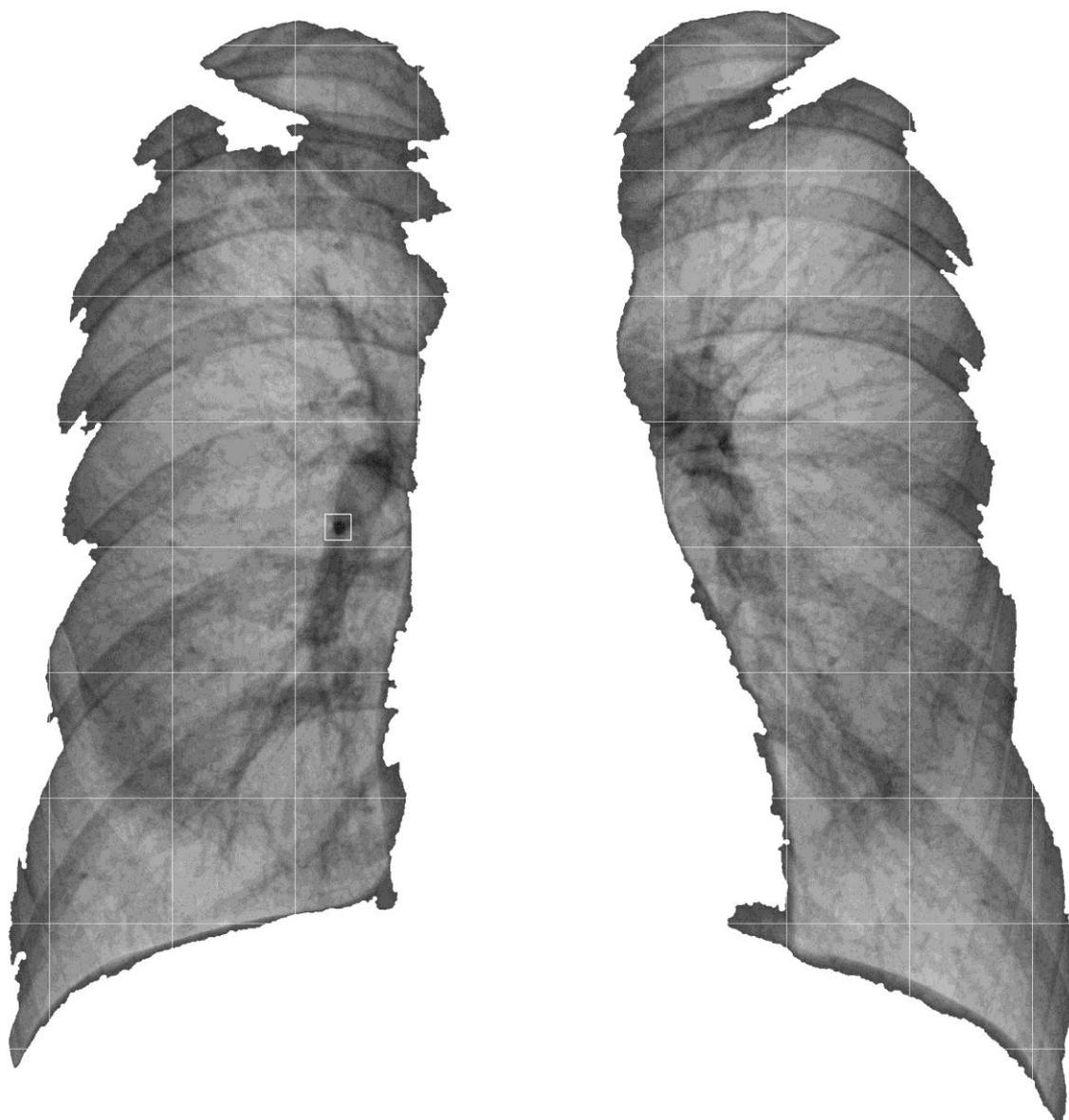


Рисунок 3.7 – Пример полученной в результате работы алгоритма области с патологией

Крупный план области с найденной на рисунке 3.7 патологией представлен на рисунке 3.8.

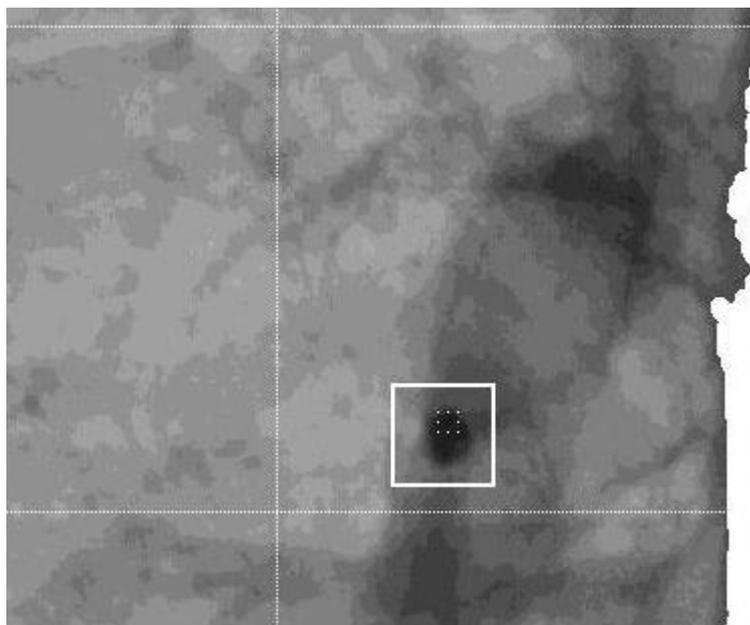


Рисунок 3.8 – Крупный план области с найденной на изображении патологией

3.3 Сравнение скорости работы старых и новых алгоритмов программного обеспечения

Для сравнения скорости работы программного обеспечения был осуществлен поиск патологий на изображениях с использованием нового и основного алгоритмов.

Примеры выполнения работы нового и основного алгоритмов представлены на рисунке 3.9. В левой части представлена информация о работе нового алгоритма, в правой части представлена информация о работе основного алгоритма программного обеспечения.

```

Command Window
>> Начало выполнения программы
Количество изображений: 3

Изображение: N5806DF.di.jpg
Результат: ядро патологии НЕ найдено

Изображение: N5836DM-1.di.jpg
Результат: найдено ядро патологии
Координаты ядра: [996,946]
Количество масок в базе данных: 3
Проверка маски: ячейка [1] - FALSE
Проверка маски: ячейка [2] - FALSE
Проверка маски: ячейка [3] - FALSE

Изображение: P5755D.di.jpg
Результат: найдено ядро патологии
Координаты ядра: [1171,794]
Количество масок в базе данных: 3
Проверка маски: ячейка [1] - FALSE
Проверка маски: ячейка [2] - TRUE
>> Создание новой маски
Маска №4 добавлена в базу данных.
Количество записей: 4
>> Выделение границ патологии

Создано новых масок: 1
>> Конец выполнения программы
Elapsed time is 3.537062 seconds.
fx >>

Command Window
>> Начало выполнения программы
Количество изображений: 3

Изображение: N5806DF.di.jpg
Проверка маски: Mask_1.jpg - FALSE
Проверка маски: Mask_2.jpg - FALSE
Проверка маски: Mask_3.jpg - FALSE

Изображение: N5836DM-1.di.jpg
Проверка маски: Mask_1.jpg - FALSE
Проверка маски: Mask_2.jpg - FALSE
Проверка маски: Mask_3.jpg - FALSE

Изображение: P5755D.di.jpg
Проверка маски: Mask_1.jpg - FALSE
Проверка маски: Mask_2.jpg - TRUE
>> Создание новой маски
>> Выполнено
>> Выделение границ патологии
>> Выполнено

Создано новых масок: 1
>> Конец выполнения программы
Elapsed time is 95.233348 seconds.
fx >>

```

Рисунок 3.9 – Примеры выполнения работы нового и основного алгоритмов

Программные средства языка технических расчетов MATLAB позволяют измерять время выполнения работы программы с помощью команд «tic» (указание точки начала отсчета) и «toc» (указание точки окончания отсчета). На рисунке 3.9 в последней строке указана информация о времени, затраченном на поиск патологии на трех флюорографических снимках для каждого алгоритма, с использованием указанных ранее команд отсчета времени. Для основного алгоритма затраченное время на поиск патологий составило более 95 секунд, а для нового алгоритма затраченное время на поиск патологий составило 3.5 секунды, что позволяет отметить скорость работы нового алгоритма более чем в 20 раз быстрее основного.

Так как на флюорографическом снимке N5806DF.di.jpg не было найдено ядро патологии, поиск патологии по маскам в новом алгоритме не был осуществлен, что позволяет отсеять текущее изображение и существенно сократить время работы программы.

На флюорографическом снимке N5836DM-1.di.jpg было найдено ядро патологии по координатам 996,946, где новым алгоритмом был произведен поиск патологии в соответствующем указанным координатам секторе. Поиск патологии на конкретном участке флюорографического снимка, а не на всем изображении позволяет существенно сократить время поиска патологии. Однако для текущего изображения поиск патологии по шаблонам масок был отрицательным.

На флюорографическом снимке P5755D.di.jpg также было найдено ядро патологии по координатам 1171,794, где новым алгоритмом был произведен поиск патологии в соответствующем указанным координатам секторе. Для текущего изображения поиск патологии по шаблонам масок был положительным.

3.4 Обновление интерфейса программного обеспечения

Разработанные алгоритмы для удобства использования необходимо поместить в графический интерфейс пользователя, в следствие чего появилась потребность в обновлении существующего интерфейса программного обеспечения для выявления потенциальной патологии на легочном (сосудистом) рисунке.

Данный интерфейс содержит следующие функциональные кнопки: «Добавление элемента в БД», «Удаление элемента из БД» и «Поиск патологий в секторе», каждая из которых осуществляет запуск одного из алгоритмов программного обеспечения для выявления потенциальной патологии на легочном (сосудистом) рисунке.

Каждая кнопка ассоциирована с одним из полей, где требуется ввод данных со стороны пользователя. Для запуска алгоритма «Добавления элемента

в БД» пользователю требуется ввести имя элемента. Для запуска алгоритма «Удаления элемента из БД» пользователю требуется ввести номер удаляемого элемента. Для запуска алгоритма «Поиск патологий в секторе» пользователю требуется ввести количество разбиений изображения.

Также с кнопкой «Поиск патологий в секторе» ассоциированы три переключателя, с помощью которых возможно изменять макет получаемого изображения.

Над кнопкой «Добавление элемента в БД» имеется специальное поле, где требуется ввести имя шаблона маски с патологией для его добавления в базу данных. Значение из этого поля присваивается переменной «name» в алгоритме программы.

Пример работы алгоритма добавления записи в базу данных в окне интерфейса пользователя представлен на рисунке 3.10.

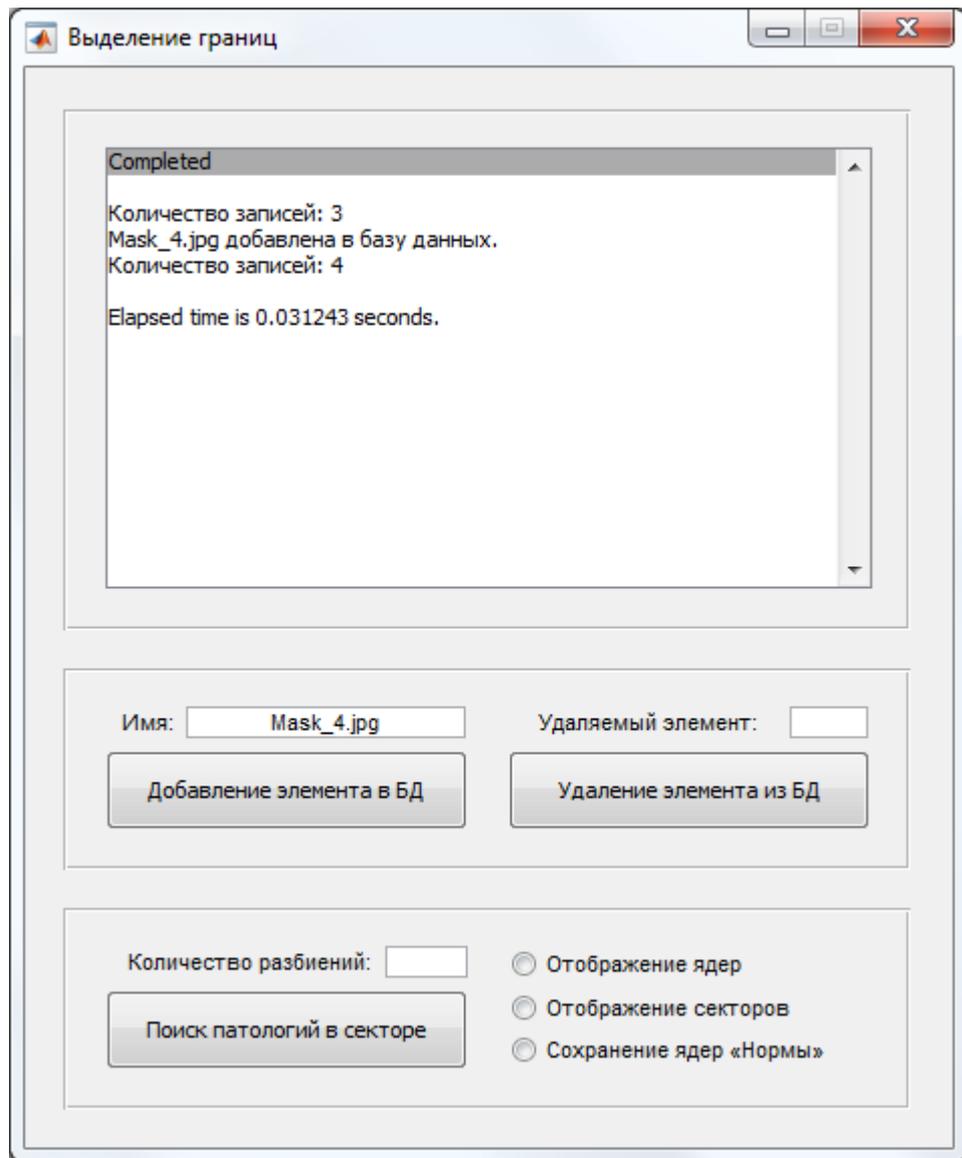


Рисунок 3.10 – Пример работы алгоритма добавления записи в базу данных в окне интерфейса пользователя

В базе данных программного обеспечения была добавлена новая запись в 4 ячейку массива. Обновленный массив базы данных представлен на рисунке 3.11.

{ } 1x4 cell

	1	2	3	4	5	6	7
1	32x32 uint8	34x34 uint8	40x40 uint8	40x40 uint8			
2							
3							
4							
5							
6							
7							
8							
9							
10							

Рисунок 3.11 – Обновленный массив базы данных

Новая запись в массиве базы данных также будет использована для выявления потенциальной патологии на флюорографических снимках при запуске соответствующего алгоритма.

Над кнопкой «Удаление элемента из БД» имеется специальное поле, где требуется ввести номер элемента, который в последующем будет удален из базы данных. Значение из этого поля присваивается переменной «del» в алгоритме программы.

Пример работы алгоритма удаления записи из базы данных в окне интерфейса пользователя представлен на рисунке 3.12.

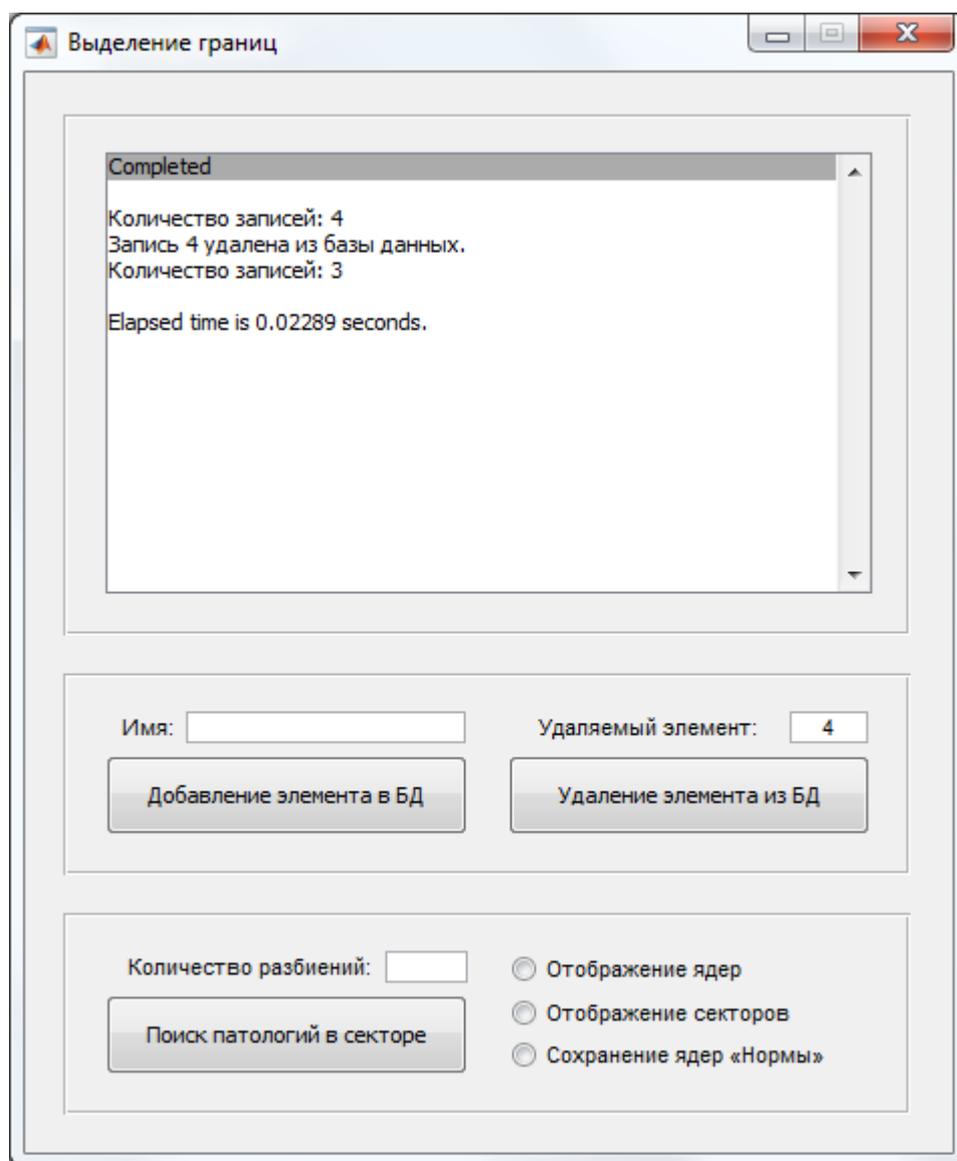


Рисунок 3.12 – Пример работы алгоритма удаления записи из базы данных в окне интерфейса пользователя

В базе данных программного обеспечения для выявления потенциальной патологии на легочном (сосудистом) рисунке была удалена запись из 4 ячейки массива, теперь массив содержит только 3 ячейки.

Обновленный массив базы данных представлен на рисунке 3.13.

1x3 cell

	1	2	3	4	5	6	7
1	32x32 uint8	34x34 uint8	40x40 uint8				
2							
3							
4							
5							
6							
7							
8							
9							
10							

Рисунок 3.13 – Обновленный массив базы данных

Следовательно, для выявления потенциальной патологии на флюорографических снимках будут использованы только оставшиеся в массиве базы данных записи.

Над кнопкой «Поиск патологий в секторе» имеется специальное поле, где указывается количество разбиений флюорографического изображения на секторы, внутри которых производится поиск патологий при ранее найденном ядре патологии. Значение из этого поля присваивается переменной «nSector» в алгоритме программы.

Пример работы алгоритма поиска патологий в секторе в окне интерфейса пользователя представлен на рисунке 3.14.

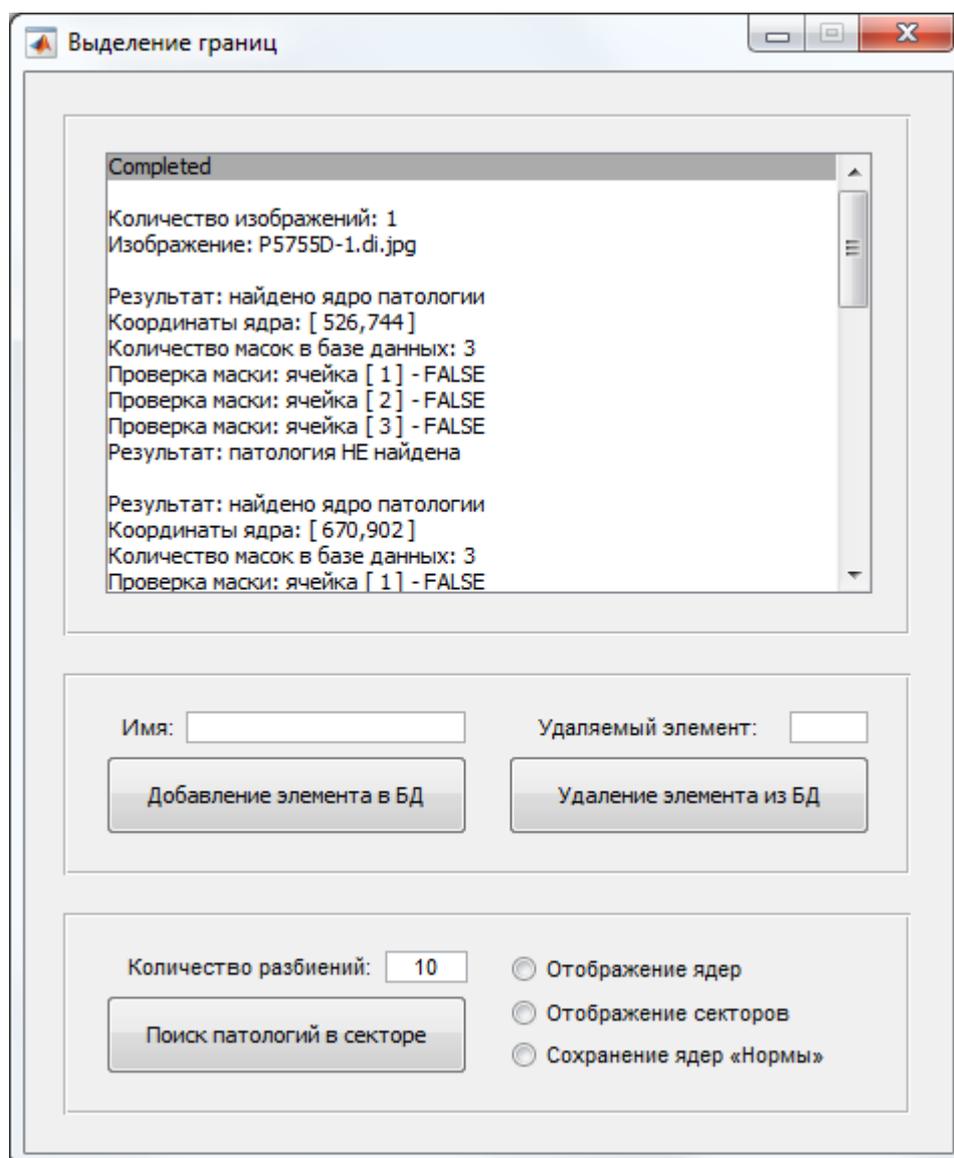


Рисунок 3.14 – Пример работы алгоритма поиска патологий в секторе в окне интерфейса пользователя

В интерфейсе пользователя также имеются три переключателя для управления макетом выходящего изображения при использовании алгоритма поиска патологий в секторе: «Отображение ядер», «Отображение секторов» и «Сохранение ядер "Нормы"».

Если пользователю требуется вывод изображения с найденными патологиями без учета ядер (потенциальных патологий), необходимо сделать активным переключатель «Отображение ядер». Значение этого переключателя присваивается логической переменной «checkKernel» в алгоритме программы. В этом случае после указания количества разбиений и выполнения работы

программы поиска патологий в секторе на полученном изображении будут отображены только найденные патологии.

Пример полученного изображения с активным переключателем «Отображение ядер» представлен на рисунке 3.15.

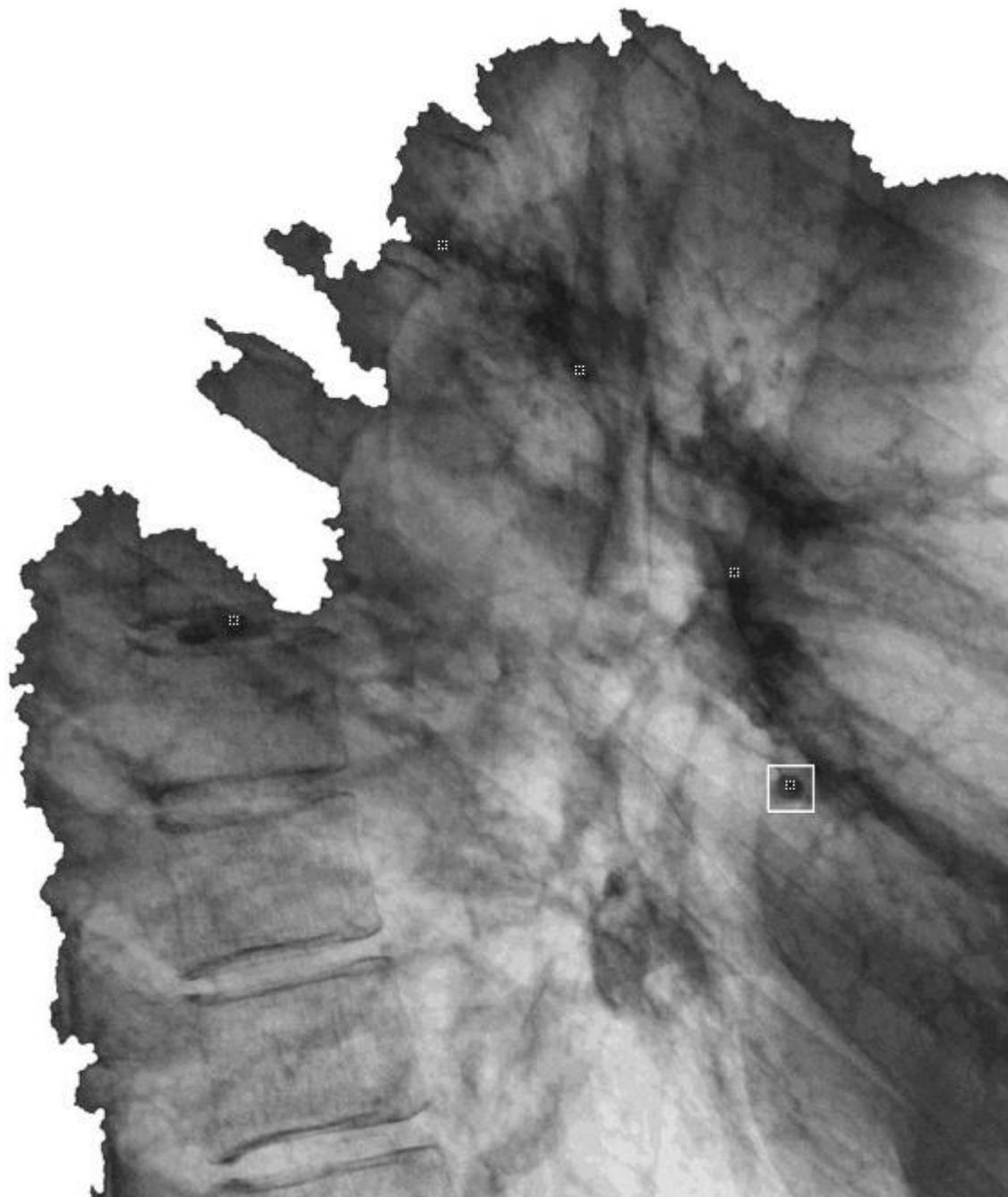


Рисунок 3.15 – Пример полученного изображения с активным переключателем «Отображение ядер»

Если пользователю требуется вывод изображения с отображением секторов, необходимо сделать активным переключатель «Отображение

секторов». Значение этого переключателя присваивается логической переменной «checkSector» в алгоритме программы. В этом случае после указания количества разбиений и выполнения работы программы поиска патологий в секторе на полученном изображении будет отображено разбиение на секторы.

Пример полученного изображения с активным переключателем «Отображение секторов» представлен на рисунке 3.16.

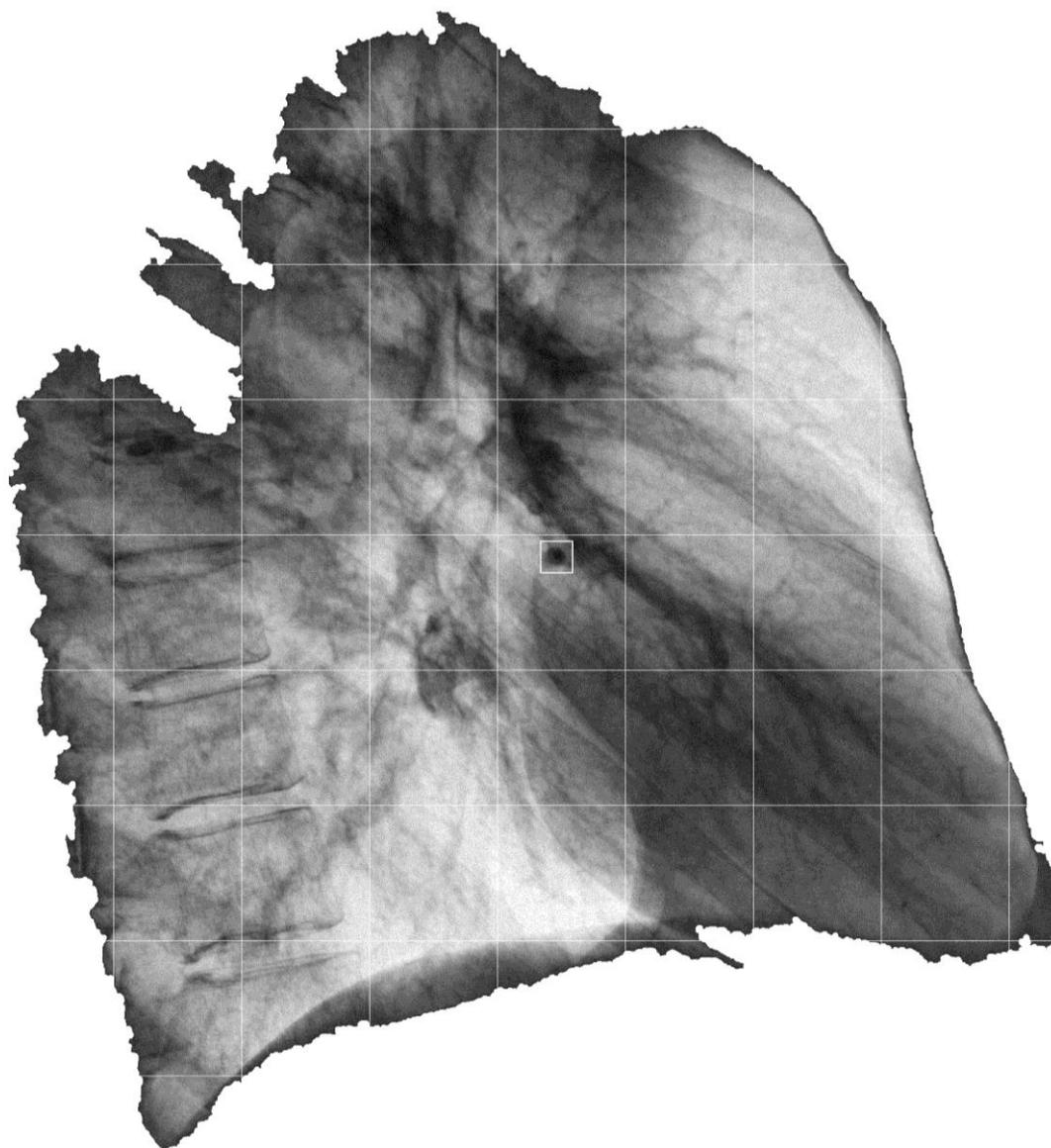


Рисунок 3.16 – Пример полученного изображения с активным переключателем «Отображение секторов»

Если пользователю требуется вывод изображения, где патология не была найдена, но обнаружено одно или несколько ядер (потенциальная патология), необходимо сделать активным переключатель «Сохранение ядер "Нормы"». Значение этого переключателя присваивается логической переменной «saveNormal» в алгоритме программы. В этом случае после указания количества разбиений и выполнения работы программы поиска патологий в секторе на полученном изображении «нормы» будут отображены все найденные ядра патологии.

Пример полученного изображения с активным переключателем «Сохранение ядер "Нормы"» представлен на рисунке 3.17.

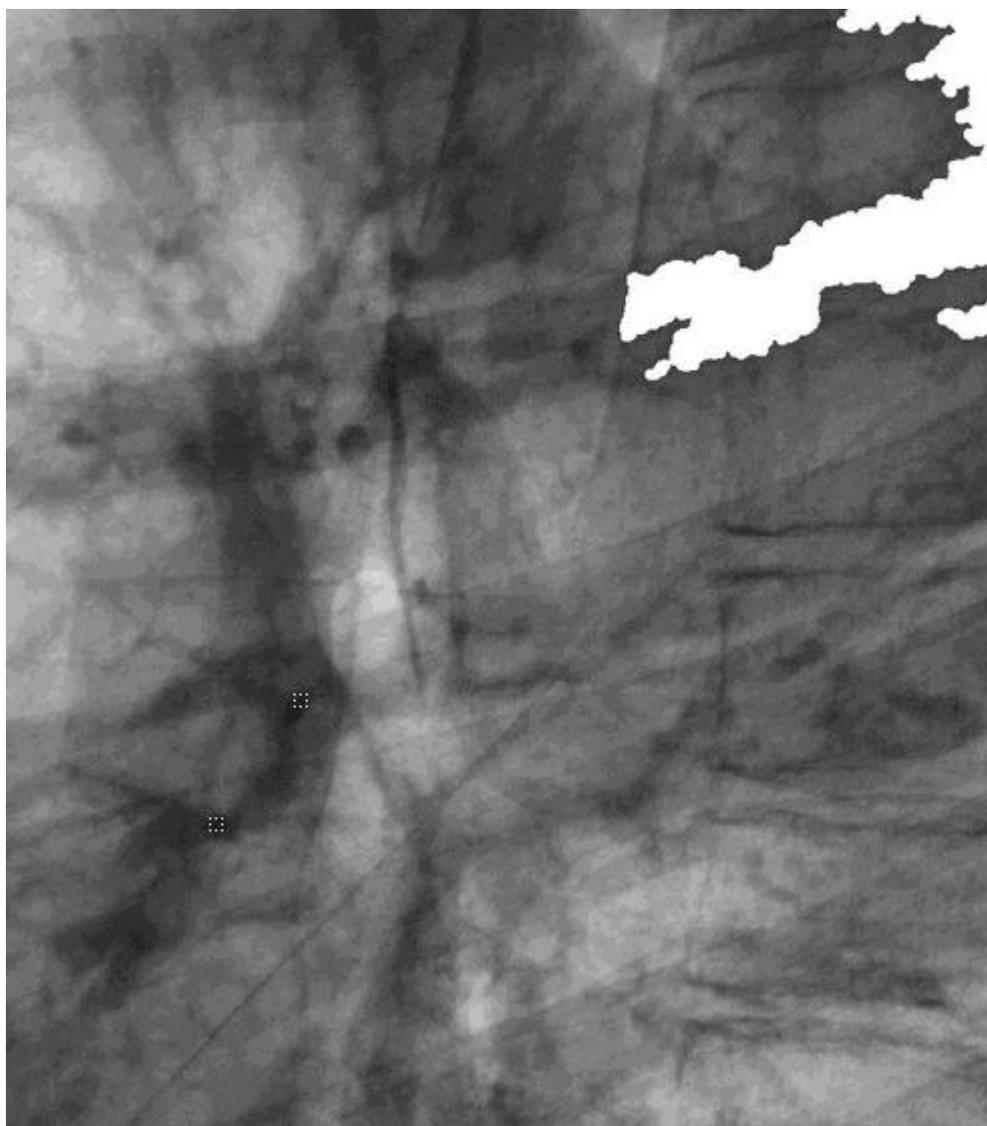


Рисунок 3.17 – Пример полученного изображения с активным переключателем «Отображение секторов»

Разработанные алгоритмы соответствуют формализованным требованиям. Произведено тестирование программного обеспечения с применением новых алгоритмов, некорректное поведение программы не было обнаружено. Произведено сравнение скорости работы нового и основного алгоритмов, поиск патологии новым алгоритмом осуществляется значительно быстрее основного. Произведено обновление интерфейса для программного обеспечения в соответствии с новыми алгоритмами. Продемонстрированы результаты тестирования каждого элемента на интерфейсе.

ЗАКЛЮЧЕНИЕ

В ходе диссертационного исследования была описана актуальность рассматриваемой темы, определены объект и предмет магистерской диссертации, поставлена цель и выявлены задачи. Также был проведен анализ состояния вопроса, а именно – проведен обзор ранее разработанного программного обеспечения для выделения границ легочного (сосудистого) рисунка, выявлены его слабые стороны и предложены варианты оптимизации работы. Кроме того, были формализованы требования к разрабатываемому программному продукту, описан язык программирования, с помощью которого разработаны новые алгоритмы программы, а также была описана работа данного программного обеспечения.

В результате разработки программного обеспечения были созданы новые алгоритмы, описанные в формализации требований к разрабатываемому программному продукту, а работа алгоритмов программы полностью соответствует указанным требованиям. Также была протестирована работа программного обеспечения с помощью новых алгоритмов на наличие сбоев и ошибок, некорректное поведение работы алгоритмов не было обнаружено. Произведено обновление интерфейса для программного обеспечения в соответствии с новыми алгоритмами.

Разработанное в ходе диссертационного исследования программное обеспечение является самообучаемым, имеет высокую скорость работы алгоритма поиска патологий за счет предварительного поиска ядра патологии и разбиения изображения на секторы, а также использования базы данных с шаблонами масок с патологиями. Все алгоритмы программного обеспечения смогут помочь специалистам в области поиска патологий быстрее и точнее обнаружить усиления на флюорографических снимках.

Однако данное программное обеспечение имеет недостаток, а именно – строго заданные погрешности поиска патологии без возможности их

регулировки из графического интерфейса, что компенсируется возможностью выявления ядра (потенциальной патологии).

Данное программное обеспечение имеет перспективу улучшения и развития и может быть дополнено новыми функциональными алгоритмами, а все необходимые значения и погрешности могут быть заданы в окне пользовательского интерфейса.

Апробация основных алгоритмов программного обеспечения была представлена в научно-практической конференции «Студенческие Дни науки в ТГУ – 2017», которая проводилась в Тольяттинском государственном университете с 3 по 15 апреля 2017 года на кафедрах институтов в виде семинаров.

Дополнительно апробация основных алгоритмов поиска патологии была представлена и опубликована на III научно-практической всероссийской конференции (школе-семинаре) молодых ученых «Современные исследования в области естественных и технических наук», которая проводилась в Тольяттинском государственном университете с 24 по 25 апреля 2017 года в заочной форме.

Кроме того, апробация основных алгоритмов разработанного продукта была представлена и опубликована на XIV Всероссийской студенческой конференции «Информационные технологии в современном мире – 2017», которая проводилась организаторами АНО ВО «Гуманитарный университет» 5 мая 2017 года в очно-заочной форме.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ 2.105 – 95. Общие требования к текстовым документам [Текст]. – М.: Изд-во стандартов, 1996. – 29 с. – (Единая система конструкторской документации).
2. ГОСТ 7.1–2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Изд-во стандартов, 2004. – 48 с.
3. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.
4. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов.
5. ГОСТ 19.701 – 90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807–85) [Текст]. Введен 1992–01–01. – М.: Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации).

Научная и методическая литература

6. Алексеев Е.Р., Чеснокова О.В. MATLAB 7. Самоучитель. Издательство: «НТ Пресс», 2006 г. – 464 стр.
7. Браженко Н.А. Туберкулез органов дыхания. Издательство: СпецЛит, 2012 г. – 368 стр.
8. В. Дьяконов, И. Абраменкова. MATLAB. Обработка сигналов и изображений. Специальный справочник. Издательство: Питер, 2002 г. – 608 стр.
9. В.П. Дьяконов. MATLAB 6.0/6.1/6.5/6.5 + SP1 + Simulink 4/5. Обработка сигналов и изображений. Москва: СОЛОН-Пресс, 2004 г. – 592 стр.
10. В.П. Дьяконов. Matlab 6.5 SP1/7 + Simulink 5/6. Работа с изображениями и видеопотоками. Москва: СОЛОН-Пресс, 2005 г. – 400 стр.

11. Зими́на В.Н., Коше́чкин В.А., Кра́вченко А.В. Туберкулез и ВИЧ-инфекция у взрослых. Руководство. Издательство: ГЭОТАР-Медиа, 2014 г. – 224 стр.
12. Краснов Д.И. Выделение патологий на полутоновых изображениях рентгенограмм грудной клетки/ Д.И.Краснов; науч. рук. М.Г. Лисовская // Прикладная математика и информатика: современные исследования в области естественных и технических наук : Материалы III научно-практической всероссийской конференции (школы-семинара) молодых ученых, Тольятти, 24-25 апреля 2017 г. – Тольятти : Изд-во ТГУ, 2017. – [С. 292-295].
13. Краснов Д.И. Обнаружение патологий и выделение их границ на флюорографических снимках/ Д.И.Краснов // Информационные технологии в современном мире – 2017 : Материалы XIV Всероссийской студенческой конференции, Екатеринбург, 5 мая 2017 г. – Тольятти : Изд-во АНО ВО ГУ, 2017. – [С. 72-75].
14. Курбатова Е.А. MATLAB 7. Самоучитель. Издательство: Вильямс, 2005 г. – 256 стр.
15. Максимова Е.И. Алгоритм обнаружения образований на снимках компьютерного томографа с использованием кластеризации / Е.И. Максимова, П.А. Хаустов; // Современные техника и технологии : сборник трудов XXI международной научной конференции студентов, аспирантов и молодых ученых, Томск, 5-9 октября 2015 г. : в 2 т. – Томск : Изд-во ТПУ, 2015. – Т. 2. – [С. 39-42].
16. Матиас Хофер. Рентгенологическое исследование грудной клетки. Практическое руководство. Издательство: «Медицинская литература», 2008 г. 224 стр.
17. Поршне́в С.В. MATLAB 7. Основы работы и программирования. Учебник. Издательство: «Бином. Лаборатория знаний», 2006 г. – 320 стр.
18. П. Рудаков, В. Сафонов. Обработка сигналов и изображений с использованием Matlab 5.x. Издательство: Диалог-МИФИ, 2000 г. – 416 стр.

19. Р. Гонсалес, Р. Вудс, С. Эддинс. Цифровая обработка изображений в среде технических расчетов MATLAB. Москва: Техносфера, 2006 г. – 616 стр.

20. Хофер Матиас. Компьютерная томография. Базовое руководство. Издательство: Медицинская литература, 2011 г. – 232 с.

Электронные ресурсы

21. Всемирная организация здравоохранения. Туберкулез [Электронный ресурс]. – Электрон. дан. – [2017]. – Режим доступа : <http://www.who.int/mediacentre/factsheets/fs104/ru>.

22. Медицинская энциклопедия. Проведение рентгенологического исследования [Электронный ресурс]. – Электрон. дан. – [2017]. – Режим доступа : http://www.medical-enc.ru/16/rentgenologicheskoe_issledovanie.shtml.

23. Медицинская энциклопедия. Определение флюорографии [Электронный ресурс]. – Электрон. дан. – [2017]. – Режим доступа : <http://www.medical-enc.ru/20/fluorography.shtml>.

24. Поташев А.В. Интеграция математического моделирования и инновационных подходов к обучению в образовании [Электронный ресурс]: Поташев А.В., Поташева Е.В., Сулейманова Д.Ю. – Электрон. текстовые данные. – М.: Русайнс, 2015. – 96 стр.

25. MathWorks. MATLAB [Электронный ресурс]. – Электрон. дан. – [2017]. – Режим доступа : <http://matlab.ru/products/matlab>.

Литература на иностранном языке

26. Gérard Blanchet. Digital Signal and Image Processing using MATLAB, Volume 2: Advances and Applications: The Deterministic Case, 2e. John Wiley & Sons, Inc., 2015.

27. Gérard Blanchet. Digital Signal and Image Processing Using MATLAB, Volume 3: Advances and Applications, The Stochastic Case, 2nd Edition. John Wiley & Sons, Inc., 2015.

28. Maurice Charbit, Gérard Blanchet. Digital Signal and Image Processing using MATLAB. ISTE Ltd, 2006.

29. Maurice Charbit, Gérard Blanchet. Digital Signal and Image Processing using MATLAB, Volume 1. ISTE Ltd, 2015
30. Tamalika Chaira. Medical Image Processing: Advanced Fuzzy Set Theoretic Techniques. CRC Press, Inc., 2015.

ПРИЛОЖЕНИЕ А

Код алгоритма создания базы данных

```
1 % Создание базы данных
2 clc;
3 clear;
4 close all;
5
6 %создание динамического массива
7 Matrix = {};
8
9 %наименование базы данных
10 name = 'DataBase.mat';
11
12 %сохранение базы данных
13 save(name, 'Matrix', '-mat');
14
15 disp('Database created');
16 disp('Completed');
```

ПРИЛОЖЕНИЕ Б

Код алгоритма добавления записи в базу данных

```
1 % Добавление записи в базу данных
2 clc;
3 clear;
4 close all;
5
6 %загрузка базы данных
7 DataBase = load('DataBase.mat', '-mat');
8 Matrix = DataBase.Matrix;
9 disp(['Количество записей: ' num2str(length(Matrix))]);
10
11 %имя добавляемого элемента
12 name = 'Mask_3.jpg';
13
14 %цикл записи элемента в базу данных
15 if (exist(name, 'file') ~=0)
16     Mask = imread(name);
17     Matrix{length(Matrix) + 1} = Mask;
18     save('DataBase.mat', 'Matrix', '-mat');
19     disp([num2str(name) ' добавлена в базу данных.']);
20     disp(['Количество записей: ' num2str(length(Matrix))]);
21 end
22
23 disp('Completed');
```

ПРИЛОЖЕНИЕ В

Код алгоритма удаления записи из базы данных

```
1 % Удаление записи из базы данных
2 clc;
3 clear;
4 close all;
5
6 %загрузка базы данных
7 DataBase = load('DataBase.mat', '-mat');
8 Matrix = DataBase.Matrix;
9 disp(['Количество записей: ' num2str(length(Matrix))]);
10
11 %номер удаляемой записи
12 del = 4;
13
14 %цикл удаления записи из базы данных
15 if ((del > 0) && (del <= length(Matrix)))
16     Matrix(del) = [];
17     save('DataBase.mat', 'Matrix', '-mat');
18     disp(['Запись ' num2str(del) ' удалена из базы данных.']);
19     disp(['Количество записей: ' num2str(length(Matrix))]);
20 end
21
22 disp('Completed');
```

ПРИЛОЖЕНИЕ Г

Код алгоритма разбиения изображения на секторы

```
1 name = 'image.jpg';
2 img = imread(name);
3 nSector = 10; %(nSector x nSector)
4 space = 1; %расстояние между точками
5 [row,column] = size(img);
6
7 for i = 1 + round(row / nSector) : round(row / nSector) : row
8     for j = 1 : space : column
9         img(i,j) = 255;
10    end
11 end
12 for i = 1 : space : row
13     for j = 1 + round(column / nSector) : round(column / nSector) : column
14         img(i,j) = 255;
15    end
16 end
17
18 imwrite(img, 'sectors.jpg', 'JPEG');
```

ПРИЛОЖЕНИЕ Д

Код алгоритма поиска ядра патологии

```
1 img = imread('image.jpg');
2 [rowImg,columnImg] = size(img);
3
4 R = 5;
5 threshold = 40;
6 borderPercent = 0.1;
7 rowImgBorder = round(rowImg * borderPercent);
8 columnImgBorder = round(columnImg * borderPercent);
9
10 flag = false;
11 for i = rowImgBorder : (rowImg - rowImgBorder)
12     if (flag == true)
13         break
14     end
15     for j = columnImgBorder : (columnImg - columnImgBorder)
16         if (flag == true)
17             break
18         end
19         for m = 1 : R
20             if ((img(i,j) < threshold) && ((img(i+m,j)) < threshold) &&
21                 ((img(i,j+m)) < threshold) && ((img(i-m,j)) < threshold) &&
22                 ((img(i,j-m)) < threshold) && (m == R))
23
24                 img(i+m,j) = 255;
25                 img(i,j+m) = 255;
26                 img(i-m,j) = 255;
27                 img(i,j-m) = 255;
28                 img(i+m,j+m) = 255;
29                 img(i-m,j-m) = 255;
30                 img(i+m,j-m) = 255;
31                 img(i-m,j+m) = 255;
32                 flag = true;
33                 disp('Найдено ядро патологии');
34                 disp(['Координаты ядра: [' num2str(j) ','
35                     num2str(i) ']']);
36
37                 iKernel = i;
38                 jKernel = j;
39             end
40         end
41     end
42 end
```

ПРИЛОЖЕНИЕ Е

Код алгоритма установки границ поиска патологий

```
1 img = imread('image.jpg');
2 [row,column] = size(img);
3 nSector = 10;
4
5 for i = 1 + floor((iKernel - 1) / round(row / nSector)) *
6     round(row / nSector) : 1 : (1 + floor((iKernel - 1) /
7     round(row / nSector))) * round(row / nSector)
8
9     for j = 1 + floor((jKernel - 1) / round(column / nSector)) *
10        round(column / nSector) : 1 : (1 + floor((jKernel - 1) /
11        round(column / nSector))) * round(column / nSector)
12
13     end
14 end
```