



## СОДЕРЖАНИЕ

<b>Введение .....</b>	<b>4</b>
<b>1 Анализ предметной области.....</b>	<b>7</b>
1.1 Анатомия позвоночника .....	7
1.2 Анализ позвоночника с использованием компьютерного зрения .....	9
1.3 Методы обнаружения анатомических частей позвоночника .....	10
1.4 Модели формирования радиологических параметров .....	11
1.5 Сверточные нейросети и области их применения .....	13
1.6 Сверточные нейросети в обработке медицинских изображений.....	15
1.7 Сравнительный анализ сверточных нейросетевых архитектур .....	18
1.7.1 Архитектура U-Net.....	18
1.7.2 Архитектура PSPNet .....	19
1.7.3 Архитектура FPN .....	20
1.7.4 Архитектура Linknet .....	21
1.7.5 Результаты сравнительного анализа нейросетевых архитектур .	23
1.8 Сравнительный анализ существующих аналогов.....	23
1.8.1 Программное обеспечение RadiAnt DICOM Viewer .....	24
1.8.2 Информационная система Horos .....	25
1.8.3 Программное обеспечение Navegatum .....	26
1.8.4 Программное обеспечение ПОЗВОНОЧНИК 3D.....	26
1.8.5 Результаты сравнительного анализа аналогов.....	27
<b>2 Проектирование системы .....</b>	<b>30</b>
2.1 Назначение системы .....	30
2.2 Цели создания системы .....	30
2.3 Перечень подсистем.....	31
2.4 Роли пользователей и их функциональные возможности .....	32
2.5 Описание предметной области посредством диаграмм IDEF .....	33
2.6 Описание предметной области посредством диаграмм UML.....	37
2.7 Проектирование схемы базы данных .....	40
2.8 Выбор технологий разработки .....	41
2.9 Результаты проектирования информационной системы .....	45

<b>3 Результаты разработки .....</b>	<b>46</b>
3.1 Формирование набора данных.....	46
3.1.1 Набор данных Genodisc .....	46
3.1.2 Набор данных TwinsUK.....	47
3.1.3 Набор данных OSCLMRIC.....	49
3.1.4 Набор данных Lumbar MRI Dataset .....	50
3.1.5 Результаты обзора наборов данных .....	51
3.2 Система локализации межпозвоночных дисков .....	52
3.2.1 Структура работы нейросети .....	53
3.2.2 Извлечение информации о межпозвоночных дисках.....	54
3.2.3 Функция потерь .....	55
3.2.4 Реализация нейросети .....	55
3.2.5 Оценка точности работы нейросетевой модели .....	58
3.2.6 Разработка системы анализа межпозвоночных дисков.....	59
3.3 Разработка веб-приложения .....	60
<b>Заключение.....</b>	<b>65</b>
<b>Список использованных источников.....</b>	<b>66</b>
Приложение А Экранные формы информационной системы.....	70
Приложение Б Исходный код нейронной сети .....	73
Приложение В Исходный код веб-приложения.....	76

## ВВЕДЕНИЕ

Боль в спине – одна из самых часто встречающихся проблем со здоровьем в мире. Согласно данным Росдравнадзора распространенность проблемы в Российской Федерации составляет около 60%. Каждый шестой человек в России, вероятно, имеет патологии, связанные с болью в спине. Несмотря на часто встречающиеся отклонения, чаще всего, этимология заболеваний остается загадкой [2]. У 85% пациентов с болями в спине не может быть диагностирована конкретная патология. Для данной преобладающей группы диагностируется неспецифическая боль в спине.

В настоящее время одной из наиболее распространенных областей исследований изучения данной проблемы являются средства, основанные на радиологических методах анализа [1]. Существует два основных метода визуализации в рамках радиологических исследований: магнитно-резонансная томография (МРТ) и компьютерная томография (КТ). МРТ наиболее популярен из-за того, что это единственная официально признанная методика диагностики, поэтому именно этот метод визуализации чаще всего применяется в клиниках [5]. Рентгенологический анализ позвоночника с применением МРТ, как правило, является индивидуально анатомически специфичным, что является основным преимуществом данного метода. При этом наиболее популярными объектами исследований являются межпозвоночные диски (фиброзно-хрящевое образование между телами двух соседних позвонков позвоночника). Существует несколько мер и установленных шкал оценок, созданных для измерения уровней дегенерации диска (шкалы уникальные для метода МРТ). Результаты анализов уровней вырождения дисков, которые составляют врачи-рентгенологи, часто прилагаются в истории болезней пациентов. [16]

Однако до сих пор нет официально подтвержденных исследований, связывающих между собой боли в спине и какие-либо из многочисленных оценок разных радиологических шкал [4]. Одни исследования показывают, что результативные оценки могут отображать процесс дегенерации дисков в связи со старением пациента, и в свою очередь никак не связаны с болью в спине. В свою очередь другие исследования показали, что оценки вырождения межпозвоночных дисков на основе МРТ являются наиболее вероятным прогнозом спинных болей. Существует основное различие между этими исследованиями. В работах, где отрицается связь между радиологическими оценками и наличием патологий, не определяются многие параметры. Например: дефекты, которые можно увидеть в концевой пластине позвонка, также не определяется наличие грыж Шморля, не учитываются возможные переломы позвоночника и т.п [27]. Поэтому данный вид исследований будем считать неполным.

Основной проблемой изучения снимков МРТ позвоночника является наличие межреберной линии, которая позволяет отличать деформированные и здоровые межпозвоночные диски. Это приводит к разности в показаниях одного и того же снимка. Проводилось исследование, когда один пациент был направлен на обследование в 10 поликлиник (с безопасным для здоровья обследуемого интервалом, естественно). Было обнаружено, что согласование между 10 различными радиологическими исследованиями практически отсутствовало. В связи с этим, было решено сопровождать радиологические параметры оценкой достоверности [8]. Данная мера необходима для обеспечения надежности при постановлении диагноза. Для каждого радиологического параметра существует свой доверительный диапазон соответственно.

Вторая проблема - это изменчивость изображений. Снимки МРТ, снятые с разных аппаратов не обладают одинаковыми характеристиками.

Это приводит к разнице в проекциях снимков, разному расстоянию между пикселями и т.п. Эти отличия могут привести к постановке неправильного диагноза. Например, снимок МРТ с помощью одного аппарата может показать наличие деформации верхних границ межпозвоночных дисков [9]. Хотя нет никакой гарантии, что данное отклонение будет присутствовать при сканировании другим аппаратом МРТ. Разрабатываемая модель, которая должна прогнозировать патологии пациента, должна быть устойчивой к подобному рода изменениям.

Разрабатываемое программное обеспечение необходимо для автоматизации задач врача-рентгенолога. Оценки, которые будут производиться системой, инвариантны к внешним факторам. Кроме того, автоматизированная система должна иметь возможность поддерживать принятие клинических решений. Постановка окончательного диагноза принимается врачом-рентгенологом. Но в связи с тем, что система может применяться за пределами отделения радиологии, например, в отделении кардиологии и неврологии, становится все более важным дополнять количественные радиологические показания объективными качественными оценками. В дополнении к количественному результату отчет будет более полным, так как вместо малопонятных радиологических параметров будут присутствовать доступные другим специалистам показатели о наличии патологий у пациентов.

Целью выпускной квалификационной работы является автоматизация процесса сегментации и анализа снимков МРТ поясничного отдела позвоночника. Для достижения поставленной цели ставятся такие задачи как: разработка системы, в которой будут храниться и обрабатываться необходимые данные о пациентах и их обследованиях; создание алгоритма для локализации и последующего анализа межпозвоночных дисков на снимках МРТ поясничного отдела

позвоночника, а также вывода серии количественных радиологических параметров.

## **1 Анализ предметной области**

В первую очередь необходимо кратко рассмотреть анатомию позвоночника человека. Затем необходимо описать основные методы компьютерного зрения, используемые при анализе снимка МРТ позвоночника. Будет рассмотрена локализация и официальное наименование позвонков. Также будет описана классификация, регрессия; рассмотрены основные радиологические оценки и шкалы измерений. Далее необходимо кратко обсудить актуальный существующий прогресс глубокого обучения в области компьютерного зрения. В частности, будет рассмотрен процесс глубокого обучения в области обработки медицинских изображений.

### **1.1 Анатомия позвоночника**

Наглядную детальную иллюстрацию анатомии позвоночника человека можно увидеть на рисунке 1.

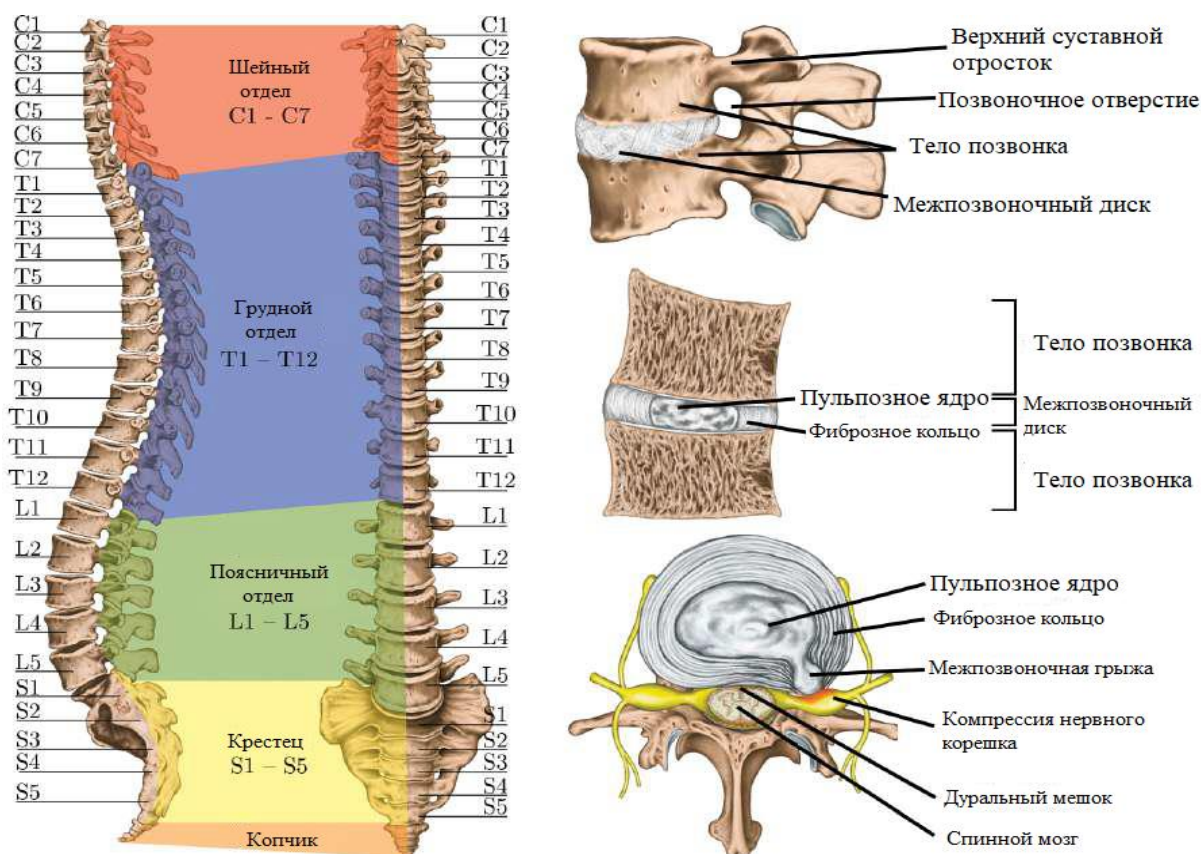


Рисунок 1 - Иллюстрация анатомии позвоночника человека

Слева на рисунке 1 изображена сагиттальная и передняя проекции позвоночника. Справа вверху представлен межпозвоночный диск и основные структурные части 2 позвонков [10]. В центре справа изображен сагиттальный вид межпозвоночного диска вместе с пульпозным ядром и фиброзным кольцом. Справа внизу: осевой вид диска с грыжей Шморля и компрессией нервного корешка. Далее представлено описание основных структурных элементов позвоночника. Знание устройства нижеописанных элементов необходимо для разработки алгоритма сегментации и анализа снимков МРТ.

Нормальный позвоночник человека содержит 33 позвонка, хотя их число может варьироваться. Они сгруппированы в пять основных областей, соответствующих изгибам позвоночника: шейный, грудной, поясничный, крестцовый и копчиковый. Есть семь шейных (C1-C7), двенадцать грудных (T1-T12), пять поясничных (L1-L5), пять крестцовых



(S1-S5), и четыре копчиковых позвонка. Один позвонок состоит из двух основных частей: тела позвонка и позвоночной дуги [19].

Между парами позвонков находятся межпозвоночные диски. Диски отмечаются в соответствии с соседними позвонками, например, межпозвоночный диск L5-S1 - это диск между позвонками L5 и S1 соответственно. Диск состоит из двух основных частей: пульпозного ядра и фиброзного кольца [20]. Хотя пульпозное ядро и фиброзное кольцо по структуре своей сильно различаются. Однако на снимках МРТ данные элементы практически невозможно рассмотреть и дифференцировать. Область, соединяющая диск и тело позвонка, называется концевой пластиной позвонка.

Спинной мозг - это связка нервной ткани, предназначенная для передачи сигналов от мозга к телу. Конкретная область позвоночника управляет определенной двигательной функцией, например, область L1-L4 контролирует сгибание бедра. Спинной мозг окружен мембранной оболочкой, называемой дуральным мешком [22].

## **1.2 Анализ позвоночника с использованием компьютерного зрения**

Анализ изображения позвоночника обычно сводится к исследованию конкретных анатомических частей позвоночника. Межпозвоночные диски и тела позвонков являются наиболее популярными объектами изучения. Общий подход автоматизированного анализа можно разбить на два основных этапа: этап локализации (сегментации) и этап классификации по радиологическим параметрам.

Стандартный подход в локализации анатомических частей позвоночника состоит в том, чтобы разделить проблему сегментации на два основных шага. Первый шаг включает в себя обнаружение

необходимых анатомических частей позвоночника на снимке МРТ. Вторым шагом представляет собой маркировку обнаруженных элементов. Этот подход возник из структуры, которую предоставляла модель, основанная на алгоритме Felzenszwalb, еще в 2005 году. Для обнаружения и маркировки межпозвоночных дисков модель была адаптирована лишь только к 2007 году.

Обнаружение анатомических частей позвоночника необходимо, так снимки МРТ обычно содержат несколько элементов, которые различаются по своему положению в зависимости от модальности снимка, типа сканирования и других параметров. После обнаружения элементы помечаются таким образом, чтобы было возможно отличить одну анатомическую часть от другой на одном снимке.

### **1.3 Методы обнаружения анатомических частей позвоночника**

Существуют основные методы для сегментации структурных элементов позвоночника:

- 1) Корреляционный фильтр. Данный метод был предложен в 2010 году для автоматической локализации позвонков для снимков МРТ в осевых проекциях. Данный метод не позволяет идентифицировать 15% снимков на исследуемом наборе данных. Однако границы позвонков определяются достаточно четко.
- 2) Каскады Хаара. Модель для данной предметной области была разработана еще в 2009 году. Данный метод отлично определяет позвонки в сагиттальных сканограммах, но слабо применим для

определения анатомических частей позвоночника на снимках МРТ в поперечной проекции.

- 3) Гистограмма ориентированных градиентов (HOG). В 2013 году группа исследователей предложила систему, предназначенную для обнаружения позвонков с помощью алгоритма HOG [31]. Результаты показали 5% ошибку на наборе данных.

Вышеописанные методы показывают неплохие результаты на представленных наборах данных. Однако, возможно добиться более высокой точности при сегментации снимков МРТ, используя современный нейросетевой подход на основе сверточных нейронных сетей. Описание и анализ основных нейросетевых моделей будут представлены впоследствии.

После локализации анатомических частей позвоночника обычно следует шаг, связанный с маркировкой каждого сегментированного элемента изображения. Обычно маркировка осуществляется с использованием графической модели, которая в свою очередь основывается на пространственном соотношении между сегментированными объектами и эталонным изображением в нужной проекции. Также модели необходимо анализировать пространство вокруг каждого объекта для определения, например, номера позвонка и проекции снимка.

#### **1.4 Модели формирования радиологических параметров**

Когда все анатомические части позвоночника локализованы и помечены, становится возможным сформировать классификатор основных радиологических параметров. Это важный шаг в создании автоматической системы анализа позвоночника. Ключевой информацией, основываясь на которой становится возможным строить объективные оценки о состоянии сегментированных частей позвоночника, является: форма субъекта и

интенсивность области вокруг сегментированного элемента. Для последующего анализа необходимо воспользоваться моделью классификации исследуемых объектов.

Основные классификаторы, которые можно использовать при формировании аналитической модели, описаны ниже:

- 1) Модель, основанная на экспертных знаниях. В 2017 году данная модель была использована при диагностике стеноза поясничного отдела позвоночника. Суть метода заключается в использовании отмеченных рентгенологами ранее радиологических признаков для построения дерева принятия решений. В методе применяются количественные характеристики, например, площадь позвоночного канала, а также качественные признаки.
- 2) Модель, основанная на формах исследуемых элементов. Для распознавания грыж Шморля в 2014 году предложили использовать функцию определения формы контура. Данная модель позволяет определять уровень деформации межпозвоночных дисков с точностью до 93,9%. Для выявления сколиоза была предложена геометрическая модель, основанная на характеристике искривления позвонков. Общая идея использования параметрической кривой хорошо работает для определения деформаций позвоночника. Для диагностики стеноза позвоночника, используются в качестве признаков диаметры и длины определяемых моделью ориентиров, окружающих позвоночный канал [23].
- 3) Модель, основанная на интенсивности изображения. В 2012 году была предложена система, использующая соотношение пикселей диска, позвонков и спинного мозга в пределах области интересов,

для классификации грыжи Шморля с точностью до 99%. Процесс не является автоматическим, но прекрасно работает с оператором.

- 4) Формоинтенсивная модель. В 2014 году был изобретен метод, который использовал гистограмму информации об интенсивности межпозвоночных дисков вместе с формой контуров для построения функции классификации. Точность данной автоматизированной модели составляет 94,9%.

В целом, нынешний подход определения радиологических параметров заключается в индивидуальной настройке для решения конкретных задач. Это отнимает много времени, особенно если нужно классифицировать несколько снимков с разных аппаратов МРТ. Также любая классификация подвержена ошибкам и не является абсолютно точной. Обеспечение стопроцентной точности в систематизировании означает значительное снижение производительности при работе алгоритмов. В связи с этим приходится мириться с возможными ошибками при классифицировании.

Для разрабатываемой информационной системы будет использоваться формоинтенсивная модель классификации, так как она является полностью автоматизированной и обладает наиболее высокими показателями точности.

## **1.5 Сверточные нейросети и области их применения**

Наиболее современным подходом для решения задач локализации объектов являются сверточные нейронные сети (CNN). CNN - это тип нейронной сети, предназначенный для работы с данными, которые пространственно связаны между собой, например, изображения, видео, аудиофайлы. Все это является частью предметов исследования машинного обучения, которое теперь также называется глубоким обучением. Стандартная модель CNN состоит из пяти элементов: ядра свертки,

входного слоя, выходного слоя, функции активации (обычно ReLU) и внутренних слоев. При обучении нейросетевой модели наборы весов изменяются так, что входные данные становятся разделимыми на выходном слое CNN [6]. Основные области применения сверточных нейронных сетей описаны ниже.

Сверточные нейронные сети применяются в задачах классификации. Архитектура AlexNet, впервые представленная в 2012 году, является одной из самых известных структур CNN для классификации изображений. Модель достигает высокого уровня производительности, как в задачах классификации, так и в задачах локализации. Исследование проводилось на крупномасштабной базе данных изображений в рамках проекта ILSVR. Модель использовала дропаут вместо регуляризации для избежания переобучения [29]. Также авторами используется перекрытие объединений для уменьшения размера нейросети. Успех данной архитектуры повысил популярность методов глубокого обучения в задачах компьютерного зрения. В результате чего, методы глубокого обучения стали преобладать в решении задач проекта ILSVR с 2013 года.

Архитектура нейросетевой модели AlexNet представлена на рисунке 2.

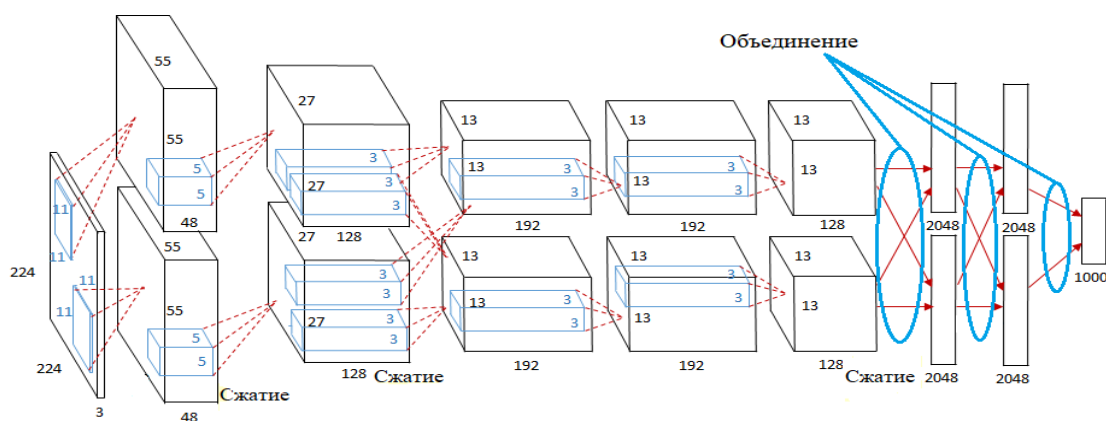


Рисунок 2 – Архитектура нейросетевой модели AlexNet

В 2013 году в рамках ILSVR была разработана другая модель на основе сверточных нейронных сетей, которая решала задачу классификации лучше, чем AlexNet. В данной модели использовалась деконволюционная сеть, на основе которой, в дальнейшем были разработаны такие популярные нейросетевые архитектуры как VGGNet и GoogleNet. В обоих алгоритмах использовались ядра свертки размера (3 x 3) и большое число скрытых слоев. Данные разработки послужили фундаментом для создания лучшей архитектуры классификации изображений на сегодняшний день под названием ResNet [25].

Также сверточные нейронные сети активно применяются в задачах сегментации изображений. Одним из лучших методов семантической сегментации является метод, основанный на сверточной нейросетевой модели. Метод использует семантическую пиксельную классификацию. Для реализации вышеописанной концепции используется объединение неглубоких слоев [26]. Эти слои близки по структуре к входному изображению. Интересно отметить то, что для данной модели не замечается улучшения точности сегментации при добавлении данных для обучения (что можно наблюдать в моделях для классификации изображений). Точность сегментации составляет 62,2% на наборе данных PASCAL VOC 2012 года.

Также набирают популярность нейросетевые сверточные модели, реализующие процесс сегментации благодаря предшествующему процессу классификации. Было разработано несколько методов локализации изображений с использованием CNN, используя параллельно обученную модель классификации. Наиболее известны 3 метода: метод обратного распространения ошибок по Симоняну (разработан в 2014 году), который вычисляет градиент прогноза оценки класса по отношению к входному изображению; метод управляемого обратного распространения по Спингенбергу, который изменяет обратный проход функции активации ReLU; метод обратного распространения ошибки по Чжану, который изменяет градиенты скрытых слоев модели [24]. Существует еще один метод, который не рассматривается в данной работе - это метод Class Activation Maps (CAM), который применяется для перерасчета градиентов первых и последних скрытых слоев нейросети.

Все вышеописанные архитектуры, модели и алгоритмы активно применяются в обработке медицинских изображений. Об этом далее.

## **1.6 Сверточные нейросети в обработке медицинских изображений**

Сверточные нейросетевые модели стали все больше и больше использоваться в медицинских системах, диагностирующих проблемы со зрением. Существует два основных способа реализовать свою сверточную нейросеть: использовать заготовленную весовую модель, либо обучить сеть с нуля.

Идея повторного использования нейросетей, обученных до этого на другом наборе данных, называется трансферным обучением. Было обнаружено, что трансферное обучение для сверточных нейросетевых моделей чрезвычайно эффективно. Особенно, когда модель была



предварительно обучена на большом наборе данных, например, таком как ImageNet [21].

Были созданы высокоточные модели, предварительно обученные на медицинских изображениях, предназначенные для классификации болезней легких и выявления рака кожи. Однако если предварительная модель была обучена на неспециализированном наборе данных, то применение такой нейросети становится нежелательным для использования в медицинских задачах. Это связано с различиями структуры данных изображений. Медицинские изображения могут быть объемными по своей природе, и включать множество дополнительных данных. В то время как обычные снимки, как правило, состоят из 3 цветовых каналов. Еще одним отличием является память, выделяемая для хранения каналов цветового диапазона, который составляет 24 бита для обычного изображения (8 бит на канал). Для медицинских изображений отличается память, выделяемая для хранения каналов диапазона. Для изображений КТ используется 12 бит на канал, а для МРТ 16 бит на канал. Еще одно существенное различие заключается в том, что медицинские изображения очень похожи во внешней структуре. То есть набор данных со снимками МРТ поясничного отдела позвоночника очень однороден по внешнему виду в отличие от снимков, представленных в ImageNet. Однако, эти различия не означают то, что целесообразнее создавать специфичные для каждого аппарата МРТ нейросетевые модели. Используя общепризнанные архитектуры для сверточных нейронных сетей, вполне возможно создать относительно универсальную и точную модель, позволяющую сегментировать снимки, предоставляемые с разных аппаратов МРТ.

Если набор исходных данных достаточно велик, то лучше обучать нейросетевую модель с нуля. Существует способ разрешить проблему недостаточного объема данных для задач сегментации. Метод заключается

в том, чтобы разделять исходные изображения на патчи. Одним из успешных примеров решения вышеописанной проблемы является модель, которая победила в рамках соревнования MICCAI 2013. Модель предназначалась для сегментации разделов мозга. При решении данной задачи сегментации использовалась одна из самых ранних архитектур CNN. Архитектура использовала операции подвыборки объединения в качестве средства для обеспечения трансляционной дисперсии входного изображения. Наряду с этим, архитектура использует скользящее окно для получения небольших участков изображения. Самый центральный пиксель затем классифицируется как определенный класс, в дальнейшем определяющий маски границ в дальнейшем процессе сегментации. В 2015 году была предложена новая нейросетевая архитектура для сегментации медицинских изображений, состоящая из сверточных и деконволюционных фильтров, с пропускными соединениями, соединяющими фильтры вместе. Вместо пиксельной классификации данная архитектура CNN создает карту сегментации с помощью деконволюции [7]. Данная архитектура создает более гладкую маску сегментации, но может потерять выделяемые объекты, если область сегментации мала. Также данная модель использует агрессивный режим обучения. Исходная выборка увеличивается благодаря эластичному масштабированию. Однако данный процесс требует сильного пиксельного контроля.

Идея многозадачности в глубоком обучении была исследована как на обычных изображениях, например, на наборе данных UberNet так и на медицинских снимках. На основе набора данных UberNet была создана нейросеть, которая была обучена распознавать изображения 7 различных классов, а также применять к классифицированным изображениям семантическую сегментацию [30]. Однако позже было обнаружено, что

обучение по нескольким критериям оказалось пагубным для точности модели. Также для многозадачных моделей общая производительность оказалась ниже, чем для однозадачных моделей. Можно выдвинуть гипотезу, что многозадачность полезна только тогда, когда и данные, используемые в обучении, достаточно тесно взаимосвязаны между собой.

## **1.7 Сравнительный анализ сверточных нейросетевых архитектур**

Сегодня существует множество архитектур CNN, решающих задачу сегментации. Помимо определения класса изображения, определенные архитектуры сверточных нейронных сетей позволяют создавать маски, делящих изображение на классы. Наиболее популярные архитектуры CNN описаны ниже.

### **1.7.1 Архитектура U-Net**

Архитектура U-Net представляет собой кодирующую-декодирующую нейросеть, состоящую из двух частей:

- 1) Стягивающего пути (кодирования) для получения контекста изображения с помощью компактной карты объектов;
- 2) Симметрично расширяющегося пути (декодера), который позволяет точно сегментировать необходимый объект благодаря сохранению пространственной информации об исследуемом объекте.

Данная архитектура быстро обучается на небольшом количестве изображений сквозным способом. Также сеть на основе U-Net очень быстро сегментирует изображения. Доказательством состоятельности данной архитектуры, является победа в конкурсе ISBI в 2015 году при решении задачи обнаружения клеток. Таким образом, можно резюмировать характерные особенности U-Net:

- Получение высоких результатов на реальных задачах (в том числе и при обработке медицинских изображений);
- Малый объем данных необходимый для высокоточного обучения сети.

На рисунке 3 изображен пример архитектуры U-Net для изображения размером (512 x 512).

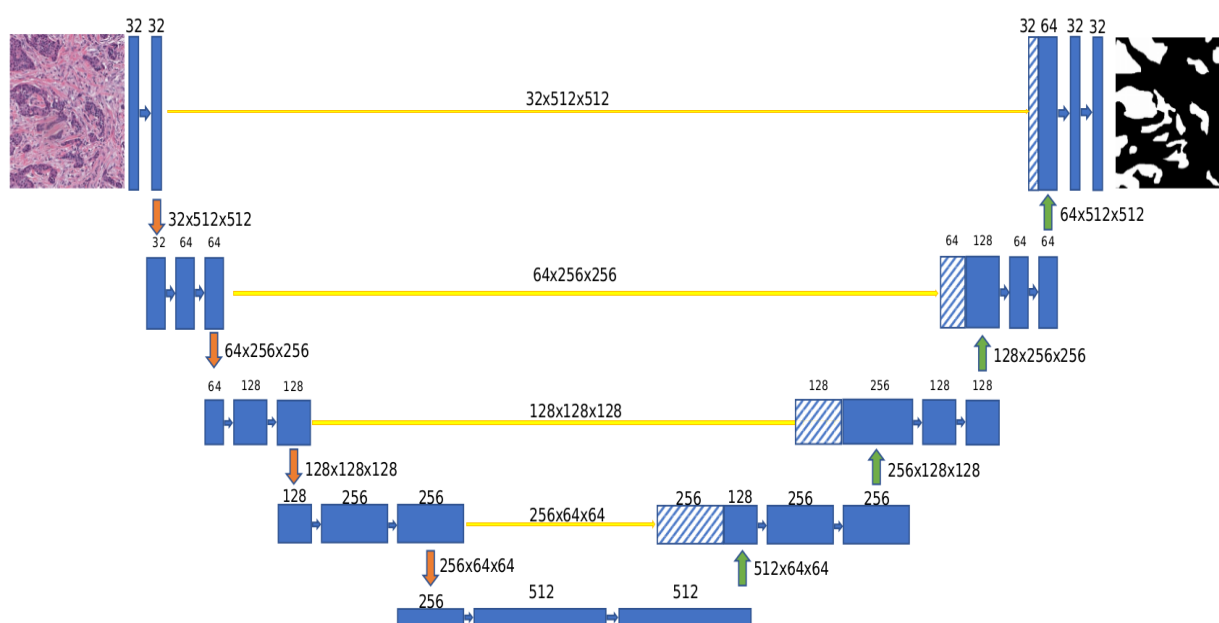


Рисунок 3 – Архитектура U-Net

### 1.7.2 Архитектура PSPNet

Определенные модели глубокого обучения предоставляют возможность работать с разномасштабной информацией. Нейросеть Pyramid Scene Parsing (PSPNet) изначально применяет операцию объединения (с использованием функции поиска максимума или среднего значения) используя разноразмерные ядра. После используется билинейная интерполяция. Использование этой функции приводит к пересчету размерности всех выходов с pooling слоя, а также к перерасчету отображения карт признаков [28]. Затем модель интегрирует новые выходы вдоль оси канала. И для генерации конечной карты признаков последняя свертка применяется уже на объединенном выходе.

Данная архитектура позволяет сегментировать группы объектов с достаточной точностью. Но для распознавания 1 класса объектов данная модель является избыточной.

На рисунке 4 показан пример вышеописанной работы архитектуры PSPNet.

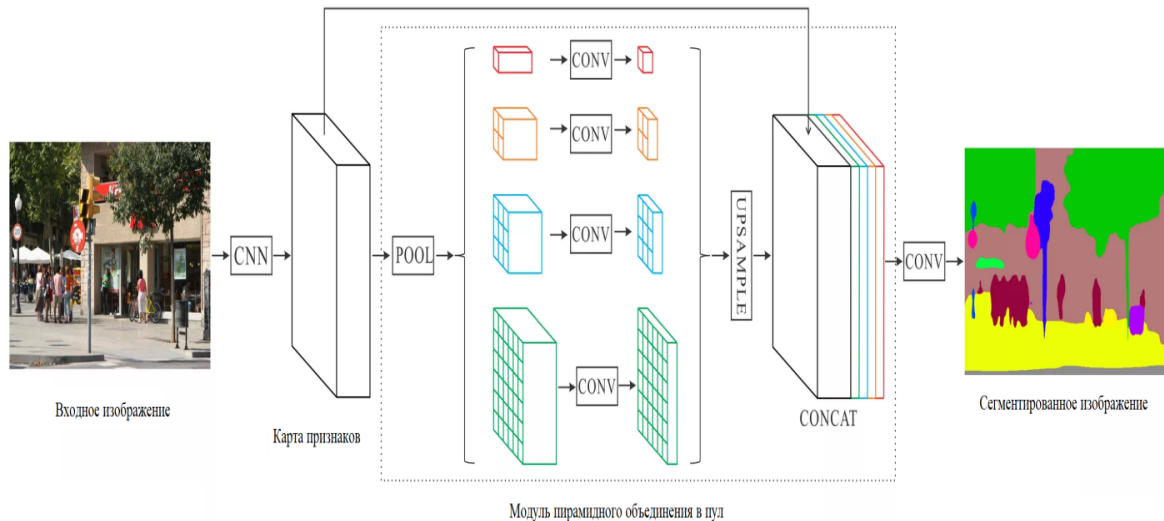


Рисунок 4 – Архитектура PSPNet

### 1.7.3 Архитектура FPN

Для архитектуры Feature Pyramid Network (FPN) характерен следующий принцип работы. Сформированные скрытыми слоями сверточной нейронной сети карты признаков, с симметрично сокращающейся размерностью, иерархически формируются в пирамиду, под названием bottom-up pathway. Организованные карты признаков нижних и верхних уровней bottom-up pathway обладают своими характеристиками. Нижние уровни имеют малый семантический уровень, но обладают высоким разрешением, верхние уровни – наоборот [15].

Архитектура FPN объединяет преимущества нижних и верхних слоев. Для этого добавляются 2 промежуточных соединения: top-down pathway connection и lateral connection. Карта признаков каждого дочернего слоя масштабируется до размера родительского и их элементы

суммируются. Для формирования финального предсказания используются карты признаков со всех уровней пирамиды.

Вышеописанная нейросетевая модель, как и предыдущая, необходима для сегментации изображений, содержащих большое количество объектов разных классов. Для разрабатываемой модели, данная архитектура будет излишней. На рисунке 5 изображен процесс работы архитектуры FPN.

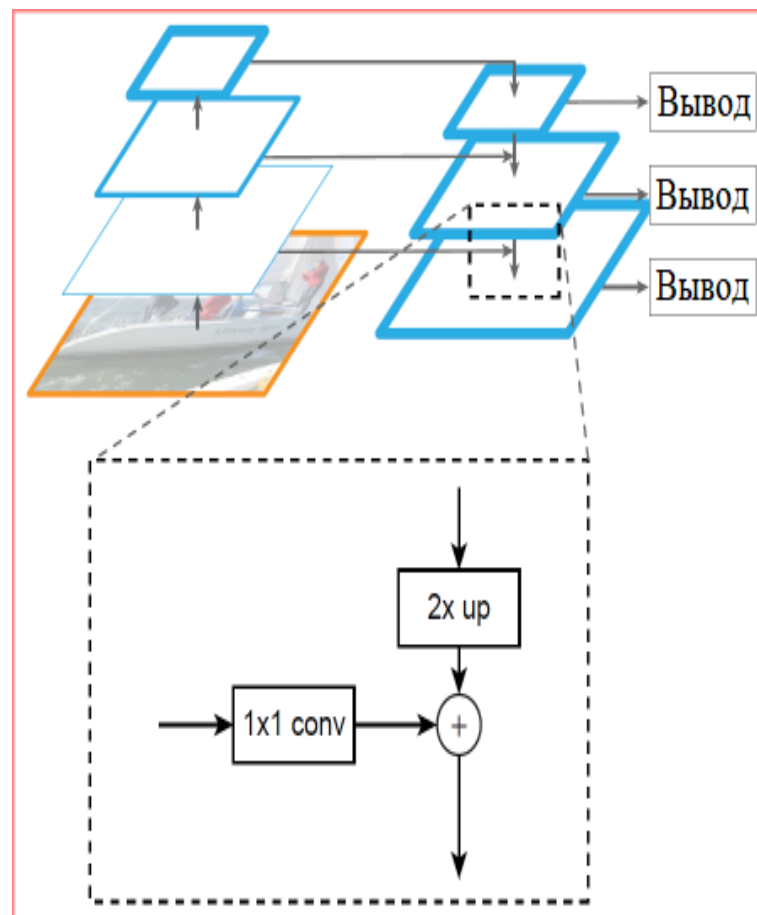


Рисунок 5 – Архитектура сети FPN

#### 1.7.4 Архитектура Linknet

Существующие алгоритмы семантической сегментации хоть и являются высокоточными, но по большей части не ориентированы на эффективное использование параметров нейронной сети. В результате, нейросетевые модели, основанные на таких алгоритмах огромны по числу

параметров и количеству выполняемых операций. Следовательно, данные модели медлительны. Linknet представляет собой архитектуру нейронной сети, схожей с U-Net [14]. Однако данная архитектура позволяет модели обучаться без значительного увеличения числа параметров. Linknet использует только 11 миллионов параметров для обработки изображения с разрешением (640 x 360).

Несмотря на скорость обучения и малое число параметров, данная архитектура обладает малой точностью при сегментации изображений. Поэтому ее использование не применимо для обработки медицинских изображений. На рисунке 6 изображен пример архитектуры Linknet.

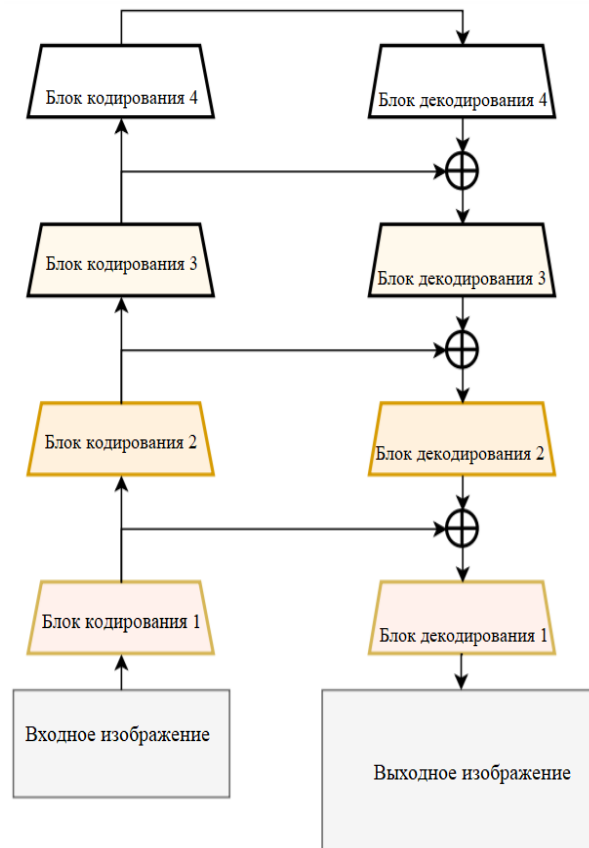


Рисунок 6 – Архитектура сети Linknet

### 1.7.5 Результаты сравнительного анализа нейросетевых архитектур

В таблице 1 представлены характеристики архитектур сверточных нейронных сетей, которые непосредственно учитывались при выборе архитектуры для разрабатываемой нейросетевой модели.

Таблица 1 – Характеристики архитектур сверточных нейронных сетей

Архитектура \ Характеристика	U-Net	Linknet	FPN	PSPNet
<i>Высокая точность на малом наборе данных</i>	+	+	-	+
<i>Высокая производительность</i>	+	+	-	+
<i>Простота реализации</i>	+	+	-	-
<i>Применимость для сегментации медицинских изображений</i>	+	-	+	+

Исходя из вышеописанных характеристик архитектур сверточных нейронных сетей, в разрабатываемой системе сегментации снимков МРТ позвоночника, будет использоваться архитектура U-Net в связи с простотой реализации, отсутствием необходимости в большом наборе данных, а также широкой применимостью при решении задач сегментации медицинских изображений.

### 1.8 Сравнительный анализ существующих аналогов

Описываемая предметная область исследовалась неоднократно. Необходимо провести сравнительный анализ существующего программного обеспечения и информационных систем сегментации и анализа снимков МРТ позвоночника, определить слабые и сильные



стороны аналогов, чтобы разработать высокоточную и универсальную систему. Наиболее популярные программные решения описаны далее.

### **1.8.1 Программное обеспечение RadiAnt DICOM Viewer**

RadiAnt DICOM Viewer – это программное обеспечение, предназначенное для просмотра и обработки медицинских изображений стандарта DICOM.

Данное программное обеспечение обладает рядом следующих преимуществ:

- Набор инструментов для обработки изображений и измерения их основных параметров;
- Масштабирование и панорамирование снимков;
- Возможность корректировки контрастности и яркости изображений;
- Определение величин нормального угла и угла Кобба;
- Определение значений плотностей тканей в единицах Хаунсфилда;
- RadiAnt DICOM Viewer полноценно функционирует на рабочих станциях даже с 512 Мб оперативной памяти;
- Программное обеспечение может поддерживать разные типы DICOM файлов;
- Присутствует экспорт DICOM файлов в изображения;
- Наличие многоязычного интерфейса;
- Присутствует движок для 3D рендеринга;
- Возможно совмещение изображений ПЭТ-КТ.

На рисунке 7 показан пример отображения серии снимков МРТ позвоночника в программе RadiAnt DICOM Viewer.



Рисунок 7 – Работа программы Radiant DICOM Viewer

### 1.8.2 Информационная система Horos

Horos – это бесплатная информационная система с возможностью просмотра медицинских изображений формата DICOM с открытым исходным кодом. Horos представляет собой настольное и мобильное приложения, позволяющие формировать отчеты по радиологическим обследованиям. Horos основан на OsiriX и других библиотеках медицинской визуализации.

Данная информационная система обладает приятным пользовательским интерфейсом, большим набором инструментов для обработки DICOM файлов и изображений, однако не портирована на ОС Windows и Linux, и доступна только на macOS от версии 10.11 и выше. Также у Horos отсутствует поддержка многоязычности. Но в отличие от предыдущего программного обеспечения в системе Horos можно автоматически формировать отчеты по проведенным исследованиям.

### 1.8.3 Программное обеспечение Navegatum

Navegatum - это бесплатное программное обеспечение для просмотра DICOM файлов. Navegatum позволяет визуализировать, анализировать и структурировать файлы DICOM. Также имеется интеграция с наиболее популярными PACS (системами передачи и архивации DICOM изображений).

Данное программное обеспечение обладает схожим функционалом с RadiAnt DICOM Viewer, однако в Navegatum отсутствует многоязычный интерфейс. На рисунке 8 приведен пример работы программы Navegatum.



Рисунок 8 – Интерфейс программы Navegatum DICOM Viewer

### 1.8.4 Программное обеспечение ПОЗВОНОЧНИК 3D

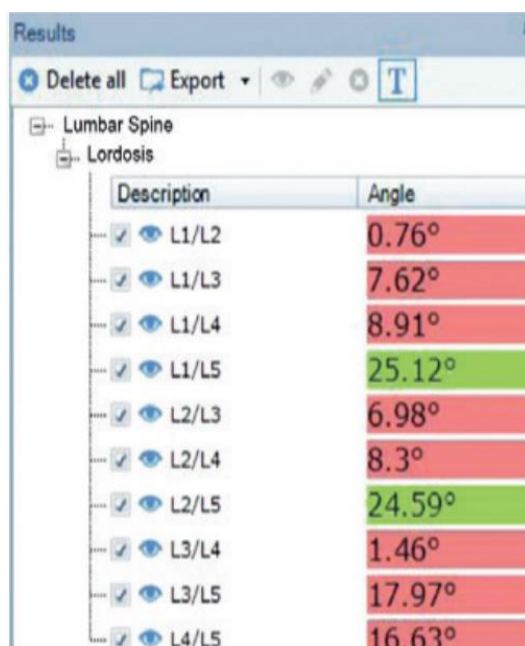
Программное обеспечение ПОЗВОНОЧНИК 3D позволяет проводить качественные анатомической оценки, определять измерения позвоночника. Возможна работа в 3D пространстве с данными КТ и МРТ и классическими 2D снимками (например, в формате DICOM или JPEG).

Программа имеет современный, интуитивный пользовательский интерфейс, а также позволяет проводить удобную интеграцию с

существующей PACS-системой. Программное обеспечение обладает следующими характеристиками:

- Автоматическое сегментирование снимков МРТ;
- Автоматическое определение размеров анатомических частей позвоночника;
- Классические радиологические измерения;
- Визуализация измеренных параметров;
- Интерактивная помощь;
- Подключение к регистрам.

На рисунке 9 изображена экранная форма определения состояния межпозвоночных дисков программного обеспечения ПОЗВОНОЧНИК 3D.



Description	Angle
<input checked="" type="checkbox"/> <input type="checkbox"/> L1/L2	0.76°
<input checked="" type="checkbox"/> <input type="checkbox"/> L1/L3	7.62°
<input checked="" type="checkbox"/> <input type="checkbox"/> L1/L4	8.91°
<input checked="" type="checkbox"/> <input type="checkbox"/> L1/L5	25.12°
<input checked="" type="checkbox"/> <input type="checkbox"/> L2/L3	6.98°
<input checked="" type="checkbox"/> <input type="checkbox"/> L2/L4	8.3°
<input checked="" type="checkbox"/> <input type="checkbox"/> L2/L5	24.59°
<input checked="" type="checkbox"/> <input type="checkbox"/> L3/L4	1.46°
<input checked="" type="checkbox"/> <input type="checkbox"/> L3/L5	17.97°
<input checked="" type="checkbox"/> <input type="checkbox"/> L4/L5	16.63°

Рисунок 9 – Интерфейс программы ПОЗВОНОЧНИК 3D

### 1.8.5 Результаты сравнительного анализа аналогов

Анализируя основные преимущества и недостатки вышеописанных программных решений, предназначенных для работы со снимками МРТ, были выделены основные характеристики, которыми должна обладать разрабатываемая система. Такими параметрами являются:

- Независимость от оборудования;
- Наличие обработки результата;
- Вычисление параметров и диагностика межпозвоночных дисков;
- Учет пациентов.

В таблице 2 представлены параметры вышеописанных программных решений.

Таблица 2 – Анализ наличия выделенных параметров программных решений

Характеристика Система	<i>RadiAnt DICOM Viewer</i>	<i>Horos</i>	<i>Navegatum</i>	<i>ПОЗВОНОЧНИК 3D</i>
<b>Независимость от оборудования</b>	+	-	-	-
<b>Наличие обработки результата</b>	+	+	+	+
<b>Вычисление параметров межпозвоночных дисков</b>	-	-	-	+
<b>Диагностика межпозвоночных дисков</b>	-	-	-	-
<b>Учет пациентов</b>	-	+	-	+

Анализируя программные решения, можно сделать вывод, что во всех вышеописанных системах, за исключением ПО ПОЗВОНОЧНИК 3D, отсутствует семантическая сегментация, следовательно, и дальнейший анализ исследуемых анатомических частей. Но недостатками программы ПОЗВОНОЧНИК 3D являются наличие зависимости от аппаратуры МРТ, а также отсутствие предварительной диагностики о возможных патологиях межпозвоночных дисков.

Разрабатываемое программное обеспечение должно обладать возможностью автоматически сегментировать межпозвоночные диски на

снимках МРТ поясничного отдела позвоночника. Также необходимо анализировать параметры локализованных объектов и представлять предварительные диагнозы о состояниях сегментированных частей позвоночника в виде удобочитаемого отчета. Необходимо учитывать возможные ошибки, которые могут возникать при сегментировании снимков. Будет необходимо учитывать отклонения, которые будут учитываться визуально врачом-рентгенологом, а также дифференцировать количественные показатели оценки точности локализации межпозвоночных дисков.

Дополнительной задачей ставится визуализация результатов работы системы сегментации. Для возможности зрительного анализа рентгенологом снимков на наличие неточностей необходимо накладывать исходное и локализованное изображения друг на друга. Также система должна предоставлять информацию об исходном и локализованном изображениях. Для формирования количественных оценок о точности сегментации необходимо учитывать локализуемые границы межпозвоночных дисков с точностью до пикселя. Благодаря информации о точных границах межпозвоночных дисков для определения процента точности сегментации будет вычисляться разница между сегментированным и скорректированным изображениями.

В дальнейшем, возможно расширение системы, для анализа других анатомических частей на изображениях МРТ.

## **2. Проектирование системы**

Исходя из вышеописанного анализа предметной области, необходимо разработать программное обеспечение, позволяющее вести учет пациентов, историю их болезней и обследований. Основной функциональной составляющей ПО - будет являться система сегментации и анализа снимков МРТ позвоночника поясничного отдела пациента. Рентгенолог должен получать данные в виде отчета, по результатам проведенного им обследования.

### **2.1 Назначение системы**

Проектируемая система предназначена для комплексной автоматизации работы специалистов рентгенологов-технологов в поликлиниках, в части исполнения следующих процессов:

- передача и учет поступающих снимков МРТ;
- сегментация межпозвоночных дисков на снимках поясничного отдела;
- анализ сегментированных снимков на наличие грыж Шморля;
- ведение административного документооборота по пациентам;
- ведение диагностики показателей пациентов;
- ведение архивов с анамнезами пациентов без ограничения сроков давности.

### **2.2 Цели создания системы**

Основными целями создания ИС являются:

- Повышение эффективности исполнения процессов, описанных выше, путем сокращения непроизводительных операций, выполняемых вручную;

- Оптимизация информационного взаимодействия участников процессов;
- Повышение качества принятия экспертных решений благодаря оперативности представления, удобству форматов отображения информации.

Для реализации поставленных целей система должна решать следующие основные задачи:

- Ввод снимков МРТ позвоночника поясничного отдела;
- Внесение и редактирование информации о пациентах;
- Построение медицинских отчетов с результатами диагностики.

### **2.3 Перечень подсистем**

В состав проектируемой ИС должны входить следующие подсистемы:

- Подсистема хранения данных;
- Подсистема сегментации снимков МРТ;
- Подсистема анализа сегментированных снимков;
- Подсистема формирования отчетности.

Подсистема хранения данных предназначена для хранения оперативных данных, основной информации о пациентах и их историй болезни, данных аналитических отчетов, ключевые данных о сотрудниках поликлиники (рентгенологах и регистраторах).

Подсистема сегментации снимков МРТ поясничного отдела позвоночника предназначена для первичной обработки изображений. Данная подсистема должна убирать шумы и лишнюю информацию со снимков. Далее должен выполняться процесс выделения межпозвоночных дисков на обработанных изображениях. Также, подсистема отвечает за



передачу данных в подсистему анализа выделенных межпозвоночных снимков.

Подсистема анализа предназначена для анализа основных радиологических параметров обработанных сегментированных данных со снимков МРТ и представления их в удобочитаемом виде для дальнейшей передачи в подсистему формирования отчетности.

Подсистема формирования отчетности предназначена для создания, понятных радиологам, отчетов по проведенным исследованиям, формирования и предоставления по запросам рентгенологов аналитических и статистических отчетов в различных форматах (включая графические).

## **2.4 Роли пользователей и их функциональные возможности**

В проектируемой системе будет обеспечено разделение ролей пользователей, администраторов и лиц, обеспечивающих функционирование информационной системы, в соответствии с их должностными обязанностями. Основные роли и их функциональные возможности описаны ниже:

- Гость
  - Возможность авторизации;
  - Просмотр гостевой информации;
  - Возможность отправить отзыв администратору;
- Пациент
  - Доступен основной функционал роли «Гость»;
  - Пользователи с данной ролью регистрируются в системе секретарем;
  - Возможность получать результаты обследования на почту;

- Секретарь
  - Доступен основной функционал роли «Гость»;
  - Регистрация пациентов;
  - Учет истории болезней пациентов;
  - Назначение обследований для рентгенологов;
- Рентгенолог
  - Доступен основной функционал роли «Гость»;
  - Возможность начать или отклонить, назначенное обследование;
  - Возможность загрузки снимков пациента в систему анализа и сегментации;
  - Оформление отчета и закрытие обследования;
- Администратор
  - Доступен основной функционал роли «Гость»;
  - CRUD операции над всеми сущностями системы.

## **2.5 Описание предметной области посредством диаграмм IDEF**

Представление модели предметной области на основе графических языков позволяет сделать работу более информативной. Функционирование любой системы удобно описывать на основе нотации методологии структурного анализа SADT. Используемый стандарт описания IDEF0 позволяет строить иерархию диаграмм. IDEF0 – модель предназначена для описания функционирования организации на основе подхода «от общего к частному». Такие модели считаются наиболее наглядными, причем для их построения не требуется сложных построений.

Методология IDEF0 широко используется благодаря простой и понятной для понимания графической нотации, применение которой для

построения модели очень удобное [11]. На диаграммах отображаются функции системы. На рисунке 10 изображена начальная контекстная диаграмма системы.

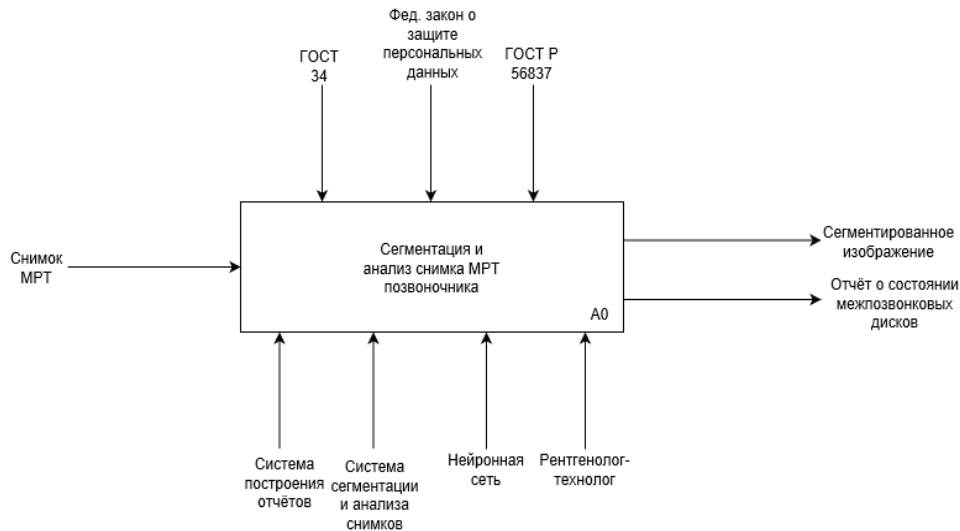


Рисунок 10 – Начальная контекстная диаграмма

На диаграмме показана общая постановка задачи: «Сегментация и анализ снимка МРТ позвоночника». Для расшифровки задачи используется диаграмма уровня А0.

На рисунке 11 представлена контекстная диаграмма IDEF0 модели.

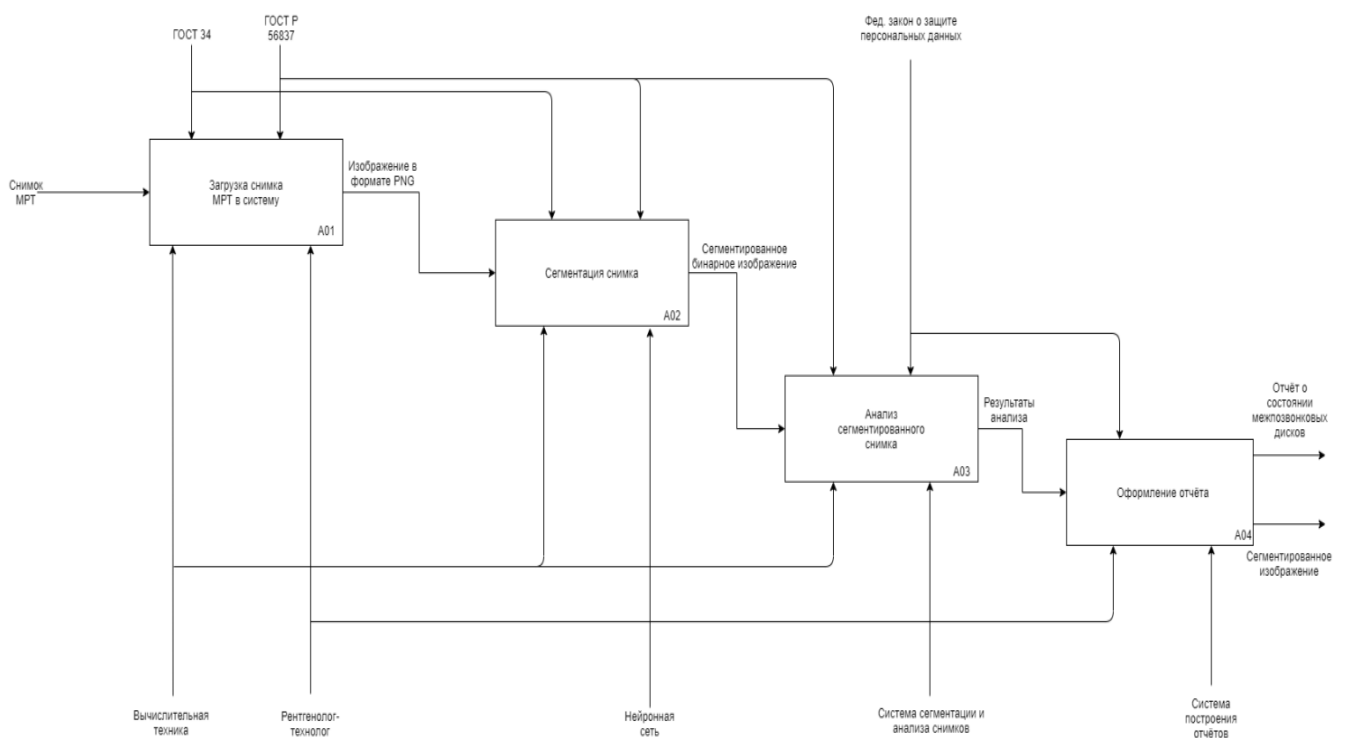


Рисунок 11 - Диаграмма уровня А0 для информационной системы

IDEF3 - методология моделирования и стандарт документирования процессов, происходящих в системе. IDEF3 широко применяется при разработке информационных систем [12]. При этом используется инструмент визуального моделирования бизнес-процессов.

Для процесса загрузки снимка в систему и его анализа до получения сегментированного изображения, которое будет анализироваться, построена диаграмма семейства IDEF3 и представлена на рисунке 12.

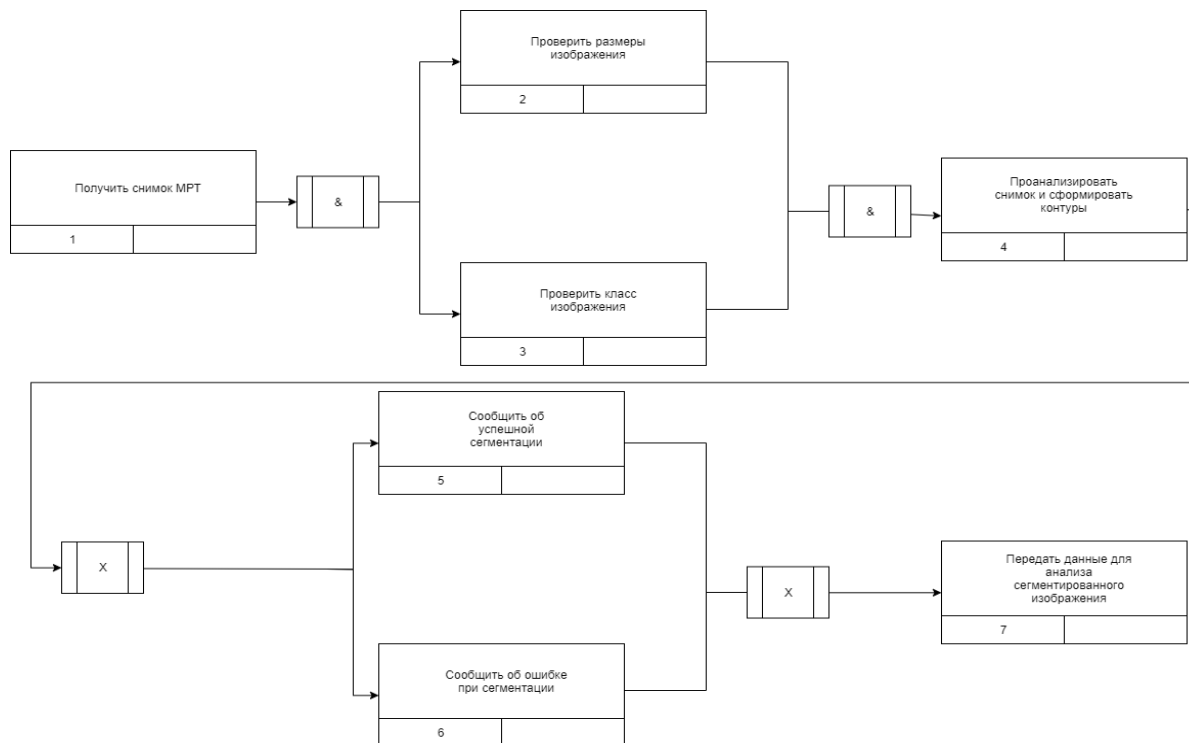


Рисунок 12 - Диаграмма IDEF3 процесса сегментации снимка МРТ

На диаграмме представлено описание технологических процессов информационной системы, с указанием того, что происходит на каждом этапе процесса. На диаграмме присутствуют разворачивающие потоки данных соединения, предназначенные для завершения одного действия и вызова начала выполнения других. А также сворачивающие соединения, выполняющие обратные действия.

Для процесса анализа сегментированного изображения и построения на выходе отчета для рентгенолога-технолога построена диаграмма семейства IDEF3 и представлена на рисунке 13.

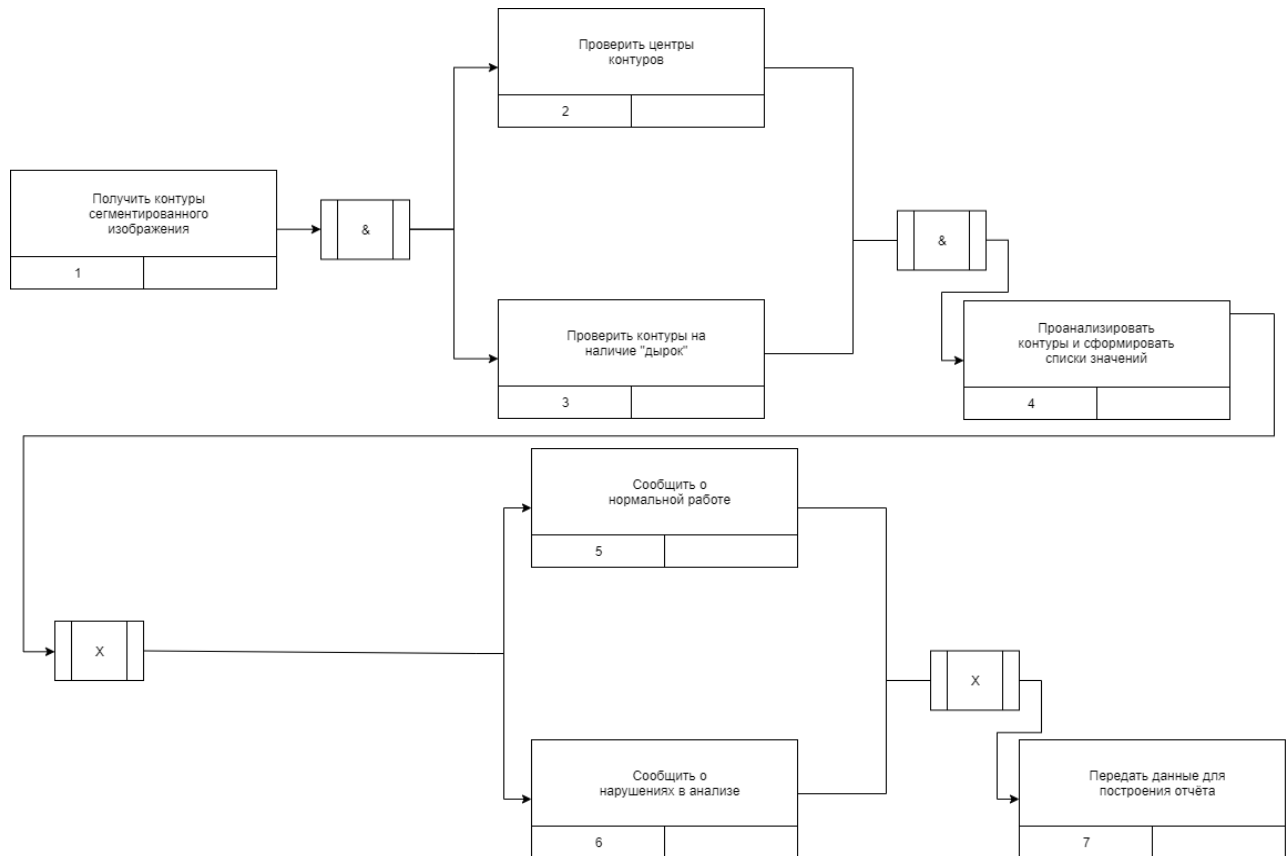


Рисунок 13 - Диаграмма IDEF3 для процесса анализа сегментированного изображения

Для уточнения описания предметной области разработки информационной системы используется методология диаграмм потоков данных (DFD – Data Flow Diagram).

Нотации DFD предназначены в первую очередь для представления реально существующих в организации потоков данных или материальных потоков, или их совокупности. Диаграммы потоков данных показывают, как каждый процесс преобразует свои входные данные в выходные, и выявляют отношения между этими процессами. DFD представляет моделируемую систему как сеть связанных работ.

Модель предметной области, разработанная на основе методологий DFD, представлена на рисунке 14.

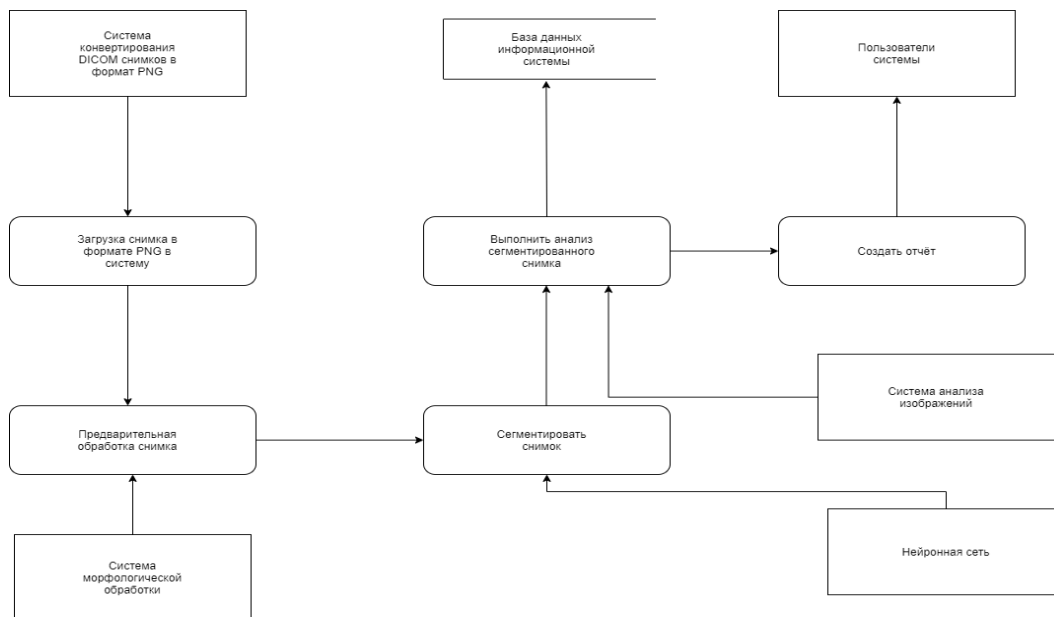


Рисунок 14 - Диаграмма DFD для разрабатываемой информационной системы

## 2.6 Описание предметной области посредством диаграмм UML

Для корректного дополнения описания предметной области используется унифицированный язык моделирования UML. Описание производится с помощью диаграмм прецедентов, диаграмм классов, а также диаграмм последовательности и других менее популярных диаграмм. Благодаря подробному описанию предметной области средствами UML появляются все условия для создания информационной системы, которая будет удовлетворять требованиям заказчика [13].

Моделирование предметной области с помощью языка UML включает в себя концептуальную и логическую диаграммы вариантов использования UML. Первым делом необходимо учесть главных действующих пользователей системы и описать их функции.

На рисунке 15 представлена концептуальная диаграмма вариантов использования UML, включающая такие действующие лица, как администратор базы данных информационной системы, пациент и рентгенолог-технолог.

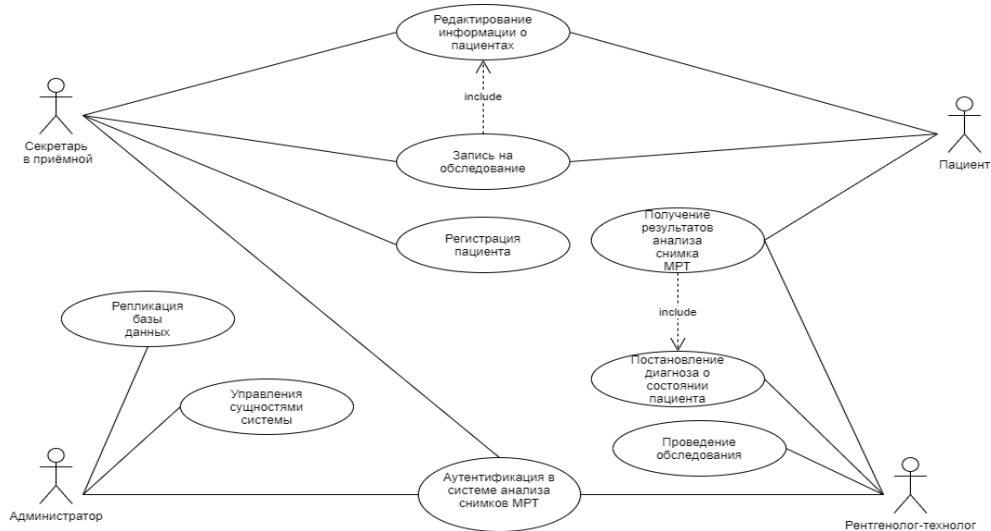


Рисунок 15 - Концептуальная диаграмма вариантов использования

Также при проектировании системы необходимо учесть и выделить основные сущности, которые помогут с дальнейшей реализацией базы данных для информационной системы [3].

На рисунке 16 представлена диаграмма классов для разрабатываемой системы.

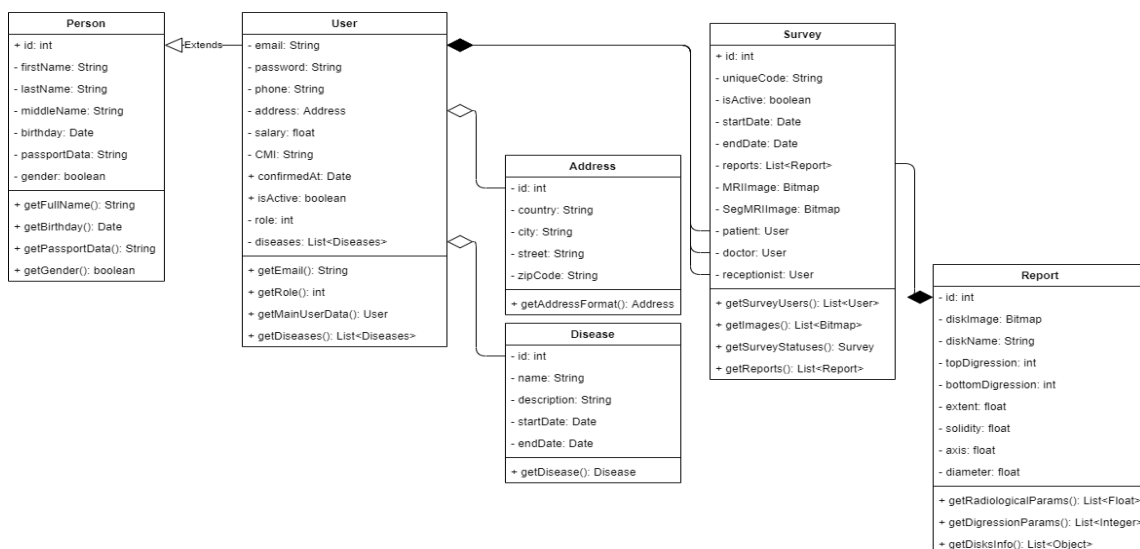


Рисунок 16 - Диаграмма классов проектируемой системы

Для более детального описания предметной области была составлена диаграмма последовательностей, представляющая собой описание жизненного цикла основных объектов системы [18]. На рисунке 17 представлена данная диаграмма.

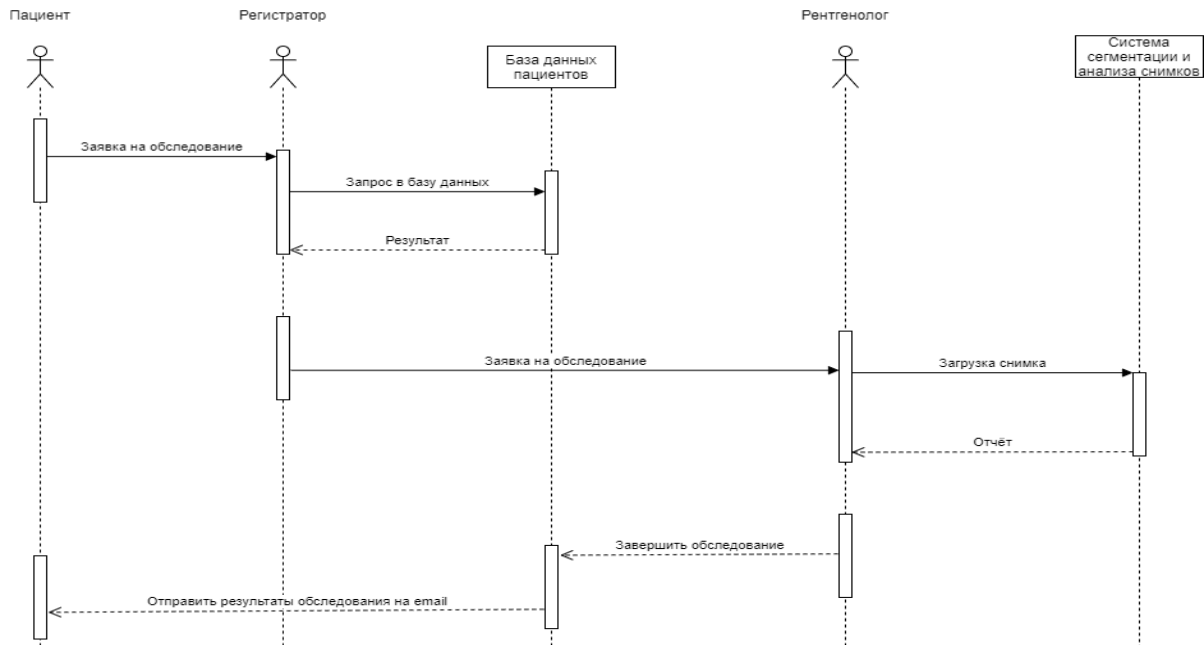


Рисунок 17 - Диаграмма последовательностей для разрабатываемой системы

Таким образом, на основе проделанной работы по описанию предметной области можно сделать следующие выводы:

- 1) Представленное в первой главе словесное описание предметной области позволяет достаточно подробно рассмотреть предметную область разработки информационной системы.
- 2) С помощью графической нотации методологии IDEF0 и IDEF3 информационную систему можно представить перед разработчиками и заказчиками в виде набора взаимосвязанных функций. Методология DFD позволяет описывать потоки данных, существующие в предметной области.
- 3) Стандарт UML позволяет не только дополнить описание предметной области, но и описать поведение системы.



Используя спроектированные схемы, становится необходимым, произвести проектирование схемы базы данных, разрабатываемой информационной системы.

## 2.7 Проектирование схемы базы данных

База данных представляет собой источник получения информации основных сущностей системы, а также является средством изменения необходимых

данных. Основная информация дифференцирована на таблицы, которые позволяют описывать конкретные бизнес-процессы.

Для описания концептуальной схемы предметной области используются ER-модели. Данная модель данных позволяет выделять основные сущности и связи между ними.

На рисунке 18 изображена ER-диаграмма проектируемой базы данных.

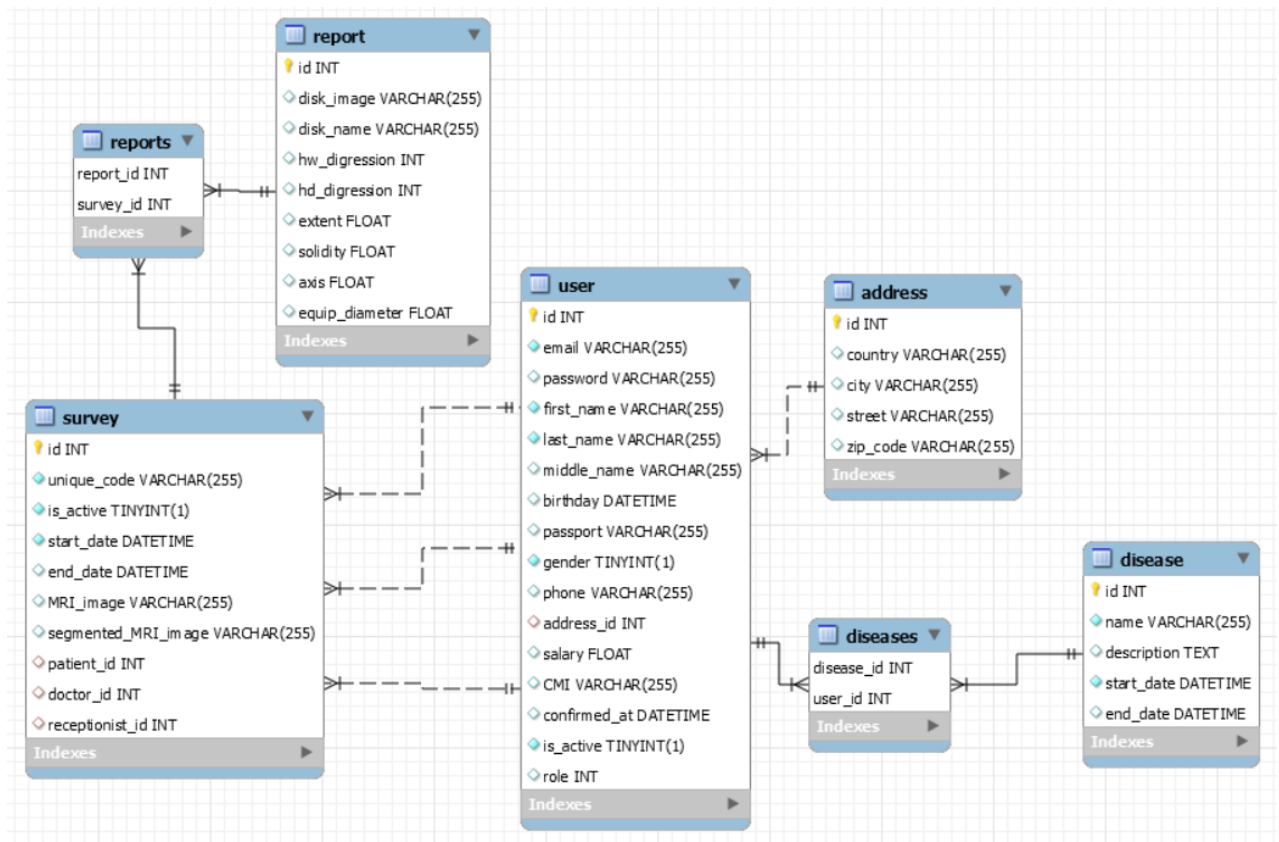


Рисунок 18 – ER-диаграмма проектируемой базы данных

Таблицы базы данных информационной системы и их описание представлены ниже.

- **Таблица user** – содержит в себе основную информацию о пользователях системы: рентгенологах, регистраторах, пациентах, администраторах. На основе данной таблицы пользователям будет предоставляться ролевой функционал к системе.
- **Таблица address** – хранит в себе ключевую информацию о местах проживания пользователей.
- **Таблица survey** – данная таблица содержит в себе информацию об обследованиях пациента. Модель содержит в себе пути к снимкам МРТ.
- **Таблица disease** – содержит истории болезней пациентов.
- **Таблица report** – данная таблица содержит отчеты о состояниях межпозвоночных дисков и их основных радиологических параметров.
- **Таблица reports** – представляет собой таблицу-ассоциацию между сущностями обследования (survey) и отчета (report).
- **Таблица diseases** – данная модель также является ассоциирующей таблицей между сущностями пользователя (user) и истории болезней (disease).

## 2.8 Выбор технологий разработки

Для реализации спроектированной базы данных была выбрана система управления базами данных MySQL версии 8.0.19. Данная СУБД является реляционной, то есть данные хранятся в абстрактных таблицах, благодаря этому достигается гибкость хранилища и его производительность.

Основными преимуществами данной СУБД являются:

- 1) Наличие пользовательской поддержки;

- 2) СУБД MySQL – имеет бесплатную community лицензию;
- 3) Данная система управления базами данных имеет широкое распространение, ее внедрение не занимает сложностей;
- 4) Программная система имеет средства репликации и архивирования информации;
- 5) Для данной СУБД существует огромное количество драйверов для разных языков программирования;
- 6) MySQL использует в своей основе сетевую архитектуру, возможно наличие огромного количества одновременных подключений;
- 7) Горизонтальная масштабируемость – мощности СУБД MySQL можно увеличивать установкой новых вычислительных машин, а не улучшением существующих;
- 8) MySQL обладает гибкой и удобной системой настроек, позволяющей определять размеры пакетов и запросов для наибольшей оптимизации.

Оптимизируемые запросы будут положительно влиять на экономическую составляющую системы, так как это позволит сэкономить серверные ресурсы.

Для визуального проектирования базы данных использовался поставляемый с СУБД MySQL программное обеспечение – Workbench 8.0 CE. Данный инструмент объединяет в себе такие процессы как проектирование, создание, моделирование и эксплуатацию над базами данных. Программа предоставляет набор инструментов для администрирования и настройки сервера и огромное количество других возможностей. MySQL Workbench является кроссплатформенной утилитой.

Программное обеспечение позволяет создавать комплексные ER-модели, проводить прямое и обратное проектирование, а также предоставляет возможность легко производить многосвязные изменения в базе данных, которые, как правило, требуют много усилий и времени.

Также программа MySQL Workbench позволяет в удобном текстовом редакторе создавать, выполнять и оптимизировать SQL-запросы. В программе присутствует панель переключения между разными базами данных, а также между разными подключениями к СУБД. Компонента отображения объектов обеспечивает мгновенный доступ к базам данных, их таблицам и содержимому таблиц.

Утилита MySQL Workbench позволяет упростить сопровождение и поддержку базы данных. ПО дает возможность архитекторам баз данных в удобной визуальной форме предоставлять требования и мгновенно разрешать проблемы до того, как будет потрачено большое количество времени и ресурсов. С помощью данного программного обеспечения становится возможным создавать надежные, грамотно структурированные БД. Также разрабатываемые базы данных весьма гибкие. И при наличии изменений бизнес-требования будут полностью удовлетворяться.

Для реализации процесса обучения нейросети используется язык программирования Python. В связи с наличием для данного языка программирования удобных и популярных фреймворков и библиотек машинного обучения. Такой библиотекой, предназначенной для глубоко машинного обучения, является Keras, который в свою очередь базируется на фреймворке TensorFlow [17]. Библиотека Keras содержит в себе множество алгоритмов для обучения нейросетевых моделей, а также позволяет очень удобно конфигурировать разрабатываемую модель. Благодаря абстракции, которую обеспечивает Keras, разработка

нейросетевой модели будет проходить безболезненно. Основными преимуществами данного фреймворка являются:

- 1) Модульность. Модули, необходимые для построения нейронной сети, заключены в простой интерфейс, что делает Keras простым в использовании;
- 2) Расширяемость. Легко создаваемые новые модули для Keras и их моментальная пригодность для предварительных исследований;
- 3) По сравнению с TensorFlow, Keras использует все преимущества фундаментальной библиотеки и стремится сформировать у разработчика положительный опыт;
- 4) Поскольку Keras - это библиотека Python, она доступна для широкого круга разработчиков в связи с присущей Python простоты и популярности как языка программирования.

Пользовательский интерфейс для нейросетевой модели и системы анализа будет представлять собой веб-приложение в связи с гетерогенностью платформ и преимуществом в виде единообразия веб-платформы.

Интерфейс должен обладать следующим минимальным функционалом: возможность выбрать изображение, отправить снимок на сегментацию, получить основную информацию о снимке, сформированного системой анализа.

Для разработки веб-интерфейса был выбран фреймворк Flask. Данный фреймворк является микро-фреймворком, который предоставляет экосистему компонентов, которые можно комбинировать и сопоставлять по мере необходимости, сосредоточившись на подключении лишь нужным компонентов.

Данный фреймворк обладает следующими преимуществами:

- 1) Масштабируемость. Фреймворк Flask более масштабируем, чем монолитные фреймворки. Сегодня приложения часто работают в контейнерах или используют облачные вычисления с автоматическим масштабированием. Приложения обычно не масштабируются сами по себе, масштабируется инфраструктура. С помощью небольшого приложения проще развертывать контейнеры на тысячах серверов, чтобы легко обрабатывать увеличенный трафик;
- 2) Простота изучения. При достаточном уровне знания языка программирования Python фреймворк Flask легко изучается;
- 3) Гибкость. Лишь небольшое число компонентов фреймворка Flask интегрируются с трудом;
- 4) Малый уровень абстракции. Существует меньше уровней абстракции между вами и базой данных, запросами, кэшем и т. д. по сравнению с монолитным фреймворком аналогом Django. Таким образом, производительность на порядок выше;
- 5) Модульность. Модульный код дает огромное количество преимуществ. С помощью Flask становится возможным создавать несколько приложений, распределенных по большой сети серверов, каждый из которых имеет определенные цели. Это повышает эффективность, приводит к лучшей тестируемости и производительности.

## **2.9 Результаты проектирования информационной системы**

Были сформированы основные требования к разрабатываемой системе. Все бизнес-процессы системы представлены в виде диаграмм семейств IDEF и UML. Для четкого определения этапов и сроков разработки был составлен план учета планирования. Также определен стек технологий, на которых будет реализовываться система.

## **3 Результаты разработки**

В данной главе будет представлен процесс формирования набора данных для обучения нейросети, результат обучения сверточной нейросети архитектуры U-Net, работа системы анализа снимков МРТ и веб-приложения.

### **3.1 Формирование набора данных**

Будут рассматриваться три набора данных: Genodisc, TwinsUk, OSCLMRIC и Lumbar MRI Dataset. Все выборки данных, содержат МРТ поясничного отдела позвоночника. Датасеты формировались на основе снимков из разных поликлиник и, таким образом, содержат различные виды снимков МРТ.

#### **3.1.1 Набор данных Genodisc**

Набор данных Genodisc содержит данные 2635 пациентов с диагнозом боли в спине. Из 2635 только 2079 были обследованы рентгенологом, и из этих 2079 только 2009 имеют необходимые срезы. Каждый срез содержит до шести поясничных дисков. Всего существует 12018 дисков, каждый из которых имеет несколько радиологических градаций. Этот набор данных был собран одноименным проектом в 2009 году. Проект Genodisc формировал датасет из снимков пациентов, которые повторно обращаются за помощью по поводу боли в спине. Снимки были получены из поликлиник Англии, Венгрии, и Италии. Изображения поступали с разных машин МРТ. Из-за этого различия в сканировании могут быть весьма заметны, например, толщина среза сагиттального сканирования может существенно отличаться. Все испытуемые сканировались в лежачем положении. 54,6% испытуемых - женщины.

Подробности об испытуемых и их сканировании можно увидеть в таблице 3.

Таблица 3 – Сводка по исследованиям в наборе данных Genodisc

Параметр	Среднее значение	Медианное значение	Ранжирование значений
Возраст пациента	50.1	50.0	14.0-87.0
Вес пациента (кг)	77.8	80.0	43-130
Толщина сагиттального среза (мм)	4.1	4.0	1.2-6.0
Сагиттальное расстояние между пикселями (мм)	0.7	0.6	0.3-1.6
Количество сагиттальных срезов	12.3	11.0	8.0-25.0
Толщина осевого среза (мм)	4.5	4.0	1.3-10.0
Осевое расстояние между пикселями (мм)	0.5	0.5	0.2-1.5
Количество осевых срезов	12.5	12.0	9-22

Каждая серия снимков содержит два типа сканирований: осевые и сагиттальные. Осевое сканирование обычно содержит информацию только о нижних трех поясничных дисках, L3-L4, L4-L5 и L5-S1. Срезы осевого сканирования могут быть получены в двух различных ориентациях: в блоках, где все осевые срезы параллельны друг другу, или где срезы, относящиеся к каждому диску, выровнены так, чтобы быть параллельными ориентации диска. Сагиттальные сканирования будут использоваться при обучении нейросети, поскольку большинство радиологических параметров можно определить именно с помощью сагиттальных сканирований.

### 3.1.2 Набор данных TwinsUK

Набор данных TwinsUK был сформирован отделом исследований близнецов Королевского колледжа Лондона. Данный датасет состоит из



1016 снимков. В отличие от набора данных Genodisc, который содержал данные лишь о пациентах с болью в спине, набор данных TwinsUK также включает в себя снимки МРТ здоровых пациентов. Каждый снимок содержит до пяти межпозвоночных дисков от L1-L2 до L5-S1. Всего имеется 6327 дисков с градациями.

Помимо базового сканирования, 423 серии для конкретных пациентов имеют последующие сканирования, сделанные через 8-12 лет после первоначального сканирования. Исходные снимки были сделаны с помощью аппарата МРТ 1.0-Tesla, а последующие - с помощью аппарата 1.5-Tesla, но и старые, и новые снимки содержат одинаковый протокол сканирования. Хотя исходная информация и последующие данные были получены с двух разных машин, но из-за такого применения схожего протокола сканирования различия в сканировании довольно малы. Как и в случае с исследованием проекта Genodisc, испытуемые сканировались в положении лежа на спине. 96,7% испытуемых - женщины.

Сводку по данному набору данных можно посмотреть в таблице 4 (в данном датасете представлены только саггитальные срезы позвоночника).

Таблица 4 – Сводка по исследованиям в наборе данных TwinsUK

Параметр	Среднее значение	Медианное значение	Ранжирование значений
Возраст пациента	56.1	56.0	19.0-79.0
Вес пациента (кг)	67.6	65.0	44-128
Толщина саггитального среза (мм)	4.0	4.0	4-5
Саггитальное расстояние между пикселями (мм)	0.9	0.9	0.8-1.2
Количество саггитальных срезов	11.0	11.0	7.0-16.0

Снимки МРТ пациентов, данные которых были добавлены в датасет TwinsUK, были исследованы рентгенологами. Радиологические оценки были аннотированы рентгенологом и заполнялись для каждого диска. 920 из 1016 испытуемых имеют дегенерацию межпозвоночных дисков разной тяжести.

### 3.1.3 Набор данных OSCLMRIC

Датасет OSCLMRIC содержит снимки МРТ поясничного отдела позвоночника от 1689 пациентов. В отличие от Genodisc, где субъекты обычно имеют только одну серию сканирований, субъекты в OSCLMRIC могут содержать более одной серии, подобно TwinsUK. Количество серий сканирований в OSCLMRIC не ограничивается только двумя или тремя. 530 испытуемых имеют более одного сканирования; 366 - 2 сканирования, 104 - 3 сканирования, 35 - 4 сканирования, 11 - 5 сканирований, 8 - 6 сканирований и 6 испытуемых с более чем 6 сканированиями. Радиологическая градуировка сканирований представлена отчетами рентгенологов. Параметры определены для 83 снимков. 57,8% испытуемых - женщины. Как и в случае с датасетом Genodisc, стандартизация протоколов сбора данных отсутствует. В таблице 5 представлено описание основных параметров набора данных OSCLMRIC.

Таблица 5 – Сводка по исследованиям в наборе данных OSCLMRIC

Параметр	Среднее значение	Медианное значение	Ранжирование значений
Возраст пациента	51.8	51.0	16.0-92.0
Вес пациента (кг)	81.7	80.0	31.8-163.0
Толщина сагиттального среза (мм)	3.9	4.0	1.0-6.0
Сагитальное расстояние между пикселями (мм)	0.7	0.7	0.4-1.3
Количество сагитальных срезов	14.7	15.0	11.0-24.0

### Продолжение таблицы 5

Толщина осевого среза (мм)	4.1	4.0	3.0-7.0
Осевое расстояние между пикселями (мм)	0.6	0.6	0.3-1.1
Количество осевых срезов	19.3	17.0	3.0-80.0

### 3.1.4 Набор данных Lumbar MRI Dataset

Датасет Lumbar Spine MRI Dataset, находится в открытом доступе на портале Mendeleu. Набор данных содержит анонимизированные клинические МРТ-исследования 515 пациентов с симптоматическими болями в спине. С данными каждого пациента может быть связано одно или несколько МРТ-исследований. Каждое исследование содержит последовательности срезов.

Осевые срезы позвоночника в данном наборе представлены последними тремя сериями. Все остальные серии содержат саггитальные срезы позвоночника. В большинстве случаев общее количество снимков в одной серии колеблется от 12 до 15. Однако есть серии в которых содержится 20 снимков, поскольку исследование распространялось не только на поясничный отдел позвоночника. Снимки в сагиттальной проекции в большинстве случаев содержат 5 поясничных позвонков и 2 крестцовых.

В наборе данных имеется в общей сложности 48345 снимков МРТ. Большинство срезов имеют разрешение изображения (320 x 320) пикселей, однако есть снимки из трех исследований с разрешением (320 x 310) пикселей. Пиксели на всех снимках имеют 12 битную точность на пиксель, что выше, чем в стандартных 8 битных изображениях в оттенках серого.

Основная информация о сериях снимков МРТ в данном датасета представлена в таблице 6.

Таблица 6 – Сводка по исследованиям в наборе данных Lumbar MRI Dataset

Параметр	Среднее значение	Медианное значение	Ранжирование значений
Возраст пациента	49.8	50.0	14.0-78.0
Вес пациента (кг)	63.7	64.0	50.8-122.3
Толщина сагиттального среза (мм)	4.0	4.0	1.1-5.2
Сагитальное расстояние между пикселями (мм)	0.7	0.7	0.4-1.3
Количество сагиттальных срезов	12.6	12.8	11.3-23.4
Толщина осевого среза (мм)	3.9	4.0	3.2-7.2
Осевое расстояние между пикселями (мм)	0.49	0.52	0.3-1.0
Количество осевых срезов	2.2	2.0	1.0-3.0

### 3.1.5 Результаты обзора наборов данных

Были проанализированы четыре набора данных, каждый из которых включает в себя поясничные МРТ-сканирования, содержащие межпозвоночные диски.

Самым важным показателем при выборе датасета является количество серий в наборе данных. Количество доступных сканирований непосредственно влияет на точность классификатора. Однако так как для обучения используется сверточная нейросетевая модель архитектуры U-Net небольшое количество снимков (от 50) достаточно для обучения классификатора. Если задачей ставится разработка полностью переносимого классификатора, размер набора данных должен быть значительно больше, соизмеримым с размером датасета Genodisc. Чем

больше гетерогенных серий снимков в датасета, тем выше точность классификатора.

Удовлетворяющий поставляемым выше требованиям датасет, представлен набором данных Lumbar MRI Dataset. Основным преимуществом данного набора стало бесплатное распространение на портале Mendeley. Конвертация снимков из формата DICOM в PNG будет производиться посредством специализированного модуля, описываемого в 1 главе программного обеспечения RadiAnt DICOM Viewer.

### **3.2 Система локализации межпозвоночных дисков**

В данном разделе описывается, основанная на CNN, структура для классификации межпозвоночных дисков в сагиттальной проекции снимков МРТ поясничного отдела позвоночника. Нейросеть архитектуры U-Net обучается, проверяется и тестируется на наборе данных Lumbar MRI Dataset. Разделение на обучающееся, валидационное и тестовое подмножества производится по принципу 50:10:10 снимков соответственно. Обученная модель предсказывает, для каждого из шести поясничных межпозвоночных дисков, шесть отдельных параметров, являющимися различными радиологическими оценками. Вместо обучения моделей для определения каждого параметра используется единая модель, которая вычисляет все параметры одновременно.

Упрощенная схема сегментации и определения радиологических параметров межпозвоночных дисков представлена на рисунке 19.



Рисунок 19 – Схема сегментации и калькуляции радиологических параметров межпозвоночных дисков

### 3.2.1 Структура работы нейросети

Стандартный снимок МРТ поясничного отдела позвоночника обычно содержит несколько межпозвоночных дисков. Количество дисков варьируется в зависимости от патологии пациента и стандартов протокола клинического центра, но обычно снимки содержат шесть поясничных дисков от диска T12-L1 до диска L5-S1. Цель исследования состоит в том, чтобы предсказать границы межпозвоночных дисков и связанных с ними позвонков. В качестве входных данных для CNN служит область вокруг каждого диска, которая включает сам диск и часть позвонков выше и ниже конкретного диска. Нейросетевая модель предсказывает все параметры дисков одновременно. При подаче снимка всего позвоночника в качестве входных данных было обнаружено, что наблюдается значительное снижение точности классификации. Нейросеть обучается одновременно для прогнозирования всех радиологических параметров с помощью многозадачной функции потерь.

Несмотря на преимущества и высокую производительность сверточных нейронных сетей, одной из проблем использования CNN в медицинской визуализации остается потребность в больших учебных наборах данных. Эта проблема решается частично за счет агрессивного увеличения набора данных, а также благодаря использованию

многозадачности. Когда одна архитектура нейросетевой модели служит для решения нескольких проблем и, следовательно, поддерживает несколько управляющих сигналов для общего магистрального канала, т. е. сверточных слоев. Многозадачность может улучшить точность классификации. Объекты и пространственные отношения, изученные слоями сверточной нейросети, обобщаются и распространяются на другие связанные подзадачи. Поскольку каждая классификационная задача является уникальной, то ее решение одновременно сродни многозадачности. Так как рентгенологи обязаны оценивать состояние всех анатомических областей в пределах изображения, разработка методов, которые могут предсказать несколько характеристик анатомических тел представляет высокий интерес. В разрабатываемой системе каждый межпозвоночный диск имеет радиологические оценки с описанием своего состояния дегенерации.

### **3.2.2 Извлечение информации о межпозвоночных дисках**

Радиологические параметры межпозвоночных дисков определяются следующим образом: область с изображением диска поворачивается в пределах сагиттального среза так, чтобы межпозвоночный диск находился в горизонтальном положении и был отцентрирован. У выделяемых областей меняется размер, сохраняя при этом соотношение сторон 1:2. Это соотношение должно гарантировать, что область диска не будет включать верхние и нижние концевые пластины тел позвонков. На рисунке 20 показан пример обработки конкретного диска.

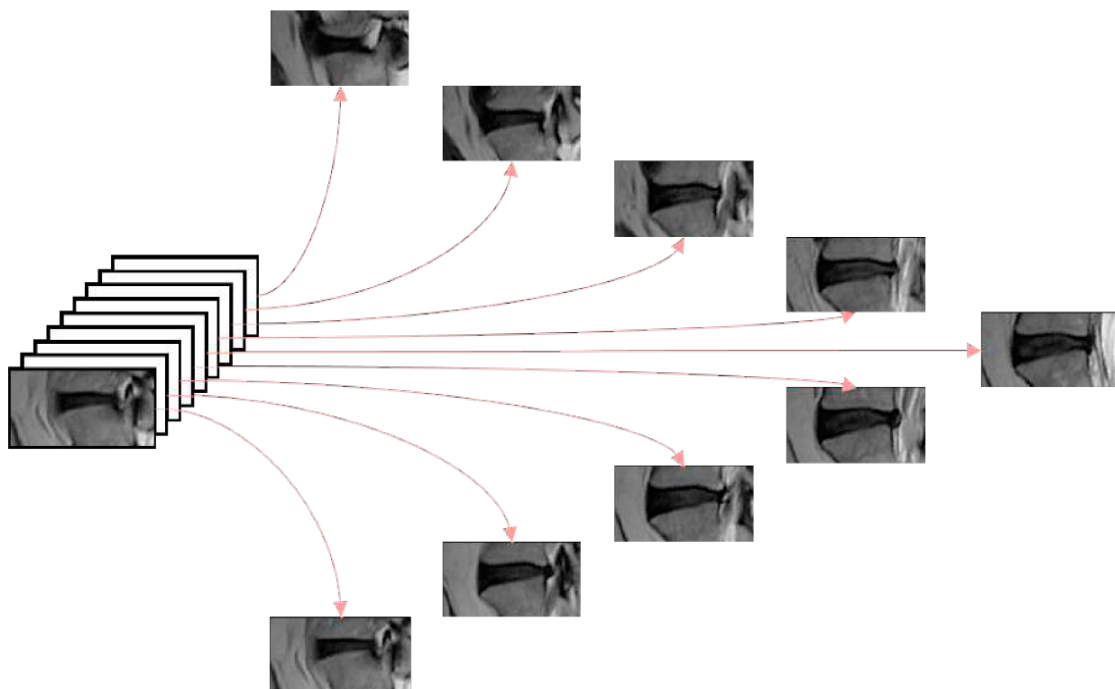


Рисунок 20 – Схема деформации дисков

### 3.2.3 Функция потерь

Функция потерь при обучении нейросети должна быть связана как с несколькими классификациями различных параметров, так и с балансом вклада каждого класса при определении радиологических характеристик.

Поскольку каждая из шести радиологических градаций является классификационной задачей, исследуется общепринятой практикой обучения сети путем минимизации логарифмических потерь softmax (кросс-энтропия). Каждая характеристика рассматривается как отдельная задача. Для каждой задачи сеть выводит вектор, который соответствует количеству параметров описанных ранее.

Поскольку данная задача является несбалансированной, что является распространенной проблемой в задачах медицинской классификации, используется сбалансированная по классам функция потерь во время обучения. Для каждой подзадачи функция потерь рассчитывается таким



образом, чтобы совокупные потери были минимальны и в итоге сбалансированы.

### **3.2.4 Реализация нейросети**

Разработанная нейросетевая архитектура представляет собой модифицированную версию нейросети VGG-M. Входной размер изображения межпозвоночного диска составляет (40 x 20). Слои сверточной сети изменяются, чтобы принимать стандартные 3 канала для изображений RGB. Не используется локальная нормализация на первых 2 слоях, чтобы гарантировать, что никакая информация не будет потеряна после слоя объединения на 5 слое. Основная конфигурация, которая используется для многозадачного определения параметров, - это точка ветвления на 5 слое нейросети. Слои, следующие за точкой ветвления, идентичны по количеству связей для каждой подзадачи.

Необходимо сформировать маски снимков. Так как в наборе данных отсутствуют сегментированные изображения, формирование масок производится вручную.

На рисунке 21 представлен исходный снимок МРТ и сформированная вручную маска сегментации для данного изображения.



Рисунок 21 – МРТ снимок позвоночника и маска сегментации

Так как для обучения нейросетевой модели архитектуры U-Net необходимо небольшое количество снимков (около 50) ограничимся формированием набора данных из 71 снимка. Для создания высокоточной модели каждый объект сегментированной маски был вынесен на отдельное изображение. Сформировав снимки и маски локализации в единый набор данных можно приступить к обучению нейросети.

Обучение нейросети выполняется с помощью стохастического градиентного спуска с нулевым импульсом без какой-либо предварительной заготовки весов. Входные данные нормируются с помощью определения среднего значения в цветовом канале.

При обучении сверточной нейросетевой модели первым делом следует верно реализовать выбранную архитектуру. Благодаря используемой библиотеке Keras сделать это очень просто. Было добавлено 5 сжимающих и 5 расширяющих скрытых слоев. Каждый из слоев выполняет изменение размера свертки на степень двойки. Обучение происходит минибатчами. Это означает, что изображение разбивается на сегментированные объекты, а общие для всего снимка веса

корректируются в конце итерации. Всего таких итераций обучения 200 (см. приложение Б).

Обучение модели происходило на рабочей станции с процессором Intel Core i7-6700HQ, 16 GB RAM, видеоадаптером NVIDIA GeForce GTX 960M. Сам процесс занял 2 часа. После этого модель была сохранена в формате h5 для дальнейшего использования. На обучающем множестве была достигнута точность 86%. На тестовом множестве точность составила 75%.

На рисунке 22 можно наблюдать сегментированное изображение позвоночника из тестовой выборки.



Рисунок 22 – МРТ снимок позвоночника и предсказанное сегментированное изображение

### **3.2.5 Оценка точности работы нейросетевой модели**

Для полноценного проведения рентгенологом анализа снимка МРТ задачей ставится визуальная оценка точности границ межпозвонкового диска, определенных с использованием нейросетевых методов. Для

решения данной задачи достаточно совместить исходный снимок МРТ с сегментированным изображением. Неточности на изображении, полученном при определении нейросетевой моделью, можно будет визуально заметить на совмещенном снимке.

Также задачей ставится разработка количественной оценки точности используемого нейросетевого метода при определении границ межпозвоночного диска. Для реализации данной задачи снимки из обучающей выборки, для которых были составлены маски изображений (описывались ранее) необходимо сравнить с предсказанным нейросетевой моделью изображением. Количество белых пикселей на изображении будет показывать количественную точность предсказанного изображения. На рисунке 23 представлена гистограмма соотношения черных пикселей к белым для конкретного снимка МРТ поясничного отдела позвоночника.

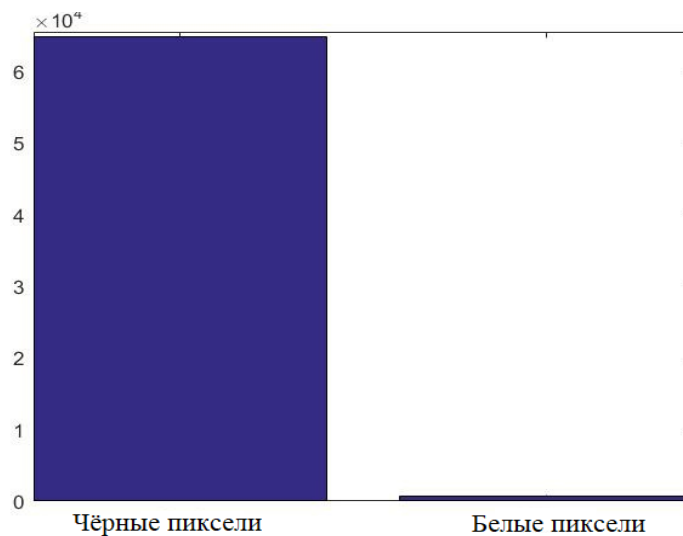


Рисунок 23 – Гистограмма количественной ошибки сегментации нейросетью

Отношение белых пикселей к общему числу пикселей на изображении и будет являться количественной мерой ошибки сегментации.

В связи с наличием ошибок сегментации необходимо разработать алгоритма определения реальных границ межпозвоночного диска, основываясь на результатах обнаружения границ с помощью нейросети. Однако данная задача является трудно решаемой в связи с гетерогенностью снимков МРТ. С помощью морфологических операций и постпроцессинга невозможно с достаточной точностью корректировать границы сегментированных межпозвоночных дисков. Возможным решением может стать использование существующих нейросетевых моделей для сегментации тел позвонков. На основе результатов работы, связанных с локализацией позвонков, станет возможным корректировка границ межпозвоночных дисков.

### **3.2.6 Разработка системы анализа межпозвоночных дисков**

Полученное в результате работы нейросети сегментированное изображение, можно анализировать с помощью морфологических операций и структур данных, высчитываемых нейросетью. Основные анализируемые параметры, которые будут изучаться для каждого отдельного объекта, на результирующем изображении, представлены ниже:

- Экстент - отношение площади объекта (межпозвоночного диска) к площади описываемого прямоугольника;
- Плотность - отношение площади объекта к площади его выпуклой оболочки;
- Осевой коэффициент - отношение максимальной и минимальной осей сегментированного объекта;
- Эквивалентный диаметр - это диаметр окружности, площадь которой совпадает с площадью объекта.

Данные параметры могут использоваться рентгенологом для определения состояния межпозвоночного диска.

Также необходимо анализировать межпозвоночные диски на наличие грыж Шморля. Определение данной патологии описывалось в 1 главе. Для определения грыжи Шморля для отдельного межпозвоночного диска используется следующий алгоритм:

- 1) Исходное изображение диска автоматически поворачивается на нужный угол, чтобы формируемая функция была одномерной;
- 2) Попиксельно создается массив высот для каждой точки диска от середины до верхней и нижней границ соответственно;
- 3) Определяется точка максимума для границ.

Было проанализировано, что отклонение более чем на 10 условных единиц высот в 90% случаев подразумевает под собой наличие грыжи Шморля для данного межпозвоночного диска.

Для удобного представления данных о сегментированных изображениях, необходимо разработать пользовательский интерфейс, позволяющий получать основную информацию, предоставляемую системой анализа (см. приложение В).

### **3.3 Разработка веб-приложения**

Пользовательский интерфейс для разработанной нейросетевой модели и системы анализа состояния межпозвоночных дисков будет представлять собой веб-приложение. Преимущества создания интерфейса в среде веб описаны в главе 2.

Интерфейс должен обладать следующим минимальным функционалом: возможность выбрать изображение, отправить снимок на

сегментацию, получить основную информацию о снимке, сформированного системой анализа.

Как описывалось ранее, для разработки веб-интерфейса был выбран фреймворк Flask. Данная технология позволит быстро и качественно развернуть приложение на рабочий сервер, написав немного кода, построить точную маршрутизацию запросов, а также с помощью шаблонизатора Jinja 2, сделать интерфейс визуально приятным и понятным пользователю.

На рисунке 24 представлен интерфейс загрузки снимка МРТ позвоночника для проведения дальнейшего исследования.

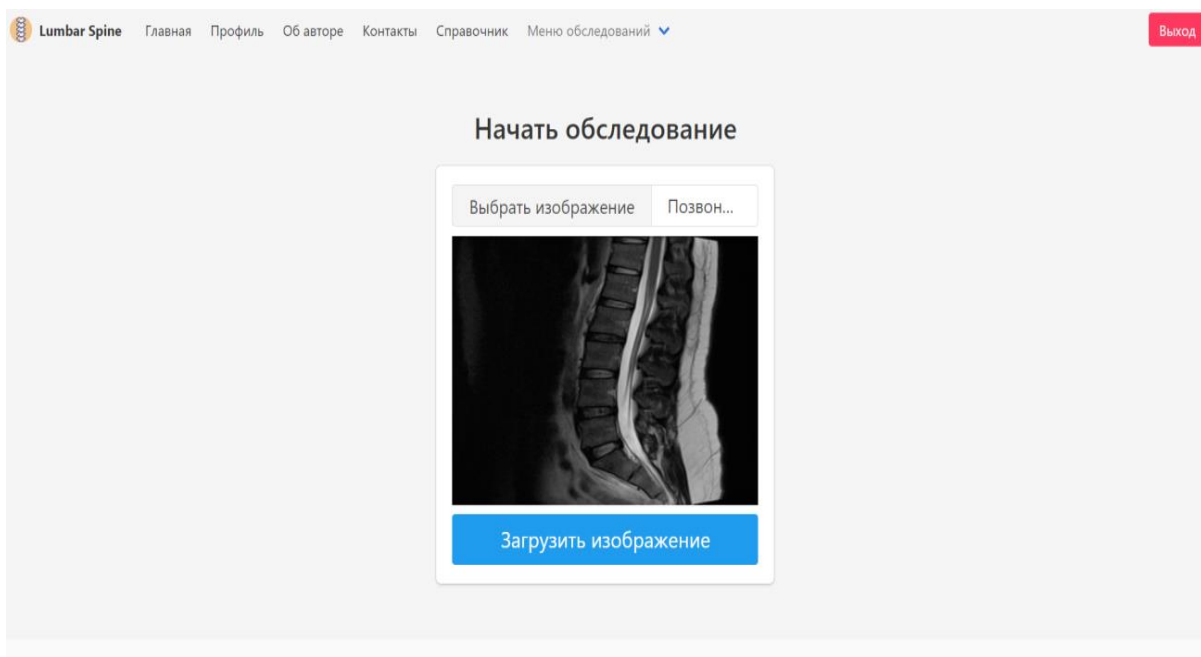


Рисунок 24 – Экранная форма загрузки снимка МРТ для дальнейшей обработки

После нажатия кнопки «Upload MRI image» в роутинге запускается процесс конвертации изображения к нужному формату, сохранению его в рабочее пространство. Далее запускается процесс работы нейросети, которая в итоге сохраняет предсказанное изображение в необходимую директорию. Далее с использованием библиотеки OpenCV выполняет свою

работу система анализа. В итоге, пользователь оказывается на странице с сформированным результатом, представленной на рисунке 25.

Изображение диска	Наименование диска	Верхняя регрессия	Нижняя регрессия	Экстенг	Плотность	Осевой коэффициент	Экв. диаметр
	S4-S5	10	11	0.59026	481.5	0.62857	24.0559
	S3-S4	6	9	0.77679	452.0	0.45714	23.5342
	S2-S3	7	7	0.79314	424.0	0.44118	22.6942
	S1-S2	7	7	0.71619	397.5	0.42857	21.8801
	L4-L5	7	9	0.61029	365.5	0.53125	20.56
	L3-L4	7	7	0.60108	308.5	0.48387	18.8645

Рисунок 25 – Экранная форма с результатами обработки снимка МРТ

Данный минимальный пользовательский интерфейс является частью проектируемой информационной системы. На рисунке 26 изображена страница со списком пациентов и доступными над ними операциями. Данная страница полностью реализует функционал регистратора в поликлинике.

Email	Имя	Фамилия	Отчество	Дата рождения	Паспортные данные	Пол	Телефон	ОМС	Страна	Город	Улица	Почтовый индекс	Редактировать
james@mail.ru	Александр	Петров	-	-	-	Мужчина	-	-	-	-	-	-	Редактировать
morgan@gmail.com	Николай	Сидоров	-	-	-	Мужчина	-	-	-	-	-	-	Редактировать

Рисунок 26 – Экранная форма со списком пациентов



Для каждого пациента доступно ведение истории болезней. На рисунке 27 изображена экранная форма с данным функционалом.

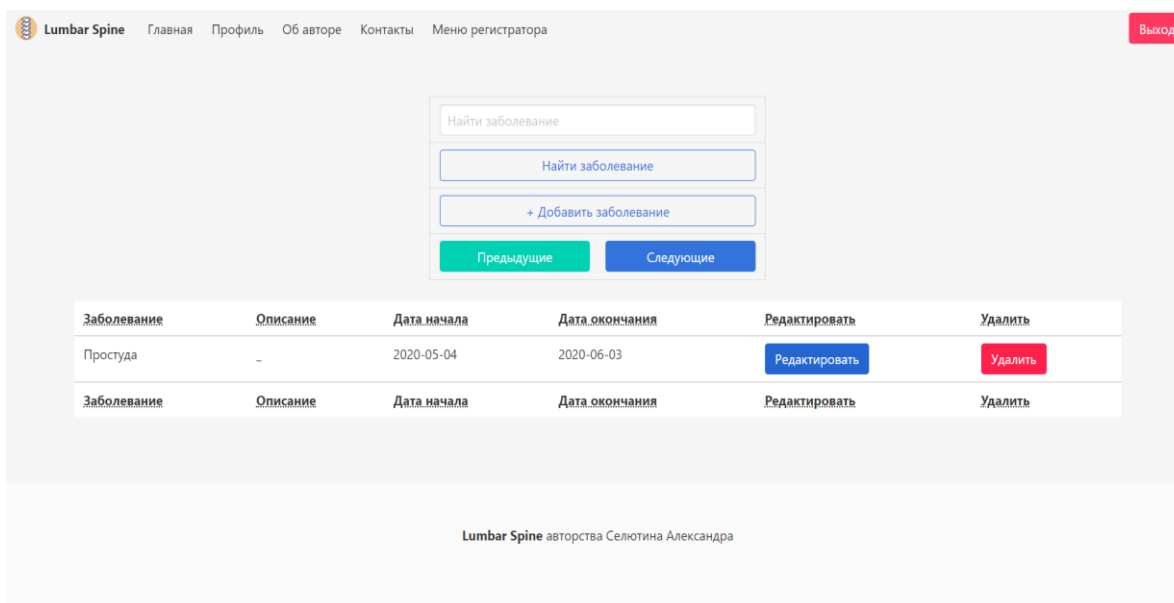


Рисунок 27 – Экранная форма с историей болезней пациента

Также для пользователя с функциональной ролью администратора была разработана система, позволяющая вести полный учет основных сущностей системы.

На рисунке 28 представлена экранная форма панели администратора, отвечающая за редактирование пользователей системы.

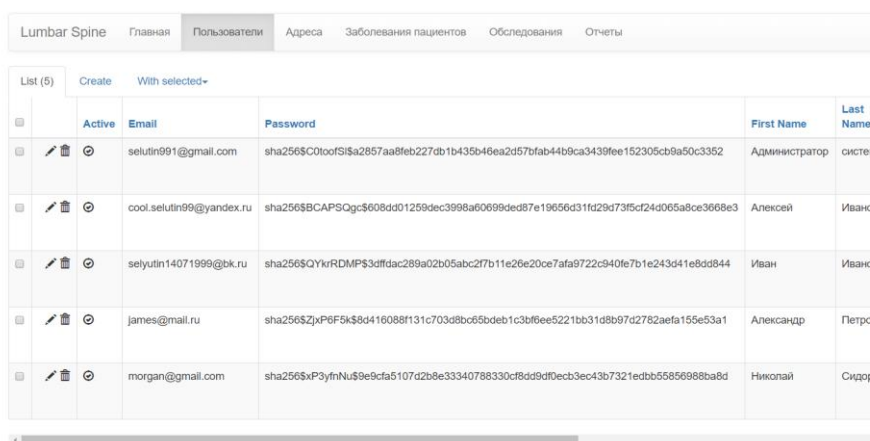


Рисунок 28 – Экранная форма панели администратора

Другие экранные формы представлены в приложении А.

Основные функции, описываемые во 2 главе, были успешно реализованы. Веб-приложение имеет инсталляционные скрипты, которые позволяют быстро разворачивать систему на производственных серверах. Подразумевается, что созданная система будет разворачиваться во внутренней сети поликлиники, и будет иметь лишь внешний доступ к популярным почтовым серверам (однако, настройки доступа будут определяться администратором сети). Открытый доступ к почтовым серверам необходим для обеспечения процесса регистрации пользователей системы.

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы было создано веб-приложение для сегментации и анализа изображений МРТ позвоночника поясничного отдела. Был реализован интерфейс загрузки изображения МРТ, процесс сегментации снимка, анализ основных параметров межпозвоночных дисков, а также процесс построения разностных функций для определения грыж Шморля. Также была реализована функциональность панели администратора, отвечающая за управление основными сущностями системы. Были выполнены все конфигурации для настройки программного обеспечения и размещения его в системе контроля версий. Данное приложение будет использоваться для автоматизации работы врача рентгенолога. Ранее снимки рассматривались врачами вручную. Разработанное программное обеспечение позволит ускорить процесс диагностики, повысить точность постановления диагноза. Для получения наиболее полных знаний в области разработки медицинского программного обеспечения в области распознавания изображений был прослушан теоретический курс лекций по компьютерной обработке изображений, а также курс по проектированию сверточных нейронных сетей.

Поставленные цели в главе 1, а также основные требования к проектируемой системе, составленные во 2 главе, были выполнены и реализованы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Абрамова, Т.А., Нейроиммунологическая характеристика больных ревматоидным артритом / Татьяна Абрамова und Сергей Абрамов. - М.: Palmarium Academic Publishing, 2012. - 220 с.
2. Антонов, И. П. Поясничные боли / И.П. Антонов, Г.Г. Шанько. - М.: Беларусь, 2016. - 144 с.
3. Арлоу, Джим UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Джим Арлоу , Айла Нейштадт. - М.: Символ-плюс, 2019. - 624 с.
4. Басиева, О. О. Ревматоидный артрит / О.О. Басиева. - М.: Феникс, 2017. - 192 с.
5. Бердяев, А. Ф. Болезни и повреждения суставов и костей / А.Ф. Бердяев. - М.: Государственное издательство медицинской литературы, 2015. - 244 с.
6. Васильев, А. Н. Принципы и техника нейросетевого моделирования / А.Н. Васильев, Д.А. Тархов. - Москва: РГГУ, 2018. - 218 с.
7. Гелиг, А. Х. Введение в математическую теорию обучаемых распознающих систем и нейронных сетей. Учебное пособие / А.Х. Гелиг, А.С. Матвеев. - М.: Издательство СПбГУ, 2018. - 224 с.
8. Давид, М. Рэйд Справочник по остеопорозу / Давид М. Рэйд. - М.: Практическая медицина, 2015. - 128 с.
9. Заболотных, И. И. Болезни суставов / И.И. Заболотных. - М.: СпецЛит, 2016. - 256 с.

10. Иванов Д.И., Частные феномены системного воспаления при спондилоартритах / Иванов Дмитрий, Евгений Гусев, Людмила Соколова. - М.: LAP Lambert Academic Publishing, 2015. - 192 с.
11. Йордон, Эдвард Объектно-ориентированный анализ и проектирование систем / Эдвард Йордон , Карл Аргила. - М.: ЛОРИ, 2020. - 264 с.
12. Кватрани, Терри Визуальное моделирование с помощью Rational Rose 2002 и UML / Терри Кватрани. - М.: Вильямс, 2015. - 192 с.
13. Киммел, Пол UML. Основы визуального анализа и проектирования / Пол Киммел. - М.: НТ Пресс, 2018. - 272 с.
14. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. - М.: Горячая линия - Телеком, 2019. - 382 с.
15. Латыпова, Рамиля Нейронные сети / Рамиля Латыпова. - М.: LAP Lambert Academic Publishing, 2015. - 217 с.
16. Лечение и профилактика болезней суставов / В.И. Мазуров и др. - М.: Диалект, 2015. - 322 с.
17. Любанович, Билл Простой Python. Современный стиль программирования / Билл Любанович. - М.: Питер, 2016. - 480 с.
18. Мюллер, Роберт Дж. Проектирование баз данных и UML / Мюллер Роберт Дж.. - М.: ЛОРИ, 2017. - 749 с.
19. Носков, Сергей Михайлович Консервативное лечение остеоартроза / Носков Сергей Михайлович. - М.: ГЭОТАР-Медиа, 2016 . - 484 с.

20. Окорочков, А. Н. Руководство по лечению внутренних болезней. Том 4. Лечение ревматических болезней / А.Н. Окорочков. - М.: Медицинская литература, 2016. - 494 с.

21. Олескин, А. В. Сетевые структуры в биосистемах и человеческом обществе / А.В. Олескин. - М.: Едиториал УРСС, Либроком, 2018. - 304 с.

22. Рамешвили, Т.Е. Дегенеративно-дистрофические поражения позвоночника (лучевая диагностика, осложнения после дисэктомии) / Т.Е. Рамешвили и др. - М.: ЭЛБИ-СПб, 2016. - 224 с.

23. Раскина, Т.А. Проблема остеопороза у мужчин с ревматоидным артритом / Татьяна Алексеевна Раскина, Ирина Степановна Дыдыкина, Марина Васильевна Летаева. - М.: LAP Lambert Academic Publishing, 2015. - 112 с.

24. Редько, В.Г. Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кибернетики / В.Г. Редько. - Москва: ИЛ, 2017. - 224 с.

25. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. - М.: Горячая линия - Телеком, 2016. - 384 с.

26. Рышард, Тадеусевич Элементарное введение в технологию нейронных сетей с примерами программ / Тадеусевич Рышард. - М.: Горячая линия - Телеком, 2015. - 541 с.

27. Сигидин, Я. А. Биологическая терапия в ревматологии / Я.А. Сигидин, Г.В. Лукина. - М.: Практическая медицина, 2015. - 304 с.

28. Таганов, Александр Иванович Нейросетевые системы искусственного интеллекта в задачах обработки изображений / Таганов Александр Иванович. - М.: Горячая линия - Телеком, 2016. - 533 с.

29. Тархов, Д.А. Нейросетевые модели и алгоритмы. Справочник / Д.А. Тархов. - М.: Радиотехника, 2014. - 726 с.

30. Хайкин, Саймон Нейронные сети. Полный курс / Саймон Хайкин. - М.: Вильямс, 2016. - 162 с.

31. Шайдуров, Александр Нейросетевой анализ медицинских данных / Александр Шайдуров. - М.: LAP Lambert Academic Publishing, 2016. - 140 с.

32. Шилин, Кирилл Юрьевич Макропроектирование компьютерных распределенных систем / Шилин Кирилл Юрьевич. - М.: Дело, 2016. - 517 с.

## ПРИЛОЖЕНИЕ А

### Экранные формы информационной системы

The screenshot shows the login page of the Lumbar Spine system. At the top left, there is a logo and the text "Lumbar Spine" followed by navigation links: "Главная", "Об авторе", and "Контакты". At the top right, there are links for "Войти" and "Регистрация". The main heading is "Вход". Below it, there is a form with two input fields: "Ваша почта" and "Ваш пароль". Under the password field, there are two links: "Сбросить пароль" and "Переотправить ссылку". Below these links is a checkbox labeled "Запомнить меня". At the bottom of the form is a blue button labeled "Войти". At the bottom of the page, there is a footer: "Lumbar Spine авторства Селютина Александра".

Рисунок А.1 – Форма входа в систему

The screenshot shows the registration page of the Lumbar Spine system. At the top left, there is a logo and the text "Lumbar Spine" followed by navigation links: "Главная", "Об авторе", and "Контакты". At the top right, there are links for "Войти" and "Регистрация". The main heading is "Регистрация". Below it, there is a form with six input fields: "Email", "Имя", "Фамилия", "Отчество", "Пароль", and "Капча". Below the "Капча" field is a CAPTCHA image showing the numbers "85 84 6". At the bottom of the form is a blue button labeled "Зарегистрироваться".

Рисунок А.2 – Форма регистрации в системе



## ПРИЛОЖЕНИЕ А (стр. 2)

The screenshot shows the user profile page for 'Lumbar Spine'. The navigation bar includes 'Главная', 'Профиль', 'Об авторе', 'Контакты', 'Справочник', and 'Меню обследований'. A 'Выход' button is in the top right. The main heading is 'Профиль' with a welcome message: 'Добро пожаловать, Иванов Алексей Иванович'. Below this are three buttons: 'Действия с аккаунтом', 'Редактировать профиль', and 'Запросить права'. The profile form contains the following fields:

<b>Имя</b>	<b>Дата рождения</b>	<b>Страна</b>
Алексей	02.03.1982	Страна
<b>Фамилия</b>	<b>Паспортные данные</b>	<b>Город</b>
Иванов	Паспортные данные	Город
<b>Отчество</b>	<b>Телефон</b>	<b>Улица</b>
Иванович	+79241233212	Улица
<b>ОМС</b>	<b>Пол</b>	<b>Почтовый индекс</b>
2332311	Мужчина	Почтовый индекс

At the bottom, there is a 'Ваш пароль' label.

Рисунок А.3 – Экранная форма профиля пользователя системы

The screenshot shows the X-ray examination list page. The navigation bar is the same as in Figure A.3. A search bar at the top contains 'По коду' and 'Искать'. The main content area features a central card for an examination:

- UVpltm** **Активное**
- Обследование назначено пациенту: **Александр Петров**
- Регистратор: **Иван Иванов**
- Обследование назначено на: **2020-06-04**

At the bottom of the card are two buttons: 'Начать' and 'Отклонить'. The background image shows a 3D model of a human spine.

Рисунок А.4 – Экранная форма списков обследований рентгенолога

Lumbar Spine Главная Профиль Об авторе Контакты Меню регистратора Выход

### Добавить обследование

**Врач-рентгенолог**

Алексей Иванов ▼

**Дата начала обследования**

04.06.2020

[Добавить обследование](#)

Lumbar Spine авторства Селютина Александра

Рисунок А.5 – Экранная форма добавления обследования регистратором

## Исходный код нейронной сети

### Листинг Б.1 – Исходный код сверточной нейронной сети на языке Python

```

# Imports
import os
import random

import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from tensorflow.python.keras.models import load_model

# My imports
from UNet import UNet
from data_generator import DataGen

# Seeding
seed = 2019
random.seed = seed
np.random.seed = seed
tf.seed = seed

# Hyperparameters
# Shows source image and mask overlay
image_size = 256
train_path = "new_data/"
epochs = 5
batch_size = 2

# Training Ids
train_ids = next(os.walk(train_path))[1]

# Validation Data Size
val_data_size = 5

valid_ids = train_ids[:val_data_size]
train_ids = train_ids[val_data_size:]

gen = DataGen(train_ids, train_path, batch_size=batch_size, image_size=image_size)
x, y = gen.__getitem__(0)
print(x.shape, y.shape)

r = random.randint(0, len(x) - 1)

fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.4)

ax = fig.add_subplot(1, 2, 1)
ax.imshow(x[r])
ax = fig.add_subplot(1, 2, 2)
ax.imshow(np.reshape(y[r], (image_size, image_size)), cmap="gray")

plt.show(fig)

# UNet
model = UNet()
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["acc"])
model.summary()

# Training the model

```

```

train_gen = DataGen(train_ids, train_path, image_size=image_size,
batch_size=batch_size)
valid_gen = DataGen(valid_ids, train_path, image_size=image_size,
batch_size=batch_size)

train_steps = 150 # len(train_ids) // batch_size
valid_steps = 150 # len(valid_ids) // batch_size

model.fit_generator(train_gen, validation_data=valid_gen,
steps_per_epoch=train_steps,
validation_steps=valid_steps, epochs=epochs)

# Save the Weights
model.save("model.h5")
# model = load_model("model.h5")

# Dataset for prediction
x, y = valid_gen.__getitem__(0)
print(x.shape)

result = model.predict(x)
result = result > 0.5

fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.4)

ax = fig.add_subplot(1, 2, 1)
ax.imshow(np.reshape(x[0], (image_size, image_size, 3)), cmap="gray")

ax = fig.add_subplot(1, 2, 2)
ax.imshow(np.reshape(result[0] * 255, (image_size, image_size)), cmap="gray")

plt.show(fig)

```

## Листинг Б.2 – Исходный код класса для формирования сегментированных изображений

```

import os

import cv2
import numpy as np
from tensorflow import keras

class PredictGen(keras.utils.Sequence):
    def __init__(self,
                 # ids,
                 path,
                 batch_size=8,
                 image_size=256):
        # self.ids = ids
        self.path = path
        self.batch_size = batch_size
        self.image_size = image_size
        # self.on_epoch_end()

    def __load__(self):
        # Path
        image_path = os.path.join(self.path)

        # Reading Image
        image = cv2.imread(image_path, 1)

```

## ПРИЛОЖЕНИЕ Б (стр. 3)

```
image = cv2.resize(image, (self.image_size, self.image_size))

# Normalizaing
image = image / 255.0

return image

def __getitem__(self):
    image = []
    _img = self.__load__()
    image.append(_img)

    image = np.array(image)

    return image
```

Исходный код веб-приложения

Листинг В.1 – Исходный код роутинга обработки снимка МРТ

```

@doctor.route('/process_survey', methods=['POST'])
@login_required
@doctor_required
def process_survey():
    survey_id = request.form.get('survey_id', type=int)
    survey = Survey.query.filter(Survey.id == survey_id).first()
    if survey_id is not None and survey is not None and survey.is_active:
        if 'mri_image' not in request.files:
            flash('Нет снимка МРТ')
            return redirect(url_for('doctor.start_survey', survey=survey_id))
        file = request.files['mri_image']
        if file.filename == '':
            flash('Не загружен файл')
            return redirect(url_for('doctor.start_survey', survey=survey_id))
        if file and allowed_file(file.filename):
            original_image_path = save_original_image(file)
            with graph.as_default():
                segmented_image_path = predict_image(original_image_path)
            if analyze_image(survey_id, segmented_image_path):
                overlay_image_path = overlay_image(original_image_path,
                segmented_image_path)

                process_survey_images(survey, original_image_path,
                segmented_image_path, overlay_image_path)
                app.logger.info('Go to the survey finish page')
                return redirect(
                    url_for(
                        'doctor.finish_survey',
                        mi=get_file_name_from_path(original_image_path),
                        smi=get_file_name_from_path(segmented_image_path),
                        oi=get_file_name_from_path(overlay_image_path),
                        survey=survey_id
                    )
                )
            else:
                flash('Не могу сформировать финальный отчет')
                return redirect(url_for('doctor.start_survey', survey=survey_id))
    else:
        app.logger.warning('Cannot find required request parameter survey')
        return render_template('errors/404.html')

```

Листинг В.2 – Исходный код сервиса обработки изображений

```

import os

import cv2
import numpy as np
from tensorflow.python.keras.models import load_model

from app.config.constants import IMAGE_EXTENSION
from app.config.settings import COMMON_DIRECTORY, MODEL_NAME,
STORED_IMAGES_PARENT_DIRECTORY, \
    SEGMENTED_IMAGES_DIRECTORY, UPLOADED_IMAGES_DIRECTORY, APP_DIRECTORY,
STATIC_DIRECTORY, DISKS_IMAGES_DIRECTORY

```

## ПРИЛОЖЕНИЕ В (стр. 2)

```
from app.utils.image_processing_service.predict_generator import PredictGenerator
from app.utils.utils import generate_unique_code

def upload_model():
    model_path = os.path.join(
        os.path.split(os.environ['VIRTUAL_ENV'])[0],
        APP_DIRECTORY,
        STATIC_DIRECTORY,
        COMMON_DIRECTORY,
        MODEL_NAME
    )
    return load_model(model_path)

def generate_image_name(save_extension=False):
    return generate_unique_code(length=20) + IMAGE_EXTENSION if not save_extension
else generate_unique_code(length=20)

def prepare_image_path(is_generating=True, image_class='segmented'):
    return os.path.join(
        os.path.split(os.environ['VIRTUAL_ENV'])[0],
        APP_DIRECTORY,
        STATIC_DIRECTORY,
        COMMON_DIRECTORY,
        STORED_IMAGES_PARENT_DIRECTORY,
        get_image_class(image_class),
        generate_image_name() if is_generating else ''
    )

def predict_image(image_path, image_size=256, batch_size=2):
    predict_generator = PredictGenerator(image_path, image_size=image_size,
batch_size=batch_size)
    image = predict_generator.__getitem__ ()

    result = upload_model().predict(image)
    result = result > 0.5

    result = np.reshape(result[0] * 255, (image_size, image_size))
    result = clean_image(result)
    segmented_image_path = prepare_image_path(image_class='segmented')
    cv2.imwrite(segmented_image_path, result)
    return ''.join(segmented_image_path)

def overlay_image(original_image_path, segmented_image_path):
    original_image = cv2.imread(original_image_path)
    segmented_image = cv2.imread(segmented_image_path)

    overlay = cv2.addWeighted(original_image, 0.5, segmented_image, 0.5, 0)
    overlay_image_path = prepare_image_path(image_class='uploaded')
    cv2.imwrite(overlay_image_path, overlay)
    return ''.join(overlay_image_path)

def clean_image(image):
    image_copy = np.uint8(image)
    grayscale_image = cv2.cvtColor(cv2.cvtColor(image_copy, cv2.COLOR_BGR2RGB),
cv2.COLOR_RGB2GRAY)
    _, binary = cv2.threshold(grayscale_image, 225, 255, cv2.THRESH_BINARY)

    kernel = np.ones((5, 5), np.uint8)
    opening = cv2.morphologyEx(binary, cv2.MORPH_OPEN, kernel)
    closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, kernel)
```

```

    return closing

def save_original_image(image):

    temp_image_path = prepare_image_path(is_generating=False,
image_class='uploaded')
    temp_image_filename = generate_image_name(save_extension=True) + '.' +
image.filename.rsplit('.', 1)[1].lower()
    image.save(os.path.join(temp_image_path, temp_image_filename)) # Save
temporary file

    original_image_path = prepare_image_path(is_generating=True,
image_class='uploaded')
    temp_image = cv2.imread(os.path.join(temp_image_path, temp_image_filename), 1)
    cv2.imwrite(''.join(original_image_path), temp_image) # Save original file in
uploaded folder
    os.unlink(os.path.join(temp_image_path, temp_image_filename)) # Remove
temporary file
    return ''.join(original_image_path)

def save_disk_image(region_of_interest):
    region_of_interest = cv2.bitwise_not(region_of_interest)

    disk_image_path = prepare_image_path(image_class='disks')
    cv2.imwrite(disk_image_path, region_of_interest)
    return get_file_name_from_path(disk_image_path)

def get_file_name_from_path(image_path):
    return os.path.basename(image_path)

def get_image_class(image_stored_class='segmented'):
    if image_stored_class == 'segmented':
        return SEGMENTED_IMAGES_DIRECTORY
    elif image_stored_class == 'uploaded':
        return UPLOADED_IMAGES_DIRECTORY
    elif image_stored_class == 'disks':
        return DISKS_IMAGES_DIRECTORY
    else:
        raise ValueError("Illegal stored folder")

```

### Листинг В.3 – Исходный код сервиса анализа изображений

```

import cv2
import numpy as np

from app.utils.image_processing_service.process_image import save_disk_image
from app.utils.services.report_service import set_report_parameters,
create_report_relationship

def calculate_disk_deformation(disk_image, is_top=True):
    disk_image = get_binary_image(disk_image, is_reading=False, is_invert=True)
    result_deformation = []
    column, row = get_center_of_contour(disk_image)
    center_row = row

    for column_iterator in range(1, disk_image.shape[1] - 1): # iterate over x
axis to the ride side
        deformation_power = 0
        while True:

```



```

        if disk_image[row][column_iterator] == 0:
            result_deformation.append(deformation_power)
            row = center_row
            break
        else:
            if is_top:
                row -= 1
            else:
                row += 1

            deformation_power += 1
    return result_deformation

def calculate_radiologic_parameters(contour):
    x, y, w, h = cv2.boundingRect(contour)
    min_axis = w
    max_axis = h

    area = cv2.contourArea(contour)
    convex = cv2.convexHull(contour)
    convex_area = cv2.contourArea(convex)

    # Calculate parameters
    extent = round(float(area) / (h * w), 5)
    solidity = round(float(area) / 1 if convex_area == 0 else convex_area, 5)
    axis = round(max_axis / min_axis, 5)
    equip_diameter = round(np.sqrt(4 * area / np.pi), 5)

    return extent, solidity, axis, equip_diameter

def get_image_contours(image_path):
    binary = get_binary_image(image_path, is_reading=True, is_invert=True)
    contours, _ = cv2.findContours(binary.astype(np.uint8), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    return binary, contours

def get_grayscale_image(image_path, is_reading):
    if is_reading:
        image_path = cv2.imread(image_path)
        image_path = cv2.cvtColor(image_path, cv2.COLOR_BGR2RGB)
        gray = cv2.cvtColor(image_path, cv2.COLOR_RGB2GRAY)
    return gray

def get_binary_image(image_path, is_reading=True, is_invert=True):
    gray = get_grayscale_image(image_path, is_reading)
    _, binary = cv2.threshold(gray, 225, 255, cv2.THRESH_BINARY_INV if is_invert
else cv2.THRESH_BINARY)
    return binary

def get_center_of_contour(contour):
    moment = cv2.moments(contour)
    cx = (moment['m10'] / moment['m00'])
    cy = (moment['m01'] / moment['m00'])
    return int(cx), int(cy)

def analyze_image(survey_id, segmented_image_path):
    image, contours = get_image_contours(segmented_image_path)
    if len(contours) < 5:
        return False

```

```

objects = 0
index = 0
for contour in contours:
    index += 1
    if index == 1:
        continue

    x, y, w, h = cv2.boundingRect(contour)
    roi = image[y:y + h, x:x + w]

    if np.sum(roi) != 0:
        objects += 1

        if objects > 8:
            return False
        disk_image_name = save_disk_image(roi)
        extent, solidity, axis, equip_diameter =
calculate_radiologic_parameters(contour)
        try:
            top_deformation = max(calculate_disk_deformation(roi, is_top=True))
            bottom_deformation = max(calculate_disk_deformation(roi,
is_top=False))
        except:
            top_deformation = 0
            bottom_deformation = 0
        report = set_report_parameters(
            disk_image=disk_image_name,
            disk_number=objects,
            radiological_parameters=[extent, solidity, axis, equip_diameter],
            disk_deformations=[top_deformation, bottom_deformation]
        )
        create_report_relationship(survey_id, report.id)
return True

```