

Инженерная школа информационных технологий и робототехники  
 Направление 09.04.01 – Информатика и вычислительная техника  
 Отделение информационных технологий

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
<b>Математическое и программное обеспечение локализации областей инфильтрата в области легкого по данным компьютерной томографии</b>

УДК 004.415:519.876:620.179.152.1:616-07

Студент

Группа	ФИО	Подпись	Дата
8ВМ83	Проскуров Владислав Анатольевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Аксёнов Сергей Владимирович	канд. техн. наук		

#### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский Владимир Юрьевич	канд. экон. наук		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Горбенко Михаил Владимирович	канд. техн. наук		

#### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Спицын Владимир Григорьевич	д-р техн. наук		

**ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП**  
**09.04.01 Информатика и вычислительная техника**

Код результата	Результат обучения (выпускник должен быть готов)
<b>Общепрофессиональные компетенции</b>	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
<b>Профессиональные компетенции</b>	
P5	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P6	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
<b>Общекультурные компетенции</b>	
P8	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P9	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P10	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P11	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

Инженерная школа информационных технологий и робототехники  
 Направление 09.04.01 – Информатика и вычислительная техника  
 Отделение информационных технологий

УТВЕРЖДАЮ:

Руководитель ООП

В. Г. Спицын

\_\_\_\_\_  
(Подпись)

\_\_\_\_\_  
(Дата)

### ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	ФИО
8ВМ83	Проскурову Владиславу Анатольевичу

Тема работы:

Математическое и программное обеспечение локализации областей инфильтрата в области легкого по данным компьютерной томографии

Утверждена приказом директора (дата, номер)

№ 59-48/с от 28.02.2020 г.

Срок сдачи студентом выполненной работы:

01.06.2020 г.

#### ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Задание на выполнение работы, Набор данных DICOM
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ul style="list-style-type: none"> <li>– Обзор технологий обработки КТ-изображений.</li> <li>– Хранение данных в DICOM формате.</li> <li>– Разработка нейронной сети для локализации патологий.</li> <li>– Аугментация КТ-данных.</li> <li>– Программная реализация нейронной сети.</li> <li>– Тестирование модели.</li> </ul>
Перечень графического материала	Презентация работы

#### Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич канд. экон. наук, доцент.
Социальная ответственность	Горбенко Михаил Владимирович канд. техн. Наук, доцент.

<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>
Архитектура нейронной сети для локализации областей инфильтрата
Аугментация данных
Алгоритм локализации областей инфильтрата
Результат локализации областей инфильтрата

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
--	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Аксёнов Сергей Владимирович	канд. техн. наук		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ83	Проскуров Владислав Анатольевич		

Инженерная школа информационных технологий и робототехники  
 Направление 09.04.01 – Информатика и вычислительная техника  
 Уровень образования магистратура  
 Отделение информационных технологий  
 Период выполнения осенний / весенний семестр 2019/2020 учебного года

Форма представления работы:

Магистерская диссертация

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

<b>Срок сдачи студентом выполненной работы:</b>	01.06.2020
---	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
24.02.2020	Обзор технологий обработки КТ-изображений	7
02.03.2020	Хранение данных в DICOM формате	5
15.03.2020	Разработка нейронной сети для локализации патологий	18
23.03.2020	Аугментация КТ-данных	14
10.04.2020	Программная реализация нейронной сети	23
15.05.2020	Тестирование модели	14
04.06.2020	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
04.06.2020	Социальная ответственность	10

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Аксёнов Сергей Владимирович	канд. техн. наук		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Спицын Владимир Григорьевич	д-р техн. наук		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
8BM83	Проскуров Владислав Анатольевич

<b>Школа</b>	Информационных технологий и робототехники	<b>Отделение школы (НОЦ)</b>	Отделение информационных технологий
<b>Уровень образования</b>	Магистратура	<b>Направление/специальность</b>	09.04.01 Информатика и вычислительная техника

### Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Использовать действующие ценники и договорные цены на потребленные материальные и информационные ресурсы, а также указанную в МУ величину тарифа на эл. энергию
Нормы и нормативы расходования ресурсов	—
Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Действующие ставки единого социального налога и НДС, ставка дисконтирования = 0,1

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

Оценка коммерческого и инновационного потенциала НТИ	Дать характеристику существующих и потенциальных потребителей (покупателей) результатов ВКР, ожидаемых масштабов их использования
Разработка устава научно-технического проекта	Разработать проект такого устава в случае, если для реализации результатов ВКР необходимо создание отдельной организации или отдельного структурного подразделения внутри существующей организации
Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Построение плана-графика выполнения ВКР, составление соответствующей сметы затрат, расчет цены результата ВКР.
Определение ресурсной, финансовой, экономической эффективности	Оценка экономической эффективности использования результатов ВКР, характеристика других видов эффекта

### Перечень графического материала

«Портрет» потребителя результатов НТИ
Сегментирование рынка - <u>выполнить</u>
Оценка конкурентоспособности технических решений - <u>выполнить</u>
Диаграмма FAST
Матрица SWOT - <u>выполнить</u>
График проведения и бюджет НТИ - <u>выполнить</u>
Оценка ресурсной, финансовой и экономической эффективности НТИ - <u>выполнить</u>
Потенциальные риски

Дата выдачи задания для раздела по линейному графику	
--	--

### Задание выдал консультант:

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент	Коногопский Владимир Юрьевич	канд. экон. наук		

### Задание принял к исполнению студент:

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8BM83	Проскуров Владислав Анатольевич		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ВМ83	Проскуров Владислав Анатольевич

<b>Школа</b>	Информационных технологий и робототехники	<b>Отделение школы (НОЦ)</b>	Отделение информационных технологий
<b>Уровень образования</b>	Магистратура	<b>Направление/специальность</b>	09.04.01 Информатика и вычислительная техника

Тема ВКР:

Математическое и программное обеспечение локализации областей инфильтрата в области легкого по данным компьютерной томографии	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Математическое и программное обеспечение локализации областей инфильтрата в области легкого по данным компьютерной томографии
<b>Перечень вопросов, подлежащих исследованию, проектированию и разработке:</b>	
<b>1. Правовые и организационные вопросы обеспечения безопасности:</b> - специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; - организационные мероприятия при компоновке рабочей зоны.	– Трудовой кодекс РФ от 30.12.2001 197-ФЗ; – СанПиН 2.2.4.548-96; – СанПиН 2.2.2/2.4.1340-03; – ГОСТ 12.2.032-78.
<b>2. Производственная безопасность:</b> 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	– создание благоприятного микроклимата; – обеспечение электробезопасности; – организация искусственного освещения; – защита от производственных шумов.
<b>3. Экологическая безопасность:</b>	– утилизация бытовых отходов; – эксплуатация компьютера.
<b>4. Безопасность в чрезвычайных ситуациях:</b>	– возможные ЧС: пожар на рабочем месте.

Дата выдачи задания для раздела по линейному графику	
--	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент	Горбенко Михаил Владимирович	канд. техн. наук		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ВМ83	Проскуров Владислав Анатольевич		

## РЕФЕРАТ

Выпускная квалификационная работа содержит 130 страниц формата А4, 30 рисунков, 28 таблиц, 46 источников, 2 приложения.

Ключевые слова: машинное обучение, 3D свертка, аугментация данных, компьютерная томография, КТ, сегментация, медицинские изображения, инфильтрат, коронавирус, 3D convolution, U-net, ground glass, СТ, COVID-19.

Объектом исследования являются снимки компьютерной томографии содержащие зоны инфильтративных уплотнений в области легкого.

Цель работы – повышение эффективности работы врача-радиолога путем разработки архитектуры глубокой сверточной нейронной сети для автоматической локализации инфильтрата на снимках компьютерной томографии.

В процессе исследования проводился анализ современных подходов в решении задач глубокого машинного обучения, способных к локализации областей инфильтрата, с использованием операции трехмерной свертки.

В результате исследования разработан уникальный подход к аугментации многомерных изображений; разработана архитектура сверточно-разверточной нейронной сети применимой в обработки трехмерных фрагментов биомедицинских изображений; получены значимые результаты в нейросетевой обработке набора данных содержащего КТ-снимки пациентов с диагностированным COVID-19 в различных стадиях.

Область применения: медицинские учреждения, выполняющие анализ результатов томографических исследований.

Значимость работы заключается в практической реализации алгоритмического решения локализации инфильтративных зон в области легкого.

В будущем возможна разработка комплексного нейросетевого решения выполняющего интеллектуальный анализ результатов лучевой диагностики с целью выявления, локализации и классификации характерных признаков для различных инфекционных и соматических заболеваний.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

В данной работе описаны следующие термины с соответствующими определениями:  
Аугментация данных – создание дополнительных данных из имеющихся за счет применения алгоритмической обработки.

Батч – подвыборка примеров из всего обучающего набора данных, пропускаемая через нейронную сеть за одну итерацию.

Воксел – элемент объёмного изображения, содержащий значение элемента раstra в трёхмерном пространстве.

Гентри – подвижная, как правило, кольцевая часть томографического аппарата.

Датасет – набор данных используемый в процессе обучения нейронной сети.

Конкатенации – бинарная операция, определяемая для объектов имеющих последовательную структуру. В общем случае, результатом конкатенации двух объектов А и В является объект, полученный поочерёдным добавлением всех элементов объекта В в конец объекта А.

Критерий останова – условие, при исполнении которого алгоритм прекращает свое выполнение.

Легочная консолидация – уплотнение лёгочной ткани за счёт заполнения воздушных в норме альвеол содержимым (экссудатом, трансудатом, кровью, водой и т. д.).

Матовое стекло – неспецифический радиологический симптом, отражающий различные патологические изменения в лёгочной ткани на уровне альвеол.

Паренхима – Основная ткань какого-либо органа.

Предиктор – показатель; прогностический параметр; прогностический фактор; средство прогнозирования.

Пуллинг – процесс уменьшения размеров изображения путём усреднения значений его элементов.

Резистентность – способность микроорганизма переносить значительно большие концентрации препарата, чем остальные микроорганизмы данного вида. Резистентность возникает при изменении генома бактериальной клетки в результате спонтанных мутаций.

Ретикулярные изменения – изменения легочного интерстиция возникающие в результате его утолщения при заполнении жидкостью, клеточными элементами, фиброзной тканью или другим патологическим субстратом.

Тензор – многомерная матрица, содержащая элементы одного типа данных.

Томография – получение послойного изображения внутренней структуры объекта.

Фреймворк – программная платформа, определяющая структуру программной системы.

Шкала Хаунсфилда – шкала линейного ослабления излучения по отношению к дистиллированной воде, рентгеновская плотность которой была принята за 0 HU.

Штамм – чистая культура микроорганизмов одного вида, выделенная из какого-либо организма заболевшего животного, человека или из окружающей природы.

Эквидистантный – расположенный на одинаковом расстоянии от чего-либо.

ResNet – архитектура сверточной нейронной сети, позволяющая преодолеть некоторые проблемы глубоко машинного обучения за счет методики остаточного обучения.

DICOM – медицинский отраслевой стандарт создания, хранения, передачи и визуализации цифровых медицинских изображений и документов обследованных пациентов.

NIfTI – формат данных облегчающий взаимодействия функциональных пакетов программного обеспечения для анализа данных МРТ.

U-net – архитектура сверточной нейронной сети, разработанной для сегментации биомедицинских изображений.

ИНС – Искусственная нейронная сеть.

ГА – генетический алгоритм, эвристический алгоритм поиска оптимального решения, основанный на принципах генетики и естественного отбора.

КТ – компьютерная томография.

МРТ – магнитно-резонансная томография.

ПО – программное обеспечение.

ТЗР – транспортно-заготовительные расходы, связанные с заготовкой и доставкой материальных ценностей.

ЕСН – единый социальный налог.

ЯП – язык программирования.

2D, 3D – что-либо, имеющее два измерения, три измерения.

HU – денситометрический показатель по шкале Хаунсфилда.

## СОДЕРЖАНИЕ

Введение.....	13
1 Цифровые медицинские изображения .....	16
1.1 Форматы для представления цифровых медицинских изображений.....	17
1.2 Проблемы обработки биомедицинских изображений .....	18
2 Архитектура нейронной сети для локализации областей инфильтрата .....	20
2.1 Операция свертки в глубоком обучении .....	20
2.1.1 Одноканальная свертка .....	20
2.1.2 Многоканальная свертка .....	21
2.1.3 Трехмерная свертка .....	21
2.2 Разработка архитектуры нейронной сети.....	22
2.3 Реализация архитектуры нейронной сети .....	25
3 Аугментация данных .....	28
3.1 Одномерная трансформация.....	28
3.2 Двумерная трансформация .....	30
3.3 Случайные искажения.....	32
3.4 Трехмерная трансформация.....	33
3.5 Реализация алгоритма аугментации данных .....	34
4 Алгоритм локализации областей инфильтрата.....	40
4.1 Используемые функционалы качества в оценке нейросетевой сегментации .....	41
4.2 Реализация алгоритма обучения .....	44
4.3 Обработка скользящим окном .....	47
4.4 Бинаризация нейросетевой сегментации.....	52
4.5 Генерация обучающего набора данных.....	54
4.6 Обучение нейронной сети.....	57
4.7 Сглаживание результатов нейросетевой сегментации фильтром Гаусса .....	60
5 Результат локализации областей инфильтрата .....	64
6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение .....	68
6.1 Потенциальные потребители результатов исследования .....	68
6.2 Оценка конкурентоспособности технических решений.....	70
6.3 SWOT-анализ.....	72
6.4 Оценка готовности проекта к коммерциализации .....	73
6.5 Организация и планирование работ по проекту .....	74
6.5.1 Определение продолжительности этапов работ.....	75
6.6 Расчет сметы затрат на выполнение проекта .....	78
6.6.1 Расчет затрат на материалы.....	78
6.6.2 Расчет заработной платы.....	79
6.6.3 Расчет затрат на социальный налог .....	80

6.6.4	Расчет затрат на электроэнергию.....	80
6.6.5	Расчет амортизационных расходов.....	81
6.6.6	Расчет прочих расходов .....	82
6.6.7	Расчет общей себестоимости разработки.....	83
6.6.8	Оценка прибыли .....	84
7	Социальная ответственность .....	85
7.1	Правовые и организационные вопросы обеспечения безопасности. ....	85
7.2	Производственная безопасность.....	87
7.2.1	Правовые и организационные вопросы обеспечения безопасности .....	87
7.2.2	Опасность поражения электрическим током .....	89
7.2.3	Отсутствие или недостаток освещения .....	91
7.2.4	Производственные шумы.....	96
7.3	Экологическая безопасность.....	97
7.4	Безопасность в чрезвычайных ситуациях .....	98
7.4.1	Перечень возможных ЧС при разработке и эксплуатации проектируемого решения .....	98
7.4.2	Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.....	99
7.4.3	Организационные мероприятия при компоновке рабочей зоны .....	101
7.5	Заключение по разделу .....	102
	Заключение.....	103
	Список использованных источников .....	104
	Приложение А. Результат нейросетевой сегментации областей интереса.....	107
	Приложение Б. Software for localization of infiltrate areas in the lung region based on computed tomography data .....	115
	Приложение В. CD-диск.....	130

## ВВЕДЕНИЕ

В 2020 году человечество столкнулось с пандемией коронавирусной инфекции COVID-19. Данная инфекция представляет собой опасное заболевание, которое может протекать как в форме острой респираторной вирусной инфекции лёгкого течения, так и в тяжёлой форме, специфические осложнения которой могут включать вирусную пневмонию. Согласно статистике ВОЗ количество заболеваний в мире сначала возникновения первой вспышки в Китае продолжает стремительно расти. На начало мая 2020 года во многих развитых странах, таких как Германия, Китай, Южная Корея, Япония распространение вируса удалось взять под контроль. Однако, ввиду того, что данный вирус имеет высокий показатель индекса репродукции, а также является новым вирусом, для которого на данный момент не создано вакцины, возникает большая вероятность появления повторных волн распространения вируса в ближайшие годы [1]. В связи с этим становится актуальной проблема ранней и точной диагностики наличия заболевания COVID.

Наиболее оперативным методом выявления респираторных нарушений, в том числе вызванных COVID, является компьютерная томография (КТ) легких. Лучевые методы, не являясь основными в диагностике коронавирусной инфекции, стали наиболее информативными для выявления наличия и выраженности изменений в легких. Данные лучевой визуализации позволяют заподозрить поражение легких вирусной этиологией (в том числе COVID-19), влиять на лечение осложнений или на постановку альтернативного диагноза. Все исследователи подчеркивают значимость таких изменений, как «матовое стекло», консолидация, ретикулярные изменения и площадь поражения, единодушно соглашаясь, что они отражают тяжесть заболевания и позволяют прогнозировать его дальнейшее развитие. Очевидно, что существует достоверная корреляционная связь между тяжестью поражения легких, определенной методом КТ и клиническим течением заболевания, а, следовательно, целесообразность количественной оценки протяженности изменений в паренхиме легких [2].

Еще одной распространенной причиной возникновения патологий легких, на сегодняшний день, является туберкулез – инфекционное заболевание, вызываемое различными видами микобактерий. Обычно туберкулез поражает легкие, но может также поражать и другие органы и системы. Микобактерии могут вырабатывать устойчивость к противомикробным препаратам, используемым для лечения этого заболевания. В результате растущего из года в год количества резистентных штаммов микобактерий широкое распространение получил туберкулез с множественной лекарственной устойчивостью.

По данным Всемирной организации здравоохранения странами с наибольшим количеством случаев выявления туберкулеза с множественной лекарственной устойчивостью (47 % от общего числа в мире) являются Китай, Индия и Российская Федерация [3]. По статистике ВОЗ каждый год туберкулезом заражается около десяти миллионов человек и около двух миллионов погибает, что делает туберкулез одной из ведущих причин смерти во всем мире. Считается, что каждый третий человек, заразившись однажды, может являться латентным носителем, десятилетиями ничего не подозревая о заболевании. И лишь в моменты ослабления иммунитета болезнь дает о себе знать. Учитывая, что в запущенных случаях на излечение может уходить до двух лет, диагностика туберкулеза на ранних стадиях играет очень важную роль.

Томографическая диагностика играет практически роль основного метода для уточнения диагнозов при перечисленных заболеваниях. По оценкам компании BusinesStat в 2017 г. натуральный объем рынка томографической диагностики в РФ составил 10,9 млн. исследований [4]. Самым распространенным видом томографических исследований является компьютерная томография (в 2017 году – 68,5 % от натурального объема рынка).

Кроме того, BusinesStat в рамках данного исследования выяснено, что среди видов заболеваний наиболее распространенной группой являются заболевания органов дыхания: в 2017 г. доля заболеваемости по группе составила 24,8 % от общего показателя по стране. Более 1,5 млн. КТ исследований в год выполняется с целью диагностики области легкого.

В настоящее время применение сверточных нейронных сетей к обработке КТ и МРТ изображений показывает великолепные результаты в задачах сегментации костных тканей, анатомии брюшной полости, сердца, легких и головного мозга. Также в последнее время была доказана эффективность использования глубокого обучения в задачах автоматического детектирования, распознавания, сегментации и классификации патологий на медицинских изображениях, что является важным шагом в раннем выявлении и диагностике различных заболеваний. Общее описание используемых подходов и примеры решения конкретных задач можно найти в обзорной публикации [5]. Однако, эффективность использования методов машинного обучения сильно ограничена небольшими объемами тренировочных наборов данных и несогласованным качеством их аннотаций.

Целью настоящей работы является повышение эффективности работы врача-радиолога путем разработки архитектуры глубокой сверточной нейронной сети для автоматической локализации инфильтрата на КТ-изображениях. Решение задачи локализации областей инфильтрата позволит быстрее и точнее готовить медицинское

заключение о состоянии легкого, что позволит увеличить качество медицинской диагностики.

Для достижения заявленной цели необходимо выполнить следующие задачи:

- Изучить форматы хранения цифровых изображений компьютерной томографии;
- Спроектировать архитектуру сверточной нейронной сети (ИНС) для локализации патологий, содержащих инфильтрат в области легкого на КТ;
- Разработать алгоритма аугментации данных;
- Реализовать архитектуру ИНС и обучить модель;
- Протестировать модель и проанализировать результаты.

Входными данными для исследования служат снимки компьютерной томографии грудной клетки пациентов с различными патологиями легких и подтвержденными диагнозами. В работе анализируются современные методы машинного обучения, способные к локализации областей инфильтрата. Широкое распространение таких патологий обуславливает существенную значимость выполняемой работы.

## 1 ЦИФРОВЫЕ МЕДИЦИНСКИЕ ИЗОБРАЖЕНИЯ

В контексте решаемой задачи под изображением понимается упорядоченная совокупность численно измеряемых наблюдений, зафиксированных в некотором ограниченном объеме. Каждое такое наблюдение описывается кубом-вокселем (элемент объёма, volume element, voxel) с координатами  $x, y, z$ . Другими словами, каждый воксел определяется своей интенсивностью и расположением в объеме.

Такие объекты информации формируются исследователями в задачах анализа внутреннего строения объектов реального мира. Одним из методов формирования таких изображений является рентгеновская компьютерная томография (КТ), широко используемая при исследовании внутренних органов человека. КТ-снимок представляет серию двумерных изображений одинакового разрешения, полученных путем измерения и сложной компьютерной обработки разностей ослабления рентгеновского излучения различными по плотности тканями. Так как изучаемый объект располагается внутри апертуры гентри, результирующее изображение хоть и имеет квадратную форму, но все же содержит полезную информацию только внутри вписанной окружности. После формирования очередного среза объект продвигается сквозь гентри на фиксированное расстояние для формирования нового среза. Таким образом, вся серия двумерных изображений-срезов составляет один трехмерный снимок.

При анализе внутренних структур объекта довольно часто на интересующие области объекта наносится разметка. Нанесенная разметка позволяет сегментировать (разделить) объект на области, представляющие и не представляющие интерес для исследователя. Иллюстрация объемного КТ-снимка, разложенного на аксиальные срезы и с нанесенной разметкой, приведена на рисунке 1.1.

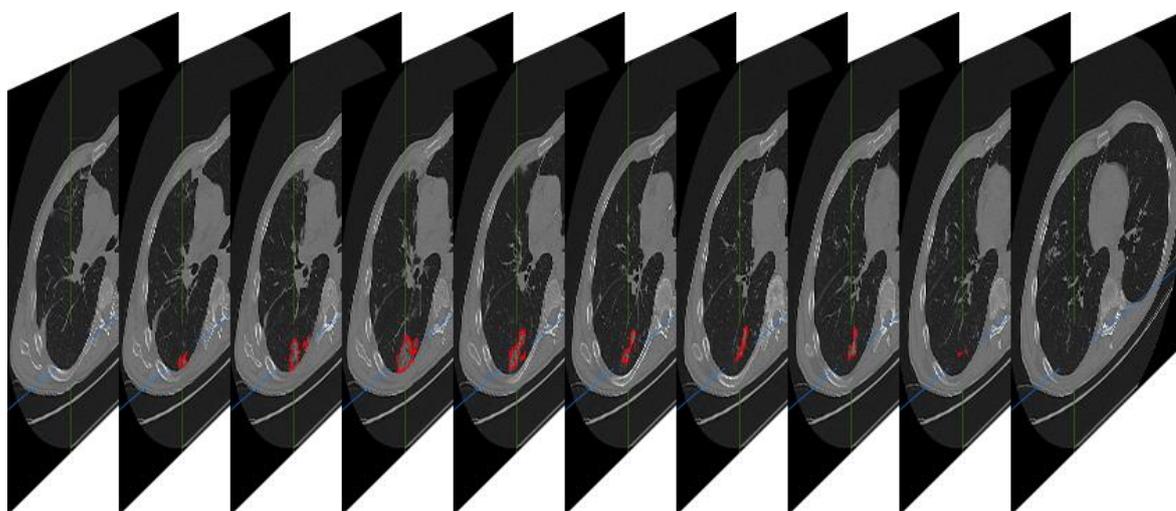


Рисунок 1.1 – Объемная разметка медицинского изображения

## 1.1 Форматы для представления цифровых медицинских изображений

В медицинской отрасли стандартным форматом хранения цифровых медицинских изображений принят формат DICOM (Digital Imaging and Communications in Medicine) [6]. Данный стандарт используется производителями медицинского оборудования в качестве формата по умолчанию для вывода результата проведенного исследования. Существует большое количество программного обеспечения, как коммерческого, так и свободно распространяемого, для работы с \*.dcm файлами. Причины создания программ, поддерживающих данный формат, обуславливаются разными задачами. Некоторые производители создают универсальное ПО для просмотра и анализа всех видов модальностей медицинских изображений, другие делают упор на обработку изображений в конкретной узкой специфике, например, в области ангиографии, виртуальной колоноскопии, перфузионного обследования тканей и др.

Также весомую популярность в научном сообществе получил формат NIfTI (Neuroimaging Informatics Technology Initiative) [7]. Он создавался для решения проблем “Вавилонской башни” (проблем совместимости данных) в задачах визуализации медицинских изображений, в частности, МРТ головного мозга. К основным преимуществам использования данного формата можно отнести следующее:

- Поддерживает все типы хранения значений переменных (8-128 бит):
  - а) целочисленные типы со знаком и без знака,
  - б) трехбайтовый RGB,
  - в) вещественные типы,
  - г) составные типы;
- Имеет стандартизированный способ хранения векторнозначных наборов данных в 1-4-мерных пространствах;
- Использует аффинные преобразования для определения пространственного положения  $(x, y, z)$  по индексам вокселей  $(i, j, k)$ ;
- Использует коды для обозначения единиц пространственно-временных измерений: миллиметры, микроны, секунды.

Для решения задач машинного обучения, связанных с классификацией или сегментацией биомедицинских изображений, перед исследователями встает проблема создания набора данных (dataset), состоящего из обучающих примеров, сегментированных экспертами.

В рамках данной работы для решения задачи ручного аннотирования обучающего набора данных, используется программный комплекс MITK Workbench [8]. Рабочее пространство программного комплекса MITK представлено на рисунке 1.2.

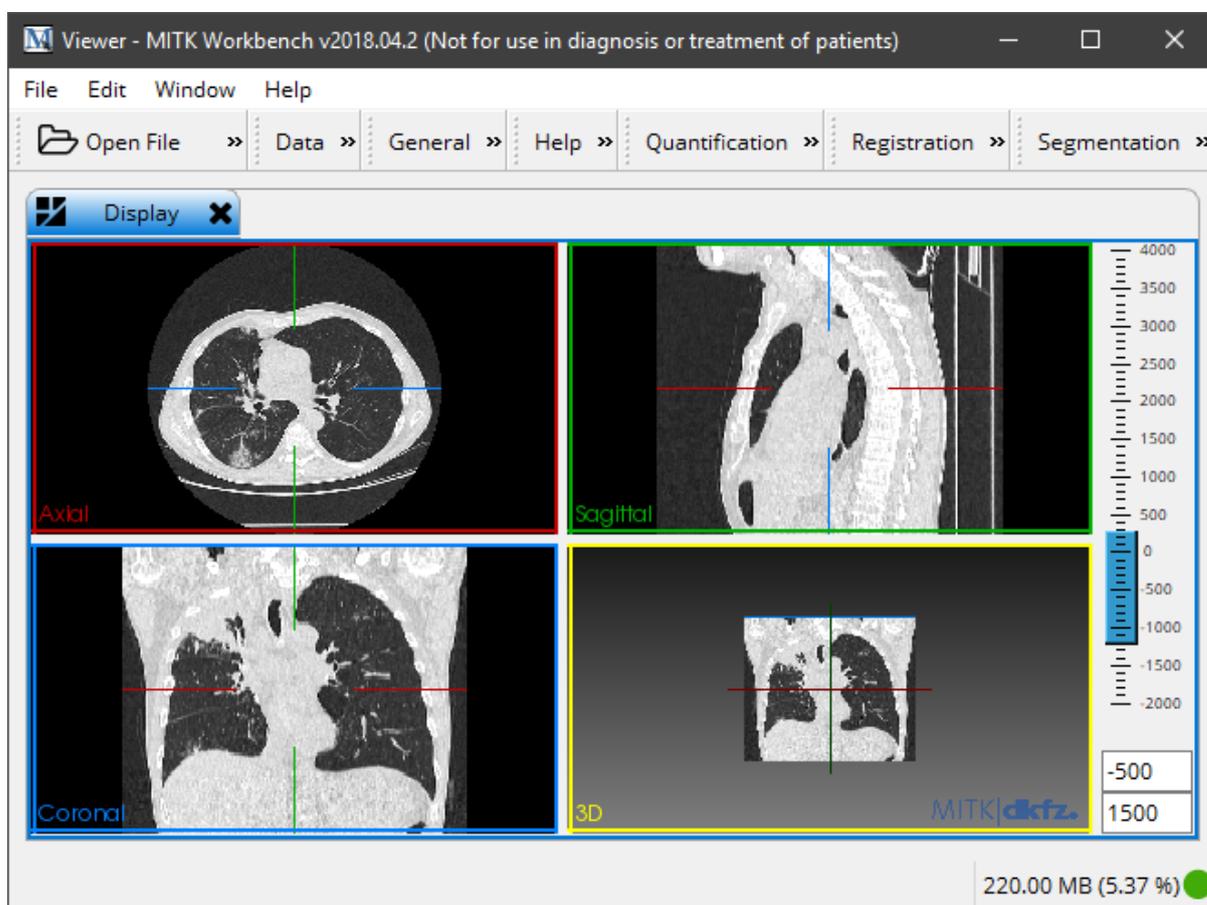


Рисунок 1.2 – Рабочее пространство программы MITK Workbench

Программный комплекс MITK поддерживает различные форматы биомедицинских изображений, в частности, им поддерживаются рассмотренные ранее форматы DICOM и NIfTI. Также данная программа предоставляет удобные инструменты визуализации объемных изображений во всех проекциях и инструменты одно-классовой и много-классовой сегментации в объеме.

## 1.2 Проблемы обработки биомедицинских изображений

Несмотря на то, что обычно диаметр апертуры гентри в среднем составляет 700 мм, это не означает, что каждый элемент итогового изображения будет отражать характеристики плотностей в окрестности данного диаметра. Перед сохранением снимка в DICOM формате, оператор КТ-аппарата самостоятельно выбирает интересующую область, это косвенно влияет на пространственные характеристики объема одного вокселя, так как

формируемое изображение сохраняется с фиксированным разрешением. Сегодня для КТ-снимка, сохраненного в DICOM-формате, чаще всего встречается разрешение  $512 \times 512$ . Но для разных пациентов это разрешение отражает совершенно разный объем. Также итоговое количество срезов, формируемых во время исследования, непостоянно. Оно зависит от длины шага продвижения объекта через гентри и протяженности исследуемой области объекта, интересующей оператора.

В связи с вышеизложенным, в контексте нейросетевой обработки биомедицинского изображения, остро встает вопрос о том, как проводить первичную подготовку данных для входа в нейросеть. С одной стороны, объемное изображение, полученное в результате КТ-исследования, изначально масштабировано таким образом, что от края до края среза располагается интересующая область и вне зависимости от возраста и комплекции тела пациента нейросетевая обработка должна проводиться относительно того масштаба, который был задан оператором КТ-аппарата. Такой подход заставляет нейронную сеть всегда работать с фрагментами изображения без представления о физическом расстоянии между извлекаемыми признаками. С другой стороны, фрагмент объемного изображения обрабатываемый нейросетью всегда должен содержать измерения, заключенные в фиксированный физический объем. Такой подход позволяет нейросети сформировать признаки, основанные не только на взаимном расположении вокселей друг от друга, но и учитывать физическое расстояние между ними.

Какой из подходов более корректен для обучения нейросети и ее использования - вопрос дискуссионный и требует дополнительного исследования. В данной работе принято решение использовать второй способ для формирования обучающего набора данных. Это означает, что нейронная сеть всегда будет обрабатывать фрагменты КТ-снимка, представляющие одинаковый физический объем с фиксированной длиной, шириной и высотой.

Для получения итоговой сегментации, КТ-снимок “разрезается” на фрагменты, соответствующие физическому объему, выбранному при обучении нейросети. Каждый фрагмент масштабируется по сетке, соответствующей размерности входа нейросети. После этого нейросеть обрабатывает каждую сетку и выдает результирующую сегментацию, заключенную в сетку той же размерности что и входные данные. Результирующие сетки масштабируются согласно размерам исходного фрагмента и “склеиваются” в той же последовательности.

## 2 АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ ДЛЯ ЛОКАЛИЗАЦИИ ОБЛАСТЕЙ ИНФИЛЬТРАТА

### 2.1 Операция свертки в глубоком обучении

Свертка является одной из основных операций, используемых в сверточных нейронных сетях [9]. Целью применения свертки является извлечение структурных признаков из входного изображения. При обработке изображений существует широкий спектр различных фильтров, которые можно выбрать для свертки. Каждый тип фильтров помогает извлекать различные аспекты или признаки из входного изображения, например, горизонтальные, вертикальные, диагональные края. Точно так же в сверточной нейронной сети различные признаки извлекаются посредством свертки с использованием фильтров, вес которых автоматически определяется во время обучения [10].

#### 2.1.1 Одноканальная свертка

Для одноканального изображения процесс вычисления свертки проиллюстрирован на рисунке 2.1.

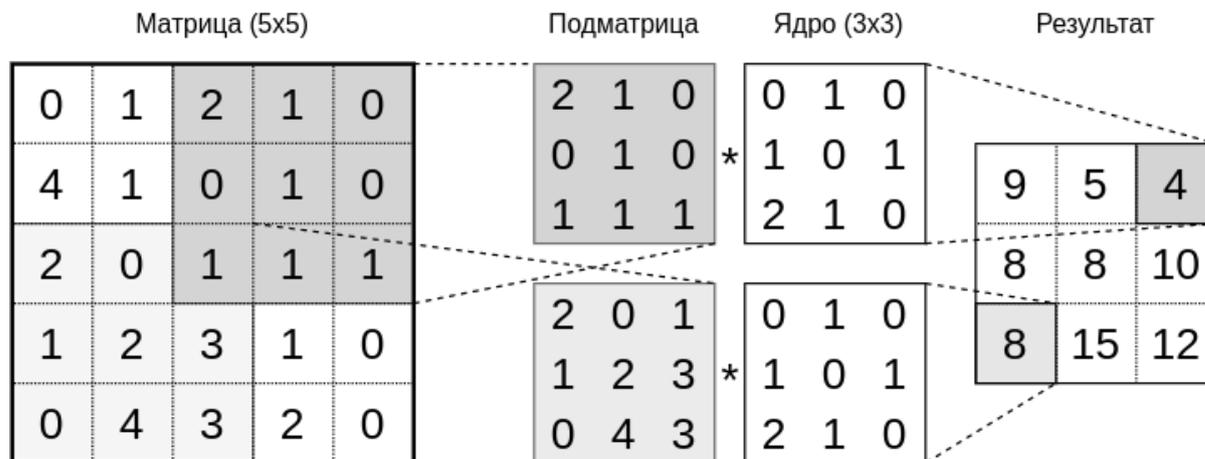


Рисунок 2.1 – Вычисление одноканальной 2D-свертки скользящим ядром

Здесь ядро (фильтр) представляет собой матрицу  $3 \times 3$  с элементами  $[[0, 1, 0], [1, 0, 1], [2, 1, 0]]$ . Фильтр скользит по входной матрице с единичным шагом (stride). Фильтр помещается в исходной матрице 9 раз. В каждой позиции выполняется поэлементное умножение элементов ядра на соответствующие элементы в матрице, после чего результаты складываются. Полученные результаты формируют матрицу  $3 \times 3$ .

### 2.1.2 Многоканальная свертка

Обычно цветные изображения представляются в цветовой модели RGB. Изображение в данной цветовой модели состоит из трёх каналов: красного, зеленого и синего. При этом каждый канал является матрицей размера, соответствующего размеру самого изображения. В контексте сверточной нейронной сети многоканальное изображение отображается в карту признаков в результате обработки фильтром. Фильтр состоит из ядер свертки, количество которых соответствует количеству каналов. Для формирования карты признаков в отношении каждого канала входного изображения выполняется одноканальная свертка с соответствующим ядром фильтра, затем результаты сверток суммируются поэлементно, данный процесс проиллюстрирован на рисунке 2.2.

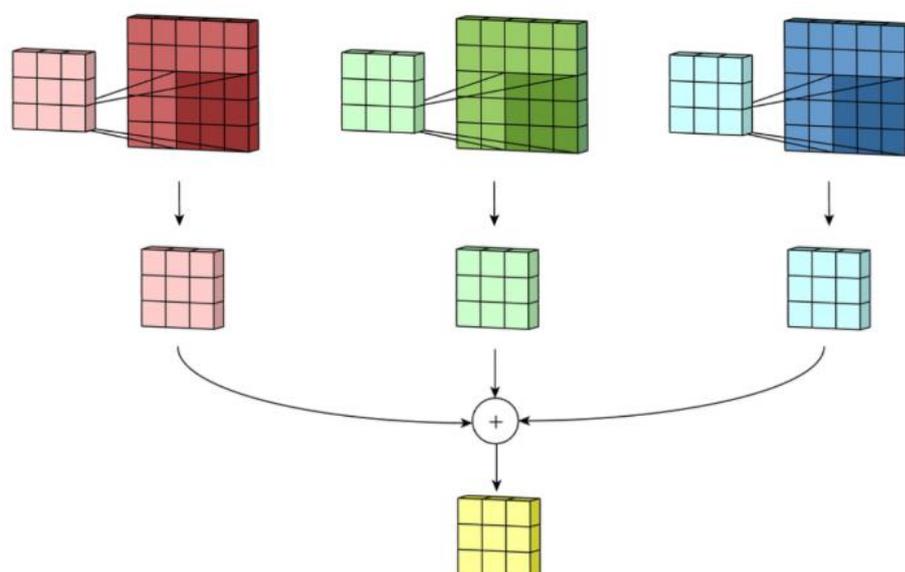


Рисунок 2.2 – Вычисление многоканальной 2D-свертки скользящим фильтром

Применение нескольких фильтров в отношении одного многоканального изображения позволяет сформировать новое многоканальное изображение, количество слоев в котором определяется количеством примененных фильтров.

### 2.1.3 Трехмерная свертка

При обработке биомедицинских изображений, таких как КТ- или МРТ-снимки, уместно использование трехмерной свертки с использованием трехмерных фильтров. Как и в двумерном случае, применение фильтра позволяет сформировать карту признаков для многоканального изображения. Отличие заключается в том, что каждое ядро, представленное в фильтре, является трехмерным.

Трехмерное ядро скользит по входному трехмерному изображению во всех трех направлениях, тем самым формируется трехмерная карта признаков. Данный процесс проиллюстрирован на рисунке 2.3.

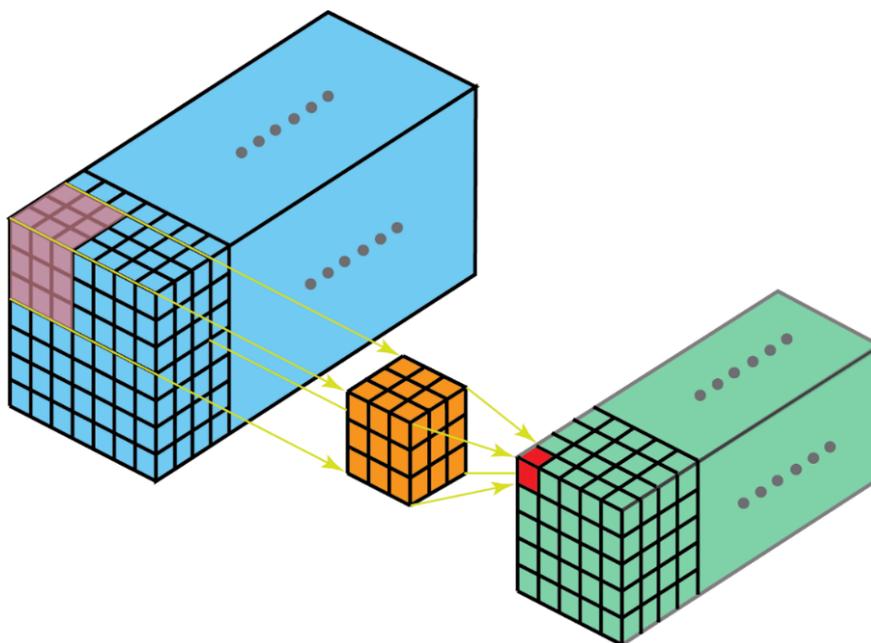


Рисунок 2.3 – Вычисления одноканальной 3D-свертки скользящим окном

Применение серии трехмерных фильтров в отношении многоканального трехмерного изображения позволяет сформировать новое многоканальное трехмерное изображение. Подобно двумерным сверткам, которые кодируют пространственные отношения объектов в двухмерной области, трехмерные свертки могут описывать пространственные отношения объектов в трехмерном пространстве.

## 2.2 Разработка архитектуры нейронной сети

Цель нейронной сети – выполнить сегментацию объемного изображения в соответствии с разметкой эксперта, предъявляемой в качестве требуемого отклика. Сегментация заключена в объем того же размера, что и вход, заполненный нулевыми и единичными значениями, где единицами обозначена область, которую выделил эксперт.

Архитектура нейронной сети, разработанной в рамках данной работы, представлена на рисунке 2.4. Данная архитектура спроектирована согласно концепции, используемой в архитектуре U-Net, представленной в оригинальных работах [11, 12]. U-net состоит из кодировщика (encoder, сжимающий путь) и декодера (decoder, расширяющий путь). Реализация данной архитектуры используется в текущей работе для сегментации КТ-снимков.

Кодировщик состоит из последовательности сверточных слоев, идущих друг за другом. При этом параметры сверточных фильтров выбраны таким образом, что очередной сверточный слой сохраняет отношение стороны аксиального среза обрабатываемого изображения к высоте этого изображения. Приведенные параметры сверток определены с использованием генетического алгоритма (ГА), цель которого заключалась в минимизации аддитивной функции потерь, отображающей общую ошибку сохранения пропорции на каждом уровне. Оптимизируемые параметры сверток ограничены размером ядра от  $2 \times 2 \times 2$  до  $7 \times 7 \times 7$ . При этом количество сверточных фильтров ограничивалось 32-мя. В результате решения данной задачи оптимальным выбором оказался выбор 15 сверточных фильтров с указанными параметрами.

После прохождения сверточного набора фильтров (сверточного слоя) результирующий тензор проходит через слой батч-нормализации [13]. Также в реализации кодировщика предусмотрена реализация принципа остаточного обучения сверточных нейронных сетей, предлагаемого концепцией архитектуры ResNet [14]. Так каждый имитированный ResNet блок формирует два пути: нелинейный, сложный путь проходящий через два сверточных слоя и простой (обходной) путь с использованием пулинг операции *FractionalMaxPool*, которая формирует тензор такого же размера, как и получаемый в результате прохождения по сложному пути. Тензоры, полученные через сложный и простой путь, складываются поэлементно. В случае несовпадения количества каналов в тензорах перед поэлементным сложением меньший тензор дополняется слоями, содержащими нулевые значения. Результирующий тензор пропускается через функцию активации *PReLU* с использованием инициализации весовых коэффициентов, предлагаемой в оригинальной работе [15]. В процессе прохождения через кодировщик в моменты уменьшения пространственных размеров тензора в четыре раза, число каналов (количество карт признаков) выходного тензора увеличивается вдвое (12, 24, 48, 96, 192).

Декодирующая часть нейросети состоит из декодирующих сверточных слоев, увеличивающих пространственные размеры тензора (*upsampling*), и уменьшающих количество каналов. Входной тензор для декодирующей свертки формируется путем конкатенации тензора, полученного из предыдущего блока, с промежуточным тензором, полученным в кодирующей части нейросети. Увеличение размера происходит за счет использования повышенного шага (*stride*), используемого ядрами развертки декодирующих слоев. После прохождения очередного декодирующего слоя к выходному тензору применяется батч-нормализация и функция активации *PReLU*. В результате прохождения последней декодирующей свертки тензор принимает размеры тензора, полученного на входе в сеть, и содержит объемную сегментацию входного изображения.

На выходе сети используется сигмоидальная функция активации *Sigmoid*. Применение данной функции активации позволяет получить единичные значения в вокселях, содержащих разметку, и нулевые в не содержащих. Так как сигмоидальная функция возвращает значения из интервала (0; 1), к выходу нейросети применяется пороговое правило.

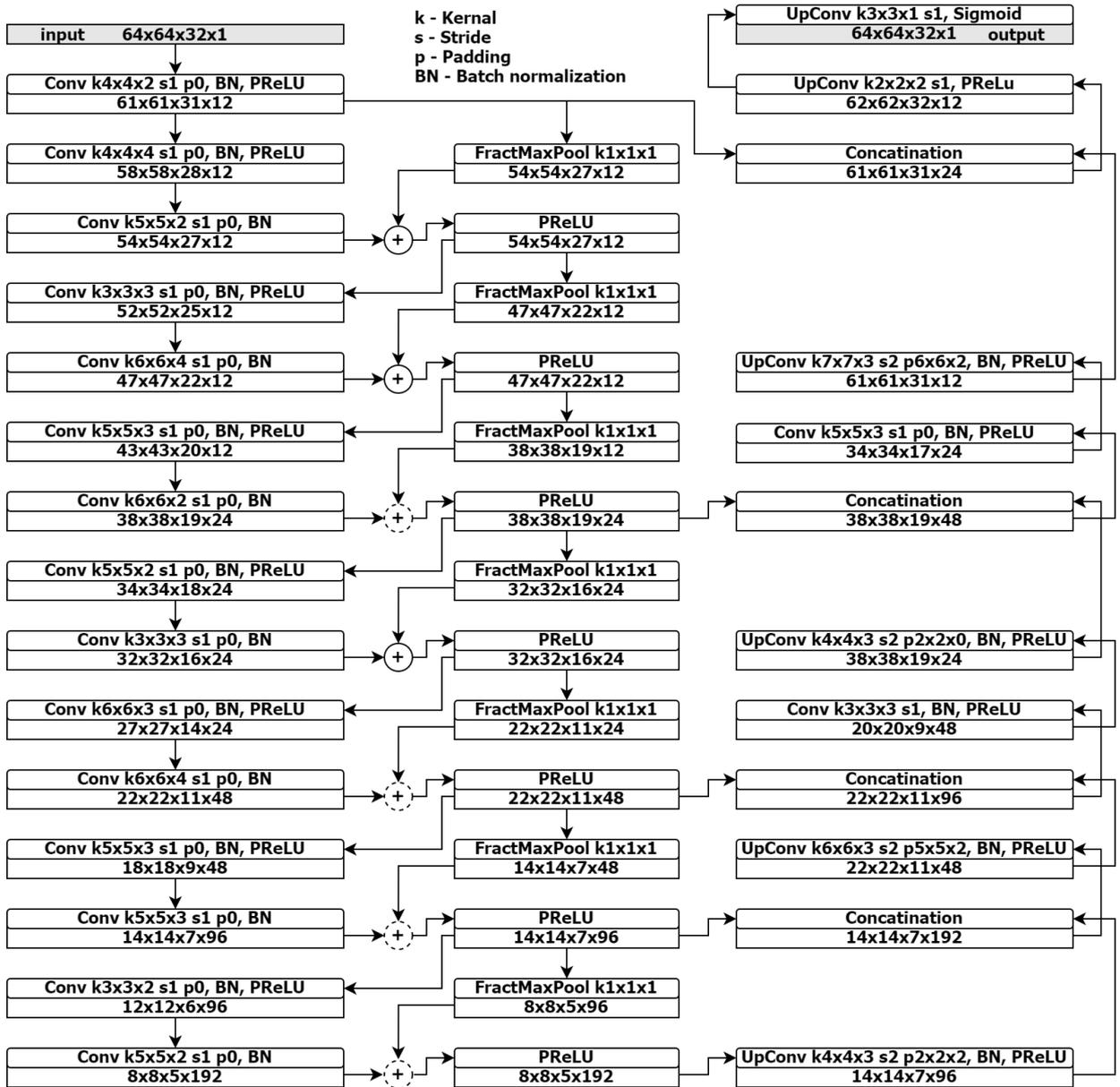


Рисунок 2.4 – Архитектура нейронной сети для сегментации КТ снимков.

## 2.3 Реализация архитектуры нейронной сети

Реализации архитектуры нейронной сети приводится на языке программирования (ЯП) Python, и основывается на использовании фреймворка (набора инструментов) машинного обучения PyTorch [16]. В листинге 2.1 приведен подключение основных модулей необходимых для реализации архитектуры ИНС.

Листинг 2.1 – Файл model.py. Подключение модулей PyTorch

```
import torch
import torch.optim as optim
import torch.nn as nn
import torch.nn.functional as F
from torch.autograd import Variable
from torch.nn.modules.conv import _ConvNd
```

Класс *ConvolutionBlock* реализует сложный и простой путь входного тензора в кодирующей части нейронной сети. Реализация класса *ConvolutionBlock* приведена в листинге 2.2.

Листинг 2.2 – Файл model.py. Класс *ConvolutionBlock*

```
class ConvolutionBlock(nn.Module):
    def __init__(self, middle_channels, out_channels,
                 kernel_size_1, kernel_size_2, stride, fract_output_size):
        super(ConvolutionBlock, self).__init__()
        self.conv1 = nn.Conv3d(middle_channels, middle_channels, kernel_size_1, stride)
        self.bn1 = nn.BatchNorm3d(middle_channels)
        self.activation = nn.PReLU()
        self.conv2 = nn.Conv3d(middle_channels, out_channels, kernel_size_2, stride)
        self.bn2 = nn.BatchNorm3d(out_channels)
        self.fract_max_pool = nn.FractionalMaxPool3d([1,1,1], fract_output_size)
        self.add_channels_for_concat = None
        if middle_channels != out_channels:
            def add_channels(x):
                out = x
                zero_pads = torch.Tensor(out.size(0), out_channels - out.size(1),
                                         out.size(2), out.size(3), out.size(4)).zero_()
                if isinstance(out.data, torch.cuda.FloatTensor):
                    zero_pads = zero_pads.cuda()
                out = Variable(torch.cat([out.data, zero_pads], dim=1))
                return out
            self.add_channels_for_concat = add_channels
        def forward(self, x):
            residual = x
            out = self.conv1(x)
            out = self.bn1(out)
            out = self.activation(out)
            out = self.conv2(out)
            out = self.bn2(out)
            residual = self.fract_max_pool(x)
            if self.add_channels_for_concat is not None:
                residual = self.add_channels_for_concat(residual)
            out+=residual
            return self.activation(out)
```

Класс *Up\_BN\_PReLU* реализует декодирующую свертку, используемую в декодирующей части нейронной сети. Реализация класса *Up\_BN\_PReLU* приведена в листинге 2.3.

Листинг 2.3 – Файл *model.py*. Класс *Up\_BN\_PReLU*

```
class Up_BN_PReLU(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size,
                 stride, padding, act_type='prelu'):
        super(Up_BN_PReLU, self).__init__()
        self.up_conv = nn.ConvTranspose3d(in_channels,
                                          out_channels,
                                          kernel_size,
                                          stride,
                                          padding)

        self.bn = nn.BatchNorm3d(out_channels)
        if act_type == 'prelu':
            self.activation = nn.PReLU()
        else:
            self.activation = nn.Sigmoid()
    def forward(self, x):
        x = self.up_conv(x)
        x = self.bn(x)
        return self.activation(x)
```

Класс *Conv\_3D* реализует непосредственно архитектуру нейронной сети, приведенной на рисунке 2.4. Реализация класса *Conv\_3D* приводится в листинге 2.4. В конструкторе класса создаются все элементы сети, используемые в кодировщике и декодере. Внутри данных элементов: сверточных и декодирующих слоев, функциях активации *PReLU*, слоях батч нормализации содержатся оптимизируемые параметры. После создания данных элементов выполняется нормализация их весов по методу Каймина (Kaiming He) [15].

Параметр *fc*, передаваемый при инициализации класса *Conv\_3D*, влияет на общее количество карт признаков, формируемых на каждом этапе свертки и развертки внутри ИНС, и, как следствие, влияет на количество скрытых параметров в архитектуре нейронной сети.

На рисунке 2.4 приведена архитектура ИНС при  $fc = 3$ . Чем больше значение *fc*, тем больше оптимизируемых параметров внутри ИНС, тем медленнее процесс обучения. Для облегчения проведения экспериментов, в случае необходимости, данная архитектура оставляет возможность кратно варьировать количеством внутренних параметров. Изменения коэффициента *fc* представляется возможным управление количеством формируемых ИНС карт признаков (feature maps). Предполагается, что увеличение карт признаков может обеспечить значимое улучшение в качестве обучаемой модели.

Метод `Conv_3D.forward`, приведенный на листинге 2.4, реализует проход входного тензора, подлежащего сегментации, через все структурные элементы нейронной сети.

Листинг 2.4 – Файл `model.py`. Класс `Conv_3D`

```

class Conv_3D(nn.Module):
    def __init__(self, fc):
        super(Conv_3D, self).__init__()
        self.conv1 = nn.Conv3d(1, 4*fc, [4,4,2], stride=1)
        self.bn1 = nn.BatchNorm3d(4*fc)
        self.activation= nn.PReLU()
        self.block1 = ConvolutionBlock(4*fc, 4*fc, [4,4,4], [5,5,2], 1, [54,54,27])
        self.block2 = ConvolutionBlock(4*fc, 4*fc, [3,3,3], [6,6,4], 1, [47,47,22])
        self.block3 = ConvolutionBlock(4*fc, 8*fc, [5,5,3], [6,6,2], 1, [38,38,19])
        self.block4 = ConvolutionBlock(8*fc, 8*fc, [5,5,2], [3,3,3], 1, [32,32,16])
        self.block5 = ConvolutionBlock(8*fc, 16*fc, [6,6,3], [6,6,4], 1, [22,22,11])
        self.block6 = ConvolutionBlock(16*fc,32*fc, [5,5,3], [5,5,3], 1, [14,14, 7])
        self.block7 = ConvolutionBlock(32*fc,64*fc, [3,3,2], [5,5,2], 1, [ 8, 8, 5])
        self.up_conv6 = Up_BN_PReLU( 64*fc,32*fc, [4,4,3], 2, padding=[2,2,2])
        self.up_conv5 = Up_BN_PReLU( 64*fc,16*fc, [6,6,3], 2, padding=[5,5,2])
        self.dec_conv1 = nn.Conv3d ( 32*fc,16*fc, [3,3,3], 1)
        self.up_conv4 = Up_BN_PReLU( 16*fc, 8*fc, [4,4,3], 2, [2,2,0])
        self.dec_conv2 = nn.Conv3d ( 16*fc, 8*fc, [5,5,3], 1, padding = 0)
        self.up_conv3 = Up_BN_PReLU( 8*fc, 4*fc, [7,7,3], 2, [6,6,2])
        self.up_conv2 = Up_BN_PReLU( 8*fc, 4*fc, [2,2,2], 1, padding = 0)
        self.up_conv1 = Up_BN_PReLU( 4*fc, 1, [3,3,1], 1, padding = 0,
                                   activation_type='sigm')

        self._init_weight()
    def _init_weight(self):
        for module in self.modules():
            if isinstance(module, _ConvNd):
                nn.init.kaiming_normal_(module.weight, mode='fan_out')
                if module.bias is not None:
                    module.bias.data.zero_()
    def forward(self,x):
        conv1 = self.conv1(x)
        conv1_bn = self.bn1(conv1)
        conv1_act = self.activation(conv1_bn)
        block1 = self.block1(conv1_act)
        block2 = self.block2(block1)
        block3 = self.block3(block2)
        block4 = self.block4(block3)
        block5 = self.block5(block4)
        block6 = self.block6(block5)
        block7 = self.block7(block6)
        up6 = self.up_conv6(block7)
        cat4 = torch.cat([up6,block6],1)
        up5 = self.up_conv5(cat4)
        cat3 = torch.cat([up5,block5],1)
        dec1 = self.dec_conv1(cat3)
        up4 = self.up_conv4(dec1)
        cat2 = torch.cat([up4,block3],1)
        dec2 = self.dec_conv2(cat2)
        up3 = self.up_conv3(dec2)
        cat1 = torch.cat([up3,conv1_act],1)
        up2 = self.up_conv2(cat1)
        up1 = self.up_conv1(up2)
        return up1

```

### 3 АУГМЕНТАЦИЯ ДАННЫХ

Аугментация данных (data augmentation) – это методика создания дополнительных обучающих данных из имеющихся. Для достижения хороших результатов глубокие нейронные сети должны обучаться на очень большом объеме данных. Следовательно, если исходный обучающий набор содержит ограниченное количество примеров, необходимо выполнить его аугментацию, чтобы улучшить результаты обучаемой модели.

При разработке моделей, осуществляющих аугментацию набора данных, состоящего из изображений, исследователи используют следующие случайные преобразования: поворот, кадрирование, масштабирование, сжатие и растяжение вдоль осей [17, 18, 19, 20], горизонтальные и вертикальные отражения [21], искажение, деформирование, зашумление, размытие, изменение яркости, насыщенности и контрастности изображения [22, 23]. Применение различных комбинаций перечисленных преобразований над изображениями из исходного набора данных позволяет значительно увеличить объем обучающей выборки.

В рамках данной работы предлагается алгоритмический подход к выполнению аугментации данных, а именно: способ деформирования и искажения объектов, определенных в  $N$ -мерном пространстве. В частности, рассматривается одномерная 1D-трансформация массивов данных, имеющих природу эквидистантных временных рядов, 2D-трансформация таких объектов как растровые изображения и 3D-трансформация объектов, представленных в объеме, например, облаком точек. Также описываемая ниже техника может быть распространена на искажения цветового пространства изображений, но рассмотрение данного вопроса выходит за рамки данной работы.

#### 3.1 Одномерная трансформация

Пусть имеется одномерный дискретный сигнал  $S(x) = (s_0, s_1, \dots, s_{N-1})$ ;  $x \in \mathbb{Z}$ , состоящий из  $N$  элементов. Каждый элемент  $S$  характеризует измерение, выполненное с фиксированным шагом в пространстве проводимого измерения относительно предыдущего элемента. Для реализации пространственных преобразований дискретного сигнала  $S$  построим его представление  $F(x)$ ;  $x \in \mathbb{R}$  в вещественном безразмерном пространстве.  $F(x)$  определим согласно (3.1):

$$F(x) = \begin{cases} s_0, & x \in (-\infty; 0,5) \\ s_1, & x \in [0,5; 1,5) \\ \dots & \\ s_{N-1}, & x \in [N - 1,5; +\infty) \end{cases} \quad (3.1)$$

В силу эквидистантной природы сигнала  $S$  приемлемы преобразования по изменению частоты его дискретизации. Для изменения частоты дискретизации  $S$  при формировании нового сигнала  $\hat{S}(x) = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{M-1})$ ;  $x \in \mathbb{Z}$  состоящего из  $M$  элементов, можно использовать формулу (3.2).

$$\hat{s}_i = F\left(\frac{N}{M} \cdot \left(i + \frac{1}{2}\right) - \frac{1}{2}\right), i \in \{0, 1, \dots, M - 1\} \quad (3.2)$$

где  $\hat{s}_i$  – значение очередного элемента в формируемом сигнале  $\hat{S}$ ;

$F(x)$  – представление исходного сигнала  $S$  в вещественном пространстве;

$N$  – количество элементов в исходном сигнале;

$M$  – количество элементов в формируемом сигнале.

В результате преобразования по формуле (3.2) сформированный сигнал  $\hat{S}$  полностью эквивалентен сигналу  $S$  при условии  $M = N$ ; имеет меньшую частоту дискретизации, а, следовательно, несет потерю информации при  $M < N$ ; и имеет большую частоту дискретизации, но не обладает большей информативностью при  $M > N$ .

Для выполнения аугментации одномерного сигнала  $S$ , а именно для осуществления случайного пространственного искажения сигнала  $S$ , внесем изменения в формулу (3.2). Примем границы области определения представления  $F$  за его истинные “опорные” точки  $P^F = \{P_0^F, P_1^F\}$ : левой опорной точкой считаем  $P_0^F = -0,5$  и правой  $P_1^F = N - 0,5$ . Внесем случайные изменения в  $P^F$ , определив новое множество опорных точек  $\widetilde{P}^F = P^F + \xi$ . Данное преобразование означает, что в окрестностях точек  $P_0^F$  и  $P_1^F$ , случайным образом внесены изменения, а именно  $\widetilde{P}_0^F = P_0^F + \xi$  и  $\widetilde{P}_1^F = P_1^F + \xi$ , где  $\xi$  является случайной величиной определяемой произвольным законом распределения. Заметим, что  $\widetilde{P}^F$  может определяться не только с помощью внесения случайных изменений. Целенаправленный выбор конкретных значений для  $\widetilde{P}^F$  может быть использован с целью “вырезания” интересующего фрагмента сигнала  $S$ . В свою очередь перестановка местами элементов  $\widetilde{P}^F$  может быть использована для выполнения операции зеркального отражения сигнала  $S$ .

Аугментацией сигнала  $S$  является сигнал  $\tilde{S}(x) = (\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{M-1})$ ,  $x \in \mathbb{Z}$ , состоящий из  $M$  элементов. Значения элементов  $\tilde{S}$  определяются по формуле (3.4). При расчете используется вспомогательная функция  $\phi$  оперирующая с интервалом  $[l; r]$ . Использование функции  $\phi$  позволяет определить координату точки, лежащей в центре отрезка с номером  $k$  внутри интервала  $[l; r]$  разделенного на  $N$  равных отрезков. Функция  $\phi$  определяется по формуле (3.3).

$$\phi(k, l, r, N) = l + \frac{r - l}{N} \cdot \left(k + \frac{1}{2}\right) \quad (3.3)$$

где  $\phi$  – вспомогательная функция;

$k$  – номер отрезка на интервале;

$l$  – координата левой границы интервала (левая опорная точка);

$r$  – координата правой границы интервала (правая опорная точка);

$N$  – количество отрезков на которое делится интервал.

$$\tilde{s}_i = F\left(\phi\left(i, \widetilde{P}_0^F, \widetilde{P}_1^F, M\right)\right), i \in \{0, 1, \dots, M - 1\} \quad (3.4)$$

где  $\tilde{s}_i$  – значение очередного элемента в формируемом сигнале  $\tilde{S}$ ;

$F(x)$  – представление исходного сигнала  $S$  в вещественном пространстве;

$\phi$  – вспомогательная функция;

$\widetilde{P}_0^F$  – левая опорная точка;

$\widetilde{P}_1^F$  – правая опорная точка;

$M$  – количество элементов в формируемом сигнале;

Заметим, что если вносимые случайные изменение  $\xi = 0$ , опорные точки сигнала не изменяются. В этом случае преобразование сигнала, выполняемое по формуле (3.4), эквивалентно преобразованию, изменяющему частоту дискретизации сигнала  $S$ , выполняемого по формуле (3.2).

В рассмотренном выше примере, аргументы  $l$  и  $r$ , передаваемые функции  $\phi$ , соответствуют координатам, определенным вдоль одной оси, но данное преобразование также применимо и для векторных вычислений, когда опорные точки делимого интервала определены в многомерном пространстве.

### 3.2 Двумерная трансформация

Растровое изображение представляется двумерной матрицей пикселей размера  $W \times H$ . Элементы изображения характеризуются двумя координатами  $(x, y)$  и значениями компонентов яркости  $C(x, y)$ . В описываемом подходе к трансформации изображений рассматриваются только искажения пространственных составляющих изображения. Пространственные преобразования не влекут за собой изменения компонентов яркости, в связи с этим количество цветовых компонент изображения не имеет значения. Фактически растровое изображение является двумерным дискретным сигналом  $S(x, y) = (s_{0,0}, s_{0,1}, \dots, s_{W-1,H-1})$ ;  $x, y \in \mathbb{Z}$  состоящим из  $W \cdot H$  элементов.

Для выполнения 2D-трансформации, как и в примере с 1D-трансформацией, перенесем двумерное растровое изображение из целочисленного пространства в вещественное, аналогичным образом построив отображение  $F(x, y); x, y \in \mathbb{R}$ . Далее, для простоты формулировок примем, что при обращении к  $F$  из аргументов  $x$  и  $y$  вычитается  $\frac{1}{2}$ . Тогда опорными точками  $F$  выступают границы области определения изображения, согласно (3.5).

$$P^F = \{P_{0,0}^F, P_{1,0}^F, P_{0,1}^F, P_{1,1}^F\} \\ P_{0,0}^F(0; 0); P_{1,0}^F(W; 0); P_{0,1}^F(0; H); P_{1,1}^F(W; H) \quad (3.5)$$

Как и раньше  $\tilde{P}^F = P^F + \xi$  определяет случайные изменения, вносимые в опорные точки  $P^F$ . Аугментацией растрового изображения (сигнала  $S$ ) является сигнал  $\tilde{S}(x, y) = (\tilde{s}_{0,0}, \tilde{s}_{0,1}, \dots, \tilde{s}_{\tilde{W}-1, \tilde{H}-1})$ ;  $x, y \in \mathbb{Z}$ , состоящий из  $\tilde{W} \cdot \tilde{H}$  элементов. При этом  $\tilde{W}$  и  $\tilde{H}$  определяют разрешение для формируемой аугментации. Расчет значений дискретного сигнала  $\tilde{S}$  выполняется согласно (3.6) или (3.7). Приведенные способы расчета значений дискретного сигнала  $S$  эквивалентны друг другу.

$$\tilde{P}_{x_0}^F(i) = \phi(i, \tilde{P}_{0,0}^F, \tilde{P}_{1,0}^F, \tilde{W}); \tilde{P}_{x_1}^F(i) = \phi(i, \tilde{P}_{0,1}^F, \tilde{P}_{1,1}^F, \tilde{W}); \\ \tilde{s}_{i,j} = F\left(\phi\left(j, \tilde{P}_{x_0}^F(i), \tilde{P}_{x_1}^F(i), \tilde{H}\right)\right) \quad (3.6)$$

$$\tilde{s}_{i,j} = F\left(\phi\left(j, \phi\left(i, \tilde{P}_{0,0}^F, \tilde{P}_{1,0}^F, \tilde{W}\right), \phi\left(i, \tilde{P}_{0,1}^F, \tilde{P}_{1,1}^F, \tilde{W}\right), \tilde{H}\right)\right) \\ i \in \{0, 1, \dots, \tilde{W} - 1\}; j \in \{0, 1, \dots, \tilde{H} - 1\}; \quad (3.7)$$

Процесс последовательного вычисления двумерной трансформации (аугментации)  $\tilde{S}$  из  $S$  проиллюстрирован на рисунке 3.1.

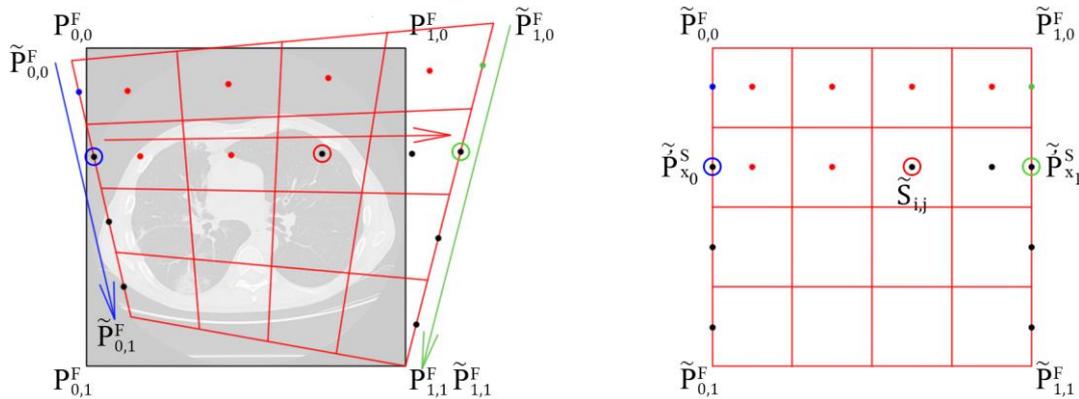


Рисунок 3.1 – Процесс выполнения 2D-трансформации

Описанный подход универсален тем, что он объединяет свойства, присущие комбинациям классических случайных преобразований двумерных изображений. Пример выполнения 2D-трансформации приведен на рисунке 3.2.

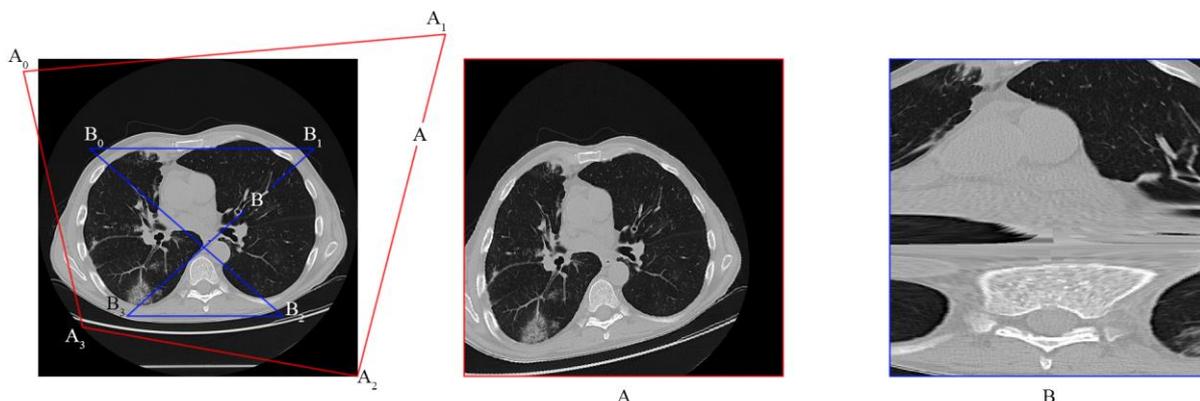


Рисунок 3.2 – Результат 2D-трансформации

При необходимости выполнения строгой операции поворота изображения, достаточно разместить опорные точки исходного изображения на описанной возле изображения окружности и рассчитать их пространственные координаты с учетом поворота по или против часовой стрелки. Для выполнения операции горизонтального или вертикального отражения достаточно изменить порядок следования опорных точек соответствующим образом. Для выполнения операции кадрирования достаточно сместить опорные точки по направлению к или от центра изображения.

### 3.3 Случайные искажения

Для внесения случайного изменения в опорные точки сигнала определенного в  $N$ -мерном пространстве определим, что вносимое изменение характеризуется случайным вектором  $\xi = (\xi_1, \dots, \xi_n)$ . Для реализации алгоритма выполнения аугментации обучающего набора данных предлагается в качестве случайного изменения, вносимого в опорные точки, использовать нормальное распределение. Тогда, если  $Z$  стандартная нормальная случайная величина, то  $\xi_i = \sigma_i \cdot Z$  будет иметь нормальное распределение с нулевым математическим ожиданием и стандартным отклонением  $\sigma$ . Иллюстрация выбора опорных точек с внесением случайного изменения приведена на рисунке 3.3.

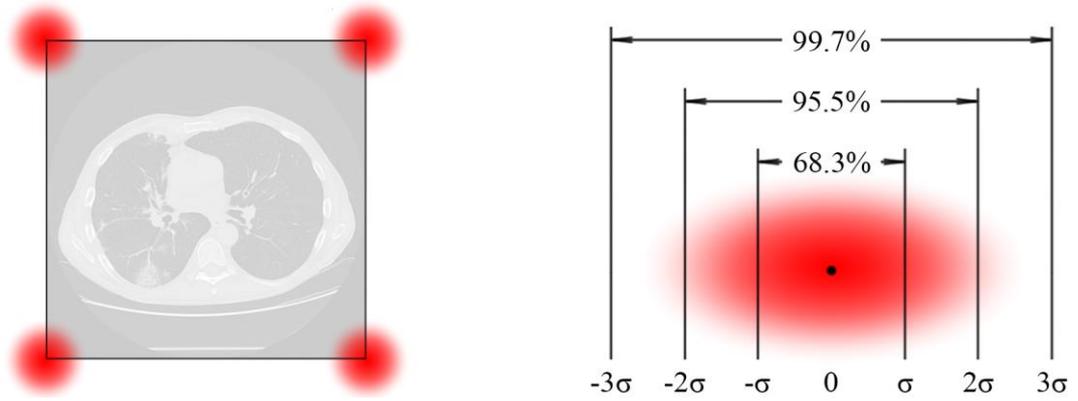


Рисунок 3.3 – Вероятностный выбор опорных точек для 2D-трансформации.

При реализации алгоритма аугментации графических наборов данных, выбор значения для параметра  $\sigma$  позволит управлять силой вносимых искажений. При использовании предложенного подхода, можно быть уверенным, что новые, случайно определенные опорные точки, с вероятностью 96 %, не отдалятся от исходной далее чем на  $2 \cdot \sigma$  и с вероятностью 68 % далее чем на  $\sigma$ .

### 3.4 Трехмерная трансформация

Биомедицинское растровое одноканальное изображение размером  $W \times H \times D$ , также как и раньше, может быть представлено в виде  $F(x, y, z); x, y, z \in \mathbb{R}$ .

Для реализации объемной 3D-трансформации определим опорные точки  $F$  являющиеся границами области определения изображения, согласно формуле (3.8).

$$\begin{aligned}
 P^F = \{ & P_{0,0,0}^F, P_{1,0,0}^F, P_{0,1,0}^F, P_{1,1,0}^F, P_{0,0,1}^F, P_{1,0,1}^F, P_{0,1,1}^F, P_{1,1,1}^F \}; \\
 & P_{0,0,0}^F(0; 0; 0); P_{1,0,0}^F(W; 0; 0); P_{0,1,0}^F(0; H; 0); P_{1,1,0}^F(W; H; 0); \\
 & P_{0,0,1}^F(0; 0; D); P_{1,0,1}^F(W; 0; D); P_{0,1,1}^F(0; H; D); P_{1,1,1}^F(W; H; D).
 \end{aligned} \quad (3.8)$$

Определим случайные изменения  $\widetilde{P}^F = P^F + \xi$  вносимые в опорные точки  $P^F$ .

Тогда расчет значений дискретного сигнала  $\widetilde{S}$  выполним по формуле (3.9).

$$\begin{aligned}
 \widetilde{P}_{x_0}^F(i) &= \phi(i, \widetilde{P}_{0,0,0}^F, \widetilde{P}_{1,0,0}^F, \widetilde{W}); \quad \widetilde{P}_{x_1}^F(i) = \phi(i, \widetilde{P}_{0,1,0}^F, \widetilde{P}_{1,1,0}^F, \widetilde{W}); \\
 \widetilde{P}_{x_2}^F(i) &= \phi(i, \widetilde{P}_{0,0,1}^F, \widetilde{P}_{1,0,1}^F, \widetilde{W}); \quad \widetilde{P}_{x_3}^F(i) = \phi(i, \widetilde{P}_{0,1,1}^F, \widetilde{P}_{1,1,1}^F, \widetilde{W}); \\
 \widetilde{P}_{y_0}^F(j) &= \phi(j, \widetilde{P}_{x_0}^F(i), \widetilde{P}_{x_1}^F(i), \widetilde{H}); \quad \widetilde{P}_{y_1}^F(j) = \phi(j, \widetilde{P}_{x_2}^F(i), \widetilde{P}_{x_3}^F(i), \widetilde{H}); \\
 \widetilde{s}_{i,j,k} &= F\left(\phi(k, \widetilde{P}_{y_0}^F(j), \widetilde{P}_{y_1}^F(j), \widetilde{D})\right) \\
 i \in \{0, 1, \dots, \widetilde{W} - 1\}; & j \in \{0, 1, \dots, \widetilde{H} - 1\}; k \in \{0, 1, \dots, \widetilde{D} - 1\};
 \end{aligned} \quad (3.9)$$

### 3.5 Реализация алгоритма аугментации данных

В рамках текущей работы предложенные алгоритмы реализованы на языке программирования Python с использованием библиотеки *numpy* [24]. Для работы с биомедицинскими изображениями, представленными в формате NIfTI, используется библиотека *nibabel* [25]. Подключение данных модулей приведено в листинге 3.1.

Листинг 3.1 – Файл ExGen.py. Подключение модулей *numpy* и *nibabel*

```
import numpy as np
import collections
import nibabel as nib
```

Функция *point\_iterator* работает с интервалом, определенном в n-мерном пространстве. Отрезок делится на *count* отрезков, координаты центральных точек данных отрезков записываются в результирующий массив. Другими словами, данная функция возвращает равномерно распределенные координаты точек на указанном интервале. Реализация функции *point\_iterator* приведена в листинге 3.2.

Листинг 3.2 – Файл ExGen.py. Функция *point\_iterator*

```
def point_iterator(start, end, count):
    #np.linspace works ~ 8 times faster
    start=np.array(start)
    end=np.array(end)
    C=(end-start)/(count)
    return start+np.array([C*(k+0.5) for k in range(count)])
```

Функция *topsIndexsByShape*, приведенная в листинге 3.3, вычисляет 8 координат вершин трехмерного массива данных на основе его размеров.

Листинг 3.3 – Файл ExGen.py. Функция *topsIndexsByShape*

```
def topsIndexsByShape(shape):
    shape=np.array(shape)
    points=[]
    if len(shape)==3:
        X,Y,Z=shape
        points=[[0,0,0],[X,0,0],[0,Y,0],[X,Y,0],
                [0,0,Z],[X,0,Z],[0,Y,Z],[X,Y,Z]]
    else:
        raise IOError("The processing form must have 3 dimensions")
    return np.array(points)
```

Функция *mixPoints*, приведенная в листинге 3.4, выполняет случайное зеркальное отражение опорных точек и используется во время генерации обучающих примеров.

Листинг 3.4 – Файл ExGen.py. Функция *mixPoints*

```
def mixPoints(p):
    if len(p)!=8: raise IOError("8 peaks expected")
    if np.random.rand()>0.5: p=np.array([p[4],p[5],p[6],p[7],p[0],p[1],p[2],p[3]])
    if np.random.rand()>0.5: p=np.array([p[2],p[3],p[0],p[1],p[6],p[7],p[4],p[5]])
    if np.random.rand()>0.5: p=np.array([p[1],p[0],p[3],p[2],p[5],p[4],p[7],p[6]])
    return p
```

Класс *AnchorPoints3DIterator* позволяет выполнять легкое итерирование по центральным точкам объемов, заключенных между узлами заданной сетки. При этом трехмерная сетка формируется по принципу, описанному в разделе, посвященном аугментации данных. Реализация класса *AnchorPoints3DIterator* приведена в листинге 3.5.

Листинг 3.5 – Файл ExGen.py. Класс *AnchorPoints3DIterator*

```
class AnchorPoints3DIterator(collections.abc.Iterator):
    """
        points[8]:=
            0-----2----
             /|      /|   Y  grid[1]
            / |     / |
            1-----3 |
             /| 4----|--6
            / | /    | /
            / 5-----7
        X /      |
         grid[0] | Z  grid[2]
    """

    def __init__(self, grid, points):
        self._p=np.array(points)
        self._g=np.array(grid)
        self._frnt_left=point_iterator(self._p[0],self._p[4],self._g[2])
        self._frnt_righ=point_iterator(self._p[1],self._p[5],self._g[2])
        self._rear_left=point_iterator(self._p[2],self._p[6],self._g[2])
        self._rear_righ=point_iterator(self._p[3],self._p[7],self._g[2])
        self._z_cur=-1; self.__next_lay()
        self._y_cur=-1; self.__next_line()
        self._x_cur=-1

    def __next_lay(self):
        self._y_cur=0
        self._z_cur+=1
        if self._z_cur>=self._g[2]:
            raise StopIteration()
        self._left=point_iterator(self._frnt_left[self._z_cur],
                                  self._rear_left[self._z_cur],self._g[1])
        self._righ=point_iterator(self._frnt_righ[self._z_cur],
                                   self._rear_righ[self._z_cur],self._g[1])

    def __next_line(self):
        self._x_cur=0
        self._y_cur+=1
        if self._y_cur>=self._g[1]:
            self.__next_lay()
        self._line=point_iterator(self._left[self._y_cur],
                                   self._righ[self._y_cur],self._g[0])
```

```

def __next__(self):
    self._x_cur+=1
    if self._x_cur>=self._g[0]:
        self.__next_line()
    return np.array([self._x_cur,self._y_cur,self._z_cur]), self._line[self._x_cur]

```

Класс *AnchorPoints3DIterator* позволяет реализовывать конструкцию вида:

```

for grid_index, point in AnchorPoints3DIteratorObject:

```

Данная конструкция позволяет получить на каждом шаге индекс внутри сетки, “натянутой” по произвольным опорным точкам, и самую координату точки, соответствующую данному индексу.

Функция *idx\_by\_grid\_and\_anchor\_points\_for\_shape*, приведенная в листинге 3.6, может использоваться для решения той же задачи, что и класс *AnchorPoints3DIterator*. Отличие заключается в том, что данная функция не позволяет выполнять поочередное итерирование, все координаты точек, соответствующие индексам сетки, вычисляются сразу.

Листинг 3.6 – Файл ExGen.py. Функция *idx\_by\_grid\_and\_anchor\_points\_for\_shape*

```

def idx_by_grid_and_anchor_points_for_shape(grid, points, origin_shape):
    ps = np.array(points,dtype=np.float32)
    g = np.array(grid)
    origin_shape = np.array(origin_shape)
    i = 0
    P = np.zeros((grid[0]*grid[1]*grid[2],3),dtype=np.int32)
    fl,s = np.linspace(ps[0],ps[1],g[0],endpoint=False,retstep=True)
    fl += s/2
    fr,s = np.linspace(ps[4],ps[5],g[0],endpoint=False,retstep=True)
    fr += s/2
    rl,s = np.linspace(ps[2],ps[3],g[0],endpoint=False,retstep=True)
    rl += s/2
    rr,s = np.linspace(ps[6],ps[7],g[0],endpoint=False,retstep=True)
    rr += s/2
    for x_cur in range(grid[0]):
        l,s=np.linspace(fl[x_cur],rl[x_cur],g[1],endpoint=False,retstep=True)
        l+=s/2
        r,s=np.linspace(fr[x_cur],rr[x_cur],g[1],endpoint=False,retstep=True)
        r+=s/2
        for y_cur in range(g[1]):
            tp, s = np.linspace(l[y_cur],r[y_cur],g[2],endpoint=False,retstep=True)
            P[i:i+g[2],:] = tp+s/2
            i += g[2]
    cond = (np.logical_and(P>=0, P<origin_shape).sum(axis=1)==3)
    P = P[cond,:].transpose()
    idx = np.meshgrid(range(grid[1]),range(grid[0]),range(grid[2]))
    IDX = np.array([idx[1].flatten(),idx[0].flatten(),idx[2].flatten()])[:,cond]
    return (IDX[0],IDX[1],IDX[2]),(P[0],P[1],P[2])

```

Функция *cutDataByAnchorPoints3D* вырезает трехмерный массив данных из входного трехмерного массива *data3d* по опорным точкам *points*, задающих искаженную сетку с заданной дискретизацией *grid*. Если опорные точки не заданы, то данная функция фактически выполняет передискретизацию трехмерного массива данных. Реализация данной функции приведена в листинге 3.7.

Листинг 3.7 – Файл ExGen.py. Функция *cutDataByAnchorPoints3D*

```
def cutDataByAnchorPoints3D(data3d, grid, points=[], fill=None):
    X, Y, Z=np.array(data3d.shape[0:3])
    if len(points)!=8:
        points=topsIndexsByShape([X,Y,Z])
    ans=None
    if len(data3d.shape)==3:
        ans=np.zeros_like(data3d, shape=(grid[0],grid[1],grid[2]))
    elif len(data3d.shape)>3:
        ans=np.zeros_like(data3d, shape=(grid[0],grid[1],grid[2],data3d.shape[3]))
    if fill!=None:
        ans[:, :, :]=fill
    idx,points=idx_by_grid_and_anchor_points_for_shape(grid=grid,
                                                       points=points, origin_shape=data3d.shape)

    ans[idx]=data3d[points]
    return ans
```

Вспомогательный класс *Conversion*, приведенный в листинге 3.8, хранит обратный коэффициент масштаба.

Листинг 3.8 – Файл ExGen.py. Класс *Conversion*

```
class Conversion:
    def __init__(self,base):
        self.base=np.array(base)
    def __call__(self, value):
        return np.array(value)/self.base
```

Класс *ExampleGenerator* позволяет генерировать случайным образом обучающие примеры из размеченных экспертом биомедицинских изображений. Конструктор генератора принимает путь к файлу с изображением и к файлу с разметкой, размер вырезаемого объема, размеры сетки, требуемые для нейронной сети, и коэффициент искажения. Реализация класса *ExampleGenerator* приведена в листинге 3.9.

Так как у биомедицинской аннотации есть особенность, заключающаяся в том, что объем сегментации может занимать менее 5 % объема всего изображения, объект класса *ExampleGenerator* хранит в себе представление о вероятностном распределении сегментации на снимке. Это позволяет управлять вероятностью предъявления обучающих примеров с содержанием разметки.

Во время генерации обучающего примера *ExampleGenerator* выполняет аугментацию данных согласно заданным параметрам. Каждый новый сгенерированный пример уникален.

Листинг 3.9 – Файл ExGen.py. Класс *ExampleGenerator*

```

class ExampleGenerator:
    def __init__(self, input_path, response_path, volume, grid, distortion):
        i = nib.load(input_path)
        o = nib.load(response_path)
        if i.header.get_zooms() != o.header.get_zooms():
            raise IOError("масштаб разметки не соответствует входному файлу")
        if i.shape != o.shape:
            raise IOError("размер разметки не соответствует входному файлу")
        zooms = np.array(i.header.get_zooms())
        size = np.array(i.shape)*zooms
        self.__Convrtter = Conversion(zooms)
        self.__volume = np.array(volume)
        self.__grid = np.array(grid)
        self.__distortion = distortion
        self.i = i.get_fdata().astype(np.float32)
        self.o = o.get_fdata().astype(np.float32)
        self.fill = self.i.min()
        self.HUmin = -1100
        self.HUmax = 600
        self.__basis=topsIndexsByShape((1,1,1))-0.5
        tgrid=np rint(size/self.__grid*4).astype(int)
        titer=AnchorPoints3DIterator(points=topsIndexsByShape(self.i.shape),grid=tgrid)
        self.__pos_points=[]
        self.__neg_points=[]
        for _,p in titer:
            idx=tuple(np rint(p).astype(int))
            if self.i[idx]<self.fill+1:
                continue
            if self.o[idx]>0.5:
                self.__pos_points.append(idx)
            else:
                self.__neg_points.append(idx)

    def __call__(self, prob):
        v=self.__pos_points
        if np.random.rand()>prob:
            v=self.__neg_points
        i=np.random.randint(len(v))
        return self.__at_point(v[i])

    def __at_point(self, cpoint):
        P = self.__basis*self.__volume
        p += np.random.normal(0, self.__distortion/2,8*3).reshape(8,3)
        p += np.random.normal(0, 0.5, 3)*self.__volume/4
        p = self.__Convrtter(p)
        p += cpoint # смещаем куб в точку с заданным центром
        p = mixPoints(p)
        ri = cutDataByAnchorPoints3D(self.i,self.__grid,p,self.fill)
        ri = (ri-self.HUmin)/(self.HUmax-self.HUmin)
        ro = cutDataByAnchorPoints3D(self.o,self.__grid,p,0)
        zoom = self.__volume/self.__grid
        return ri, ro, zoom

```

В контексте обучения нейронной сети класс *ExampleGenerator* может использоваться по-разному. С одной стороны, его можно использовать для подготовки обучающего набора в виде большого количества сохраненных файлов. Такой подход плох тем, что при долгом обучении нейросеть так или иначе обработает одни и те же примеры неоднократно. С другой стороны, его можно использовать непосредственно в момент формирования батча для очередной эпохи, такой подход гарантирует, что нейросеть всегда будет встречаться с уникальными примерами.

Генерация батча перед очередной эпохой требует значительного процессорного времени, что существенно замедляет обучение. Алгоритм генерации батча перед новой эпохой технически возможно реализовать таким образом, что генерация будет проходить в то время, когда GPU занято обучением ИНС. Разработка такого алгоритма выходит за рамки данной работы. В данной работе тестировался подход с генерацией батча перед очередной эпохой. Но по причине неэффективности предложенного алгоритма, принято решение генерировать обучающий набор данных, состоящий из большого количества примеров (более 10 тысяч), непосредственно перед обучением.

#### 4 АЛГОРИТМ ЛОКАЛИЗАЦИИ ОБЛАСТЕЙ ИНФИЛЬТРАТА

Тестирование предложенного нейросетевого подхода сегментации медицинских КТ-снимков, производится на наборе данных «MosMedData: COVID19\_1110», собранном и опубликованном «Научно-практическим клиническим центром диагностики и телемедицинских технологий Департамента здравоохранения города Москвы» [26]. Данный набор данных (датасет) содержит результаты компьютерной томографии органов грудной клетки с рентгенологическими признаками вирусной пневмонии (COVID-19), а также без признаков (норма). Датасет содержит 1110 КТ снимков в формате NIfTI, среди них 49 (с диагностированной первой стадией заболевания, КТ-1) имеют бинарную пиксельную маску разметки для областей интереса (зон уплотнений по типу «матового стекла» и консолидации). Также датасет имеет распределение на 5 классов, со следующим описанием каждого класса:

- КТ-0 (СТ-0): норма и отсутствие КТ-признаков вирусной пневмонии;
- КТ-1 (СТ-1): зоны уплотнения по типу «матового стекла». Вовлечение паренхимы легкого  $\leq 25\%$ ;
- КТ-2 (СТ-2): зоны уплотнения по типу «матового стекла». Вовлечение паренхимы легкого = 25 – 50 %;
- КТ-3 (СТ-3): зоны уплотнения по типу «матового стекла» и консолидации. Вовлечение паренхимы легкого = 50 – 75 %;
- КТ-4 (СТ-4): Диффузное уплотнение легочной ткани по типу «матового стекла» и консолидации в сочетании с ретикулярными изменениями. Вовлечение паренхимы легкого  $\geq 75\%$ .

Распределение примеров, представленных в наборе данных, по классам приведено в таблице 4.1. Пример экспертной сегментация для одного файла из набора данных показан на рисунке 4.1.

Таблица 4.1 – Распределение примеров по классам

Класс	Количество примеров
КТ-0 (СТ-0)	254
КТ-1 (СТ-1)	684
КТ-2 (СТ-2)	125
КТ-3 (СТ-3)	45
КТ-4 (СТ-4)	2

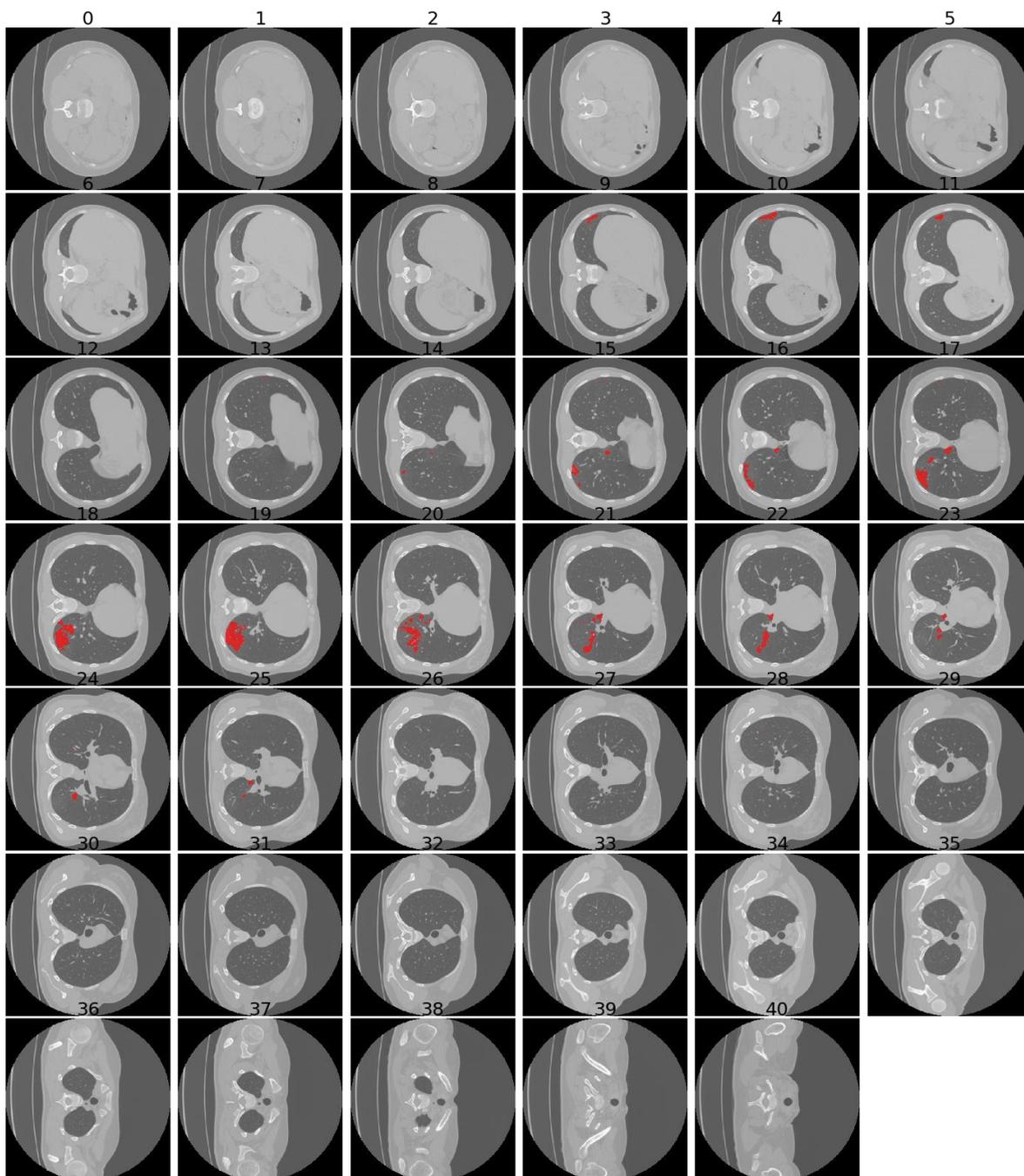


Рисунок 4.1 – Пример КТ снимка из аннотированной экспертом части набора данных (красным цветом выделена сегментация областей интереса, выполненная экспертом)

#### 4.1 Используемые функционалы качества в оценке нейросетевой сегментации

Для оценки результатов сегментации в данной работе используются различные метрики. Они вычисляются на основе матрицы ошибок бинарной классификации. Матрица ошибок – это таблица размером  $2 \times 2$ , где столбцы соответствуют фактическим значениям, а строки соответствуют спрогнозированным. Матрица ошибок приведена в таблице 4.2.

Таблица 4.2 – Матрица ошибок в задачах бинарной классификации

		<b>True condition</b> Истинное состояние	
		<b>Condition positive (P)</b> Истинное состояние положительно	<b>Condition negative (N)</b> Истинное состояние отрицательно
<b>Predicted condition</b> Предсказанное состояние	<b>Total population</b> Любое состояние		
	<b>Predicted condition Positive</b> Предсказано положительное состояние	<b>True Positive (TP), hit</b> Истинно-положительное решение	<b>False Positive (FP), false alarm</b> Ложно-положительное решение ошибка 1-го рода
	<b>Predicted condition negative</b> Предсказано отрицательное состояние	<b>False Negative (FN), miss</b> Ложно-отрицательное решение ошибка 2-го рода	<b>True Negative (TN), correct rejection</b> Истинно-отрицательное решение

Если ИНС предсказывает наличие разметки в вокселе, и при этом этот же воксел отмечен экспертом как относящийся к области интереса, то значение истинно-положительно *TP* (True Positive). В случае если ИНС предсказывает разметку, а экспертная разметка отсутствует, то значение истинно-отрицательное *FP* (False Positive). Если ИНС предсказывает отсутствие разметки в вокселе, а эксперт отмечает наличие разметки в вокселе, то значение ложно-отрицательное *FN* (False Negative). Если же ИНС предсказывает отсутствие разметки в вокселе, и экспертная разметка отсутствует в вокселе, то значение истинно-отрицательное *TN* (True Negative). На основе данных матрицы ошибок вычисляются следующие показатели качества классификации.

Точность (Precision, positive predictive value, *PPV*) показывает, сколько из предсказанных положительных примеров оказались действительно положительными. Таким образом, точность – это доля истинно положительных примеров от общего количества предсказанных положительных примеров, она используется в качестве показателя качества модели, когда цель состоит в том, чтобы снизить количество ложно-положительных примеров (то есть ошибку I рода). Значение *Precision* вычисляется по формуле (4.1).

$$Precision = PPV = \frac{TP}{TP + FP} \quad (4.1)$$

Полнота (Recall, sensitivity, hit rate, true positive rate, *TPR*) показывает, сколько от общего числа фактических положительных примеров было предсказано как положительный класс. Полнота – это доля истинно положительных примеров от общего количества фактических положительных примеров. Полнота используется в качестве показателя качества модели, когда необходимо определить все положительные примеры,

то есть, когда важно снизить количество ложно-отрицательных примеров (снизить ошибку II рода). Значение *Recall* вычисляется по формуле (4.2).

$$Recall = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \quad (4.2)$$

F-мера (F-measure,  $F_1$  score) позволяет обобщить точность и полноту. F-мера представляет собой гармоническое среднее точности и полноты. F-мера вычисляется по формуле (4.3).

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (4.3)$$

Выпадение (Fall-out, false alarm rate, false positive rate, *FPR*) показывает, сколько от общего числа фактических отрицательных примеров было предсказано как положительный класс. Выпадение – это доля ложно-положительных примеров от общего количества фактических отрицательных примеров. Значение *Fall\_out* вычисляется по формуле (4.4).

$$Fall\_out = FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \quad (4.4)$$

Intersection over Union (*IoU*, Jaccard index, коэффициент Жаккарда) – метрика, используемая в задачах сегментации. *IoU* определяется как отношение пересечения к объединению двух множеств [27]. Данная метрика отображает бинарную меру сходства между анализируемыми множествами. Аналогичными свойствами обладает метрика известная как коэффициент Дайса (Dice). Увеличение значения коэффициента Дайса как и *IoU* может свидетельствовать о увеличившейся схожести двух множеств. Расчет данных значений выполняется по формулам (4.5).

$$J = \frac{TP}{TP + FN + FP}; D = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = F_1 \quad (4.5)$$

где  $J$  – коэффициент Жаккарда (*IoU*);

$D$  – коэффициент Дайса.

Из приведенных формул (4.3) и (4.5) видно, что значение коэффициента Дайса это и есть F-мера, следовательно для F-меры характерны те же свойства, что и для *IoU*.

ROC-кривая (ROC-curve) - график, который позволяет оценить качество бинарной классификации. График показывает зависимость полноты (*TPR*) от выпадения (*FPR*) при варьировании порога. В точке (0,0) порог минимален, точно так же минимальны *TPR* и *FPR*, в свою очередь в точке (1,1) порог, *TPR* и *FPR* максимальны. Идеальным случаем для

бинарного классификатора является случай, когда ROC-кривая проходит через точку (0,1). Очевидно, что график данной функции всегда монотонно не убывает. Пример ROC-кривой приведен на рисунке 4.2.

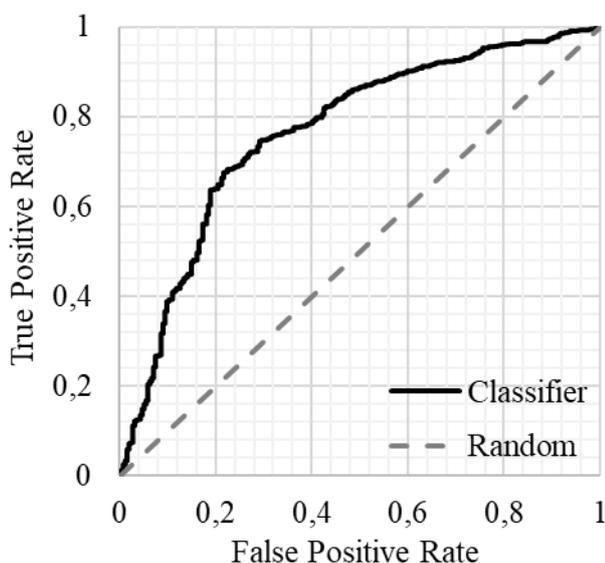


Рисунок 4.2 – Пример ROC-кривой для произвольного бинарного классификатора

Площадь под кривой (Area under curve,  $AUC$ ) – площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций. Чем выше показатель  $AUC$ , тем качественнее классификатор. При  $AUC = 0.5$  анализируемый классификатор эквивалентен случайному угадыванию.

#### 4.2 Реализация алгоритма обучения

Процесс обучения нейронной сети реализован в виде отдельной функции, для которой нужно задать *epochs* - число эпох обучения, *number\_of\_mini\_epoch* – число подэпох (мини-эпох), *number\_of\_example* – число генерируемых примеров для каждой эпохи, *batch\_size* – размер батча, *num\_workers* – число потоков загрузчика данных, *lr* – скорость обучения, *loss\_history* – словарь, в который добавляется значение ошибки обучения, *prob\_positive\_gen* – вероятность генерации положительного примера, другими словами, параметр регулирует частоту генерации обучающих примеров, которые содержат целевую сегментацию (если значение равно 1, то все примеры будут вырезаны из области КТ содержащих инфильтрат), *weight\_decay* – параметр L2 регуляризации сети.

Для начала обучения необходимо инициализировать нейронную сеть с заданными размерами входной сетки *grid* и вырезаемого объема *volume* из КТ-снимка. Данные значения выбираются согласно параметрам предварительно сгенерированного обучающего

набора данных. Для генерации обучающего набора случайным образом берутся искаженные фрагменты обучающих КТ-снимков пациентов в nii-файлах с заданными размерами (*grid*) и объемом (*volume*) и соответствующие им сегментации.

Процедура обучения нейронной сети начинается с инициализации оптимизатора и критерия оптимизации (функция ошибки). В качестве алгоритма оптимизации использовался метод оптимизации Adam [28]. Цель оптимизатора – минимизировать значение функции потерь. В данной работе используется модернизация (4.5) в виде непрерывной и всюду дифференцируемой функции потерь – *DiceLoss*. Реализация класса *DiceLoss* приведена в листинге 4.1.

Листинг 4.1 – Файл *ct\_lossFunctions.py*. Класс *DiceLoss*

```
class DiceLoss(nn.Module):
    def __init__(self): super().__init__()
    def forward(self, prediction, target):
        assert prediction.size() == target.size()
        p = prediction.view(-1)
        t = target.view(-1)
        i = (p * t).sum()
        u = ((1 - p) * t).sum() + (p * (1 - t)).sum() + i + 1e-6
        return 1 - i / u
```

В начале каждой эпохи открывается *number\_of\_example* обучающих примеров из предварительно сгенерированного набора данных. Далее, данные примеры предъявляются нейронной сети несколько раз (*number\_of\_mini\_epoch*). Во время каждой мини-эпохи выполняется шаг градиентного спуска, который обновляет скрытые параметры (веса) нейронной сети. Для каждой глобальной эпохи обучения рассчитываются среднее значения функции ошибки для обучающих и валидационных примеров.

Алгоритм обучения реализован в листинге 4.2 в возвращаемой функции замыкания *trainer* из метода *Get\_NewNetwork\_and\_Trainer*, который принадлежит классу *CT\_Network\_Creater*. Реализация класса *CT\_Network\_Creater* не приводится.

Листинг 4.2 – Файл *cnn\_ct.py*. Метод *Get\_NewNetwork\_and\_Trainer*

```
def Get_NewNetwork_and_Trainer(self):
    net = Conv_3D().to(self.device)
    def trainer(epochs, number_of_mini_epoch, number_of_example, batch_size,
                num_workers, lr, loss_history, prob_positive_gen, weight_decay=0):
        optimizator = optim.Adam(net.parameters(), lr = lr,
                                  weight_decay = weight_decay)
        criterion = DiceLoss().to(self.device)
        validation = self.Get_test_loader(batch_size)
        dataloader = self.Get_train_loader(batch_size)
        for epoch in range(epochs):
            loss_on_batch = np.array([])
            for mini_epoch in range(number_of_mini_epoch):
```

```

for inpt_, target_ in dataloader:
    inpt = inpt_.to(self.device)
    target = target_.to(self.device)
    optimizer.zero_grad()
    out = net(inpt)
    loss = criterion(out, target)
    loss_on_batch=np.append(loss_on_batch,loss.item())
    loss.backward()
    optimizer.step()
loss_on_batch=loss_on_batch.mean()
loss_history['train'].append(loss_on_batch)
loss_on_epoch_val=[]
for validation_inpt, validation_target in validation:
    validation_inpt = validation_inpt.to(self.device)
    validation_target = validation_target.to(self.device)
    validation_out = net(validation_inpt)
    validation_loss = criterion(validation_out, validation_target)
    del validation_out
    loss_on_epoch_val.append(validation_loss.item())
loss_on_epoch_val=np.array(loss_on_epoch_val).sum()/len(loss_on_epoch_val)
print('Epoch:{}\t Loss: {}'.format(epoch+1, loss_on_epoch_val))
loss_history['validation'].append(loss_on_epoch_val)
del inpt, target, out, validation_inpt, validation_target, \
    loss_on_batch, loss_on_epoch_val
gc.collect()
return
return net, trainer

```

Тестовый запуск обучения нейронной сети выполнен в два этапа по 400 эпох. Выбраны следующие параметры обучения:  $grid = (64, 64, 32)$ ;  $volume = (50, 50, 60)$ ;  $number\_of\_mini\_epoch = 5$ ;  $lr = 0,0005$ ;  $batch\_size = 8$ ;  $number\_of\_example = 40$ ;  $prob\_positive\_gen = 0,4$ . Динамика изменения ошибки приведена на рисунке 4.3.

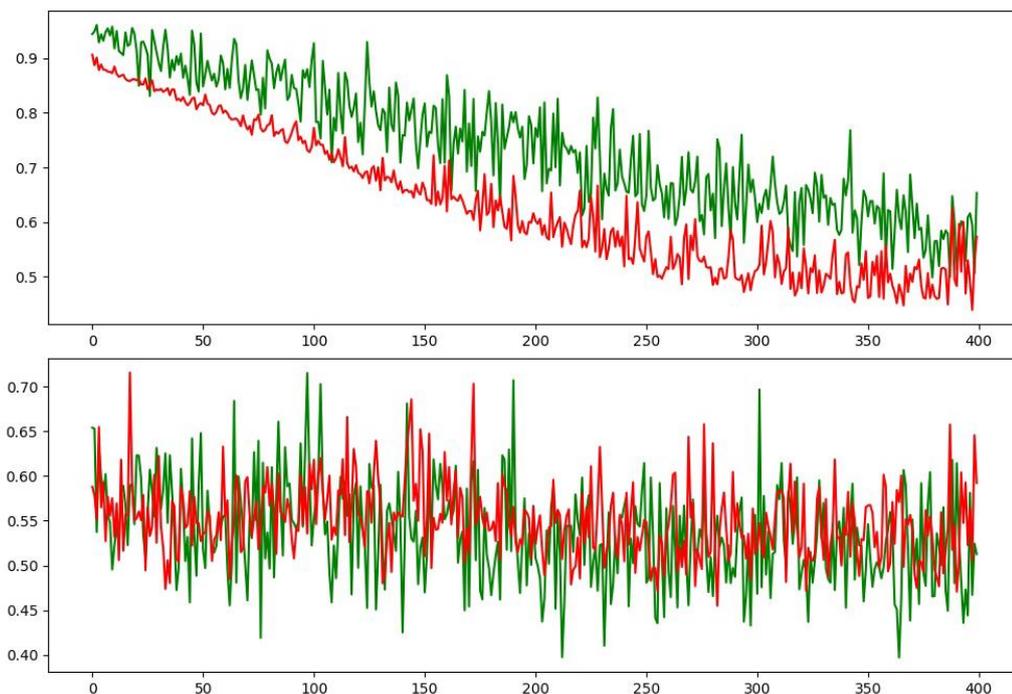


Рисунок 4.3 – Динамика изменения значения функции ошибки для тестового запуска. сверху – первый запуск обучения, снизу – дообучение ИНС. Красный – ошибка обучающего набора, Зеленый – ошибка валидационного набора.

### 4.3 Обработка скользящим окном

Обработка биомедицинского изображения скользящим окном позволяет опираться на информацию, предоставляемую в заголовках файла с изображением. Биомедицинские изображения такие как снимки компьютерной томографии всегда сопровождаются информацией о пространственных характеристиках объекта исследования, представленных в метрической системе координат. В контексте исследовательской работы по обработке медицинских изображений тестируемым алгоритмам предстоит обрабатывать фрагменты изображения, ограниченные объемом, имеющим фиксированные параметры по длине, ширине и высоте в метрической системе координат и дискретизируемые по объемной сетке, имеющей фиксированные размеры.

*Примечание* – Далее по тексту для обозначения метрических параметров фрагмента изображения таких как длина, ширина и высота в метрической системе координат используется слово “объем”. Для обозначения параметров обрабатываемого фрагмента изображения используется слово “сетка”. Объем является совокупностью трех параметров, измеряемых в миллиметрах, в свою очередь сетка описывается тремя безразмерными параметрами. Отображение фрагмента изображения, заключенного внутри объема по сетке, формирует обрабатываемый объект - “куб”, вне зависимости от того имеет ли в реальности объем или сетка форму параллелепипеда. Также любая арифметическая операция, применяемая в отношении объема или сетки, подразумевает ее применение поэлементно к каждому параметру. Например, увеличение объема в два раза подразумевает, что все параметры, характеризующие объем, увеличиваются вдвое, в реальности такое преобразование влечет численное увеличение классически понимаемого объема в восемь раз.

Для реализации эффективного по скорости алгоритма обработки трехмерного изображения скользящим окном, в первую очередь, предлагается масштабировать изображение до такого разрешения, чтобы вырезаемый по сетке окна куб соответствовал объему данного окна. Для масштабирования объемного изображения используется функция *resizeNDArray*. Функция *resizeNDArray*, приведенная в листинге 4.3, выполняет передискретизацию (масштабирование) n-мерного массива *data* к размерам указанным в аргументе *size*. Пример использования данной функции приведен в листинге 4.4. Данная функция реализована для упрощения дальнейшей обработки изображения. Заметим, что изображение, подлежащее обработке скользящим окном, описывается объемом не кратным объему окна. Так как область интереса в контексте нейросетевой обработки КТ-снимков располагается близко к центру изображения, а области близкие к

краям изображения не содержат полезной информации и подразумевается, что объем окна существенно меньше объема изображения – разумно расположить объем области, подлежащей обработке, внутри объема изображения.

Листинг 4.3 – Файл `segmentation.py`. Функция `resizeNDArray`

```
def resizeNDArray(data, size):
    data=np.array(data)
    size=np.array([int(size)] \
        if type(size) in [int, float] else \
        np.array(list(size),dtype=int)
    assert(len(data.shape) == len(size)),\
        str("Ожидалась "+str(len(data.shape))+ "D сетка")
    def linespace(data_dim, ans_dim):
        points, step = np.linspace(0, data_dim, num = ans_dim, endpoint=False,
retstep=True)
        return (points+step/2).astype(int)
    t=[linespace(data_dim, ans_dim) \
        for data_dim, ans_dim in zip(data.shape, size)]
    data_pos=tuple([f.flatten() for f in np.meshgrid(*t)])
    ans_pos= tuple([f.flatten() for f in \
        np.meshgrid(*[range(ans_dim) for ans_dim in size])])
    ans=np.zeros_like(data, shape=size)
    ans[ans_pos]=data[data_pos]
    return ans
```

Листинг 4.4 – Файл `segmentation.py`. Примеры использования функции `resizeNDArray`

```
>>> resizeNDArray([1, 2, 3, 4, 5, 6, 7, 8], 4.0)
array([2, 4, 6, 8])
>>> resizeNDArray([[1, 2, 3, 4], [5, 6, 7, 8]], (3, 12))
array([[1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4],
       [5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8],
       [5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8]])
>>> resizeNDArray([[[1, 2], [3, 4]], [[5, 6],[7,8]]],(3,2,5))
array([[1, 1, 2, 2, 2],
       [3, 3, 4, 4, 4]],
       [[5, 5, 6, 6, 6],
       [7, 7, 8, 8, 8]],
       [[5, 5, 6, 6, 6],
       [7, 7, 8, 8, 8]])
```

В связи с этим нет необходимости масштабировать все изображение, масштабированию должен быть подвергнут только объем подлежащий обработке. Также важно учитывать, что в ситуации, когда шаг окна по каждой оси не равен соответствующему параметру объема окна - объем максимально возможной области, подлежащей обработке, вписанной внутри объема изображения не кратен объему окна. Далее приводится реализация алгоритмов, выполняющих обработку изображений скользящим окном в двух режимах: с единичным шагом и половинным шагом относительно размеров окна. Алгоритм обработки с шагом равным соответствующим размерам окна по всем направлениям *segmentationOne* приведен в листинге 4.5. Алгоритм

обработки с шагом равным половине соответствующих размеров окна по всем направлениям *segmentationHalf* приведен в листинге 4.6.

Листинг 4.5 – Файл *segmentation.py*. Функция *segmentationOne*

```
def segmentationOne(input_path, output_path, volume, grid, func,
norm_function=None):
    volume=np.array(volume) # Обрабатываемый объем
    grid=np.array(grid) # Сетка
    i=nib.load(input_path) #Открываем NIfTI файл
    affine=i.header.get_best_affine()
    zooms=np.array(i.header.get_zooms()) #Воксел в мм
    convrter=Conversion(zooms)
    size=np.array(i.shape)*zooms #изображение в мм
    data=i.get_fdata().astype(np.float32)
    fill=data.min() #минимальное значение в данных
    m_grid=np.floor(size/volume).astype(np.int32)
    #Отступ для отцентровки обрабатываемого объема
    padding=convrter((size-m_grid*volume)/2).astype(int)
    p1=padding
    p2=(padding+convrter(m_grid*volume)).astype(int)
    zone=data[p1[0]:p2[0],p1[1]:p2[1],p1[2]:p2[2]]
    ro_shape=zone.shape
    ri=resizeNDArray(zone, m_grid*grid)
    del zone
    HUmin=-1100; HUmax=600
    ri-=HUmin
    ri/=(HUmax-HUmin)
    tens,deslicer=slicer(ri,grid)
    del ri
    ro=deslicer(func(tens))
    del tens, deslicer
    if norm_function!=None:
        ro=norm_function(ro)
    zone=resizeNDArray(ro, ro_shape)
    del ro
    ans=np.zeros_like(data) #массив для ответа
    ans[p1[0]:p2[0],p1[1]:p2[1],p1[2]:p2[2]]=zone
    del zone
    o=nib.Nifti1Image(ans, affine)
    nib.save(o,output_path)
    del ans, o
    return
```

Листинг 4.6 – Файл *segmentation.py*. Функция *segmentationHalf*

```
def segmentationHalf(input_path, output_path, volume, grid, func,
norm_function=None):
    volume=np.array(volume) # Обрабатываемый объем
    grid=np.array(grid) # Сетка
    i=nib.load(input_path) #Открываем NIfTI файл
    affine=i.header.get_best_affine()
    zooms=np.array(i.header.get_zooms()) #Воксел в мм
    convrter=Conversion(zooms)
    size=np.array(i.shape)*zooms #изображение в мм
    data=i.get_fdata().astype(np.float32)
    fill=data.min() #минимальное значение в данных
    m_grid=np.floor(size/(volume/2)).astype(np.int32)
    #Отступ для отцентровки обрабатываемого объема
    padding=convrter((size-m_grid*(volume/2))/2).astype(int)
```

```

p1=padding
p2=(padding+convrter(m_grid*(volume/2))).astype(int)
zone=data[p1[0]:p2[0],p1[1]:p2[1],p1[2]:p2[2]]
zone_shape=zone.shape
ri=resizeNDArray(zone, (m_grid*(grid/2)).astype(int))
ri_shape=np.array(ri.shape)
del zone
HUmin=-1100; HUmax=600
ri-=HUmin
ri/=(HUmax-HUmin)
ro=np.zeros_like(ri)
window_padding=np.array([f.flatten() for f in np.meshgrid(*[range(2)\
    for _ in range(3)])]).transpose()*(np.array(grid)/2).astype(np.int)
for wp1 in window_padding:
    w_grid=((ri_shape-wp1)/grid).astype(int)
    wp2=wp1+w_grid*grid
    w_ri=ri[wp1[0]:wp2[0],wp1[1]:wp2[1],wp1[2]:wp2[2]]
    tens,deslicer=slicer(w_ri,grid)
    del w_ri
    ro[wp1[0]:wp2[0],wp1[1]:wp2[1],wp1[2]:wp2[2]]+= \
        deslicer(func(tens))
    del tens, deslicer
ro=ro/8 #нормализация ro
if norm_function!=None:
    ro=norm_function(ro)
zone=resizeNDArray(ro, zone_shape)
del ro
ans=np.zeros_like(data) #массив для ответа
ans[p1[0]:p2[0],p1[1]:p2[1],p1[2]:p2[2]]=zone
del zone
o=nib.Nifti1Image(ans, affine)
nib.save(o,output_path)
del ans, o
return

```

Перед подачей значений плотности фрагментов в *func*, алгоритм выполняет нормализацию их значений по диапазону от минус 1100 до плюс 600 по шкале Хаунсфилда, тем самым приводя измерения плотностей в шкалу от 0 до 1 [29].

Функции *segmentationOne* и *segmentationHalf* обрабатывают КТ-снимки, представленное в формате NIFTI (расширение \*.*nii*), путь к которому указывается в первом аргументе *input\_path* [7]. Для чтения и сохранения файлов в формате NIFTI используется сторонняя библиотека *nibabel* [25]. Обработка трехмерного изображения скользящим окном выполняется путем разбиения входного изображения на фрагменты (кубы), соответствующие заданному объему *volume* и сетке *grid* передаваемых функции. Обработка кубов выполняется функцией, передаваемой по ссылке в аргументе *func*. В результате обработки основным алгоритмом, представленным в *func*, выполняется финальная обработка функцией *norm\_function*, также передаваемой в аргументах. В итоге функции сохраняют результат обработки в файл, представленный в формате NIFTI, по пути указанном в аргументе *output\_path*.

Подразумевается, что, функция *func*, ответственная за выполнение обработки фрагментов изображения, принимает на вход массив, состоящий из трехмерных кубов, подлежащих обработке. Данный массив представляется в виде четырех мерного объекта, имеющего тип данных *torch.tensor*. За разбиение обрабатываемой области на кубы отвечает функция *slicer*, исходной код которой приведен в листинге 4.7. За приведение результатов обработки к исходному виду отвечает функция замыкания *deslicer*, возвращаемая *slicer*-ом.

Листинг 4.7 – Файл *segmentation.py*. Функции *slicer* и *deslicer*

```
def slicer(data,grid):
    f=np.array(data.shape) #Форма массива
    g=np.array(grid)      #Форма фрагмента
    assert(len(f)==3), "Входной массив не 3d"
    assert(len(g)==3), "Сетка не 3d"
    assert(bool((f%g==0).prod())), "Входные пропорции не кратны сетке"
    N=int((f/g).prod()) #Количество фрагментов
    ans=torch.zeros(size=tuple(np.append(N,grid)))
    G=(f/g).astype(int) #Сетка фрагментов
    mg=np.meshgrid(*[range(r) for r in list(G)])
    idxlist=np.array([f.flatten() for f in mg]).transpose()
    idx=0
    for i in idxlist:
        p1=i*g #координата начала фрагмента
        p2=p1+g #координата конца фрагмента
        ans[idx]=torch.tensor(data[p1[0]:p2[0],p1[1]:p2[1],p1[2]:p2[2]])
        idx+=1
    def deslicer(dedata):
        idx=0
        ans=np.zeros(f)
        for i in idxlist:
            p1=i*g; p2=p1+g
            ans[p1[0]:p2[0],p1[1]:p2[1],p1[2]:p2[2]]=dedata[idx].numpy()
            idx+=1
        return ans
    return ans, deslicer
```

Основное отличие функции *segmentationHalf* от функции *segmentationOne* заключается в том, что фактически обработка вокселей, лежащих внутри обрабатываемой области, выполняется восемь раз. Это происходит за счет того, что при смещении окна с половинным шагом - области несмежные с границей обрабатываемой зоны оцениваются несколько раз. Данный эффект проиллюстрирован на рисунке 4.4 для двумерного случая.

Использование данного подхода позволяет снизить вероятность появления резких “скачков” в сегментации на границе обрабатываемого окна и получать более стабильные результаты нейросетевого сегментирования в целом.

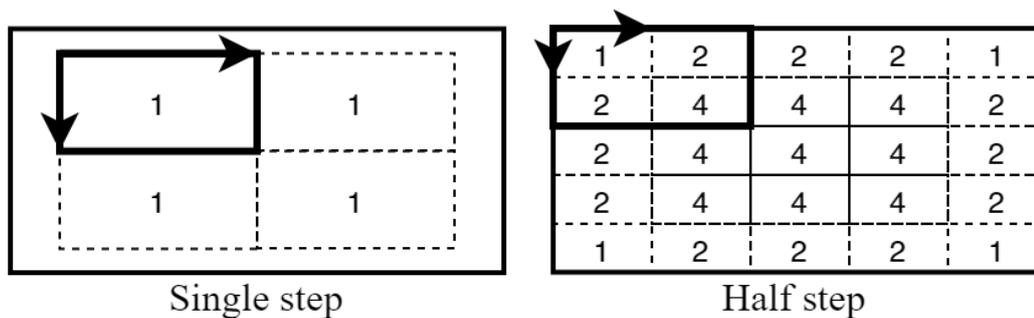


Рисунок 4.4 – Обработка двумерного изображения скользящим окном

Для двумерном случае, проиллюстрированного на рисунке 4.4 можно пронаблюдать, что обработка пикселей, попадающих в апертуру скользящего с половинным шагом окна, не граничащего с краем обрабатываемой области, выполняется 4 раза. В задачах где отсутствует необходимость обработки пикселей на границе обрабатываемой области можно пренебречь результатами, полученными при усреднении 4-х кратной нейросетевой реакции. В свою очередь, обработка изображения с шагом равным размеру окна не обладает таким недостатком.

#### 4.4 Бинаризация нейросетевой сегментации

При решении задачи бинарной сегментации выполнение неоднократной алгоритмической оценки скользящим окном с дальнейшим усреднением результатов позволяет получить более объективную оценку для анализируемой области. Но применение данной техники может сильно изменить распределение значений отклика. В связи с этим недостаточно определить оптимальное пороговое значение бинаризации для классификатора в целом. Поиск данного порога необходимо выполнить и для алгоритма, использующего единичный шаг окна, и для алгоритма с половинным шагом.

Поиск оптимального порогового значения для бинаризации результата предлагается выполнить путем варьирования порогового значения с целью оптимизации бинарной меры сходства IoU. Алгоритм, варьирующий пороговым значением с целью расчета значения IoU в директории, содержащей файлы с не бинаризованной алгоритмической разметкой и разметкой, выполненной экспертом, приведен в листинге 4.8. Функция *variate\_treshold\_for\_iou* принимает на вход путь к директории *input\_path*, содержащей изображения, на которых производится расчет, количество вычислений *num*, которые необходимо сделать на интервале допустимых пороговых значений от минимального *min\_t* до максимального *max\_t*.

Листинг 4.8 – Файл metrics.py. Функция `variate_treshold_for_iou`

```

def iou_score_numpy(y_true, y_pred):
    y_true = y_true.flatten()
    y_pred = y_pred.flatten()
    intersection = y_true@y_pred
    union = y_true.sum() + y_pred.sum() - intersection + 1e-6
    return intersection/(union)
def generate_tresholds_array(num,min_t=0,max_t=1):
    end=True
    if min_t==0 and max_t==1: end=False
    tresholds_array,step=np.linspace(min_t, max_t, num = num, endpoint=end,
retstep=True)
    if min_t==0 and max_t==1: tresholds_array+=step/2
    return tresholds_array
def variate_treshold_for_iou(input_path, num, min_t=0, max_t=1):
    files=[]
    try:
        files = os.listdir(input_path)
    except FileNotFoundError:
        raise FileNotFoundError("Неверный путь к директории с примерами: " +
input_path)
    output_files = sorted(list(filter(lambda x: x.split('_')[0].find('e') > -1,
files)))
    segmentation_files = sorted(list(filter(lambda x: x.split('_')[0].find('s') > -1,
files)))
    if len(output_files) != len(segmentation_files):
        raise FileNotFoundError("Число файлов не совпадает")
    example_paths = []
    for out_f, segment_f in zip(output_files, segmentation_files):
        example_paths.append([input_path+out_f, input_path+segment_f])
    for i, (out_f, segment_f) in enumerate(example_paths):
        e_file = nib.load(out_f)
        s_file = nib.load(segment_f)
        s = s_file.get_fdata()
        e = e_file.get_fdata()
        if i == 0:
            y_true = np.array(e.flatten())
            y_score = np.array(s.flatten())
        else:
            y_true = np.append(y_true, e.flatten())
            y_score = np.append(y_score, s.flatten())
    ious=np.array([])
    i=1
    print("{:5.1f}%".format(0),end='\r')
    tresholds=generate_tresholds_array(num,min_t,max_t)
    for t in tresholds:
        iou=iou_score_numpy(y_true, np.where(y_score<t, 0, 1))
        ious=np.append(ious,iou)
        print("{:5.1f}%".format(i/len(tresholds)*100),end='\r')
        i+=1
    return tresholds, ious

```

Основываясь на результате возвращаемом функцией `variate_treshold_for_iou`, представляется возможным определить оптимальное пороговое значение, для этого достаточно выбрать значение порога из массива `tresholds`, при котором метрика, представленная в массиве `ious` максимальна. Прежде чем принять решение об оптимальном пороге, рекомендуется визуализировать полученный результат в виде графика.

#### 4.5 Генерация обучающего набора данных

Разработанный генератор примеров позволяет варьировать вырезаемый объем из КТ-снимка и переупаковывать его содержимое в сетку заданного размера, другими словами, перепаковать вырезанный трехмерный массив из КТ снимка в трехмерный массив, соответствующий размерам входа ИНС.

В данной работе рассматривалось три варианта вырезаемого объема  $40 \times 40 \times 40$  мм (64 мл),  $60 \times 60 \times 60$  мм (216 мл),  $80 \times 80 \times 80$  мм (512 мл). Примеры входа для ИНС, соответствующие разным объемам, показаны на рисунках 4.5 – 4.7. Каждый сгенерированный пример имеет разрешение  $64 \times 64 \times 32$  вокселя.

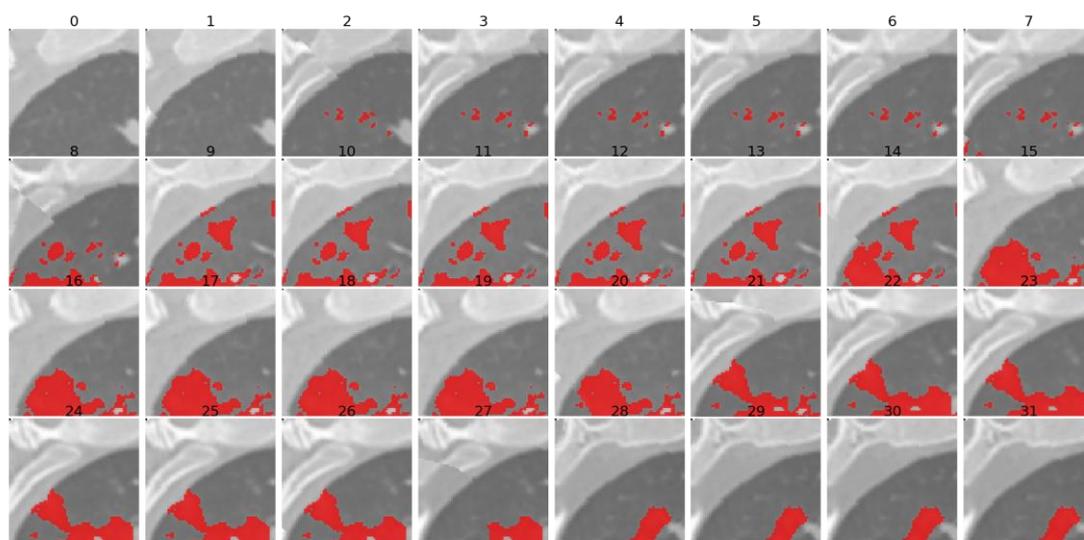


Рисунок 4.5 – Случайно сгенерированный, аугментированный пример  $40 \times 40 \times 40$  мм.

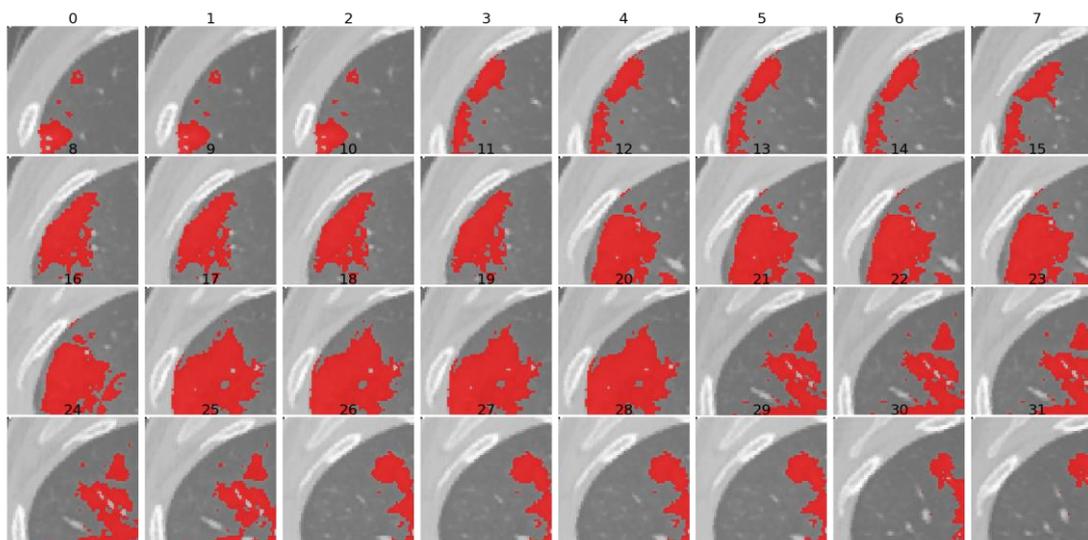


Рисунок 4.6 – Случайно сгенерированный, аугментированный пример  $60 \times 60 \times 60$  мм.

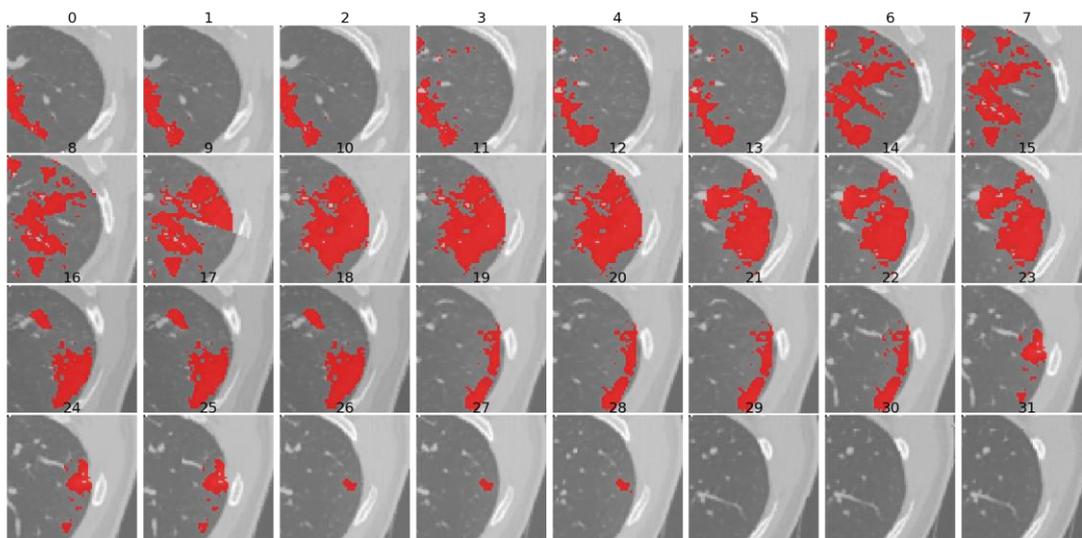


Рисунок 4.7 – Случайно сгенерированный, аугментированный пример  $80 \times 80 \times 80$  мм.

С целью изучения вопроса выбора наиболее подходящего объема для обучения итоговой нейронной сети из 44 обучающих снимков было сгенерировано по 3000 аугментированных экземпляров для каждого объема. Также для каждого объема сгенерирована валидационная выборка, состоящая из 500 примеров. Для всех вариантов объема произведено обучение ИНС. Количество эпох обучения в каждом эксперименте составило 50 эпох, скорость обучения  $lr=0,001$ . За одну эпоху в произвольном порядке нейросети предъявляются все 3000 примеров по 8 в батче, после фиксируется среднее значение ошибки обучения и валидации. Общее время, затраченное на обучение, составило 14,5 часов. Динамика изменения значения функции ошибки во время обучения для обучающего и валидационного наборов представлены на рисунках 4.8 и 4.9.

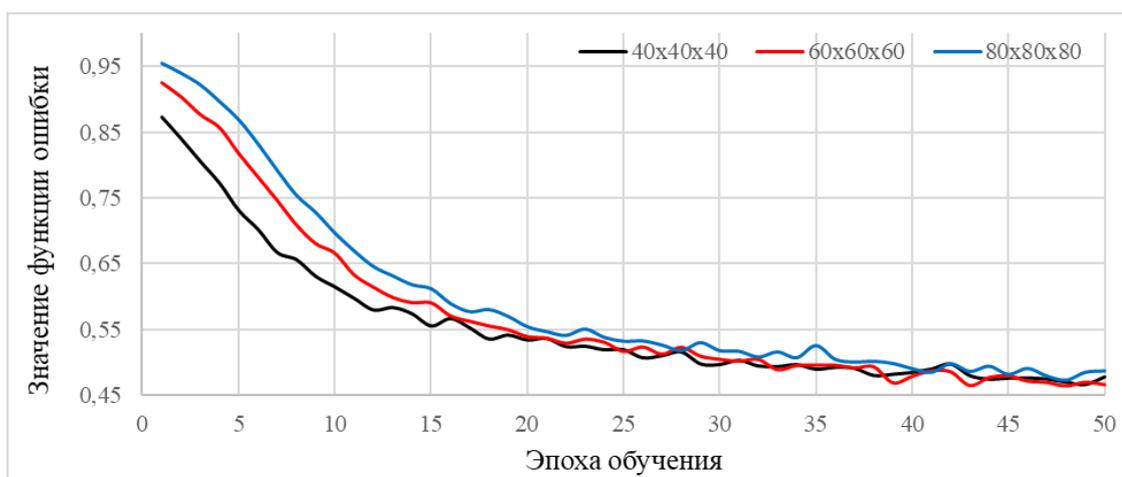


Рисунок 4.8 – Динамика изменения значения функции ошибки на обучающем наборе

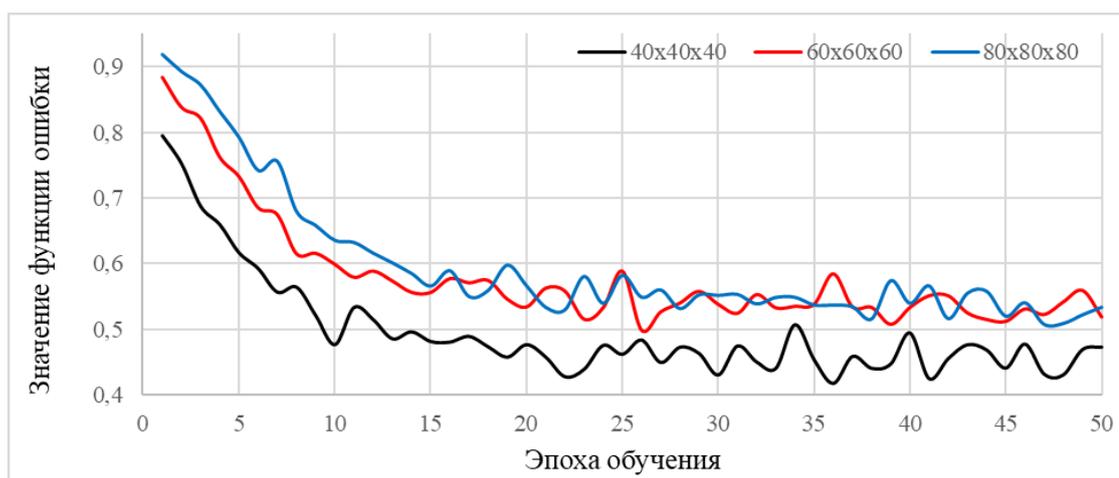


Рисунок 4.9 – Динамика изменения значения функции ошибки на валидационном наборе

После обучения для 5 тестовых КТ снимков, подсчитаны метрики качества сегментации. Сводные результаты оценки нейросетевой сегментации приведены в таблице 4.3.

Таблица 4.3 – Результат сегментации тестовой выборки для разных параметров окна

Объем, мм	Precision, PPV	Recall, TPR	F <sub>1</sub> Score	IoU
40 × 40 × 40	0,619	0,602	0,592	0,446
60 × 60 × 60	0,581	0,618	0,559	0,419
80 × 80 × 80	0,561	0,454	0,428	0,326

Примечание к таблице: значения рассчитаны для каждого КТ-снимка в тестовой выборке при выбранном пороге бинаризации равным 0,2.

Исходя из полученных результатов, однозначно можно сделать вывод о том, что выбор меньшего объема, предъявляемого ИНС, положительно сказывается на процессе обучения и результирующей точности.

Заметим, что для используемого набора данных характерна следующая особенность: в среднем одним вокселем, представленным в КТ-снике, описывается объем  $0,6 \times 0,6 \times 8,0$  мм. Это означает, что, вырезая объем размером  $40 \times 40 \times 40$  мм из исходного изображения, обрабатывается фрагмент, имеющий разрешение примерно  $66 \times 66 \times 5$  вокселей. Данный фрагмент “переупаковывается” по размерам входа для нейронной сети к разрешению  $64 \times 64 \times 32$  вокселей. Ведь размер вырезаемого фрагмента из исходного снимка максимально близок к размеру, с которым работает нейронная сеть, из всех рассмотренных вариантов. В свою очередь для объема  $60 \times 60 \times 60$  мм характерна обработка фрагментов размер которых примерно равен  $100 \times 100 \times 8$  вокселей; для объема  $80 \times 80 \times 80$  мм –  $133 \times 133 \times 10$  вокселей.

Для фиксированного входа ИНС очевидна зависимость, что чем больше объем фрагмента, тем больше информации при переупаковке будет потеряно. Соответственно, при увеличении размера фрагментов стоит ожидать ухудшения в качестве сегментации. Данное предположение подтверждается проведенными экспериментами.

Также у обрабатываемого набора данных есть еще одна особенность, в целом нехарактерная для проведения КТ исследований. Она заключается в том, что все исследования в представленном наборе данных выполнены с шагом в 8 мм между аксиальными срезами. При выполнении КТ исследования, радиологи стараются выбирать меньший шаг между срезами, обычно не превосходящий 5 миллиметров. Конечно выбор такого большего шага аргументируется меньшей дозой радиации получаемой телом пациента в течении исследования. Но при выборе шага в 8 мм, во время проведения радиологического исследования в среднем выполняется 40 сканирований, и в итоге сформированное изображение имеет разрешение  $512 \times 512 \times 40$  вокселей. Очевидно, что в данном случае детализация изображения на сагиттальных и корональных срезах очень плохая. В свете изложенного выше, можно сделать вывод, что увеличение объема вырезаемого фрагмента по высоте не должно негативно сказаться на качестве итоговой сегментации. В связи с этим принято решение, в дальнейших экспериментах использовать фиксированный объем обрабатываемой нейронной сетью –  $40 \times 40 \times 60$  мм.

#### **4.6 Обучение нейронной сети**

Рассматриваемый набор данных содержит 49 КТ-снимков, для которых представлена ручная сегментация областей интереса соответствующих уплотнениям по типу «матового стекла» для пациентов с признаками вирусной пневмонии. КТ-снимки данных пациентов используются в обучении нейронной сети. Набор данных был разделен на две части, в обучающую выборку случайным образом отобрано 44 КТ-снимка, оставшиеся 5 сформировали тестовую выборку.

Согласно предложенному алгоритму аугментации из обучающих и тестовых КТ-снимков была произведена генерация новых обучающих и валидационных примеров, соответствующих входу ИНС. Обучающий набор данных сформирован из 15000 случайно сгенерированных примеров, валидационный из – 500 примеров. Каждый пример в данных наборах имеет разрешение  $64 \times 64 \times 32$  вокселя и описывает фрагмент КТ-снимка заключенный в объем  $40 \times 40 \times 60$  мм (96 мл).

Обучение сети производилось в два этапа. Первый этап при значении  $lr = 0,001$  (скорость обучения) и количестве эпох 100. Во время второго этапа значение скорости обучения было снижено до 0,0001, количество эпох обучения составило 30. График изменения значения ошибки приведен на рисунке 4.10.

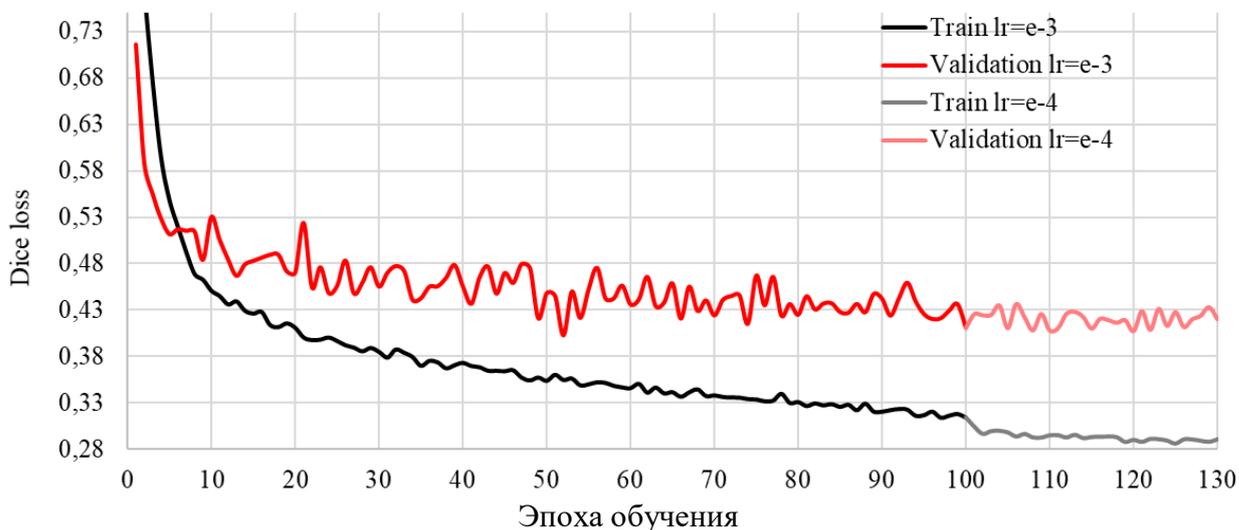


Рисунок 4.10 – Динамика изменения значения функции ошибки во время обучения ИНС

На графика изменения значения ошибки можно заметить, что в момент уменьшения значения скорости обучения, ошибка на обучающем наборе заметно уменьшилась. Также уменьшилось и значение дисперсии ошибки на валидационном наборе. Обучение остановлено в момент достижения критерия останова для ошибки на обучающем наборе. После каждой эпохи ошибка валидации рассчитывается исключительно с целью составления понимания, о том, как хорошо бы был сегментирован тестовый набор данных в случае завершения обучения на определенной эпохе. Ход обучения ИНС и принятие решения о необходимости завершить обучение никак не зависит от значения ошибки на валидационном наборе.

После обучения ИНС для 5 тестовых КТ-снимков подсчитаны средние значения функционалов качества сегментации. Сводные результаты оценки нейросетевой сегментации скользящим окном приведены в таблице 4.4.

Таблица 4.4 – Результат сегментации тестовой выборки

Precision, PPV	Recall, TPR	F1 Score	IoU
0,652	0,638	0,614	0,477

Примечание – метрики рассчитаны для каждого КТ-снимка в тестовой выборке при выбранном пороге бинаризации равным 0,2. Результаты усреднены.

Из результатов оценки нейросетевой сегментации можно сделать вывод, что по всем исследуемым метрикам по сравнению с экспериментом, в котором использовался объем  $40 \times 40 \times 40$  мм, качество нейросетевой сегментации улучшилось. Однако обучение текущей нейронной сети заняло более трех суток, тогда как обучение сети из предыдущего эксперимента составило не более пяти часов. При этом значение критерия IoU изменилось только в третьем знаке.

На рисунках 4.11 – 4.13 приведены аксиальные срезы для КТ-снимков пациентов из тестовой выборки. Желтым цветом представлено результирующее значение при нейросетевой обработке скользящим окном. Красным цветом выделен контур сегментации, выполненной экспертом.

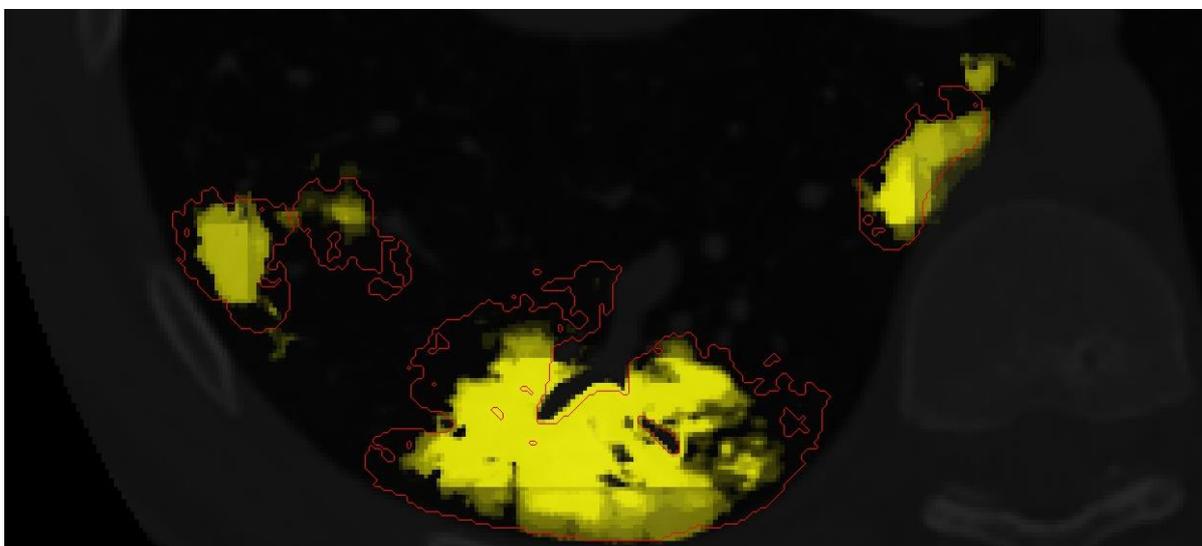


Рисунок 4.11 – Пациент № 0288, аксиальный срез №13

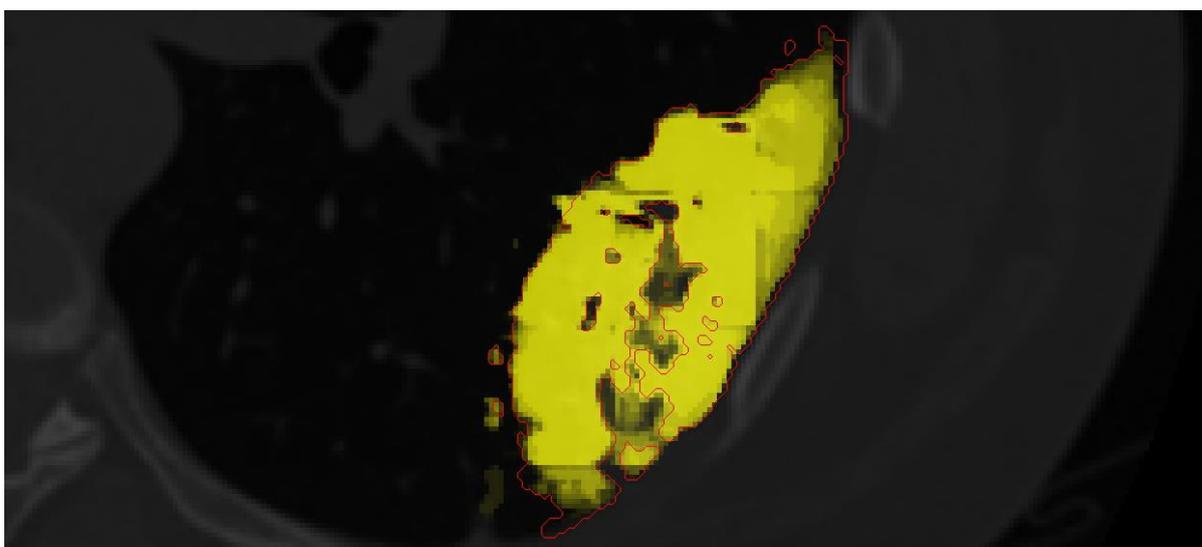


Рисунок 4.12 – Пациент № 0291, аксиальный срез №18

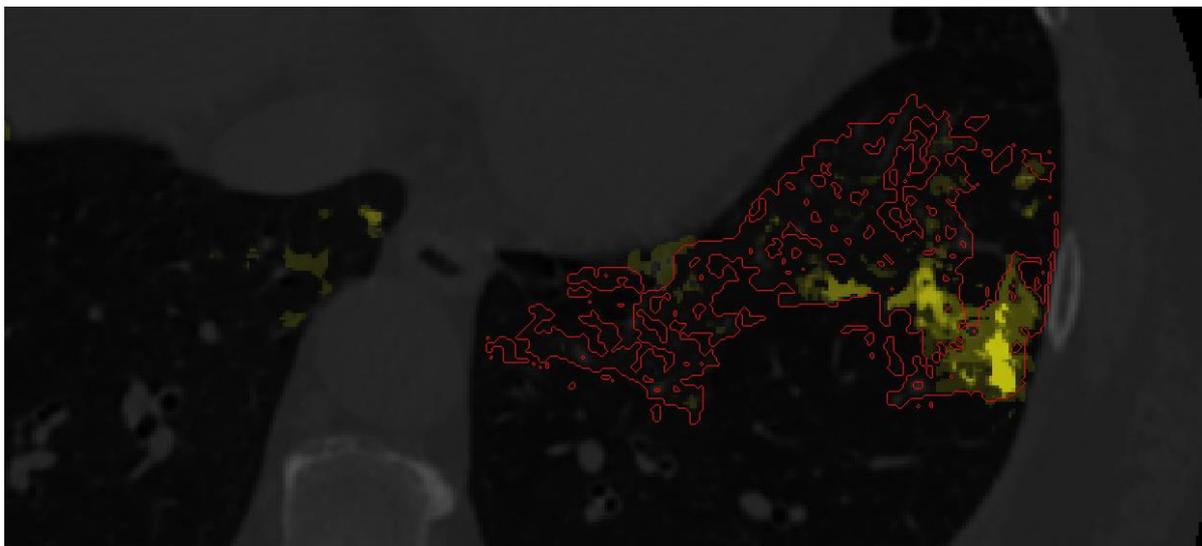


Рисунок 4.13 – Пациент № 0291, аксиальный срез №13

Визуальный анализ результатов нейросетевой обработки КТ-снимков из тестовой выборки, показывает, что нейронная сеть хорошо реагирует на области интереса. Большая часть выделяемых областей имеет значения крайне близкие к максимальному для используемой функции активации на выходе ИНС. В отношении каждого вокселя КТ-снимка, за счет особенности движения скользящего окна, нейросетевая оценка выполняется 8 раз с дальнейшим усреднением результатов. Получение максимального, для функции активации, значения после усреднения, свидетельствует о равномерности распределения сформированных внутри ИНС правил определения, характерных для областей интереса, признаков.

#### **4.7 Сглаживание результатов нейросетевой сегментации фильтром Гаусса**

На примере обработанного КТ-снимка из тестовой выборки, рассмотрим фрагмент аксиального среза, содержащего сегментацию, выполненную экспертом. Рассматриваемый фрагмент приведен на рисунке 4.14.

На рисунке 4.14 видно, что в результате нейросетевой обработки КТ-снимка скользящим окном с половинным шагом, результирующая не бинаризованная сегментация имеет остаточный эффект от резких переходов характерный для обработки скользящим окном с шагом равным размеру окна. Очевидно, что свести влияние данного эффекта к минимуму можно за счет уменьшения шага скользящего окна до пространственных размеров описываемых одним вокселем КТ-снимка, но такой подход очень сильно замедлит процесс обработки изображения.

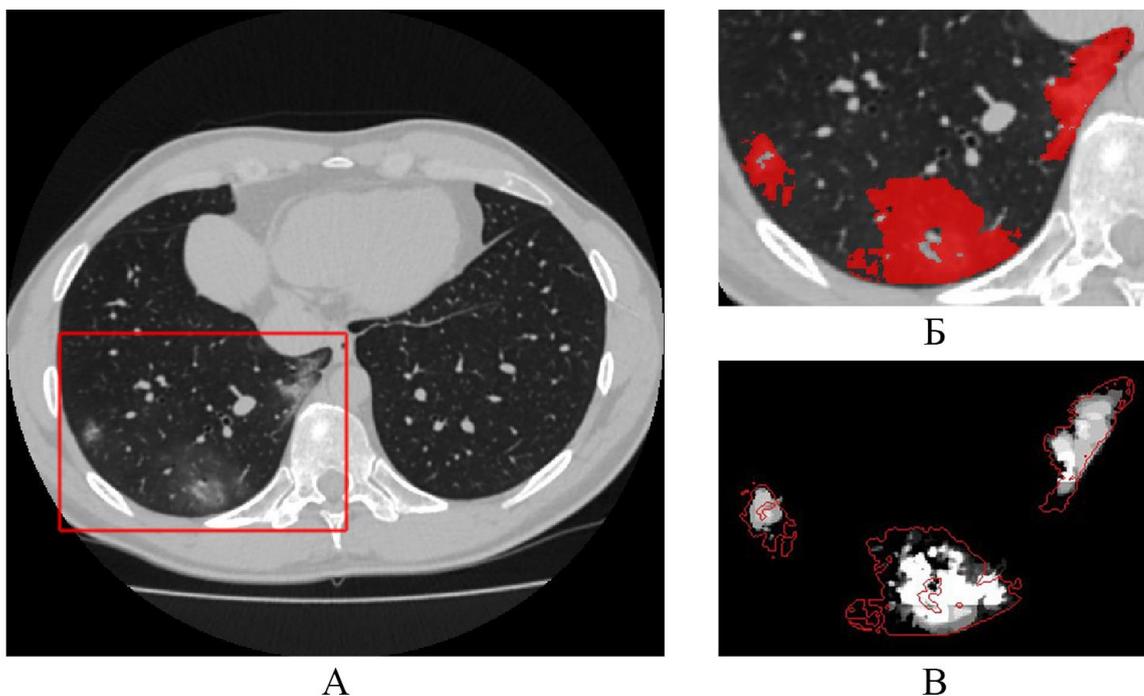


Рисунок 4.14 – Рассматриваемый фрагмент аксиального среза, содержащего сегментацию.  
 а) Аксиальный срез №15 пациента 0288 с уплотнениями по типу «матового стекла»  
 б) Разметка областей интереса, выполненная экспертом  
 в) Результат нейросетевой обработки скользящим окном

Предлагается исследовать возможность компенсации негативных эффектов, вызванных большим шагом скользящего окна с помощью пространственного усредняющего фильтра. В качестве такого фильтра предлагается использовать фильтр Гаусса. В анализе используется реализация многомерного фильтра Гаусса, представленная в библиотеке SciPy [30].

Основным параметром влияющим на результат работы фильтра является стандартное отклонение гауссова распределения  $\sigma$ . Чем больше значение данного параметра, тем сильнее будет “размыто” результирующее изображение.

Для ИНС, полученной в результате основного эксперимента, было проведено исследование зависимости качества ИНС как бинарного классификатора от используемого значения  $\sigma$ . На рисунке 4.15 приводятся результаты применения фильтра Гаусса со следующими значениями стандартного отклонения: 0,5; 1,0; 1,5; 2,0; 2,2.

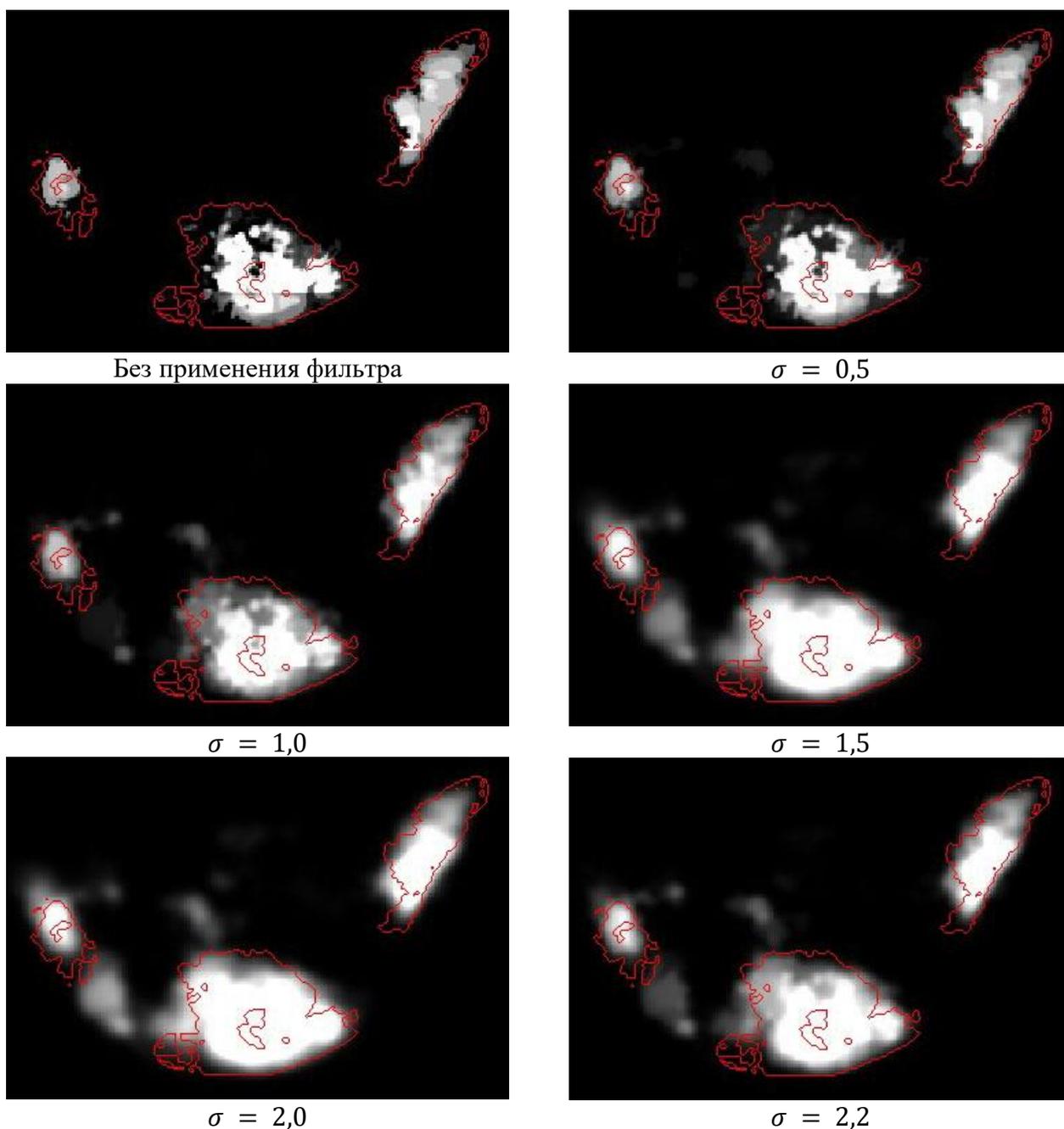


Рисунок 4.15 – Усреднение результата нейросетевой обработки фильтром Гаусса

Оценка качества ИНС, использующейся для решения задачи сегментации КТ-снимка проводится с точки зрения её оценки как бинарного классификатора. Для оценки влияния изменения параметра  $\sigma$  использовано значение площади под ROC-кривой – AUC. Построение ROC-кривых выполнено после применения фильтра Гаусса с каждым из предложенных значений  $\sigma$ , результаты приведены на рисунке 4.16. Расчетные значения AUC для каждой построенной ROC-кривой приведены в таблице 4.5.

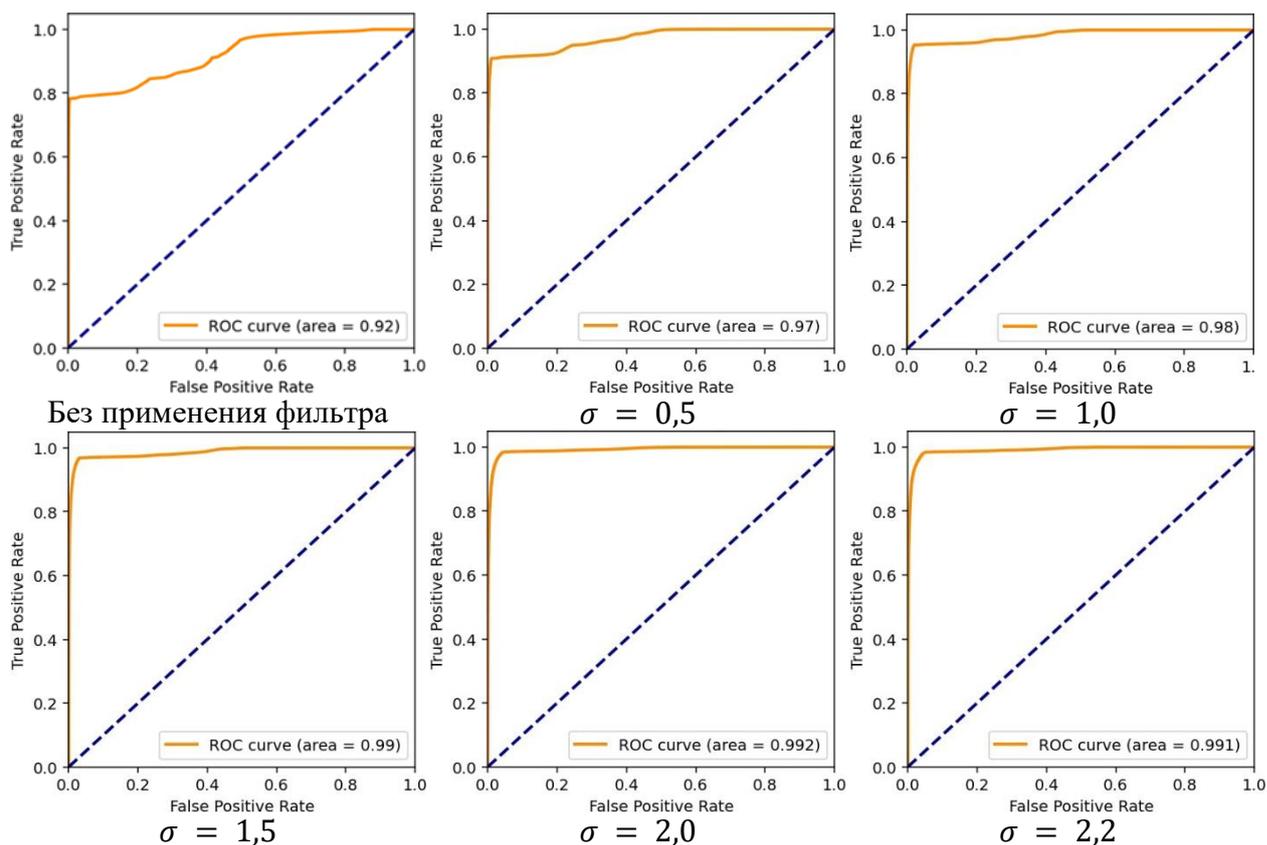


Рисунок 4.16 – ROC-кривые для рассматриваемых значений параметра  $\sigma$  фильтра Гаусса

Таблица 4.5 – AUC для рассматриваемых значений параметра  $\sigma$  фильтра Гаусса

	Без фильтра	$\sigma = 0,5$	$\sigma = 1,0$	$\sigma = 1,5$	$\sigma = 2,0$	$\sigma = 2,2$
AUC	0,923	0,969	0,979	0,986	0,992	0,991

Из полученных результатов можно сделать ошибочное заключение, о том что применение фильтра Гаусса с параметром  $\sigma = 2,0$  позволяет радикально улучшить качество выполняемой ИНС сегментации. Проведение дополнительных экспериментов, нацеленных на определение максимально возможного значения для функционала ЮУ при применении рассмотренных значений  $\sigma$ , показало, что применение фильтра Гаусса действительно положительно сказывается на сегментации при  $\sigma = 0,5$ .

## 5 РЕЗУЛЬТАТ ЛОКАЛИЗАЦИИ ОБЛАСТЕЙ ИНФИЛЬТРАТА

При постановке диагноза широко распространена практика использования объема инфильтрации легкого как предиктора, принято решение оценить возможность использования обученной нейронной сети в задаче постановки диагноза. В результате нейросетевой обработки КТ-снимка представляется возможным численно определить суммарный объем области интереса в миллилитрах. Для корректного решения данной задачи предиктором к принятию решения должен выступать не фактический объем инфильтрации, а его относительное значение к объему легких. Но так как в рамках данной работы не рассматривается задача сегментирования легкого, определить относительное значение объема инфильтрации к объему легкого не представляется возможным. Рассмотрим в качестве предиктора для постановки диагноза именно фактический объем инфильтрации.

В обрабатываемом наборе данных «MosMedData: COVID19\_1110», подробно описанном в предыдущем разделе, представлено 1110 КТ-снимков, распределенных по 5 классам. Класс СТ-0 содержит снимки пациентов, для которых эксперт не обнаружил признаков вирусной пневмонии. В свою очередь остальные классы (СТ-1, 2, 3, 4) содержат снимки пациентов, легкие которых, по мнению эксперта, содержат зоны уплотнения по типу «матового стекла». Отметим, что качество экспертизы в представленном наборе данных не может быть оценено в рамках данной работы. Среди представленных 1110 примеров в обучении нейронной сети использовались только 44 снимка, представленных в классе СТ-1, так как только для этих пациентов в используемом наборе данных представлена сегментация, выполненная экспертом.

Обработка КТ-снимка подразумевает нейросетевую обработку скользящим окном с движением апертуры окна в половину своего размера по трем осям. Фактически данный вид обработки предъявляет КТ-снимок нейронной сети для оценки восемь раз. Среднее время сегментации одного снимка составило 3 мин. 17 сек., медианное 2 мин. 57 сек., минимальное 47 секунд, максимальное 12 мин. 48 сек. Сегментирование всего набора данных заняло немногим больше 2 суток и 14 часов.

После обработки каждого снимка фиксировалось суммарное значение по всем вокселям с применением трех способов постобработки:

- 1) Без применения порога (А);
- 2) С применением оптимального порога, определенного на тестовой выборке (В);
- 3) С применением фильтра Гаусса ( $\sigma = 0,5$ ) и оптимального порога (С).

Для получения оценки фактического объема инфильтрации легких для каждого пациента, суммарное значение умножается на значение объема, описываемого одним вокселем КТ-снимка.

При решении задачи постановки диагноза первичным является принятие решения о том болен человек или нет. И только в случае если обнаружены достаточные признаки заболевания следует принимать решение о том в какой стадии развивается болезнь. В связи с этим сведем задачу оценки предложенного нейросетевого подхода для постановки диагноза к задаче бинарной классификации, объединив классы (СТ-1, 2, 3, 4) в единый класс СТ-1.

При решении задачи бинарной классификации часто для оценки качества предлагаемых классификаторов используют значение площади под ROC-кривой – AUC. Построим ROC-кривые для каждого из предложенного вида постобработки. Рассчитав значение AUC для каждого вида постобработки, сделаем вывод о том, какой из приведенных видов постобработки является более качественным в решении поставленной задачи. ROC-кривые представлены на рисунке 5.1. Расчетные значения AUC приведены в таблице 5.1.

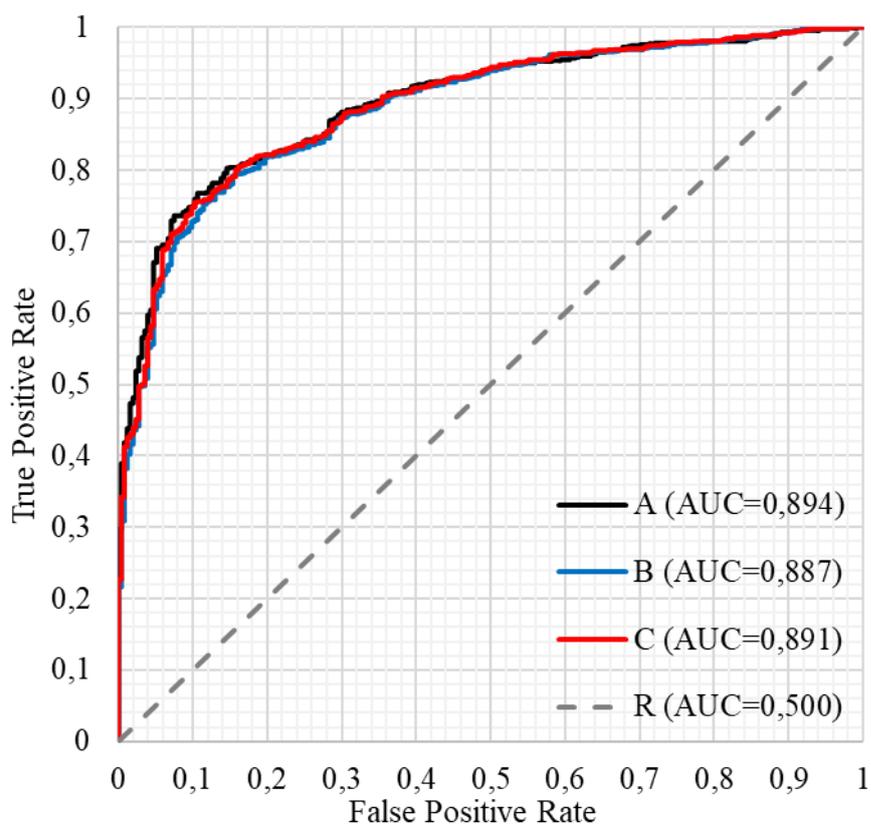


Рисунок 5.1 – ROC-кривые для рассматриваемых видов постобработки КТ-снимка.  
А) Без применения порога; В) С применением оптимального порога, определенного на тестовой выборке; С) С применением фильтра Гаусса при  $\sigma = 0,5$  и оптимального порога;  
R) Случайный выбор

Таблица 5.1 – Расчетные значения AUC для используемых способов постобработки.

Способ постобработки	Применена		AUC
	Фильтрация по Гауссу	Бинаризация	
А	нет	нет	0,894
В	нет	да	0,887
С	да	да	0,891

Полученные результаты показывают, что серьезные различия в качестве классификации для предложенных способов постобработки отсутствуют. Однако, использование постобработки КТ-снимка, хоть и незначительно, но негативно сказывается на качестве классификатора. Заметим, что в случае выполнения бинаризации, применение фильтра Гаусса, как и в задаче сегментации, положительно сказывается на качестве классификатора.

Построим распределение классов, представленных в наборе данных, по фактически детектированному с помощью ИНС объему инфильтрации в области легкого без применения постобработки. Приведенная на рисунке 5.2 стековая гистограмма построена с применением логарифмического масштабирования к протяженности интервалов группировки.

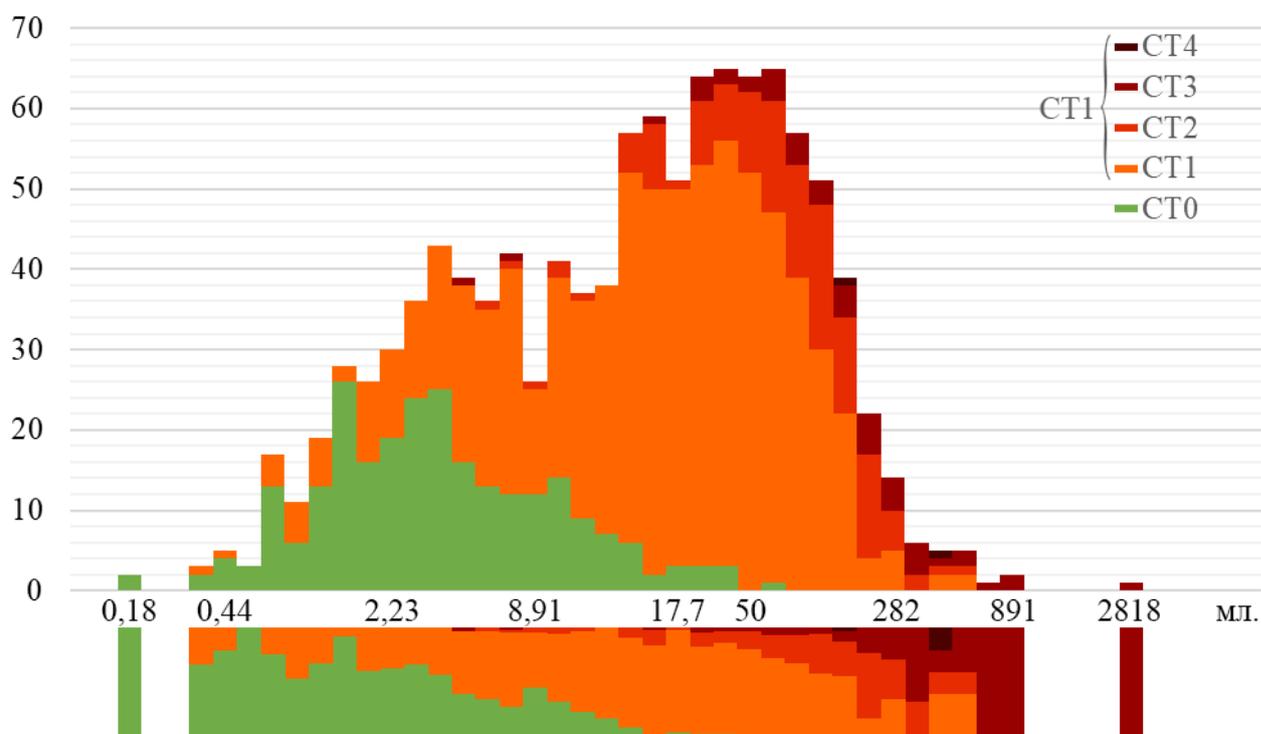


Рисунок 5.2 – Распределение классов набора данных «MosMedData: COVID19\_1110»

При определении эффективного порогового значения точность бинарной классификации составила 84,6 %. Оценочно данному порогу соответствует фактический

объем детектированного нейросетью инфильтрата в области легкого в диапазоне от 3,5 мл до 6,6 мл.

Очевидно, что качество классификации при фиксированном пороговом значении во многом зависит от реального распределения классов в генеральной совокупности. Данное распределение не известно и требует дополнительной экспертной оценки. В свою очередь, выбор оптимального порогового значения должен выполняться с учетом результатов такого вида экспертизы. Сбалансировав выборку путем 5-и кратного случайного выбора по 100 пациентов из каждого класса (СТ-0 и СТ-1), была сформирована новая (псевдо бутстрап) выборка, состоящая из 1000 пациентов. Для выборки определенно оптимальное пороговое значение, оценочно соответствующее объему инфильтрации от 25,2 мл до 38,6 мл. При принятии указанного порога точность классификации на сбалансированной выборке составила 82,2 %, а на исходной 76,8 %.

Для визуальной оценки качества, выполняемой ИНС локализации областей интереса путем сегментации КТ-снимка, в приложении А приводятся результаты сегментации для снимков пациентов из классов СТ-0 и СТ-1. Выбор пациентов основан на значении суммарно детектированного объема инфильтрата. В приложении А представлены аксиальные срезы из КТ-снимков пациентов, для которых объем инфильтрата принял минимальное, среднее и максимальное значение внутри распределения класса. Сведения о таких пациентах и ссылки на иллюстрации приведены в таблице 5.2.

Таблица 5.2 – Выбранные пациенты для визуального анализа результата

Объем детектированного инфильтрата, мл	Ближайший пациент в классе по детектированному объему инфильтрата			
	СТ-0		СТ-1	
	№ пациента	иллюстрация	№ пациента	иллюстрация
0,7	0172	A.1		
2,5	0113	A.2		
8,9	0047	A.3	0720	A.5
17,7	0152	A.4	0401	A.6
177,0			0317	A.7
707,0			0505	A.8

Визуальный анализ полученных результатов показывает, что разработанный подход к локализации областей инфильтрата на КТ-снимках хорошо справляется с поставленной задачей. Но у разработанного алгоритма сегментации есть и недостатки, например, ИНС часто принимает артефакты от сокращения сердца пациента за область интереса. Также ИНС склонна к выполнению сегментации кушетки, на которой лежит пациент при выполнении КТ-исследования.

## **6 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

Целью данного раздела является определение оценки коммерческого потенциала, перспективности и альтернатив проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения, а также планирование и формирование бюджета научных исследований, определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования. Расчет экономических показателей выполняется согласно методическим указаниям [31].

Научно-исследовательская работа направлена на разработку архитектуры нейронной сети для автоматизированной сегментации биомедицинских изображений, особое внимание уделено задаче автоматической локализации областей инфильтрата в легких по данным компьютерной томографии. Разработка на основе описываемого в данной работе алгоритма программной системы для обработки КТ-снимки позволит находить, распознавать и локализовывать не только области инфильтрата в легких, но и любые патологии и новообразования, поддающиеся аннотации, на биомедицинских изображениях. Данная разработка актуальна и востребована в радиологических отделениях медицинских учреждений. Применение данного подхода в задачах радиоизотопной диагностики позволит качественно улучшить точность подготавливаемых экспертом заключений о результатах исследования и существенно сократить время проведения исследования. Также разработанная методика автоматизированной аннотации биомедицинских изображений может стать основой для разработки обучающего программного обеспечения для практикантов в области радиологических исследований.

### **6.1 Потенциальные потребители результатов исследования**

Для анализа потребителей результатов исследования необходимо рассмотреть целевой рынок и провести его сегментирование. Целевой рынок представляет собой сегменты рынка, на котором в будущем возможно внедрение и коммерциализация разработки. Сегментирование – это разделение потребителей на однородные группы, для каждой из которых может потребоваться определенный товар.

Так как главными потребителями разработки являются медицинские учреждения, в качестве основных критериев сегментации выбраны профили медицинской помощи специализированного учреждения, и типы томографических исследований, применяемых в рамках данного профиля. Сегментирование производится с позиции взвешенной оценки между степенью распространенности и применимости вида томографического

исследования в рамках данного профиля и степенью готовности данного профиля к внедрению методики автоматизированной аннотации биомедицинских изображений. Результаты сегментирования приведены в таблице 6.1.

Таблица 6.1 – Карта сегментирования рынка по профилю медицинской помощи

		Вид исследования			
		КТ	МРТ	Рентген	УЗИ
Профиль медицинской помощи	Гастроэнтерология				
	Гематология				
	Генетика				
	Гинекология				
	Дерматология				
	Кардиология				
	Неврология				
	Онкология				
	Оториноларингология				
	Офтальмология				
	Психиатрия				
	Пульмонология				
	Ревматология				
	Стоматология				
	Терапевтия				
	Травматология и ортопедия				
	Урология				
	Хирургия				
Эндокринология					

Заинтересованные стороны проекта, лица и организации, которые активно участвуют в проекте или интересы которых затронуты в ходе исполнения или в результате завершения проекта, приведены в таблице 6.2.

Таблица 6.2 – Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
НИ ТПУ, ИШИТР	Получение нейросетевого алгоритма для определения локализации областей инфильтрата в области легкого по данным компьютерной томографии с целью его дальнейшего внедрения и представления на конференциях
ОГАУЗ «Томский Фтизиопульмонологический Медицинский Центр»	Повышение эффективности работы врача-радиолога при подготовке медицинского заключения о наличии патологий, представляющих собой скопление инфильтрата в области легкого
Медицинские научно-исследовательские группы и институты	Разработка позволит более глубоко исследовать признаки и принципы формирования разных видов патологий за счет интеллектуального нейросетевого анализа биомедицинских изображений

Компании, занимающиеся разработкой и производством медицинского оборудования	При достижении очень высокого качества в сегментации и детектировании патологий разработка может стать основой для формирования интеллектуальных отчетов, получаемых с оборудования, с помощью которого выполняется радиологическое исследование
Студент	Защита магистерской диссертации

## 6.2 Оценка конкурентоспособности технических решений

Перечень основных конкурентных технических решений приведен в таблице 6.3.

Таблица 6.3 – Перечень основных конкурентных технических решений

Код	Название продукта	Страна производителя
1	Zebra «AI1»	Израиль
2	Arterys «Lung AI»	США
3	InferVision «InferRead»	Китай
4	DeepWise «Dr. Wise»	Китай
5	Проект, предлагаемый в текущем исследовании	Россия

Для проведения конкурентного анализа выбраны следующие критерии:

- Скорость сегментации – скорость выполнения обработки изображения;
- Качество сегментации – оценка качества извлекаемой исследователем информации после обработки биомедицинского изображения. Так наименьшую оценку получает тот продукт, результатом работы которого является численная оценка вероятности о наличии признаков искомой патологии, и наибольшую - точная локализация с автоматическим определением класса патологии;

- Энергоэффективность – оценка необходимости эксплуатации оборудования, требующего большего количества вычислительных ресурсов и как следствие электроэнергии, косвенно данный параметр затрагивает необходимость закупки дополнительного дорогостоящего оборудования;

- Универсальность – определяет степень возможности применения продукта в спектре внутри медицинского профиля, а также гибкость продукта относительно новых задач, например, переобучения модели для работы с новой, еще не изученной, патологией или изменения принципа распределения признаков, определяющих патологию;

- Потенциал разработки – отражает возможности по улучшению алгоритма без внесения значительных изменений в его архитектуру;

– Доступность – оценка степени, на сколько открытой и прозрачной является разработка. Определяется доступностью документации, открытостью исходного кода и степень готовности компании-производителя к сотрудничеству.

Анализ конкурентных технических решений определяется по формуле (6.1). Результаты конкурентного анализа приведены в таблице 6.4.

$$K = \sum V_i \cdot B_i \quad (6.1)$$

где  $K$  – конкурентоспособность научной разработки или конкурента,

$V_i$  – вес показателя (в долях единицы),

$B_i$  – балл  $i$ -го показателя;

Таблица 6.4 – Результаты конкурентного анализа

Критерий оценки	Вес	Баллы					Конкурентоспособность				
		$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$K_1$	$K_2$	$K_3$	$K_4$	$K_5$
<b>Технические критерии оценки ресурсоэффективности</b>											
<b>1. Скорость сегментации</b>	0.06	5	2	3	4	4	0.30	0.12	0.18	0.24	0.24
<b>2. Качество сегментации</b>	0.26	3	2	4	5	3	0.18	0.12	0.24	0.30	0.18
<b>3. Энергоэффективность</b>	0.04	5	3	2	3	3	0.30	0.18	0.12	0.18	0.18
<b>4. Универсальность</b>	0.08	3	2	3	4	3	0.18	0.12	0.18	0.24	0.18
<b>5. Потенциал разработки</b>	0.24	3	2	3	5	5	0.18	0.12	0.18	0.30	0.30
<b>Экономические критерии оценки эффективности</b>											
<b>6. Доступность</b>	0.20	5	5	3	2	5	0.30	0.30	0.18	0.12	0.30
<b>7. Цена</b>	0.12	5	4	1	3	5	0.30	0.24	0.06	0.18	0.30
<b>Итого:</b>	1.00	29	20	19	26	28	1.74	1.20	1.14	1.56	1.68

По результатам проведённого конкурентного анализа можно сделать вывод, что предложенный метод конкурентоспособен в сравнении с другими рассмотренными методами в первую очередь за счет своей гибкости и потенциала к решению широкого спектра задач, а также своей доступностью для российского рынка. Но все же на данном этапе своего развития, представленный проект оценочно проигрывает легкодоступному и дешевому Израильскому продукту компании Zebra-Med.

Также стоит отметить, что результаты, демонстрируемые китайской компанией DeepWise ставят высокую планку для разрабатываемой технологии. Но данная компания сильно ориентирована на китайский рынок и на сегодняшний день не демонстрирует готовности к международному сотрудничеству, этот факт снижает конкурентоспособность ее продукта.

### 6.3 SWOT-анализ

Результаты SWOT-анализа учитываются при разработке структуры работ, выполняемых в рамках разработки проекта. Учитывая отраслевую специфику объекта исследования, необходимо провести комплексный анализ проекта. В ходе проведения SWOT-анализа была составлена итоговая матрица (таблица 6.5).

Таблица 6.5 – Результаты SWOT-анализа

	<b>Сильные стороны</b> С1. Высокая точность сегментации С2. Высокая скорость сегментации С3. Широкая область применения С4. Открытый исходный код С5. Отсутствие конкурентов в РФ	<b>Слабые стороны</b> Сл1. Проигрыш в точности некоторым зарубежным конкурентам Сл2. Сильная зависимость от оборудования Сл3. Необходимость в большом количестве обучающих данных
<b>Возможности</b> В1. Дальнейшее обучение сети для повышения качества сегментации В2. Стремительное развитие направления машинного обучения В3. Рост производительности аппаратных средств В4. Создание базы обучающих примеров	<b>Направления развития</b> 1. В1В2С1 - В связи с развитием направления машинного обучения и появлением новых методов и алгоритмов, возможно внедрение новых оптимизаций и эвристик в данную разработку для повышения качества сегментации 2. В3С2 - С увеличением производительности аппаратных средств компьютера, замена комплектующих ведёт к увеличению скорости сегментации 3. В1В4С3С4 - Открытый исходный код и широкая область применения позволят добавить новые функциональные возможности 4. В1В4С5 - Сильные стороны проекта позволят привлечь потенциальных заказчиков и организовать плодотворное сотрудничество с медицинскими учреждениями в РФ	<b>Сдерживающие факторы</b> В3Сл1 - Использование более производительных аппаратных средств повысит конкурентоспособность разработки В4Сл1Сл3 - Создание базы данных с большим количеством обучающих примеров очень трудоемкий процесс, требующий привлечения дорогостоящих специалистов В3Сл2 - С ростом производительности аппаратных средств уменьшается цена оборудования, в свою очередь, отказ от поддержки проекта в обновлении аппаратных средств может привести к снижению конкурентоспособности проекта
<b>Угрозы</b> У1. Появление на рынке конкурентоспособных аналогов разработки У2. Несвоевременное финансирование	1. У1С1С2С3С4С5 - высокая точность и скорость сегментации и широкая область применения позволят избежать появления конкурентоспособных аналогов и потерю актуальности разработки	1. У1Сл1Сл2Сл3 - слабые стороны проекта могут привести к появлению конкурентоспособных аналогов разработки 2. У2Сл3 - Отсутствие должного финансирования работы специалистов, подготавливающих обучающие наборы данных делает невозможным развитие проекта

На основе полученных данных можно сделать вывод, что основным недостатком данной разработки является необходимость в большем количестве аннотированных обучающих примеров, решение этой проблемы требует привлечения дорогостоящих специалистов по разметке (врачей-радиологов) к проекту. Также ввиду большой вычислительной сложности задачи обучения экспериментальных нейросетевых моделей требуется закупка современного вычислительного оборудования.

#### 6.4 Оценка готовности проекта к коммерциализации

Для оценки готовности научной разработки к коммерциализации была составлена таблица 6.6, содержащая показатели степени проработанности проекта с позиции коммерциализации и компетенций разработчика научного проекта.

Таблица 6.6 – Бланк оценки степени готовности научного проекта к коммерциализации

№	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1	Определен имеющийся научно-технический задел	4	4
2	Определены перспективные направления коммерциализации научно-технического задела	4	3
3	Определены отрасли и технологии (товары, услуги) для предложения на рынке	4	3
4	Определена товарная форма научно-технического задела для представления на рынок	2	3
5	Определены авторы и осуществлена охрана их прав	1	2
6	Проведена оценка стоимости интеллектуальной собственности	2	1
7	Проведены маркетинговые исследования рынков сбыта	2	2
8	Разработан бизнес-план коммерциализации научной разработки	1	1
9	Определены пути продвижения научной разработки на рынок	2	2
10	Разработана стратегия (форма) реализации научной разработки	1	4
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	1	2
12	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	1	1
13	Имеется команда для коммерциализации научной разработки	1	2
14	Проработан механизм реализации научного проекта	3	4
	<b>ИТОГО БАЛЛОВ</b>	<b>29</b>	<b>34</b>

После проведения анализа можно сделать вывод, что перспективность проведения коммерциализации средняя. По вопросам маркетинговых исследований, финансирования коммерциализации необходимо привлечение в команду проекта специалистов из данных областей.

## 6.5 Организация и планирование работ по проекту

При организации процесса реализации необходимо рационально планировать занятость каждого из его участников и сроки проведения отдельных работ. В данном подразделе приводится полный перечень проводимых работ, определяются их исполнители и рациональная продолжительность.

Для выполнения работ по проекту сформирована рабочая группа. Состав рабочей группы приведен в таблице 6.7. Перечень работ и продолжительность их выполнения приведен в таблице 6.8.

Таблица 6.7 – Рабочая группа проекта

№	ФИО, основное место работы, должность	Роль в проекте	Функции
1	Аксёнов С. В. к.т.н., доцент ОИТ ТПУ	Научный руководитель, руководитель проекта	Составление ТЗ, консультации, проверка полученных результатов
2	Проскуров В.А. магистрант ТПУ	Инженер, исполнитель по проекту	Анализ области разработки, разработка проекта, тестирование, подготовка документации

Таблица 6.8 – Перечень работ и продолжительность их выполнения

№	Этап работы	Исполнители	Занятость, %
1	Составление и утверждение технического задания	НР	100
2	Подбор и изучение материалов по теме	И	100
3	Изучение уже существующих решений в данной области	И	100
4	Выбор направления исследований	НР, И	100, 60
5	Календарное планирование работ по теме	НР	100
6	Теоретическая разработка структуры алгоритма, решающего поставленную задачу	И	100
7	Построение пробных вариантов алгоритма	И	100
8	Оценка результатов и итеративное внесение изменений в структуру алгоритма	НР, И	30, 100
9	Сравнение результатов работы конечного варианта алгоритма с другими решениями в данной области	И	100
10	Оценка эффективности полученных результатов	НР, И	100, 100
11	Оценка целесообразности проведения дальнейших исследований по данной теме	НР, И	100, 40

### 6.5.1 Определение продолжительности этапов работ

В силу новизны изучаемой предметной области на данном этапе у рабочей группы отсутствует достаточный опыт в реализации подобных проектов. В связи с этим расчет продолжительности этапов работ осуществляется опытно-статистическим способом. Основываясь на профессиональном опыте руководителя проекта, ожидаемая продолжительность работ  $t_{ож}$  определяется по формуле (6.2).

$$t_{ож} = \frac{t_{мин} + 4 \cdot t_{вер} + t_{макс}}{6} \quad (6.2)$$

где  $t_{ож}$  – ожидаемая продолжительность работ, дн.;

$t_{мин}$  – минимальная продолжительность работ, дн.;

$t_{вер}$  – наиболее вероятная продолжительность работ, дн.;

$t_{макс}$  – максимальная продолжительность работ, дн.;

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях  $T_{рд}$  ведется по формуле (6.3).

$$T_{рд} = t_{ож} \cdot K_{вн} \cdot K_{д} \quad (6.3)$$

где  $T_{рд}$  – продолжительность выполнения этапа работы, дн.;

$t_{ож}$  – ожидаемая продолжительность работ, дн.;

$K_{вн}$  – коэффициент влияния внешних факторов,  $K_{вн} \in [0,8; 1,2]$ ;

$K_{д}$  – коэффициент дополнительного времени,  $K_{д} \in [1,0; 1,2]$ .

Коэффициент  $K_{вн}$  позволяет учесть возможное влияние внешних факторов на соблюдение предварительно определенных длительностей. В данной работе принимается, что влияние внешних факторов отсутствует,  $K_{вн} = 1$ .

Коэффициент  $K_{д}$  позволяет учесть дополнительное время, необходимое на компенсацию непредвиденных задержек, вызванных внутренними причинами.

Расчет продолжительности этапа в календарных днях  $T_{кд}$  ведется по формуле (6.4).

$$T_{кд} = [T_{рд} \cdot T_{к}] \quad (6.4)$$

где  $T_{кд}$  – продолжительность этапа работы в календарных днях, дн.;

$T_{рд}$  – продолжительность выполнения этапа работы, дн.;

$T_{к}$  – коэффициент календарности.

Коэффициент  $T_K$  позволяет перейти от длительности работ в рабочих днях к ее аналогам в календарных днях, расчет данного коэффициента выполняется по формуле (6.5).

$$T_K = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}} \quad (6.5)$$

где  $T_K$  – коэффициент календарности;

$T_{КАЛ}$  – календарные дни, дн.;

$T_{ВД}$  – выходные дни, дн.;

$T_{ПД}$  – праздничные дни, дн.

Расчетное значение коэффициента календарности при принятых значениях в 2020 г.  $T_{КАЛ} = 366$ ,  $T_{ВД} = 52$  и  $T_{ПД} = 14$ , для 6-дневной рабочей недели составило  $T_K = 1,22$ . Данное значение будет использовано при определении календарной продолжительности периода выполнения проекта и при построении соответствующего графика.

В таблице 6.9 приводятся расчетные значения, определяющие продолжительность этапов работ и их трудоемкость по исполнителям, занятым на каждом этапе, с учетом принятых значений  $K_{ВН} = 1,0$ ,  $K_D = 1,1$  и  $T_K = 1,22$ .

Таблица 6.9 – Трудозатраты на выполнение проекта

Этап	Продолжительность работ, дн.				Трудоемкость работ по исполнителям, дн.			
					$T_{рл}$		$T_{кл}$	
	$t_{мин}$	$t_{вер}$	$t_{макс}$	$t_{ож}$	НР	И	НР	И
1	2	3	3	2,83	3,12		4	
2	12	15	20	15,33		16,87		21
3	3	4	9	4,67		5,13		7
4	3	3	5	3,33	3,67	2,20	5	3
5	2	3	4	3,00	3,30		5	
6	4	5	9	5,50		6,05		8
7	14	20	24	19,67		21,63		27
8	12	28	32	26,00	8,58	28,60	11	35
9	2	3	4	3,00		3,30		5
10	3	3	6	3,50	3,85	3,85	5	5
11	2	2	4	2,33	2,57	1,03	4	2
<b>Итого:</b>				89,17	25,08	88,66	34	113

Общая ожидаемая продолжительность работ по проекту составила 89 рабочих дней. Общая трудоемкость работ составила 25 и 89 дней для научного руководителя и инженера соответственно. Общая продолжительность рабочих этапов, с учетом возможности параллельного выполнения запланированных работ, составила 115 календарных дней. Хронологически упорядоченные значения продолжительности этапов работ сведены в линейный график осуществления проекта, приведенный в таблице 6.10.

Таблица 6.10 – Календарный план. Линейный график работ

Этап	T <sub>кд</sub>		Продолжительность выполнения работ																				
			январь			февраль				март					апрель				май				
	НР	И	13	20	27	3	10	17	24	2	9	16	23	30	6	13	20	27	4	11	18	25	1
1	4					▨																	
2		21				■	■	■	■														
3		7							■														
4	5	3							▨	■													
5	5										▨												
6		8									■	■											
7		27											■	■	■	■	■						
8	11	35															▨	■	■	■	■	■	
9		5																				■	
10	5	5																				▨	■
11	4	2																				▨	■

Примечание к таблице: ▨ – Научный руководитель, ■ – Инженер

## 6.6 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- прочие (накладные) расходы.

### 6.6.1 Расчет затрат на материалы

К данной статье расходов относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ над объектом проектирования. Сюда же относятся специально приобретенное оборудование, инструменты и прочие объекты, относимые к основным средствам, стоимостью до 40 000 руб. включительно. Цена материальных ресурсов определяется по соответствующим ценам или договорам поставки. Кроме того, статья включает так называемые транспортно-заготовительные расходы, связанные с транспортировкой от поставщика к потребителю, хранением и прочими процессами, обеспечивающими движение (доставку) материальных ресурсов от поставщиков к потребителю. Сюда же включаются расходы на совершение сделки купли-продажи (т.н. транзакции). Приблизительно они оцениваются в процентах к отпускной цене закупаемых материалов, как правило, это 5-20 %.

Таблица 6.11 – Расчет затрат на материалы

Наименование материала	Цена за ед., руб.	Кол-во	Сумма, руб.
Видеокарта Asus GeForce GTX 1070 Ti, 8 Gb	38 680,00	1 шт.	38 680,00
Блок питания Zalman Wattbit, 600 Вт	3 050,00	1 шт.	3 550,00
Оперативная память Silicon Power DDR4, 8 Gb	3 100,00	2 шт.	6 200,00
Бумага для принтера формата А4, 500 л	274,00	1 уп.	274,00
<b>Итого:</b>			48 704,00

По информации поставщика компьютерных комплектующих ТЗР на их закупку составляет 300,00 руб. Расходы на материалы с учетом ТЗР равны  $C_{\text{мат}} = 48\,704,00 + 300,00 = 49\,004,00$  руб.

### 6.6.2 Расчет заработной платы

Данная статья расходов включает заработную плату научного руководителя и инженера, включая премии, входящие в фонд заработной платы. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада.

Для 6-дневной рабочей недели среднедневная тарифная заработная плата  $ЗП_{\text{ср.д}}$  рассчитывается по формуле (6.6). Данная формула учитывает, что в 2020 году 301 рабочий день и, следовательно, в месяце в среднем 25 рабочих дней при 6-дневной рабочей неделе.

$$ЗП_{\text{ср.д}} = \frac{МО}{25} \quad (6.6)$$

где  $ЗП_{\text{ср.д}}$  – средняя тарифная заработная плата;

МО – месячный оклад.

Расчет затрат на полную заработную плату приведены в таблице 6.12. Затраты времени по каждому исполнителю в рабочих днях взяты из таблицы 6.9. Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов:  $K_{\text{пр}}=1,1$ ;  $K_{\text{доп.зп}}=1,188$ ;  $K_{\text{р}}=1,3$ . Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент  $K_{\text{и}}=1,1*1,188*1,3=1,699$ .

Таблица 6.12 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб. день	Затраты времени, раб. дни	Коэффициент	Фонд з/платы, руб.
НР	35 120	1 404,80	34	1,699	81 149,67
И	16 139	645,56	113	1,699	123 939,12
<b>Итого:</b>					205 088,79

Общие затраты на заработную плату составят  $C_{\text{зп}} = 205\,088,79$  руб.

### 6.6.3 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН) в 2020 году составляет 30 % от фонда оплаты труда. ЕСН складывается из трех разновидностей платежей:

- взноса на обязательное пенсионное страхование, 22 %;
- отчислений на обязательное медицинское страхование, 5,1 %;
- отчислений на социального страхования, 2,9 %.

Затраты на единый социальный налог составят  $C_{\text{соц.}} = C_{\text{ЗП}} * 0,3 = 205\,088,79 * 0,3 = 61\,526,63$  руб.

### 6.6.4 Расчет затрат на электроэнергию

Расчет затрат на электроэнергию включает в себя затраты на электроэнергию, израсходованную в ходе выполнения проекта на работу оборудования. Затраты на электроэнергию единицей оборудования  $C_{\text{эл}}$  рассчитываются по формуле (6.7).

$$C_{\text{эл}} = P_{\text{об}} \cdot t_{\text{об}} \cdot \text{Ц}_{\text{Э}} \quad (6.7)$$

где  $C_{\text{эл}}$  – общие затраты на электроэнергию, руб.;

$P_{\text{об}}$  – мощность, потребляемая оборудованием, кВт;

$t_{\text{об}}$  – время работы оборудования, час.;

$\text{Ц}_{\text{Э}}$  – тариф на 1 кВт·ч, руб.

Мощность, потребляемая оборудованием, определяется по формуле (6.8).

$$P_{\text{об}} = P_{\text{ном.}} \cdot K_{\text{з}} \quad (6.8)$$

где  $P_{\text{об}}$  – мощность, потребляемая оборудованием, кВт;

$P_{\text{ном.}}$  – номинальная мощность оборудования, кВт;

$K_{\text{з}}$  – коэффициент загрузки,  $K_{\text{з}} \in [0.0; 1.0]$ .

Отметим, что коэффициент загрузки оборудования  $K_{\text{з}}$ , зависит от средней степени использования номинальной мощности. Для технологического оборудования малой мощности  $K_{\text{з}}=1.0$ .

Время работы оборудования вычисляется на основе итоговых данных таблицы 6.9 для инженера  $T_{\text{рд}}$  из расчета, что продолжительность рабочего дня равна 8 часов. Время работы оборудования  $t_{\text{об}}$  рассчитывается по формуле (6.9).

$$t_{об} = T_{рД} \cdot K_t \quad (6.9)$$

где  $t_{об}$  – время работы оборудования, час.;

$T_{рД}$  – продолжительность выполнения этапа работы, дн.;

$K_t$  – коэффициент использования оборудования,  $K_t \in [0.0; 1.0]$ .

Коэффициент использования оборудования  $K_t$  зависит от средней степени использования оборудования в течение рабочего дня исполнителем проекта.

Одноставочный тариф на электрическую энергию для населения и приравненным к нему категорий потребителей в Томской области в период с 01.01.2020 по 30.06.2020 составляет  $C_э = 3,50$  руб./кВт·ч (с учетом НДС) [32].

Учитывая специфику проекта в расчете учитываются расходы на работу персонального компьютера с высокопроизводительным графическим ускорителем в течении 30 суток, без простоя. Расчет затрат на электроэнергию для технологических целей приведен в таблице 6.13.

Таблица 6.13 – Затраты на электроэнергию для технологических целей

Наименование оборудования	$P_{ном.},$ кВт	$K_э$	$P_{об},$ кВт	$T_{рД},$ час	$K_t$	$t_{об},$ час	$C_э, руб./$ кВт·ч	$C_{эл}, руб.$
Персональный компьютер научного руководителя	0,39	0,6	0,234	272	0,6	163,2	3,50	133,66
Персональный компьютер инженера	0,44	0,7	0,308	904	0,8	723,2		779,61
Рабочая станция / Персональный компьютер с установленными комплектующими, приобретёнными для выполнения проекта	0,54	0,9	0,486	н.д.	н.д.	720,0		1 224,72
Струйный принтер	0,55	1,0	0,550	н.д.	н.д.	4,0		7,70
<b>Итого:</b>								2145,69

Затраты на электроэнергию для технологических целей составят  $C_{эл} = 2 145,69$  руб.

### 6.6.5 Расчет амортизационных расходов

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта. Для расчета используется формула (6.10).

$$C_{AM} = \frac{N_A \cdot C_{об} \cdot t_{рф} \cdot n}{F_d} \quad (6.10)$$

где  $C_{AM}$  – амортизационные отчисления, руб.;

$N_A$  – коэффициент годовой нормы амортизации единицы оборудования;

$C_{об}$  – балансовая стоимость единицы оборудования с учетом ТЗР, руб.;

$t_{рф}$  – фактическое время работы оборудования, час.;

$n$  – число задействованных однотипных единиц оборудования, шт.;

$F_d$  – действительный годовой фонд времени работы оборудования, час.

Согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы», электронно-вычислительная техника относится ко второй группе оборудования со сроком полезного использования от 2-х до 3-х лет [33]. Следовательно, коэффициент годовой нормы амортизации единицы электронно-вычислительной техники  $N_A \in [0,5; 0,33]$ . Расчет затрат на амортизационные отчисления приведен в таблице 6.14.

Таблица 6.14 – Затраты на амортизационные отчисления

Наименование оборудования	$N_A$	$C_{об}$ , руб.	$t_{рф}$ , час.	$n$ , шт.	$F_d$ , час	$C_{AM}$ , руб.
Персональный компьютер научного руководителя	0,4	45 500,00	163,2	1	2400	1 237,60
Персональный компьютер инженера	0,4	48 500,00	723,2	1	2400	5 845,87
Рабочая станция	0,5	42 000,00	720,0	1	8784	1 721,31
Видеокарта Asus GeForce GTX 1070 Ti, 8 Gb	0,5	38 680,00	720,0	1	8784	1 585, 25
Блок питания Zalman Wattbit, 600 Вт	0,5	3 550,00	720,0	1	8784	145,49
Оперативная память Silicon Power DDR4, 8 Gb	0,33	3 100,00	720,0	2	8784	169,40
Струйный принтер	0,5	12 000,00	4,0	1	500	48,00
<b>Итого:</b>						<b>9 167,67</b>

Затраты на амортизационные отчисления составят  $C_{AM} = 9 167,67$  руб.

### 6.6.6 Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов. Расчет прочих расходов выполняется по формуле (6.11).

$$C_{\text{проч.}} = (C_{\text{мат}} + C_{\text{ЗП}} + C_{\text{соц.}} + C_{\text{эл}} + C_{\text{АМ}}) \cdot 0,1 \quad (6.11)$$

где  $C_{\text{проч.}}$  – затраты на прочие расходы, руб.;

$C_{\text{мат}}$  – затраты на материалы с учетом ТЗР, руб.;

$C_{\text{ЗП}}$  – затраты на заработную плату, руб.;

$C_{\text{соц.}}$  – затраты на единый социальный налог, руб.;

$C_{\text{эл}}$  – затраты на электроэнергию для технологических целей, руб.;

$C_{\text{АМ}}$  – затраты на амортизационные отчисления, руб.

Так как факт покупки дорогостоящего оборудования производится в технологических целях и не влияет на возможные дополнительные затраты по проекту, исключим их стоимость из расчета затрат на прочие расходы. Затраты по данным позициям с учетом ТЗР составили 48 730,00 руб., 10 % от данной суммы, учтенные на данный момент в затратах на прочие расходы, составляющие 4873,00 руб. вычитаются из расчета. Затраты на прочие расходы составили  $C_{\text{проч.}} = (48\,704,00 + 205\,088,79 + 61\,526,63 + 2\,145,69 + 9\,167,67) \cdot 0,1 - 4\,873,00 = 27\,790,28$  руб.

### 6.6.7 Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта «Математическое и программное обеспечение локализации областей инфильтрата в области легкого по данным компьютерной томографии». Расчет затрат на реализацию проекта приведен в таблице 6.15. Затраты на реализацию проекта составят  $C = 354\,423,06$  руб.

Таблица 6.15 – Смета затрат на реализацию проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	48 704,00
Основная заработная плата	$C_{\text{ЗП}}$	205 088,79
Отчисления в социальные фонды	$C_{\text{соц.}}$	61 526,63
Расходы на электроэнергию	$C_{\text{эл}}$	2 145,69
Амортизационные отчисления	$C_{\text{АМ}}$	9 167,67
Прочие расходы	$C_{\text{проч.}}$	27 790,28
<b>Итого:</b>		<b>354 423,06</b>

### **6.6.8 Оценка прибыли**

Прибыль от непосредственной реализации текущего проекта не ожидается. Предполагается, что реализация данного проекта способна внести существенный вклад в инициализацию большего проекта, предполагающего выполнение лучевой диагностики с использованием средств искусственного интеллекта.

По оценкам BusinesStat в 2017 г. натуральный объем рынка томографической диагностики составил 10,9 млн исследований [4]. Самым распространенным видом томографических исследований в 2017 г. стала компьютерная томография (68,5 % от натурального объема рынка). Также в рамках данного исследования выяснено, что среди видов заболеваний наиболее распространенной группой являются заболевания органов дыхания: в 2017 г. доля заболеваемости по группе составила 24,8 % от общего показателя по стране. Из приведённого исследования следует, что более 1,5 млн. КТ-исследований в год выполняется с целью диагностики области легкого, следовательно, более 11 тыс. из них проводятся на территории Томской области.

Внедрение средств анализа КТ-изображений с помощью искусственного интеллекта только в задачах анализа области легкого на территории Томской области при установленной цене 250 руб. за одно автоматизированное исследование способно обеспечить прибыль в размере более 1,3 млн. руб. в год.

## **7 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

Целью данной работы является разработка математического и программного обеспечения локализации областей инфильтрата в области легкого по данным компьютерной томографии. Для разработки данного программного обеспечения осуществлялось программирование, проектирование и анализ с использованием вычислительной техники.

В качестве рабочего места рассматривается рабочее место оператора персональной электронной вычислительной машины (ПЭВМ). Использование средств вычислительной техники, накладывает целый ряд вредных факторов на человека, что впоследствии снижает производительность его труда и может привести к существенным проблемам здоровья оператора. Обеспечение производственной и экологической безопасности является необходимым условием реализации любой деятельности, в том числе конструкторской и исследовательской. В общем, обеспечение безопасности предполагает создание безопасных и благоприятных рабочих условий для лиц, задействованных в работе над проектом, а также условий, обеспечивающих экологическую безопасность окружающей среды.

Первичным этапом в задаче обеспечения безопасности труда является выявление и анализ вредных и опасных факторов труда оператора ПЭВМ, возможных причин потенциальных аварий и пожаров, производственных травм, профессиональных заболеваний. Следующими этапами в задачи обеспечения безопасности труда являются разработка мероприятий по защите вредных и опасных факторов, оценка условий труда и микроклимата рабочей среды.

Данный раздел посвящен анализу вредных и опасных факторов производственной среды для операторов ПЭВМ, разработке программ по минимизации воздействия вредоносного и опасного влияния выявленных факторов, а также программ по снижению вредных воздействий на окружающую среду, экономии невозполнимых ресурсов и защите в чрезвычайных ситуациях.

### **7.1 Правовые и организационные вопросы обеспечения безопасности.**

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Режим труда и отдыха предусматривает соблюдение определенной длительности непрерывной работы на ПК и перерывов, регламентированных с учетом продолжительности рабочей смены, видов и категории трудовой деятельности.

Вид трудовой деятельности на персональном компьютере в рамках данной работы соответствует группе В – творческая работа в режиме диалога с ПК, категория трудовой деятельности – III (до 6 часов непосредственной работы на ПК).

При 8-часовой рабочей смене и работе на ПК, соответствующей описанным выше критериям необходимо через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва устраивать регламентированные перерывы продолжительностью 20 минут каждый час или продолжительностью 15 минут через каждый час работы. Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа. Эффективными являются нерегламентированные перерывы (микропаузы) длительностью 1-3 минуты.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 2-3 недели.

Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины, одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

При работе в ночное время продолжительность рабочей смены сокращается на один час. К работе в ночную смену не допускаются беременные женщины; работники, не достигшие возраста 18 лет; женщины, имеющие детей в возрасте до трех лет, инвалиды, работники, имеющие детей-инвалидов, а также работники, осуществляющие уход за больными членами их семей в соответствии с медицинским заключением, матери и отцы-одиночки детей до пяти лет.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях [34].

## 7.2 Производственная безопасность

Для обеспечения производственной безопасности необходимо проанализировать воздействия на человека вредных и опасных производственных факторов, которые могут возникать при разработке или эксплуатации проекта. Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме. Вредные факторы характеризуются потенциальной опасностью для здоровья, в частности способствуют развитию каких-либо заболеваний, приводят к повышенной утомляемости и снижению работоспособности. При этом, вредные факторы проявляются при определенных условиях, таких как интенсивность и длительность воздействия. Опасные производственные факторы способны моментально оказать влияние на здоровье работника: привести к травмам, ожогам или к резкому ухудшению здоровья работников в результате отравления или облучения. В таблице 7.1 представлены возможные вредные и опасные факторы, возникающие при работе за ПЭВМ.

Таблица 7.1 – Вредные и опасные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
Опасные и вредные производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды на местонахождении работающего	+	+	-	СанПиН 2.2.4.528 – 96 [35]
Опасные и вредные производственные факторы, связанные с электрическим током	+	+	+	ГОСТ Р 12.1.019 – 2009 [36]
Отсутствие или недостаток необходимого естественного освещения	+	+	-	СНиП 23-05-10 [37]
Повышенный уровень и другие неблагоприятные характеристики шума	+	+	+	ГОСТ 12.1.003-83 [38]

### 7.2.1 Правовые и организационные вопросы обеспечения безопасности

Одним из необходимых благоприятных условий труда является обеспечение в помещениях нормальных условий микроклимата, оказывающих существенное влияние на тепловое самочувствие человека. Микроклимат в производственных помещениях, зависит от особенностей технологического процесса, а также внешних условий (категории работ,

времени года, условий вентиляции и отопления). К параметрам, характеризующим микроклимат в производственных помещениях, относятся:

- Температура воздуха ( $t$ , °С);
- Температура поверхностей ( $t$ , °С);
- Относительная влажность воздуха ( $\varphi$ , %);
- Скорость движения воздуха ( $v$ , м/с);
- Интенсивность теплового облучения ( $I$ , Вт/м<sup>2</sup>).

В производственных помещениях для работы с компьютером происходит постоянное выделение тепла самой вычислительной техникой, вспомогательными приборами и средствами освещения. Поскольку оператор расположен в непосредственной близости с источниками выделения тепла, то данный фактор является одним из важнейших вредных факторов производственной среды пользователя компьютера, а высокая температура воздуха способствует быстрой утомляемости и ухудшению самочувствия [39].

Влажность оказывает большое влияние на терморегуляцию организма. Так, например, высокие показатели относительной влажности (более 85 %) затрудняют терморегуляцию, снижая возможность испарения пота, низкие показатели влажности (менее 20 %) вызывают пересыхание слизистых оболочек человека [40].

Работа программиста относится к категории Ia, которые производятся сидя и сопровождаются незначительным физическим напряжением. Интенсивность энергозатрат организма для данной категории работ составляет до 120 ккал/ч (до 139 Вт). Оптимальные значения показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia представлены в таблице 7.2 [35].

Таблица 7.2 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	22 – 24	60 – 40	0,1
Теплый	Ia	21 – 23	60 – 40	0,1

Оптимальные микроклиматические условия обеспечивают общее и локальное ощущение теплового комфорта при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

В таблице 7.3 приведены допустимые величины показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia [35].

Таблица 7.3 – Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	20 – 25	15 – 75	0,1
Теплый	Ia	21 – 28	15 – 75	0,1 – 0,2

Допустимые микроклиматические условия не вызывают повреждений или нарушений состояния здоровья, но могут приводить к возникновению общих и локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности.

Согласно требованиям, СанПиН 2.2.2/2.4.1340–03, в кабинете поддерживается температура равная 19 – 20 °С, при относительной влажности в 55 – 58 %. Для этого в помещении проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ [41].

## 7.2.2 Опасность поражения электрическим током

Повышенное значение напряжения в электрической цепи, замыкание которой, может произойти через тело человека – опасный фактор физической природы. При работе с компьютером человеку грозит поражение электрическим током в случае короткого замыкания. Это может привести к появлению ожогов, механическим повреждениям тканей, нагреву сосудов, раздражающим воздействиям на ткани.

Опасность поражения человека электрическим током оценивается величиной тока I (А), проходящего через его тело, или напряжением прикосновения U (В). Степень опасного воздействия на человека электрического тока зависит от рода и величины напряжения тока, частоты электрического тока, пути тока через тело человека, продолжительности его воздействия на организм человека, а также условий внешней среды.

Электрический ток, протекая через тело человека, производит термическое, механическое и световое воздействие – электролитическое разложение жидкости (в том числе и крови), судорожное сокращение мышц, разрыв тканей и поражение глаз.

Работа с компьютером является опасной с точки зрения поражения током, так как практически во всех частях компьютера течет электрический ток. Поражение электрическим током при работе за компьютером возможно при наличии оголенных участков на кабеле, нарушении изоляции распределительных устройств и от токоведущих частей компьютера в случае их пробоя и нарушении изоляции, при работе с компьютером во влажной одежде или влажными руками.

Помещение, где расположено рабочее место человека, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

– К мероприятиям по предотвращению возможности поражения электрическим током относятся:

– При производстве монтажных работ необходимо использовать только исправный инструмент, аттестованный службой КИПиА;

– С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;

– При включенном сетевом напряжении работы на задней панели должны быть запрещены;

– Все работы по устранению неисправностей должен производить квалифицированный персонал;

– Необходимо постоянно следить за исправностью электропроводки.

Согласно ГОСТ 12.1.038-82 на рабочем месте программиста допускаются уровни напряжений прикосновения и токов, представленные в таблице 7.4 [36].

Таблица 7.4 – Предельно допустимые напряжения прикосновения и токи

Род тока	Напряжение прикосновения, В	Ток, мА
	не более	
Переменный, 50Гц	2,0	0,3
Постоянный	8,0	1,0

Значения напряжения прикосновения и токов приведены при продолжительности не более 10 минут в сутки.

### 7.2.3 Отсутствие или недостаток освещения

Рабочее помещение должно иметь как естественное, так и искусственное освещение. Коэффициент естественного освещения должен быть не менее 1,2%. Согласно СанПиН 2.2.2/2.4.1340-03 освещенность на поверхности рабочего стола в зоне размещения документа должна быть 300 – 500 лк., что может достигаться установкой местного освещения, не создающего бликов на поверхности экрана. Освещенность поверхности экрана не должна превышать 300 лк. Яркость светящихся поверхностей (окон, светильников), находящихся в поле зрения должна быть не более 200 кд/м<sup>2</sup>. Для источников искусственного освещения следует применять люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). Коэффициент пульсации при работе с компьютером не должен превышать 5 %. Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура) за счет правильного выбора и расположения светильников, яркость бликов на экране не должна превышать 40 кд/м<sup>2</sup>. Светильники местного освещения должны иметь неприсвечивающий отражатель.

Рациональное освещение помещений и рабочих мест – одно из важнейших условий создания благоприятных и безопасных условий труда. Нерационально организованное освещение может явиться причиной травматизма: плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени и пульсации освещенности ухудшают видимость и могут вызвать неадекватное восприятие наблюдаемого объекта.

Искусственное освещение предусматривается в помещениях, в которых испытывается недостаток естественного света, а также для освещения помещения в те часы суток, когда естественная освещенность отсутствует. По принципу организации искусственное освещение можно разделить на два вида [37, 42]:

– Общее освещение. Предназначено для освещения всего помещения, оно может быть равномерным или локализованным. Общее равномерное освещение создает условия для выполнения работ в любом месте освещаемого пространства. При общем локализованном освещении светильники размещают в соответствии с расположением оборудования, что позволяет создавать повышенную освещенность на рабочих местах;

– Комбинированное освещение состоит из общего и местного. Его целесообразно устраивать при работах высокой точности, а также при необходимости создания в процессе работы определенной направленности светового потока.

Местное освещение предназначено для освещения только рабочих поверхностей и не создает необходимой освещенности на прилегающих к ним участках. Оно может быть стационарным и переносным [42].

Расчетная схема для основных расчетных параметров определяющих размещение светильника в помещении представлена на рисунке 7.1. Размещение светильника в помещении определяется следующими параметрами:

- $H$  – высота помещения;
- $h_c$  – расстояние светильников от перекрытия (свес);
- $h_n = H - h_c$  – высота светильника над полом, высота подвеса;
- $h_{рп}$  – высота рабочей поверхности над полом;
- $h = h_n - h_{рп}$  – расчётная высота, высота светильника над рабочей поверхностью;

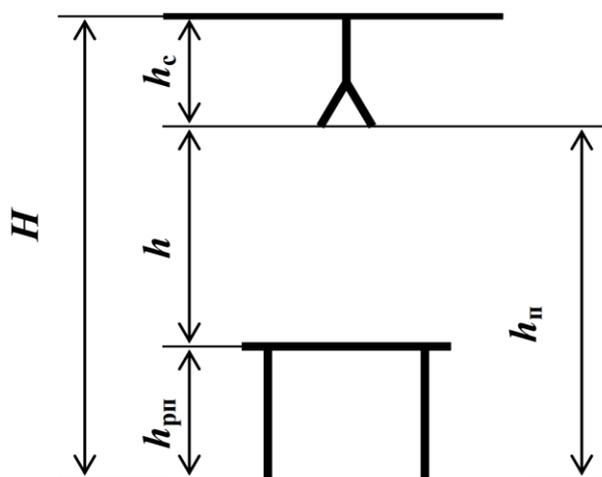


Рисунок 7.1 – Основные расчетные параметры освещения

Для освещения в помещении используются люминесцентные лампы. Схема размещения светильников в помещении приведена на рисунке 7.2, характеристики рассматриваемого помещения приведены в таблице 7.5.

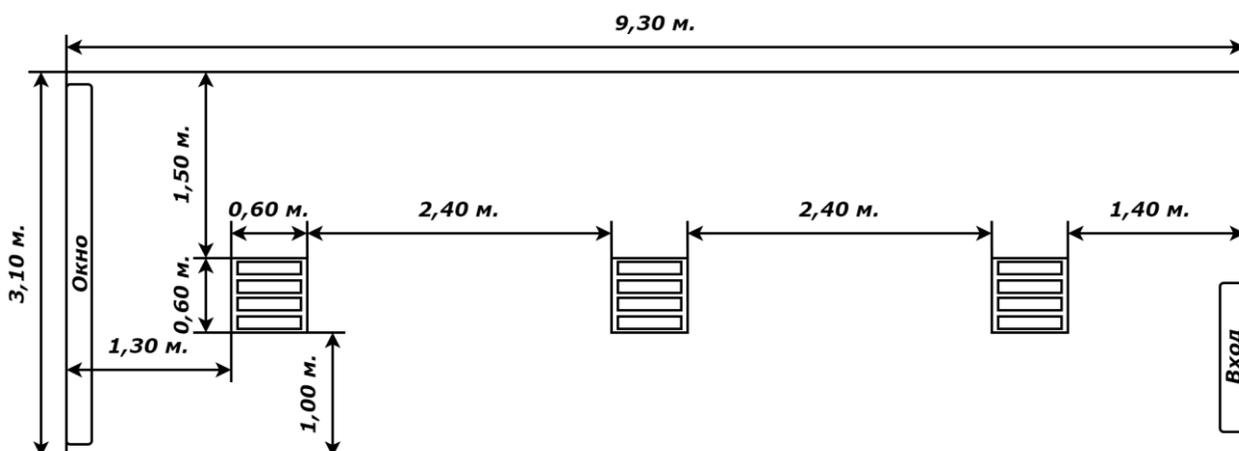


Рисунок 7.2 – План размещения общего освещения (вид сверху)

Таблица 7.5 – Исходные характеристики помещения

Размеры помещения, мм			высота рабочей поверхности над полом, мм
Длина	Ширина	Высота	
9300	3100	2400	720

В связи с тем, что в рассматриваемом помещении используется встраиваемый монтаж светильников, расстояние до светильников от перекрытия  $h_c = 0$  мм.

Так как высота всех рабочих поверхностей над полом  $h_{рп} = 720$  мм, расчётная высота светильника над рабочими поверхностями составила  $h = 1680$  мм.

Расстояние между соседними светильниками составило  $L = 2400$  мм.

Исходя из схемы размещения освещения приведенной на рисунке 7.2, рассчитано среднее расстояние от крайних светильников до стены  $l_{ср.} = (1300 + 1500 + 1400 + 1000)/4 = 1300$  мм.

Расчет параметров освещения осуществляется для светильников с зеркальной решеткой, встраиваемым в потолок типа Армстронг – ЛВО 4x18. Все светильники в рассматриваемом помещении, укомплектованы одинаковыми люминесцентными лампами. Характеристики используемых светильников приведены в таблице 7.6.

Таблица 7.6 – Характеристика используемых светильников

Количество и мощность лампы, Вт	Световой поток, лм.	Коэффициент пульсации, %	Размеры, мм		
			Длина	Ширина	Высота
4 x 18	3280	< 2	600	600	0

Расчёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента светового потока, учитывающим световой поток, отражённый от потолка и стен. Требуемый световой поток лампы накаливания или группы люминесцентных ламп светильника определяется по формуле (7.1) [43].

Коэффициент использования светового потока показывает какая часть светового потока лампы попадает на рабочую поверхность. Он зависит от индекса помещения  $i$ , типа светильника, высоты светильников над рабочей поверхностью  $h$  и коэффициента отражения стен  $\rho_c$  и потолка  $\rho_{п.}$ . Индекс помещения определяется по формуле (7.2) [43].

$$\Phi = \frac{E_H \cdot S \cdot K_3 \cdot Z}{N \cdot \eta} \quad (7.1)$$

где  $\Phi$  – Световой поток источника света, лм;

$E_H$  – нормируемая минимальная освещенность по СНиП 23-0595, лк;

$S$  – площадь освещаемого помещения, м<sup>2</sup>;

$K_3$  – коэффициент запаса, учитывающий загрязнение светильника;

$Z$  – коэффициент неравномерности освещения, отношение  $E_{ср}/E_{мин.}$ ;

$N$  – число ламп в помещении;

$\eta$  – коэффициент использования светового потока.

$$i = \frac{S}{h * (A + B)} \quad (7.2)$$

где  $i$  – индекс помещения.

$S$  – площадь освещаемого помещения, м<sup>2</sup>.

$h$  – допустимая высота подвеса светильников с люминесцентными лампами;

$A$  – ширина подвеса, м;

$B$  – длина подвеса, м;

Работа оператора ЭВМ относится к третьему разряду зрительных работ и считается работой высокой точности. Согласно СНиП 23-05-95 норма освещенности для рассматриваемого помещения при системе комбинированного освещения оставляет  $E_H = 400$  лк.

Площадь освещаемого помещения  $S = 9,3 \text{ м.} \cdot 3,1 \text{ м.} = 28,83 \text{ м}^2$ .

Коэффициент запаса  $K_3$  согласно таблице коэффициентов запаса светильников люминесцентных ламп приведенной в [43]  $K_3 = 1,5$ .

Коэффициент неравномерности освещения для люминесцентных ламп при расчетах берется равным  $Z = 1,1$  [43].

Число ламп в помещении согласно схеме приведенной на рисунке 7.2  $N = 3$ .

Согласно значениям коэффициентов отражения потолка и стен приведенным в [43] оценочно принято, что коэффициент отражения стен  $\rho_c = 30 \%$  и потолка  $\rho_{п} = 65 \%$ .

Индекс помещения  $i$  рассчитан по формуле (7.2):

$$i = \frac{9,30 * 3,10}{1,68 * (9,30 + 3,10)} = 1,38$$

Ближайшим аналогом для рассматриваемого светильника согласно [43] (основные характеристики некоторых светильников с люминесцентными лампами) является светильник типа ОД–2-40. По таблице коэффициентов использования светового потока светильников с люминесцентными лампами определен коэффициент использования светового потока  $\eta = 0,5$  [43].

Требуемое значение светового потока светильника с люминесцентными лампами светильника определено по формуле (7.1):

$$\Phi_{\text{л.расч}} = \frac{400 \text{ лк} \cdot 28,83 \text{ м}^2 \cdot 1,5 \cdot 1,1}{3 \cdot 0,5} = 12\,685 \text{ лм}$$

Согласно параметрам используемого в рассматриваемом помещении светильника, параметры которого приведенным в таблице 7.6, световой потока используемого светильника  $\Phi_{\text{исп.}} = 3\,280 \text{ лм}$ . Отклонение светового потока в рассматриваемом помещении от требуемого составило минус 287% :

$$\frac{\Phi_{\text{исп.}} - \Phi_{\text{л.расч}}}{\Phi_{\text{исп.}}} = \frac{3\,280 \text{ лм} - 12\,685 \text{ лм}}{3\,280 \text{ лм.}} \cong -287\%$$

Исходя из формулы (7.1) текущая освещенность помещения составляет  $E_{\text{исп.}} = 103 \text{ лк.}$ , что является серьезнейшим нарушением норм предписываемых СНиП 23-05-95:

$$\frac{\Phi_{\text{л.расч}}}{E_{\text{н}}} = \frac{12\,685 \text{ лм.}}{400 \text{ лк.}} = \frac{3\,280 \text{ лм.}}{E_{\text{исп.}}}; E_{\text{исп.}} \cong 103 \text{ лк}$$

Для устранения данного нарушения необходимо произвести монтаж дополнительных светильников того же типа. Требуемое количество светильников для рассматриваемого помещения определено исходя из формулы (7.1):

$$N_{\text{треб.}} = \frac{\Phi_{\text{л.расч}} \cdot N_{\text{исп.}}}{\Phi_{\text{исп.}}} = \frac{12\,685 \text{ лм} \cdot 3}{3\,280 \text{ лм}} \cong 11,6$$

С целью устранения нарушения предлагается увеличить количество светильников до 12 штук. Согласно [43] наилучшим вариантом равномерного размещения светильников является шахматное размещение. Произвести их монтаж предлагается согласно схеме приведенной на рисунке 7.3.

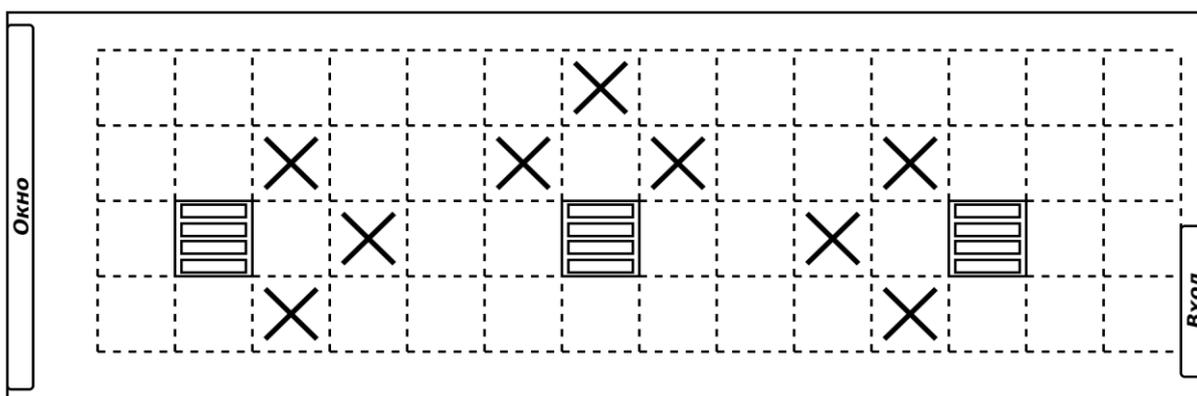


Рисунок 7.3 – План размещения дополнительных светильников.

При использовании 12-и светильников в рассматриваемом помещении, расчетное значение требуемого светового потока для одного светильника составит:

$$\Phi_{\text{л.расч}} = \frac{400 \text{ лк} \cdot 28,83 \text{ м}^2 \cdot 1,5 \cdot 1,1}{12 \cdot 0,5} = 3\,171 \text{ лм.}$$

Отклонение светового потока, при использовании 12 светильников типа ЛВО 4x18, от расчетного значения составит 3 %. Данное значение удовлетворит допустимым отклонениям в диапазоне (–10 %; +20 %).

$$\frac{\Phi_{\text{исп.}} - \Phi_{\text{л.расч}}}{\Phi_{\text{исп.}}} = \frac{3\,280 \text{ лм.} - 3\,171 \text{ лм.}}{3\,280 \text{ лм.}} \cong 3\%$$

Мощность осветительной установки составит:

$$P = 12 \cdot 4 \cdot 18 \text{ Вт} = 864 \text{ Вт}$$

## 7.2.4 Производственные шумы

Шум – это совокупность различных звуков, возникающих в процессе производства и неблагоприятно воздействующих на организм [40]. Шум может привести к нарушениям слуха (в случае постоянного нахождения при шуме более 85 децибел(dB)), может являться фактором стресса и повысить систолическое кровяное давление. Дополнительно, он может способствовать несчастным случаям, маскируя предупреждающие сигналы и мешая сконцентрироваться.

Для рассматриваемого помещения основными источниками шума являются комплектующие персонального компьютера (ПК). Нормативным документом, регламентирующим уровни шума для различных категорий рабочих мест служебных

помещений, является ГОСТ 12.1.003-83 «ССБТ. Шум. Общие требования безопасности» [38]. Помещения, в которых для работы используются ПК не должны граничить с помещениями, в которых уровни шума превышают нормируемые значения. В помещениях, оборудованных ПК, которые являются основным источником шума, при выполнении данных видов работ, уровень шума на рабочем месте не должен превышать 50 дБА [38].

Для снижения уровня шума могут быть использованы следующие средства:

- Низкошумные устройства вентиляции и кондиционирования в помещении;
- Звукопоглощающий корпус ПК;
- Звукоизолирующая прокладка для корпуса ПК;
- Низкошумные вентиляторы охлаждения корпуса ПК.

### **7.3 Экологическая безопасность**

Разработка математического и программного обеспечения локализации областей инфильтрации в области легкого происходит в офисном(нежилом) помещении. Офис является источником следующих видов отходов:

- Твердые отходы бумага, канцелярские принадлежности, комплектующие;
- Жидкие отходы, сточные воды;
- Люминесцентные лампы.

При использовании компьютера и другой вычислительной техники неизбежно устаревание моделей. В результате этого возникает необходимость замены и обновления оборудования. Чтобы уменьшить негативное воздействие на экологическую обстановку, утилизировать электронику нужно наиболее безопасным для окружающей среды способом, то есть обратиться в специализированную компанию по утилизации. Такие компании действуют на всей территории Российской Федерации, в том числе и в Томской области.

Бумага может быть переработана и использована в качестве вторсырья. Для сбора макулатуры в России существуют специальные пункты приема. Некоторые из них предоставляют услугу вывоза макулатуры.

Сточные воды – один из источников загрязнения природных водоемов, так как они содержат различные загрязнения, в том числе мусор и примеси. В результате деятельности офисного предприятия происходит образование сточных вод. Для последующей очистки в системах канализации применяются отстаивание и фильтрация. Возможна дополнительная очистка с использованием озонаторов и ультрафиолета.

Отдельного внимания заслуживает вопрос утилизации люминесцентных ламп. Они покрыты люминесцентным веществом, имеют стеклянную оболочку и электроды. Внутри

таких ламп находится инертный газ с парами ртути. В случае повреждения корпуса лампы, пары ртути попадают в атмосферу. Поэтому после окончания срока службы люминесцентные лампы необходимо сдавать на специальные предприятия по утилизации, имеющие специальную лицензию на данный вид деятельности. В Томске к таким предприятиям относятся: ООО "НПП "Экотом", АО "Полигон", ООО "Утилитсервис".

Необходимо отметить, что в целом при работе с компьютером существенного загрязнения окружающей среды не происходит и вредные выбросы не сравнимы с производственными.

## **7.4 Безопасность в чрезвычайных ситуациях**

### **7.4.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения**

Возможными чрезвычайными ситуациями могут быть:

– Техногенные:

- а) взрывы;
- б) пожары;
- в) обрушение помещений;
- г) аварии на системах жизнеобеспечения.

– Природные:

- а) наводнения;
- б) ураганы;
- в) бури;
- г) природные пожары.

– Экологические:

- а) разрушение озонового слоя;
- б) кислотные дожди.

– Биологические:

- а) эпидемии;
- б) пандемии.

– Антропогенные:

- а) война;
- б) терроризм.

Общие правила поведения при чрезвычайных ситуациях:

- 4) Не паниковать и не поддаваться панике. Призывать окружающих к спокойствию;
- 5) По возможности немедленно позвонить по телефону «01», сообщить что случилось, указать точный адрес места происшествия, назвать свою фамилию и номер своего телефона;
- 6) Включить устройства передачи звука (радио, телевизор), а также прослушать информацию, передаваемую через уличные громкоговорители и громкоговорящие устройства. В речевом сообщении будут озвучены основные рекомендации и правила поведения;
- 7) Выполнять рекомендации специалистов (сотрудников полиции, медицинских работников, пожарных, спасателей);
- 8) Не создавать условия, которые препятствуют и затрудняют действия сотрудников полиции, медицинских работников, спасателей, пожарных. Наиболее характерной для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, чрезвычайной ситуацией является пожар;

Причинами возникновения данного вида ЧС могут являться:

- Возникновением короткого замыкания в электропроводке;
- Возгоранием устройств ПЭВМ из-за неисправности аппаратуры;
- Возгоранием устройств искусственного освещения;
- Возгоранием мебели по причине нарушения правил пожарной безопасности, а также неправильного использования дополнительных бытовых электроприборов и электроустановок;

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [44].

#### **7.4.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий**

Пожарная безопасность подразумевает надлежащее состояние объекта с исключением возможности возникновения очага возгорания (пожара) и его распространения в пространстве. Обеспечение пожарной безопасности – приоритетная задача для любого предприятия. Создание системы защиты регламентировано законом и нормативными документами различных ведомств.

Каждый сотрудник организации должен быть ознакомлен с инструкцией по пожарной безопасности, пройти инструктаж по технике безопасности и строго соблюдать его.

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией. Электроустановки и бытовые электроприборы в помещениях по окончании рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легковоспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

Перед уходом из служебного помещения работник обязан провести его осмотр, закрыть окна, и убедиться в том, что в помещении отсутствуют источники возможного возгорания, все электроприборы отключены и выключено освещение. С периодичностью не реже одного раза в три года необходимо проводить замеры сопротивления изоляции токоведущих частей силового и осветительного оборудования.

Работник при обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) должен:

- Немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;
- Принять по возможности меры по эвакуации людей и материальных ценностей;
- Отключить от сети закрепленное за ним электрооборудование;
- Приступить к тушению пожара имеющимися средствами пожаротушения;
- Сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;
- При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. При организации производства следует руководствоваться

правилами противопожарного режима, регламентирующими требования к пожарной безопасности [45].

### **7.4.3 Организационные мероприятия при компоновке рабочей зоны**

Большое значение для профилактики статических физических перегрузок имеет правильная организация рабочего места человека, работающего с компьютером. Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда. Оно должно удовлетворять следующим требованиям:

- Обеспечивать возможность удобного выполнения работ;
- Учитывать физическую тяжесть работ;
- Учитывать размеры рабочей зоны и необходимость передвижения в ней работающего;
- Учитывать технологические особенности процесса выполнения работ;

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место программиста должно соответствовать требованиям СанПин 2.2.2/2.4.1340-03.

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м. Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю. Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали). Рабочие места с компьютерами при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 – 2,0 м.

## 7.5 Заключение по разделу

В разделе проанализированы вредные и опасные производственные факторы. Было установлено, что офисное помещение, в котором производилась разработка математического и программного обеспечения локализации областей инфильтрата в области легкого, не соответствует нормам СНиП и СанПиН по причине недостаточности освещения рабочего места оператора ЭВМ. Для устранения данного нарушения необходимо произвести дополнительный монтаж 9-и светильников того же типа, что и используются в офисном помещении в данный момент. Других нарушений не выявлено.

При работе в офисном помещении производится утилизация отходов: бумага, канцелярские принадлежности, люминесцентные лампы и т.д. При надлежащей утилизации этих отходов (с помощью специальных фирм, имеющих лицензию на осуществление утилизации) загрязнение окружающей среды сведено к минимуму.

Рассмотрена наиболее распространенная чрезвычайная ситуация – пожар. Пожар может быть следствием короткого замыкания или неверной эксплуатации электроприборов, а также несоблюдения техники безопасности. В связи с этим сотрудники проходят инструктаж по правилам пожарной безопасности, проходят учебные эвакуации.

Правовые и организационные вопросы обеспечения безопасности регулируются государственными органами. Правила и нормы для обеспечения нормальных условий труда всех сотрудников устанавливаются на государственном уровне.

## ЗАКЛЮЧЕНИЕ

Разработанные методы первичной подготовки биомедицинских изображений позволили обеспечить единообразное представление входных данных, подлежащих алгоритмической обработке. Предложенная концепция обеспечивает алгоритму возможность формировать правила извлечения областей интереса, учитывая не только их взаимное расположение, но и физическое расстояние между ними.

Основываясь на исследовании лучших практик, предлагаемых в последние годы при решении задач глубокого машинного обучения, разработана и реализована уникальная архитектура сверточной нейронной сети. Разработанная модель работает с трехмерным представлением обрабатываемых фрагментов КТ-снимка, данный подход позволяет формировать объемные пространственные связи между извлекаемыми признаками во внутреннем представлении ИНС. Ключевой концептуальной особенностью предлагаемой модели является внедрение идеи остаточного обучения ResNet в сжимающий путь U-net подобной сети.

Для решения задач аугментации графических наборов данных разработан и впервые предложен системный подход к реализации пространственных искажений. Реализация предложенного алгоритма аугментации данных, позволила качественно расширить весьма скромный набор данных использованный при обучении ИНС.

Реализованный алгоритм обработки объемных изображений скользящим окном,двигающимся с шагом в половину своих размеров, выполняет восьмикратную нейросетевую оценку в отношении каждого вокселя КТ-снимка. Использование фильтра Гаусса позволило минимизировать негативные эффекты, проявляющиеся в результате смещения скользящего окна.

Реализованные методы были применены к базе данных КТ-снимков пациентов с подтвержденным диагнозом COVID-19. Полученные результаты нейросетевой сегментации свидетельствует о высоком уровне качества локализации зон инфильтративных уплотнений в области легкого.

Разработанное программное обеспечение может стать основой для комплексного нейросетевого решения, выполняющего интеллектуальный анализ результатов лучевой диагностики, с целью выявления, локализации и классификации характерных признаков для различных инфекционных и соматических заболеваний.

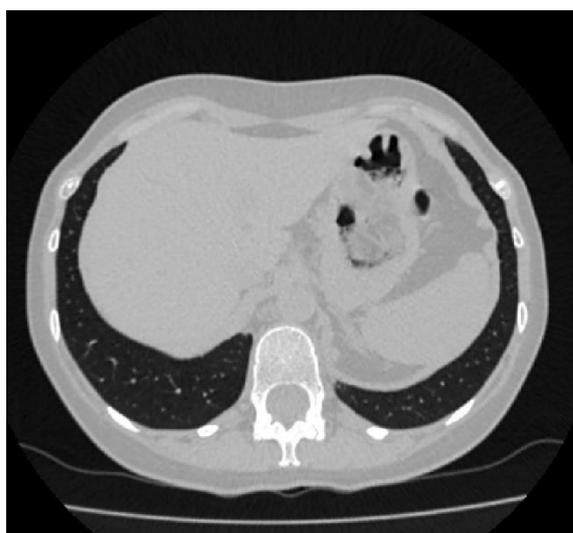
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вопросы и ответы: сходства и различия возбудителей COVID-19 и гриппа [Электронный ресурс] // Всемирная организация здравоохранения: [сайт]. [2020]. URL: <https://www.who.int/ru/news-room/q-a-detail/q-a-similarities-and-differences-covid-19-and-influenza> (дата обращения: 30.04.2020).
2. Трофимова Т.Н., Лукина О.В., Сперанская А.А., Ильина Н.А., Гаврилов П.В., Панунцева К.К. Лекция: Коронавирусная инфекция COVID-19. Часть 5. Лучевые методы исследования при COVID-19 и вирусных пневмониях. Санкт-Петербург: Первый Санкт-Петербургский государственный медицинский университет им. И.П. Павлова, 2020. 41 с.
3. What is multidrug-resistant tuberculosis (MDR-TB) and how do we control it? [Электронный ресурс] // World Health Organization: [сайт]. [2018]. URL: [https://www.who.int/news-room/q-a-detail/what-is-multidrug-resistant-tuberculosis-\(mdr-tb\)-and-how-do-we-control-it](https://www.who.int/news-room/q-a-detail/what-is-multidrug-resistant-tuberculosis-(mdr-tb)-and-how-do-we-control-it) (дата обращения: 16.02.2020).
4. BusinesStat. Анализ рынка томографической диагностики в России в 2015-2019 гг, оценка влияния коронавируса и прогноз на 2020-2024 гг // BusinesStat - готовые обзоры рынков. 2020. URL: <https://businesstat.ru/catalog/id75069/>
5. Litjens G. A survey on deep learning in medical image analysis // Medical Image Analysis, No. 42, 2017. pp. 60-88.
6. International standard to transmit, store, retrieve, print, process, and display medical imaging information [Электронный ресурс] // Digital Imaging and Communications in Medicine: [сайт]. URL: <https://www.dicomstandard.org> (дата обращения: 28.01.2020).
7. NIfTI-1 format description [Электронный ресурс] // Neuroimaging Informatics Technology Initiative: [сайт]. URL: <https://nifti.nimh.nih.gov> (дата обращения: 06.02.2020).
8. The Medical Imaging Interaction Toolkit (MITK) [Электронный ресурс] [2019]. URL: <http://mitk.org/wiki/MITK> (дата обращения: 04.02.2020).
9. LeCun Y. Gradient-based learning applied to document recognition // Proc. of the IEEE, Vol. 86, No. 11, 1998. pp. 2278-2324.
10. Николенко С., Кадурич А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. Питер, 2018. 481 с.
11. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv preprint, 2015. arXiv:1505.04597.
12. Çiçek Ö., Abdulkadir A., Lienkamp S.S. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. arXiv preprint, 2016. arXiv:1606.06650.
13. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint, 2015. arXiv:1502.03167.
14. Kaiming H., Xiangyu Z., Shaoqing R., Jian S. Deep Residual Learning for Image Recognition. arXiv preprint, 2015. arXiv:1512.03385.
15. Kaiming H., Xiangyu Z., Shaoqing R., Jian S. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. arXiv preprint, 2015. arXiv:1502.01852.
16. PyTorch [Электронный ресурс] // Open source machine learning framework: [сайт]. [2020]. URL: <https://pytorch.org> (дата обращения: 10.02.2020).
17. Simard P.Y., Steinkraus D., Platt J.C. Best practices for convolutional neural networks applied to visual document analysis. // Proceedings of International Conference on Document Analysis and Recognition. 2003.

18. Ciregan D., Meier U., Schmidhuber J. Multicolumn deep neural networks for image classification // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2012. pp. 3642–3649.
19. Wan L., Zeiler M., Zhang S., Le Cun Y., Fergus R. Regularization of neural networks using dropconnect // In International Conference on Machine Learning. 2013. pp. 1058-1066.
20. Sato I., Nishimura H., Yokoi K. Apac: Augmented pattern classification with neural networks. arXiv preprint, 2015. arXiv:1505.03229.
21. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems. 2012.
22. DeVries T., Taylor G.W. Improved regularization of convolutional neural networks with cutout. arXiv preprint, 2017. arXiv:1708.04552.
23. Zhong Z., Zheng L., Kang G., Li S., Yang Y. Random erasing data augmentation. arXiv preprint, 2017. arXiv:1708.04896.
24. NumPy [Электронный ресурс] // NumPy is the fundamental package for scientific computing with Python: [сайт]. [2020]. URL: <https://numpy.org/> (дата обращения: 05.01.2020).
25. NiBabel [Электронный ресурс] // NIPY community: [сайт]. [2019]. URL: <https://nipy.org/nibabel/> (дата обращения: 12.02.2020).
26. Набор данных MosMedData: COVID19\_1110 [Электронный ресурс] // Центр диагностики и телемедицины: [сайт]. [2020]. URL: <https://mosmed.ai> (дата обращения: 18.05.2020).
27. Елисеева И.И., Рукавишников В.О. Группировка, корреляция, распознавание образов: (статистические методы классификации и измерения связей). Москва: Статистика, 1977. 143 с.
28. Diederik P.K., Jimmy B. Adam: A Method for Stochastic Optimization. arXiv preprint, 2015. arXiv:1412.6980.
29. Илясова Е.Б., Чехонацкая М.Л., Приезжева В.Н. Лучевая диагностика: учебное пособие. Москва: ГЭОТАР-Медиа, 2009. 280 с.
30. Multidimensional Gaussian filter [Электронный ресурс] // SciPy: open-source software for mathematics, science, and engineering: [сайт]. [2019]. URL: [https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian\\_filter.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian_filter.html) (дата обращения: 05.05.2020).
31. Конотопский В.Ю. Методические указания к выполнению раздела "Финансовый менеджмент, ресурсоэффективность и ресурсосбережение" для всех специальностей. Томск: Изд-во Томского политехнического университета, 2015. 29 с.
32. Департамент тарифного регулирования Томской области. Приказ №6-585. О тарифах на электрическую энергию для населения и приравненных к нему категорий потребителей Томской области на 2020 год. Томск. 2019. 8 с.
33. Правительство Российской Федерации. Постановление Правительства РФ от 01.01.2002 N 1 (ред. от 27.12.2019) "О Классификации основных средств, включаемых в амортизационные группы".
34. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020) // Сборник основных федеральных законов РФ - 24.04.2020.
35. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений [Электронный ресурс] // Электронный фонд правовой и нормативно-технической документации: [сайт]. URL: <http://docs.cntd.ru/document/901704046> (дата обращения: 26.05.2020).

36. ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ).  
Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов (с Изменением N 1) [Электронный ресурс] // Электронный фонд правовой и нормативно-технической документации: [сайт]. URL: <http://docs.cntd.ru/document/5200313> (дата обращения: 26.05.2020).
37. Приказ Минстроя России от 07.11.2016 N 777/пр. "СНиП 23-05-95\* Естественное и искусственное освещение" (Изменение № 1).
38. ГОСТ 12.1.003-83. «ССБТ. Шум. Общие требования безопасности» [Электронный ресурс] // Электронный фонд правовой и нормативно-технической документации: [сайт]. URL: <http://docs.cntd.ru/document/5200291> (дата обращения: 26.05.2020).
39. Ефремова О.С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. 2-е изд., перераб. и доп.-е изд. «Альфа-Пресс», 2008.
40. Назаренко О.Б., Амелькович Ю.А. Безопасность жизнедеятельности: учебное пособие. 3-е изд., перераб. и доп.-е изд. Томск: Томский политехнический университет, 2013.
41. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [Электронный ресурс] // Электронный фонд правовой и нормативно-технической документации: [сайт]. URL: <http://docs.cntd.ru/document/901865498> (дата обращения: 26.05.2020).
42. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*. Свод правил естественное и искусственное освещение. 2017.
43. Назаренко О.Б. Безопасность жизнедеятельности. Расчёт искусственного освещения. Методические указания к выполнению индивидуальных заданий для студентов дневного и заочного обучения всех специальностей. Томск: ТПУ, 2008.
44. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности (с Изменением N 1) [Электронный ресурс] // Электронный фонд правовой и нормативно-технической документации: [сайт]. URL: <http://docs.cntd.ru/document/1200071156> (дата обращения: 26.05.2020).
45. Правительство Российской Федерации. Постановление Правительства РФ от 25 апреля 2012 г. N 390 (ред. от 23.04.2020 г.) «О противопожарном режиме» // Собрание законодательства РФ, 2020.
46. NiBabel [Электронный ресурс] // Access a sacophony of neuro-imaging file formats: [сайт]. [2019]. URL: <https://nipy.org/> (дата обращения: 07.01.2020).

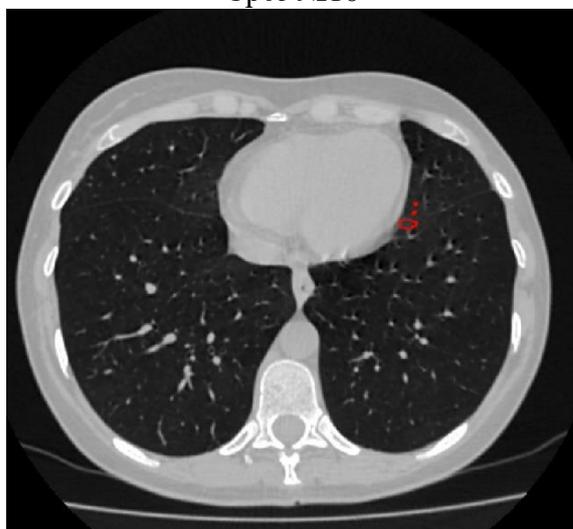
Приложение А  
Результат нейросетевой сегментации областей интереса



Срез №10



Срез №14



Срез №17



Срез №23

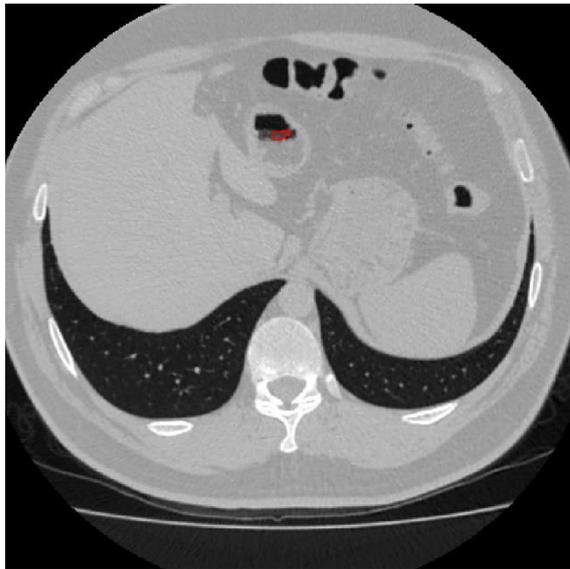


Срез №28



Срез №33

Рисунок 1 – Нейросетевая сегментация областей интереса. Класс СТ-0. Пациент № 0172.



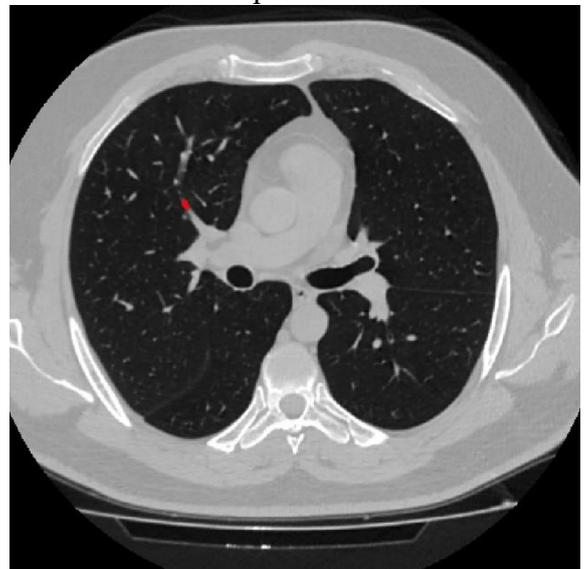
Срез № 8



Срез № 9



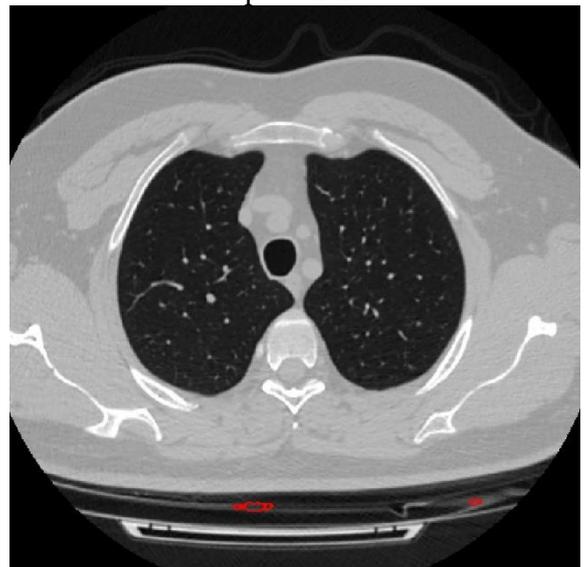
Срез № 17



Срез № 27

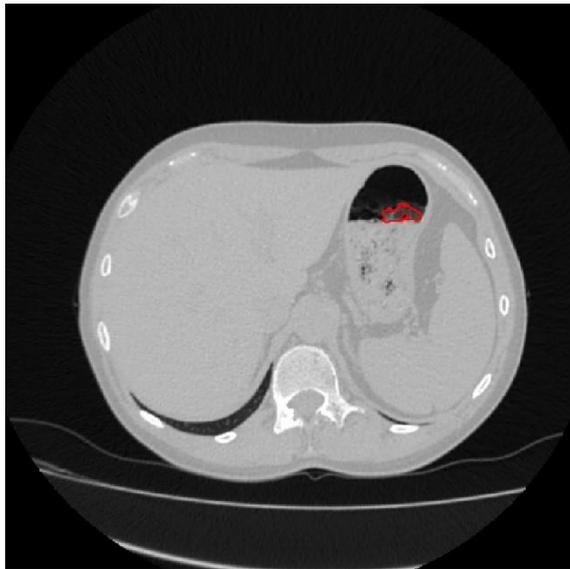


Срез № 30

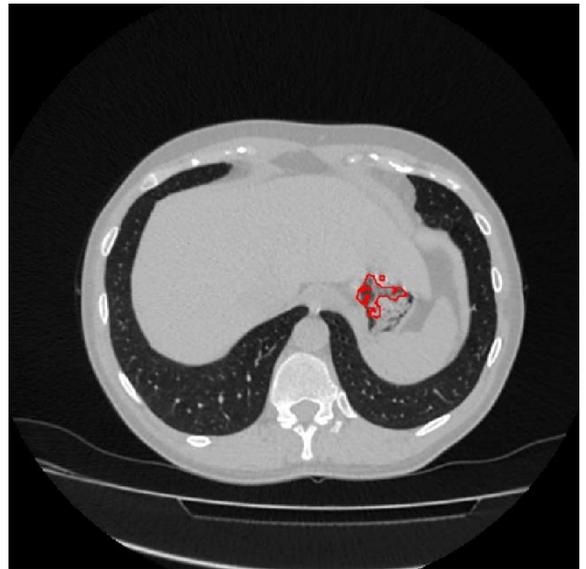


Срез № 36

Рисунок 2 – Нейросетевая сегментация областей интереса. Класс СТ-0. Пациент № 0113.



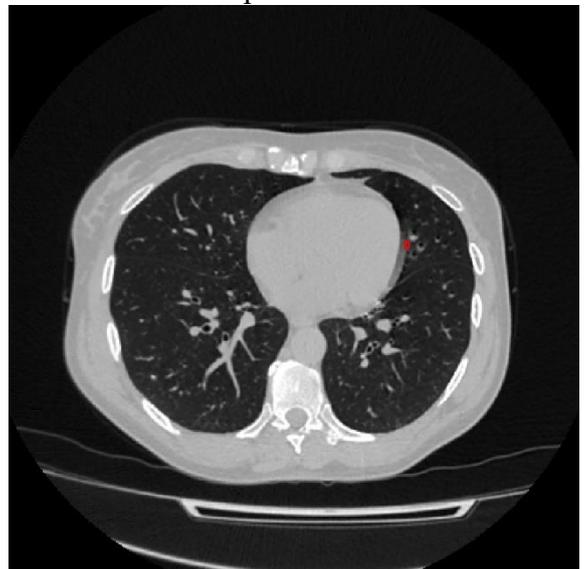
Срез № 9



Срез № 13



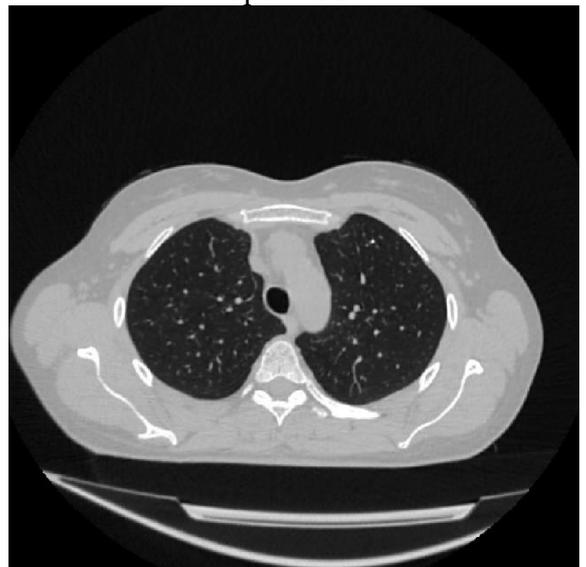
Срез № 14



Срез № 20

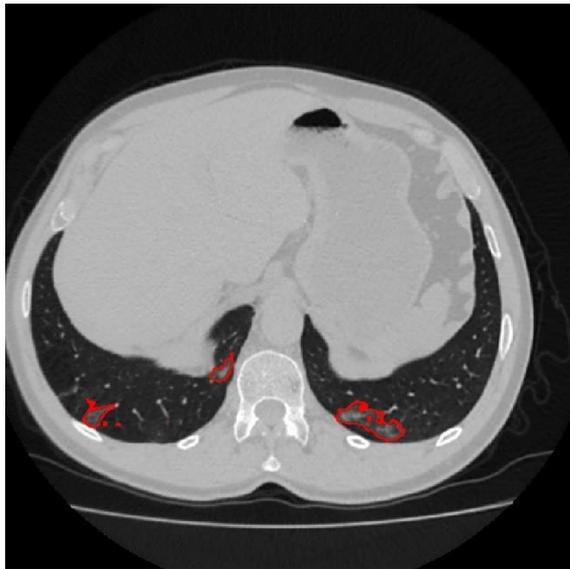


Срез № 28

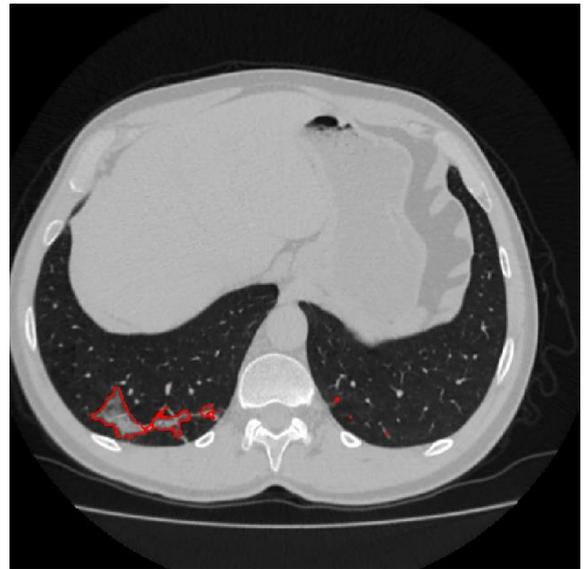


Срез № 33

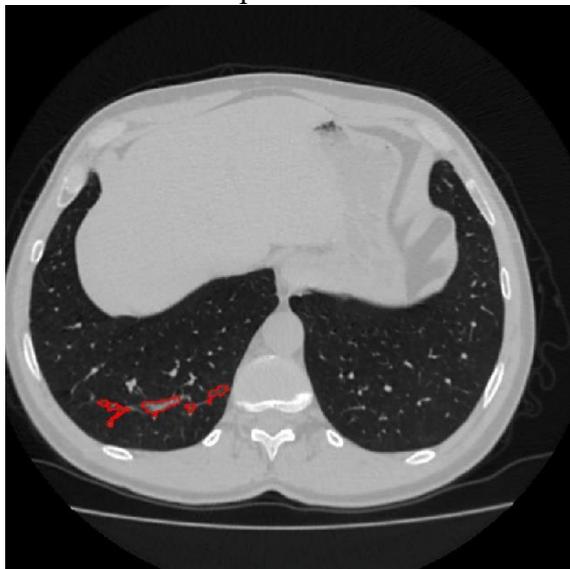
Рисунок 3 – Нейросетевая сегментация областей интереса. Класс СТ-0. Пациент № 0047.



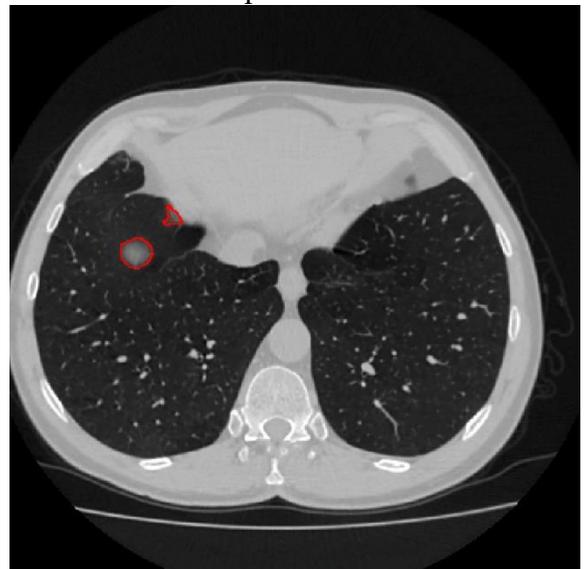
Срез № 11



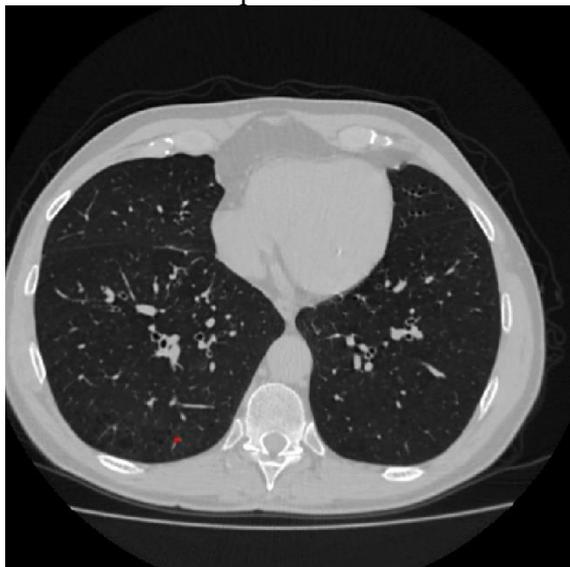
Срез № 12



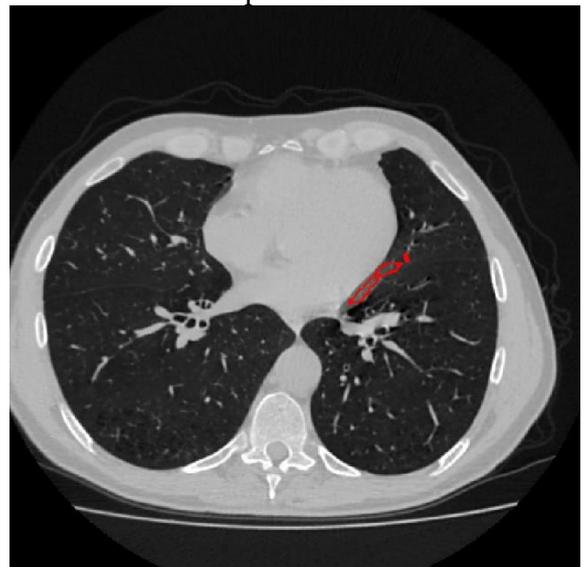
Срез № 13



Срез № 15



Срез № 19

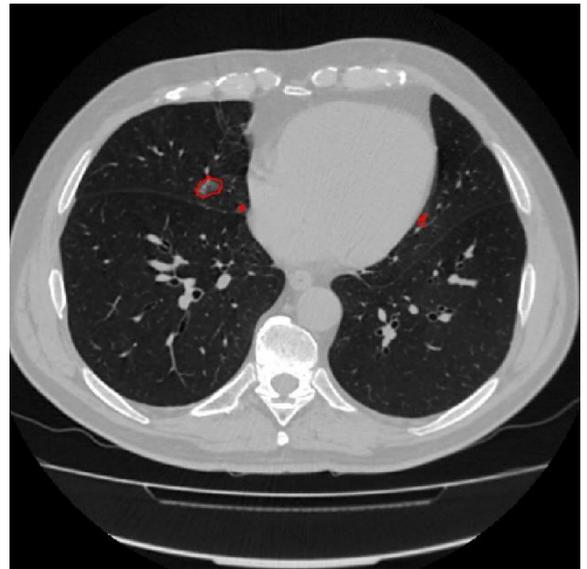


Срез № 22

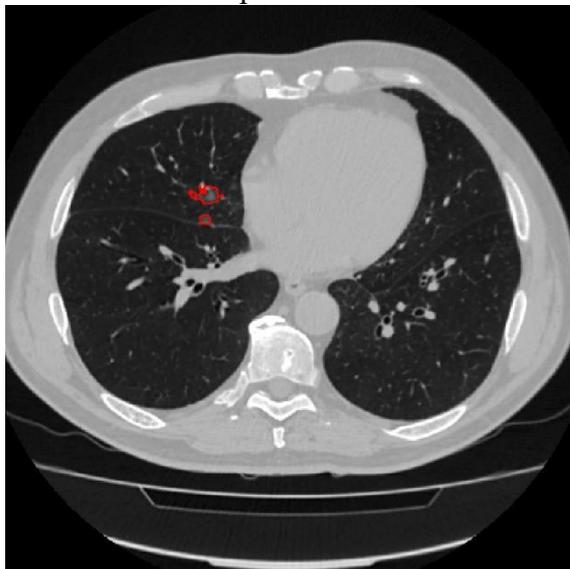
Рисунок 4 – Нейросетевая сегментация областей интереса. Класс СТ-0. Пациент № 0152.



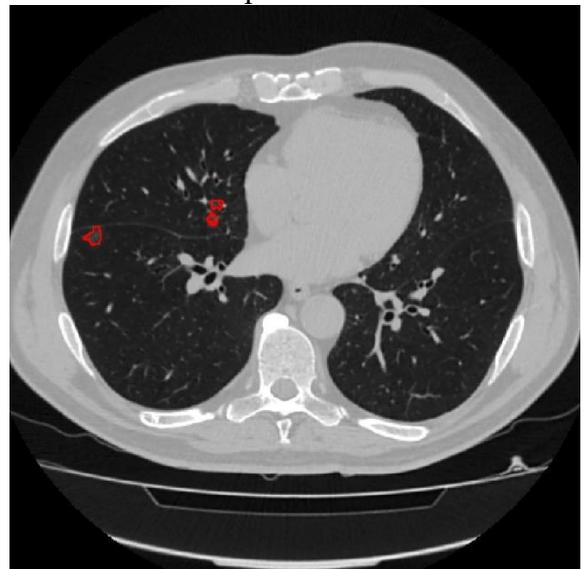
Срез № 15



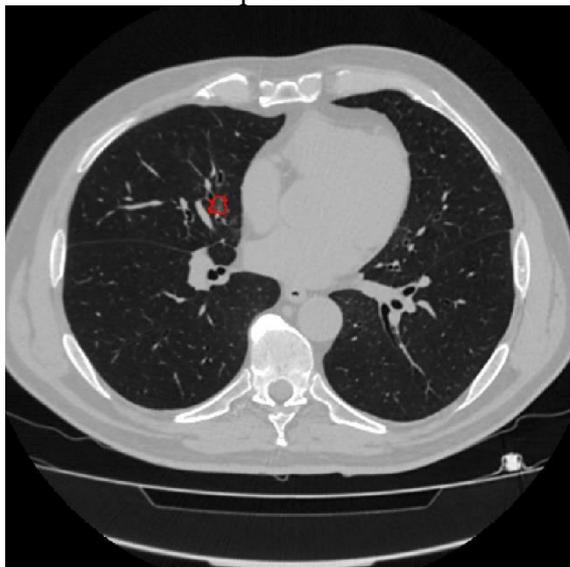
Срез № 17



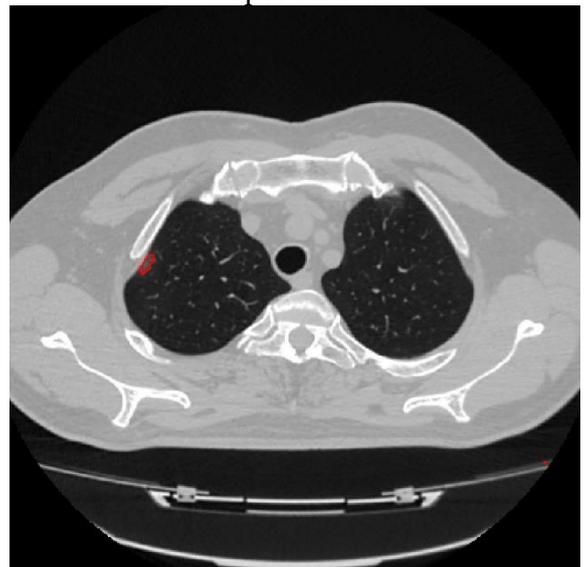
Срез № 18



Срез № 19

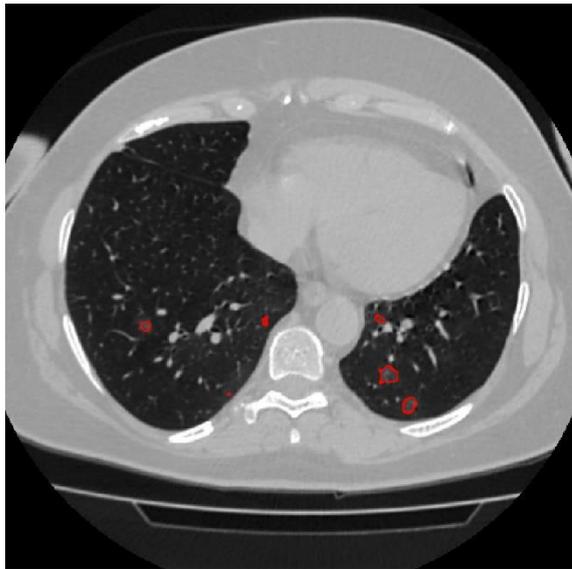


Срез № 20

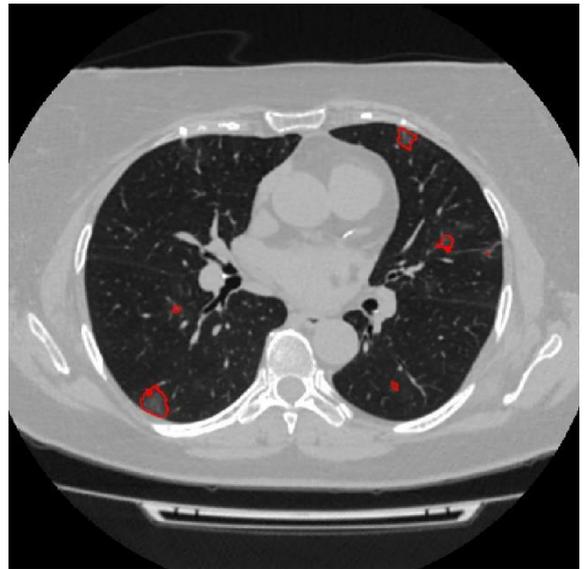


Срез № 35

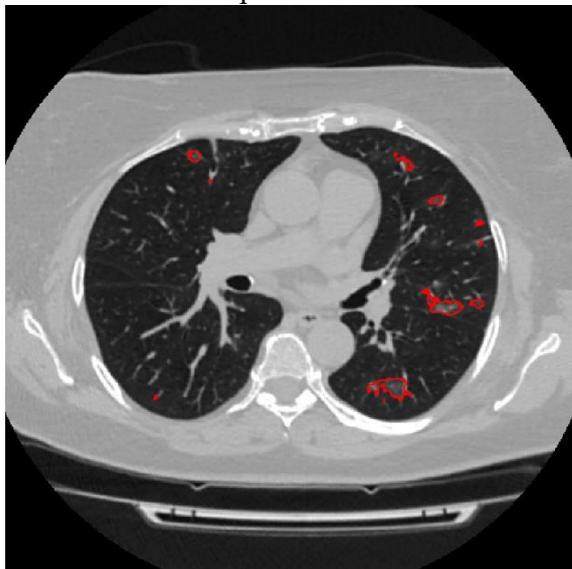
Рисунок 5 – Нейросетевая сегментация областей интереса. Класс СТ-1. Пациент № 0720.



Срез № 14



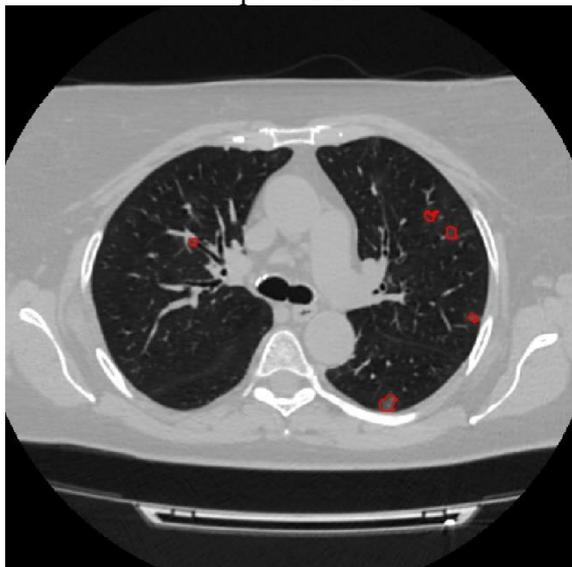
Срез № 21



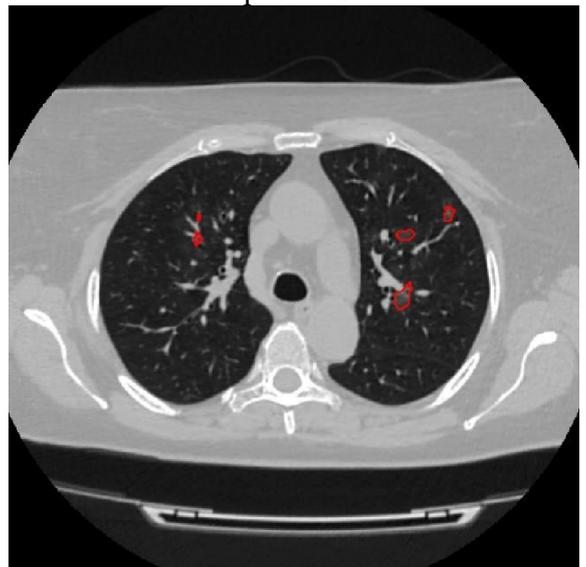
Срез № 22



Срез № 23

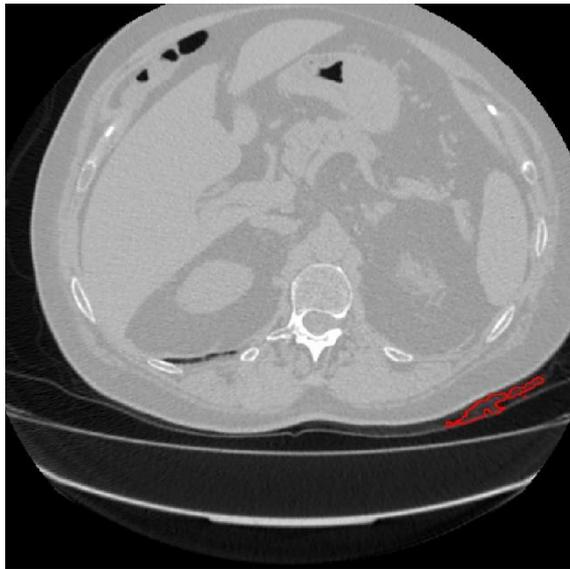


Срез № 25

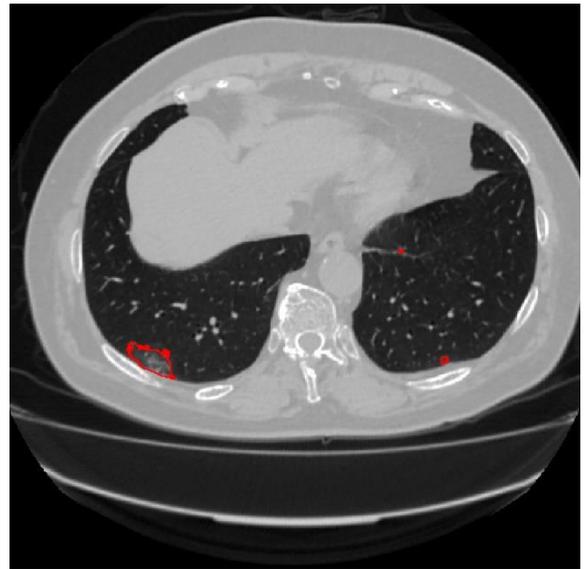


Срез № 26

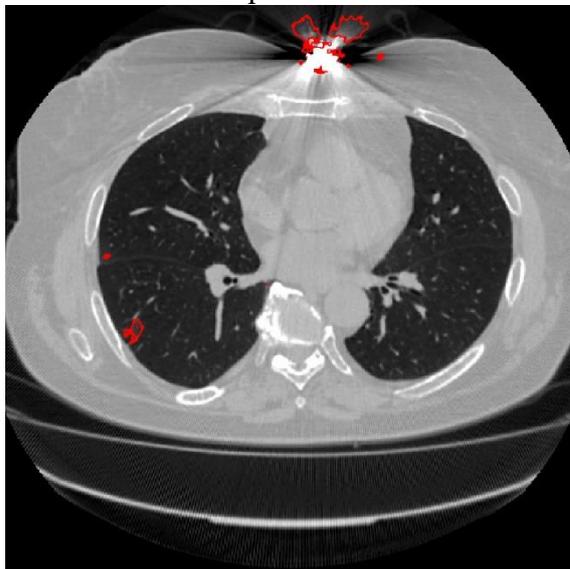
Рисунок 6 – Нейросетевая сегментация областей интереса. Класс СТ-1. Пациент № 0401.



Срез № 8



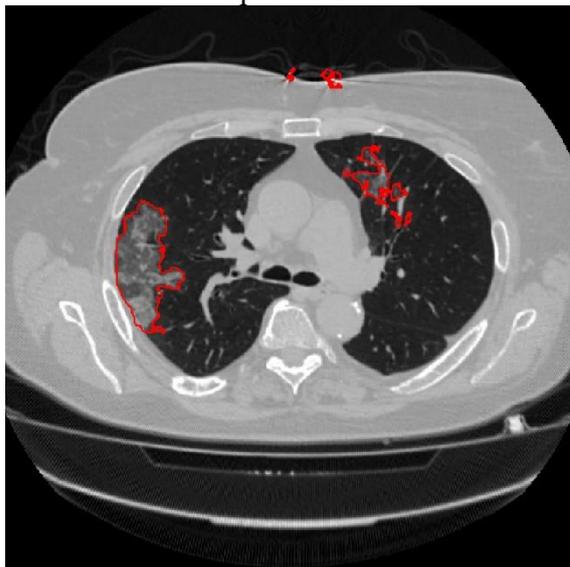
Срез № 16



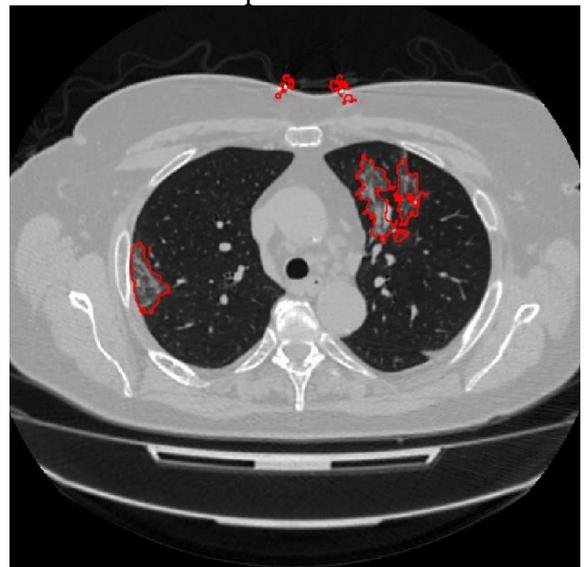
Срез № 23



Срез № 26

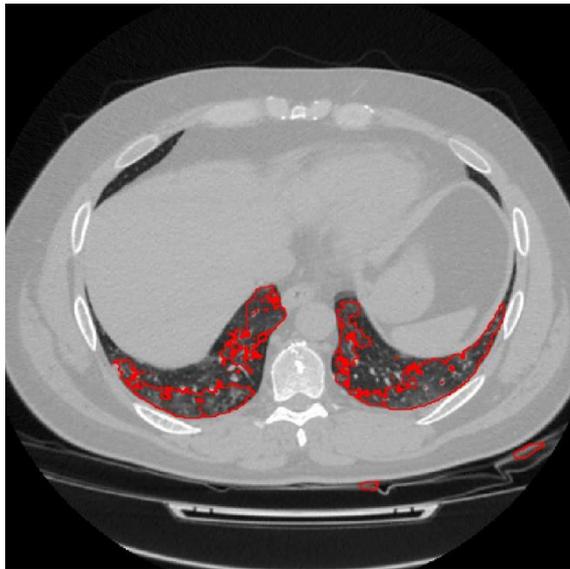


Срез № 29

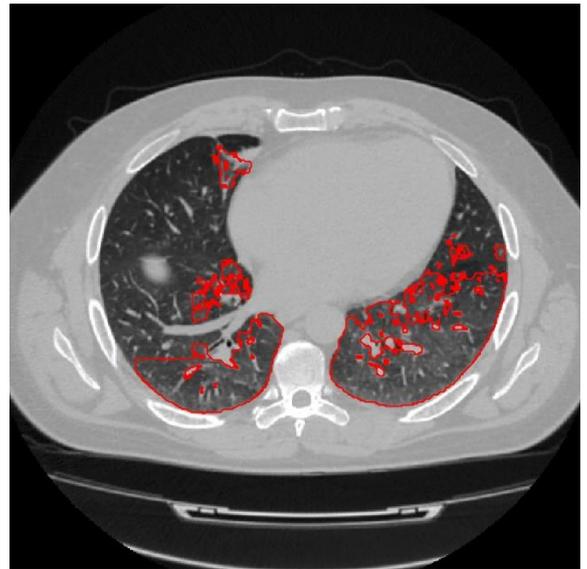


Срез № 31

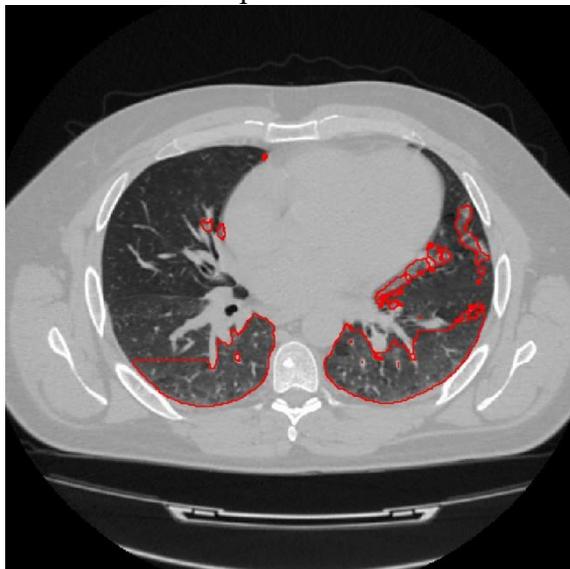
Рисунок 7 – Нейросетевая сегментация областей интереса. Класс СТ-1. Пациент № 0317.



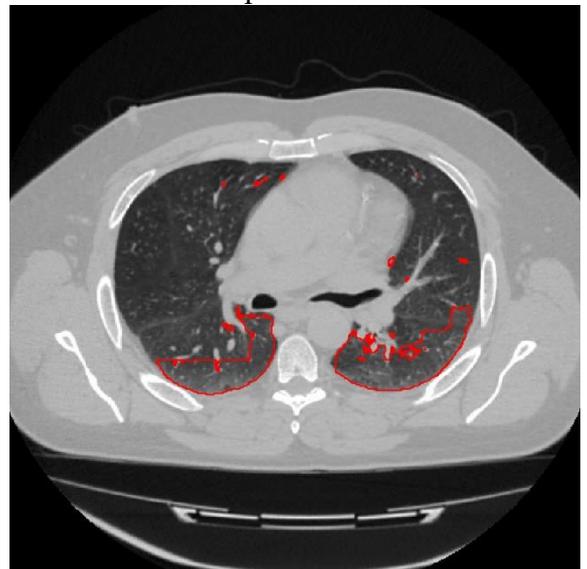
Срез № 13



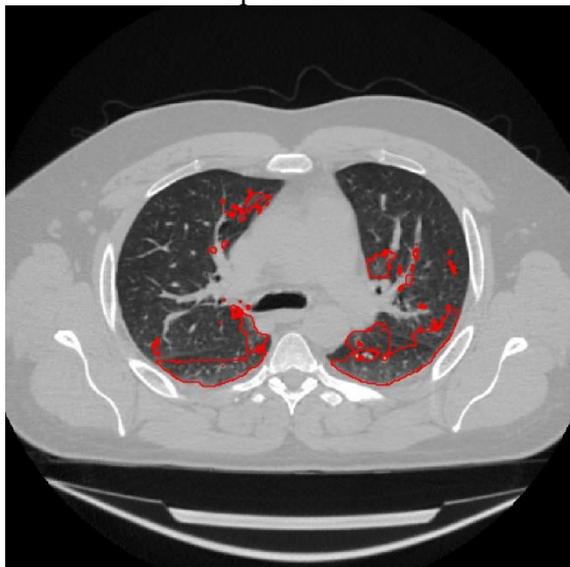
Срез № 17



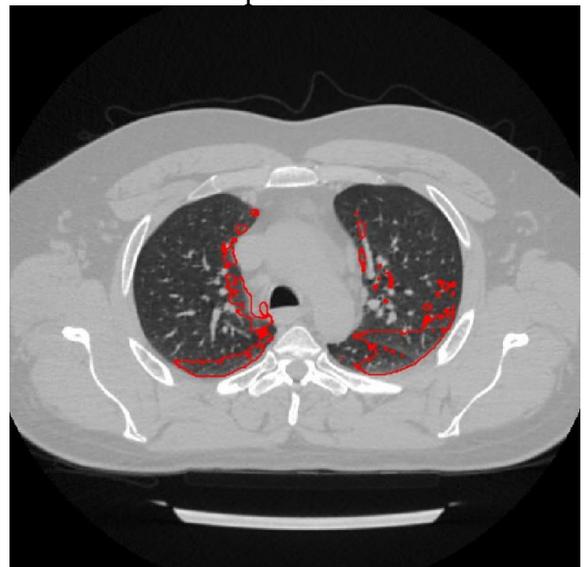
Срез № 19



Срез № 22



Срез № 24



Срез № 26

Рисунок 8 – Нейросетевая сегментация областей интереса. Класс СТ-1. Пациент № 0305.

## Приложение Б

### Software for localization of infiltrate areas in the lung region based on computed tomography data

Студент:

Группа	ФИО	Подпись	Дата
8BM83	Проскуров Владислав Анатольевич		

Руководитель ВКР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Аксёнов Сергей Владимирович	канд. техн. наук		

Консультант-лингвист отделения иностранных языков ШБИП:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Аксёнова Наталия Валерьевна	канд. филол. наук		

## **Introduction**

A common cause of lung pathology is tuberculosis. Tuberculosis is an infectious disease caused by various types of mycobacteria. Tuberculosis usually affects the lungs, but it can also affect other organs and systems. The number of resistant Mycobacterium strains is growing. As a result, multidrug-resistant tuberculosis is widespread.

According to the data of the World Health Organization for the 2018 year, the countries with the greatest number of cases of the development of tuberculosis with the plural medicinal stability (47 % of the total number in the world) are China, India and the Russian Federation [1]. Each year, about 10 million people become infected with tuberculosis. About 2 million people die that makes tuberculosis one of the leading causes of death worldwide.

In 2020, humanity faced a pandemic of the coronavirus infection COVID-19. This infection is a dangerous disease. It can occur as an acute respiratory viral infection of a light course, or in a severe form. For example, specific complications may occur in the form of viral pneumonia. According to statistics from the world health organization, the number of diseases in the world continues to grow. This virus has a high reproduction index. COVID-19 is a new virus for which there is currently no vaccine. The probability of new waves of infection over the next few years is very high [2]. In this regard, the problem of early and accurate diagnosis of COVID-19 becomes urgent.

Computed tomography (CT) of the lungs can detect pathological changes even before making a final diagnosis. CT is the most rapid method of detection of respiratory disorders. Radiation methods are not the main ones in the diagnosis of coronavirus infection. Unfortunately, they have become the most informative in identifying changes in the respiratory system. Radiation imaging data suggest a viral etiology of lung damage (COVID-19) and it makes possible to influence the management of a specific patient in the treatment of complications. All researchers emphasize the significance of changes such as ground-glass opacity (GGOs), compaction, reticular changes, and the area of the lesion. They agree that these signs determine the severity of the disease and make it possible to predict its further development. It means that there is a reliable correlation between the severity of lung damage determined by CT and the clinical course of the disease. So, the feasibility of quantifying the degree of changes in the lung parenchyma occurs.

Currently, the use of convolutional neural networks for processing CT and MRI images shows excellent results in processing biomedical images. Today, bone segmentation, anatomy of the abdominal cavity, heart, lungs and brain are almost solved. The effectiveness of using deep learning in the tasks of detecting pathologies in medical images is still being proved. It is an important stage in the early detection and diagnosis of various diseases. The effectiveness of using

machine learning methods is limited by the modest volumes of data sets and the quality of their annotations.

The purpose of this work is to improve the efficiency of the radiologist. It is necessary to develop a deep convolutional neural network architecture for automatic localization of infiltration on CT images. Solving the problem of localization of infiltration will speed up the preparation of medical reports. Also, solving this problem will improve the quality of medical diagnostics.

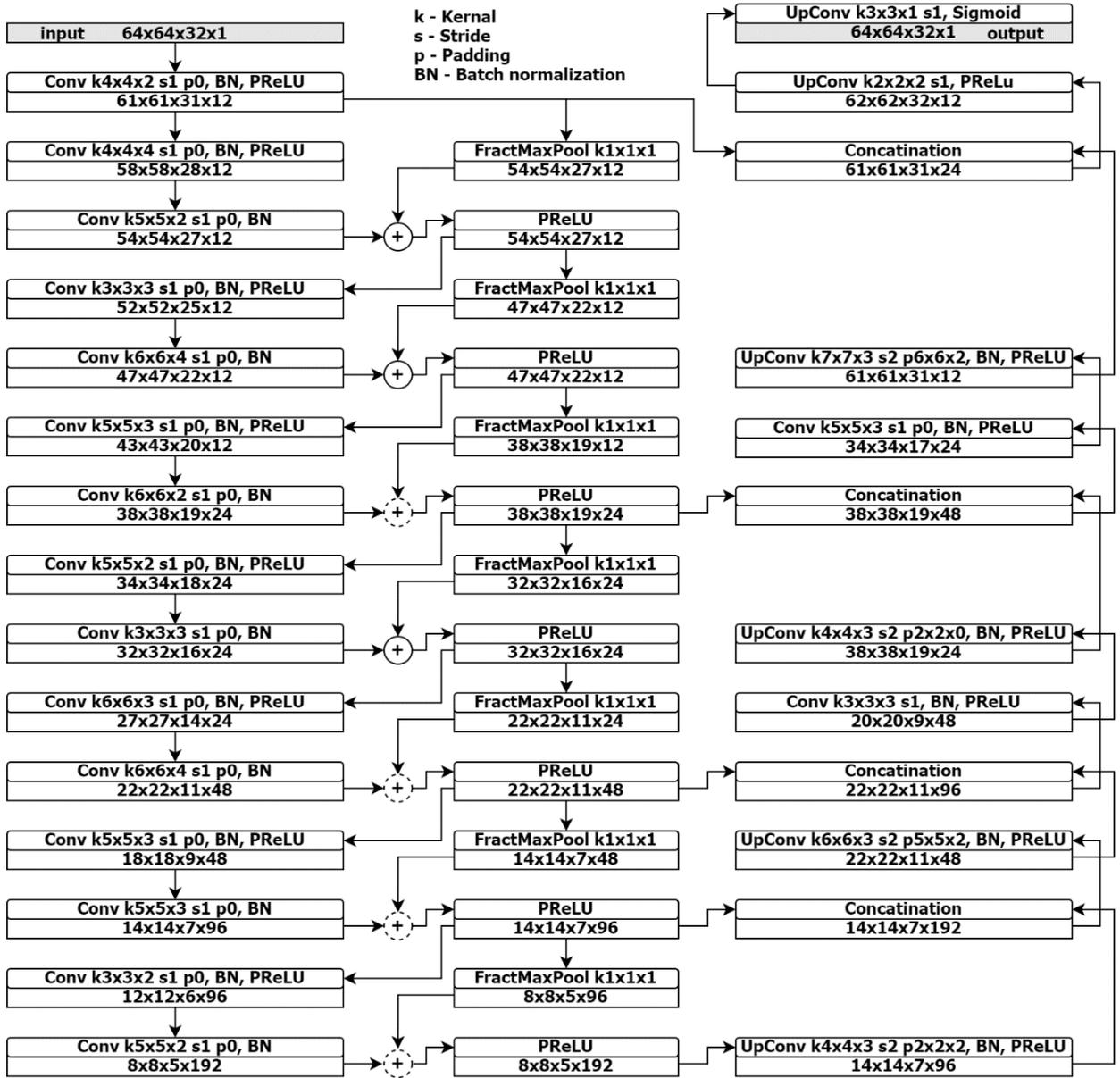
## **1 Network Architecture**

The goal of a neural network is to segment a three-dimensional image depending on how the expert does it. It is achieved by generalizing the rules that guide the expert in performing segmentation. Expert segmentation is presented to the neural network during training as the desired response. The segmentation is filled with zero's and one's values, where the ones indicate the area selected by the expert.

The architecture of the neural network developed in this work (see Figure 1) is designed according to the concept used in the U-Net architecture presented in the original works [3,4]. U-net consists of an encoder (the encoder is a compressing path) and a decoder (the decoder is an expanding path). The implementation of this architecture is used in the current work for the segmentation of CT images.

The encoder consists of a sequence of convolutional layers that follow each other. The parameters of convolutional filters are selected so that the next convolutional layer preserves the ratio of the side of the axial slice of the processed image to the height of this image.

After passing the convolutional set of filters (convolutional layer), the resulting tensor passes through the Batch-normalization layer [5]. The implementation of the encoder also provides for the implementation of the principle of residual learning of convolutional neural networks, proposed by the concept of the ResNet architecture [6]. The resulting tensors are passed through the activation function *PReLU* using the initialization of weight coefficients proposed in the original paper [7]. While passing through the encoder, the number of channels (the number of feature maps) of the output tensor is doubled when the spatial dimensions of the tensor are reduced by four times (12, 24, 48, 96, 192).



**Fig. 1.** Neural network architecture for segmentation of CT images

The decoding part of the neural network consists of decoding convolutional layers that increase the spatial dimensions of the tensor (upsampling) and reduce the number of channels. The input tensor for decoding convolution is formed by combining a tensor obtained from the lower level with an intermediate tensor obtained in the encoding part of the neural network. As a result, of passing the last decoding convolution, the tensor takes the size of the tensor received at the input to the network and contains the volume segmentation of the input image. The output of the neural network uses a sigmoid activation function *Sigmoid*. When using this activation function, the resulting voxel values equal to one show the segmentation of the areas of interest. Since the sigmoid function returns values from the interval (0; 1), a threshold rule must be applied to the output of the neural network.

## 2 Dataset

The proposed neural network approach to segmentation of medical CT images is tested on the «MosMedData dataset: COVID19\_1110», collected and published by the «Scientific and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Department of Health» [8]. The dataset contains the results of computed tomography of chest organs with x-ray signs of viral pneumonia (COVID-19), as well as without signs (norm). The data set contains 1110 CT images in NIfTI format, of which 49 (with the first stage diagnosed, CT-1) have binary segmentation for fillings of interest by type of ground-glass opacity and consolidation. The dataset also has a distribution of 5 classes (see Table 1), with the following description of each class:

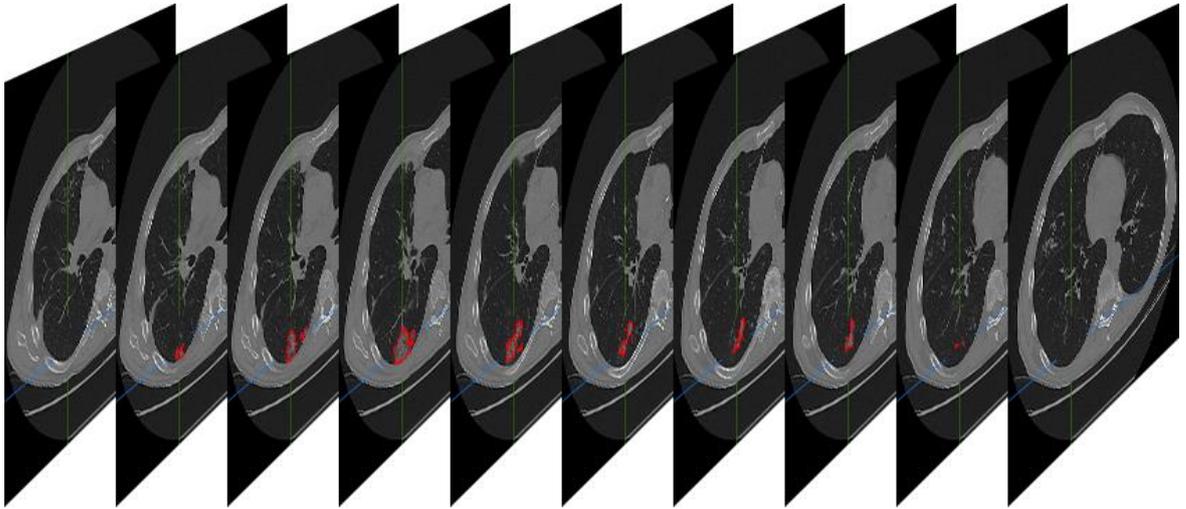
- CT-0: normal and no CT-signs of viral pneumonia;
- CT-1: includes ground-glass opacities.  
Involvement of the lung parenchyma  $\leq 25\%$ ;
- CT-2: includes ground-glass opacities.  
Involvement of the lung parenchyma = 25 – 50 %;
- CT-3: includes ground-glass opacities and consolidation.  
Involvement of the lung parenchyma = 50 – 75 %;
- CT-4: Diffuse compaction of the lung tissue includes ground-glass opacities and consolidation in combination with reticular changes.  
Involvement of the lung parenchyma  $\geq 75\%$ .

**Table 1.** Distribution of patients by class

Class	Number of patients
CT-0	254
CT-1	684
CT-2	125
CT-3	45
CT-4	2

A CT image is a series of two-dimensional images with the same resolution. They are obtained by measuring and complex computer processing differences in the attenuation of x-ray radiation by tissues of different densities.

When analyzing the internal structures of a product, segmentation is often applied to its areas of interest (see Figure 2). Segmentation divides an object into areas that are or are not of interest to the researcher.



**Fig. 2.** Three-dimensional segmentation of a medical image

A fragment of a three-dimensional image processed by a neural network must always contain dimensions enclosed in a fixed physical volume. This approach allows the neural network to generate features based not only on the relative location of voxels from each other but also to take into account the physical distance between them.

To get the final segmentation, the CT image is "cut" into fragments corresponding to the physical volume selected during neural network training. Each fragment is scaled on a grid corresponding to the dimension of the neural network input. The neural network then processes each fragment and outputs the resulting segmentation enclosed in a grid of the same size as the input data. The resulting grids are scaled according to the size of the original fragment and "glued" in the same sequence.

### **3 Data augmentation**

Data augmentation is a technique for creating additional training data from existing data. To achieve good results, deep neural networks must be trained on a very large amount of data. Therefore, if the source training set contains a limited number of examples, you must augment it to improve the results of the training model.

When developing models that augment a data set consisting of images, researchers use the following random transformations: rotation, framing, scaling, compression, and stretching along axes [9-12], horizontal and vertical reflections [13], distortion, distortion, noise, blurring, changing the brightness, saturation and contrast of the image [14, 15]. Applying various combinations of these transformations to images from the source data set can significantly increase the size of the training sample.

In this paper, we propose an algorithmic approach to perform data augmentation, namely, a method for deforming and distorting objects defined in N-dimensional space. In particular, we

consider one-dimensional 1D transformation of data arrays that have the nature of equidistant time series, 2D transformation of objects such as bitmaps, and 3D transformation of objects represented in a volume, such as a point cloud. The technique described below can also be extended to distorting the color space of images, but this issue is beyond the scope of this work.

Let there be a one-dimensional discrete signal  $S(x) = (s_0, s_1, \dots, s_{N-1}); x \in \mathbb{Z}$  consisting of  $N$  elements. Each element  $S$  characterizes a measurement performed with a fixed step in the space of the measurement being performed relative to the previous element. To implement spatial transformations of a discrete signal  $S$ , we construct its representation  $F(x); x \in \mathbb{R}$  in a real dimensionless space. Define  $F(x)$  as:

$$F(x) = \begin{cases} s_0 & , x \in (-\infty; 0,5) \\ s_1 & , x \in [0,5; 1,5) \\ \dots & \\ s_{N-1} & , x \in [N - 1,5; +\infty) \end{cases} \quad (1)$$

Signal augmentation is determined by adding random spatial distortions to the signal.

Take as reference points  $P^F = \{P_0^F, P_1^F\}$  the boundaries of the representation definition area  $F$ : the left reference point is  $P_0^F = -0,5$  and the right reference point is  $P_1^F = N - 0,5$ . We make random changes to  $P^F$  by defining a new set of reference points  $\widetilde{P}^F = P^F + \xi$ . This transformation means that changes are made randomly in the vicinity of points  $P_0^F$  and  $P_1^F$ . That is,  $\widetilde{P}_0^F = P_0^F + \xi$  and  $\widetilde{P}_1^F = P_1^F + \xi$ , where  $\xi$  is a random variable determined by an arbitrary distribution law. Note that  $\widetilde{P}^F$  can be determined not only by making random changes. Targeted selection of specific values for  $\widetilde{P}^F$  can be used to cut out a fragment of the signal  $S$ . The permutation of elements  $\widetilde{P}^F$  can be used to perform the operation of mirroring the signal  $S$ .

An augmentation of the signal  $S$  is a signal  $\tilde{S}(x) = (\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{M-1}), x \in \mathbb{Z}$ , consisting of  $M$  elements. The values of the elements  $\tilde{S}$  are determined by the formula (3). The calculation uses an auxiliary function  $\phi$  that operates with the interval  $[l; r]$ . Using the function  $\phi$ , the coordinate of the point lying in the center of the segment with the number  $k$  inside the interval  $[l; r]$ , divided into  $N$  equal segments, is determined. Define the function  $\phi$  as:

$$\phi(k, l, r, N) = l + \frac{r - l}{N} \cdot \left(k + \frac{1}{2}\right) \quad (2)$$

where  $\phi$ : auxiliary function,  $k$ : number of the segment in the interval,  $l$ : coordinate of the left border of the interval (left reference point),  $r$ : coordinate of the right border of the interval (right reference point),  $N$ : the number of segments that the interval is divided into.

$$\tilde{s}_i = F\left(\phi\left(i, \widetilde{P}_0^F, \widetilde{P}_1^F, M\right)\right), i \in \{0, 1, \dots, M-1\} \quad (3)$$

where  $\tilde{s}_i$ : value of the next element in the augmented signal  $\tilde{S}$ ,  $F(x)$ : representation of the original signal  $S$  in real space,  $\phi$ : auxiliary function,  $\widetilde{P}_0^F$ : left reference point,  $\widetilde{P}_1^F$ : right reference point,  $M$ : number of elements in the augmented signal.

Note that if the random change  $\xi = 0$ , the reference points of the signal do not change. In this case, the signal conversion performed by formula (3) is equivalent to the conversion that changes the sample rate of the signal  $S$ .

In the example above, the arguments  $l$  and  $r$  passed to the function  $\phi$  correspond to coordinates defined along the same axis. This transformation is also applicable for vector calculations, when the reference points of a divisible interval are defined in a multidimensional space.

A bitmap image is represented by a two-dimensional matrix of  $W \times H$  pixels. Image elements are characterized by two  $x$  and  $y$  axis coordinates and an intensity value  $C(x, y)$ . Only distortions of image spatial components are considered in the described method of images transforming. Spatial transformations do not involve changes in brightness components. Therefore, the count of image color components does not matter. Actually, the raster image is two-dimensional discrete signal  $S(x, y) = (s_{0,0}, s_{0,1}, \dots, s_{W-1,H-1})$ ;  $x, y \in \mathbb{Z}$  consisting of  $W \cdot H$  elements.

To perform a 2D-transformation as in example with 1D-transformation, a two-dimensional bitmap image from an integer space will transfer to real one by building a map  $F(x, y)$ ;  $x, y \in \mathbb{R}$ . Next, we assume that when referring to  $F$ ,  $\frac{1}{2}$  is subtracted from  $x$  and  $y$ . Then, the reference points  $F$  are the boundaries of the image definition area, according to (4).

$$P^F = \{P_{0,0}^F, P_{1,0}^F, P_{0,1}^F, P_{1,1}^F\} \\ P_{0,0}^F(0; 0); P_{1,0}^F(W; 0); P_{0,1}^F(0; H); P_{1,1}^F(W; H) \quad (4)$$

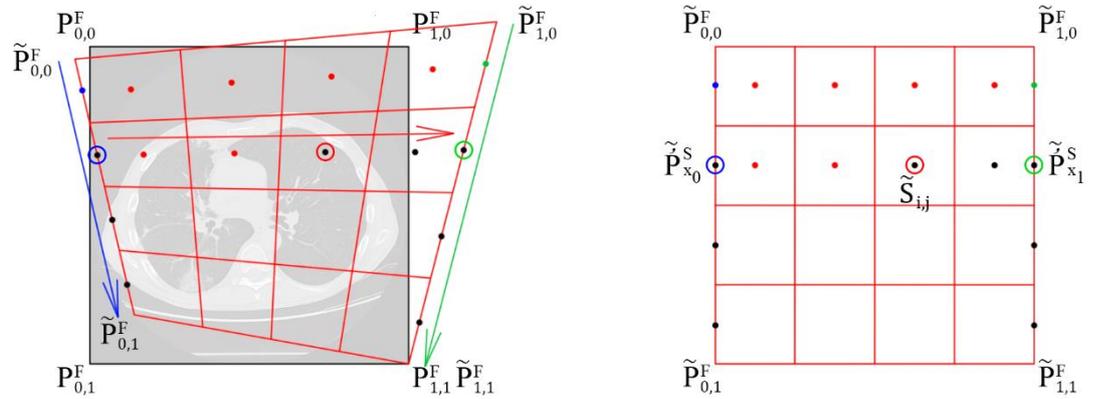
As before  $\widetilde{P}^F = P^F + \xi$  determines random changes made to reference points  $P^F$ . Augmentation of raster image (signal  $S$ ) is  $\tilde{S}(x, y) = (\tilde{s}_{0,0}, \tilde{s}_{0,1}, \dots, \tilde{s}_{\widetilde{W}-1, \widetilde{H}-1})$ ;  $x, y \in \mathbb{Z}$  signal, consisting of  $\widetilde{W} \cdot \widetilde{H}$  elements.  $\widetilde{W}$  and  $\widetilde{H}$  determines the resolution for the generated augmentation. The values of the discrete signal  $\tilde{S}$  are calculated according to (5) or (6). The given methods of calculating the discrete signal  $S$  values are equivalent to each other.

$$\widetilde{P}_{x_0}^F(i) = \phi(i, \widetilde{P}_{0,0}^F, \widetilde{P}_{1,0}^F, \widetilde{W}); \widetilde{P}_{x_1}^F(i) = \phi(i, \widetilde{P}_{0,1}^F, \widetilde{P}_{1,1}^F, \widetilde{W}); \\ \tilde{s}_{i,j} = F\left(\phi\left(j, \widetilde{P}_{x_0}^F(i), \widetilde{P}_{x_1}^F(i), \widetilde{H}\right)\right) \quad (5)$$

$$\tilde{s}_{i,j} = F \left( \phi \left( j, \phi \left( i, \tilde{P}_{0,0}^F, \tilde{P}_{1,0}^F, \tilde{W} \right), \phi \left( i, \tilde{P}_{0,1}^F, \tilde{P}_{1,1}^F, \tilde{W} \right), \tilde{H} \right) \right) \quad (6)$$

$$i \in \{0, 1, \dots, \tilde{W} - 1\}; j \in \{0, 1, \dots, \tilde{H} - 1\};$$

The process of sequential calculation of two-dimensional transformation (augmentation)  $\tilde{S}$  of  $S$  (see Figure 3).



**Fig. 3.** The process of 2D-transformation

The described method is universal because it combines the properties inherent in combinations of classical random transformations of two-dimensional images (see Figure 4).

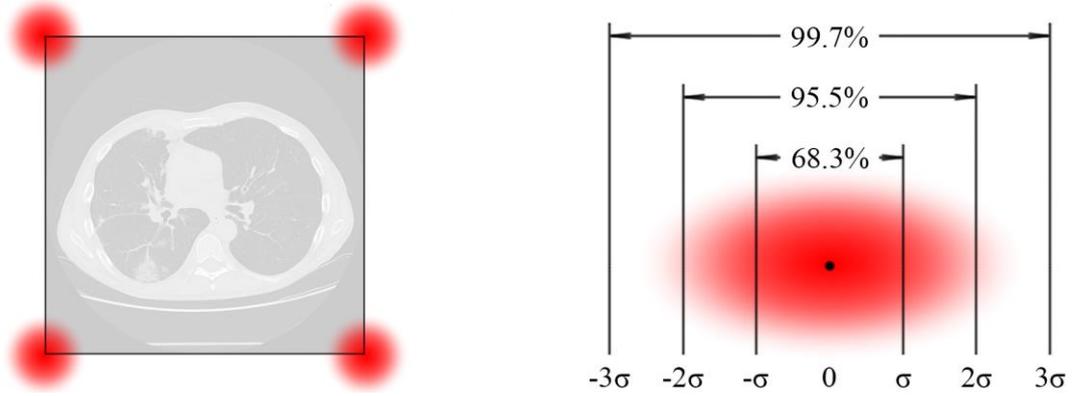


**Fig. 4.** Result of 2D transformation

If it necessary to perform a strict image rotation operation, it is enough to place the reference points of the source image on the circle described near the image and calculate their spatial coordinates taking into account the clockwise or counterclockwise rotation. The operation of horizontal or vertical reflection is to change the order of reference points respectively. The framing operation is to shift the reference points towards or away from the center of the image.

To make a random change to the reference points of a signal in  $N$ -dimensional space, it was determined that the change produced is characterized by a random vector  $\xi = (\xi_1, \dots, \xi_n)$ . To

implement the algorithm for performing augmentation of the training dataset, the normal distribution was used as a random change made to the reference points. Then, if  $Z$  is a standard normal random variable, then  $\xi_i = \sigma_i \cdot Z$  will have a normal distribution with zero expectation and standard deviation  $\sigma$  (See Figure 5). Selecting a value for the parameter  $\sigma$  allows controlling the strength of the introduced distortion.



**Fig. 5.** Probabilistic selection of reference points for 2D transformation

Using the proposed method, one can be sure that new randomly defined reference points with a probability of 96 % will not move further than  $2 \cdot \sigma$  from the original point and with a probability of 68 % further  $\sigma$ .

A biomedical raster single-channel image of the size  $W \times H \times D$  (for example, a CT-image) can be represented as  $F(x, y, z); x, y, z \in \mathbb{R}$ . To implement a 3D transformation, the reference points  $F$  (which are the boundaries of the image definition area) are defined as:

$$\begin{aligned}
 P^F = \{ & P_{0,0,0}^F, P_{1,0,0}^F, P_{0,1,0}^F, P_{1,1,0}^F, P_{0,0,1}^F, P_{1,0,1}^F, P_{0,1,1}^F, P_{1,1,1}^F \}; \\
 & P_{0,0,0}^F(0; 0; 0); P_{1,0,0}^F(W; 0; 0); P_{0,1,0}^F(0; H; 0); P_{1,1,0}^F(W; H; 0); \\
 & P_{0,0,1}^F(0; 0; D); P_{1,0,1}^F(W; 0; D); P_{0,1,1}^F(0; H; D); P_{1,1,1}^F(W; H; D).
 \end{aligned} \tag{7}$$

As before, random changes are made to the reference points  $\widetilde{P}^F = P^F + \xi$  and the calculating the values of a three-dimensional discrete signal  $S$  as:

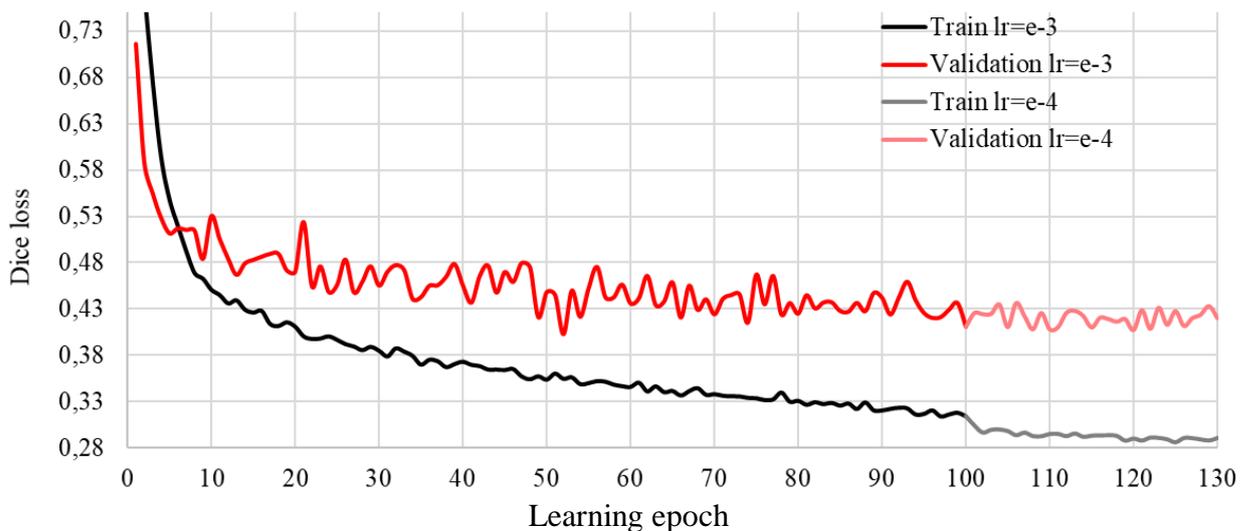
$$\begin{aligned}
 \widetilde{P}_{x_0}^F(i) &= \phi(i, \widetilde{P}_{0,0,0}^F, \widetilde{P}_{1,0,0}^F, \widetilde{W}); \widetilde{P}_{x_1}^F(i) = \phi(i, \widetilde{P}_{0,1,0}^F, \widetilde{P}_{1,1,0}^F, \widetilde{W}); \\
 \widetilde{P}_{x_2}^F(i) &= \phi(i, \widetilde{P}_{0,0,1}^F, \widetilde{P}_{1,0,1}^F, \widetilde{W}); \widetilde{P}_{x_3}^F(i) = \phi(i, \widetilde{P}_{0,1,1}^F, \widetilde{P}_{1,1,1}^F, \widetilde{W}); \\
 \widetilde{P}_{y_0}^F(j) &= \phi(j, \widetilde{P}_{x_0}^F(i), \widetilde{P}_{x_1}^F(i), \widetilde{H}); \widetilde{P}_{y_1}^F(j) = \phi(j, \widetilde{P}_{x_2}^F(i), \widetilde{P}_{x_3}^F(i), \widetilde{H}); \\
 \widetilde{s}_{i,j,k} &= F\left(\phi(k, \widetilde{P}_{y_0}^F(j), \widetilde{P}_{y_1}^F(j), \widetilde{D})\right) \\
 i \in \{ & 0, 1, \dots, \widetilde{W} - 1 \}; j \in \{ 0, 1, \dots, \widetilde{H} - 1 \}; k \in \{ 0, 1, \dots, \widetilde{D} - 1 \};
 \end{aligned} \tag{8}$$

## 4 Training

Dataset contains 49 CT images for which manual segmentation of the areas of interest corresponding to ground-glass opacity seals for patients with signs of viral pneumonia is presented. CT scans of patient data are used in neural network training. The data set was divided into two parts, 44 CT images were randomly selected for the training sample, and the remaining 5 formed a test sample.

The training data set is formed from 15,000 randomly generated augmented examples, and the validation set is formed from 500 examples. Each example in these sets has a resolution of  $64 \times 64 \times 32$  voxels and describes a fragment of a CT image enclosed in a volume of  $40 \times 40 \times 60$  mm (96 ml).

The network was trained in two stages (see Figure 6). The first stage was conducted for 100 epochs with the value  $lr = 0,001$  (learning rate). At the second stage, the learning rate was reduced to 0,0001, and the number of learning epochs was 30.



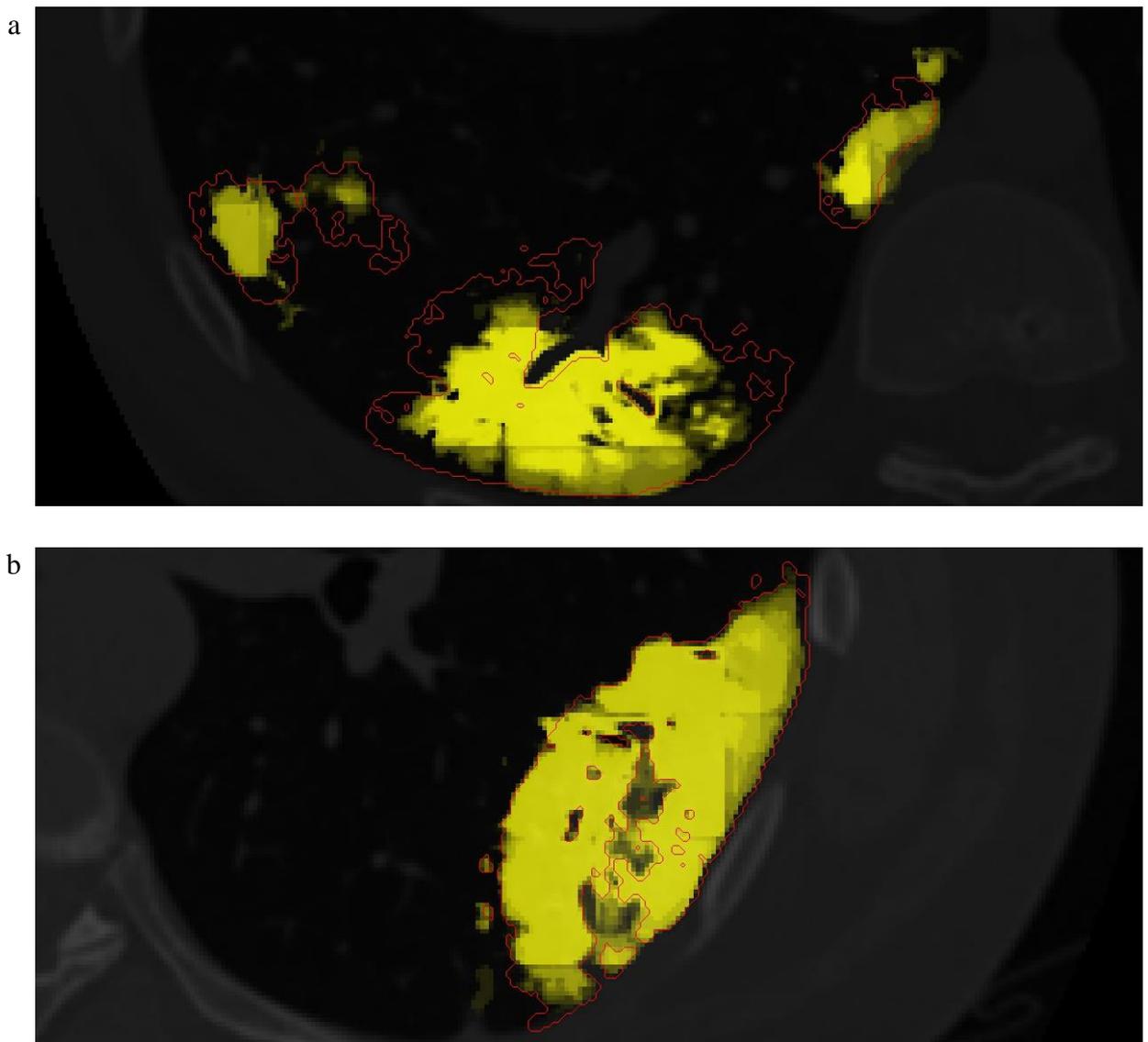
**Fig. 6.** The dynamics of changing the value of the error function

After training the neural network using 5 test CT images, the average values of metrics displaying the quality of segmentation were obtained (see Table 2). The following values are calculated: Precision, Recall, F-score, and IoU.

**Table 2.** Result of segmentation of the test sample

Precision, PPV	Recall, TPR	F <sub>1</sub> Score	IoU
0.652	0.638	0.614	0.477

Visual analysis of the results of the neural network processing of CT images from the test sample (see Figure 7) shows that the neural network responds well to the areas of interest. Most of the selected areas have values very close to the maximum for the activation function used at the output of the neural network. Each voxel of the CT image is evaluated by the neural network 8 times; this is due to the feature of the sliding window movement. Obtaining the maximum value for the activation function after averaging indicates the uniformity of distribution of the rules for determining features characteristic of the areas of interest formed in the representation of the neural network.



**Fig. 7.** Axial slices of patients from the test sample Yellow is the result of processing a neural network with a sliding window. The contour of the segmentation done by the expert are highlighted in red. (a) Patient 0288, slice 13. (b) Patient 0291, slice 13.

## 5 Experiments

Often, when making a diagnosis, the volume of lung infiltration is used as a predictor. It was decided to evaluate the possibility of using a trained neural network in solving this problem. As a result of neural network processing of the CT image, it is possible to numerically determine the total volume of the area of interest in milliliters.

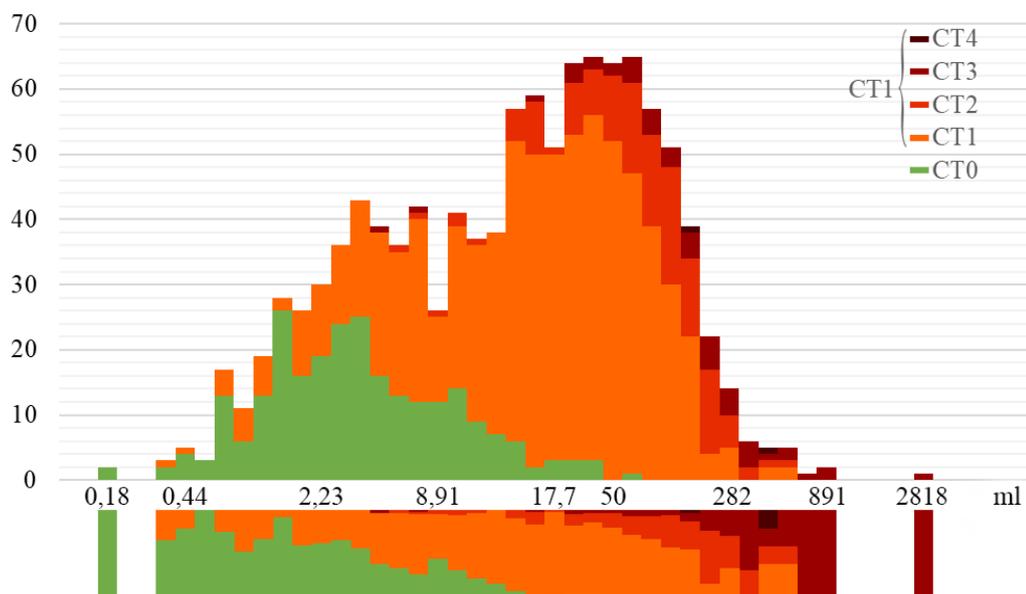
The processed data set «MosMedData: COVID19\_1110» contains 1110 CT images distributed in 5 classes. The CT-0 class contains CT images of patients for whom the expert found no signs of viral pneumonia. In turn, other classes (CT-1, 2, 3, 4) contain images of patients' lungs, which, according to the expert, contain areas of ground-glass turbidity.

The average segmentation time per image was 3 minutes. 17 seconds, median of 2 minutes. 57 seconds, a minimum of 47 seconds, a maximum of 12 minutes 48 seconds. Segmentation of the entire data set took more than 2 days and 14 hours.

To get an estimate of the actual volume of infiltration for each patient, the total value is multiplied by the value of the volume described by a single voxel of CT image.

When solving the problem of making a diagnosis, the primary task is to decide whether a person is ill or not. And only if there are enough signs of the disease, the stage at which the disease develops is determined. In this regard, we reduce the problem of evaluating the proposed neural network approach for diagnosis to the problem of binary classification by combining the classes (CT-1, 2, 3, 4) into a single class CT-1.

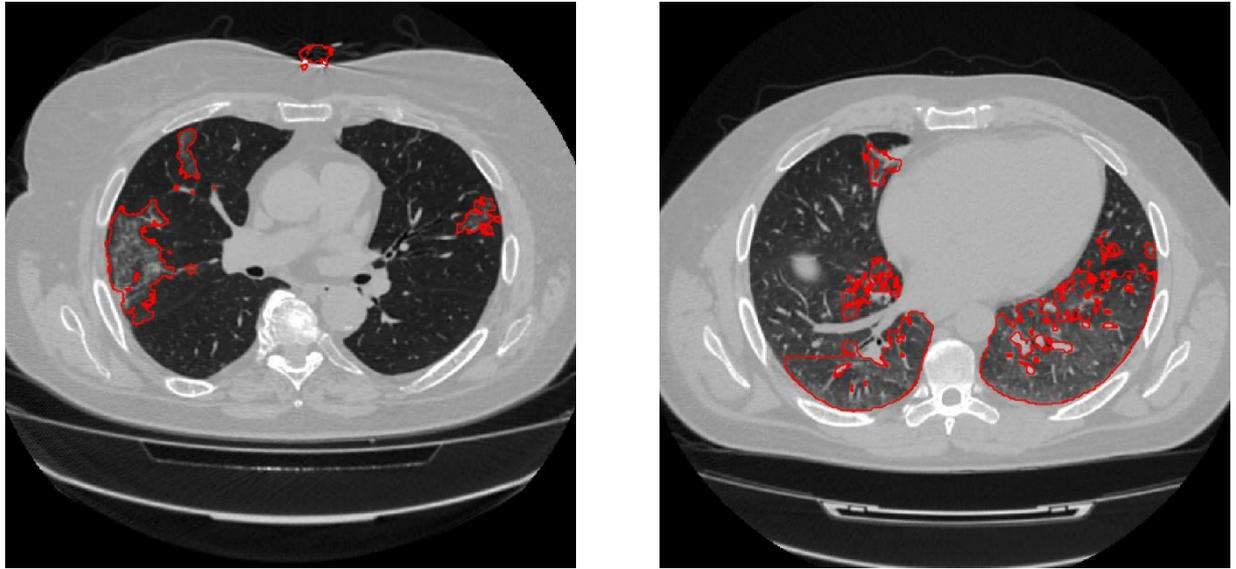
Distribution of classes represented in the data set by the volume of infiltration detected by the neural network in the lung region. The stacked histogram (see Figure 8) shown is constructed by applying logarithmic scaling to the length of grouping intervals.



**Fig. 8.** Distribution of dataset classes. MosMedData: COVID19\_1110

When determining the effective threshold value, the accuracy of the binary classification was 84.6 %. This threshold is estimated to correspond to the actual volume of infiltrating detected by the neural network in the lung region in the range from 3.5 ml to 6.6 ml.

Visual analysis of the obtained segmentation results (see Figure 9) shows that the developed approach to localization of infiltration areas on CT images copes well with the task.



**Fig. 9.** The result of segmentation of randomly selected patients. MosMedData COVID19\_1110

However, the developed segmentation algorithm has disadvantages, for example, the neural network often takes artifacts from the patient's heart contraction as an area of interest. The neural network also tends to perform segmentation of the couch on which the patient is lying.

## Conclusion

Research on the best practices offered in recent years for solving deep machine learning problems has allowed us to develop and implement a unique architecture of a convolutional neural network. The developed model works with a three-dimensional representation of the processed fragments of the CT image. This approach allows us to form high-quality three-dimensional spatial relationships between selected features in the internal representation of the neural network. The key conceptual feature of the proposed model is the implementation of the idea of residual ResNet training in the compression path of a U-net similar network.

Visual analysis of the results shows that the trained neural network copes well with the problem of localization of the "ground-glass" sealing zone in the lung area.

## References

1. "What is multidrug-resistant tuberculosis (MDR-TB) and how do we control it?" 2018. [Online]. Available: [https://www.who.int/news-room/q-a-detail/what-is-multidrug-resistant-tuberculosis-\(mdr-tb\)-and-how-do-we-control-it](https://www.who.int/news-room/q-a-detail/what-is-multidrug-resistant-tuberculosis-(mdr-tb)-and-how-do-we-control-it). [Accessed 16 02 2020].
2. «Q&A: Influenza and COVID-19 - similarities and differences» 2020. [Online]. Available: <https://www.who.int/en/news-room/q-a-detail/q-a-similarities-and-differences-covid-19-and-influenza>. [Accessed: 30 04 2020].
3. O. Ronneberger, P. Fischer and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv preprint, 2015.
4. Ö. Çiçek, A. Abdulkadir и S. S. Lienkamp, 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, arXiv preprint, 2016.
5. S. Ioffe and C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv preprint, 2015.
6. H. Kaiming, Z. Xiangyu, R. Shaoqing и S. Jian, Deep Residual Learning for Image Recognition, arXiv preprint, 2015.
7. H. Kaiming, Z. Xiangyu, R. Shaoqing and S. Jian, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, arXiv preprint, 2015.
8. Dataset «MosMedData: COVID19\_1110» 2020. [Online]. Available: <https://mosmed.ai>. [Accessed: 18 05 2020].
9. Simard P.Y., Steinkraus D., Platt J.C. Best practices for convolutional neural networks applied to visual document analysis. // Proceedings of International Conference on Document Analysis and Recognition. 2003.
10. Ciregan D., Meier U., Schmidhuber J. Multicolumn deep neural networks for image classification // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2012. pp. 3642–3649.
11. Wan L., Zeiler M., Zhang S., Le Cun Y., Fergus R. Regularization of neural networks using dropconnect // In International Conference on Machine Learning. 2013. pp. 1058-1066.
12. Sato I., Nishimura H., Yokoi K. Apac: Augmented pattern classification with neural networks. arXiv preprint, 2015. arXiv:1505.03229.
13. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems. 2012.
14. DeVries T., Taylor G.W. Improved regularization of convolutional neural networks with cutout. arXiv preprint, 2017. arXiv:1708.04552.
15. Zhong Z., Zheng L., Kang G., Li S., Yang Y. Random erasing data augmentation. arXiv preprint, 2017. arXiv:1708.04896.

**Приложение В**  
**CD-диск**