

Санкт-Петербургский государственный университет

Программная инженерия

Кафедра системного программирования

Фадеева Анастасия Владимировна

Разработка системы анализа медицинских
изображений для решения задач
кардиологии

Бакалаврская работа

Научный руководитель:
ст. преп. Смирнов М. Н.

Рецензент:
инженер-программист ООО «Системы КМ» Монькин С. А.

Санкт-Петербург
2019

SAINT-PETERSBURG STATE UNIVERSITY

Software engineering

Anastasiia Fadeeva

Development of a Medical Image Analysis System for Solving Problems in Cardiology

Graduation Thesis

Scientific supervisor:
senior lecturer M.N. Smirnov

Reviewer:
software engineer CM Systems LLC S.A. Monkin

Saint-Petersburg
2019

Оглавление

Введение	4
Постановка задачи	6
1. Обзор	7
1.1. Обзор существующих решений	7
1.2. Обзор методов сегментации изображений	7
1.3. Используемые технологии	14
2. Архитектура и особенности реализации	16
2.1. Модуль анализа изображений	17
2.2. Взаимодействие с пользователем	19
3. Обучение модели сегментации	23
3.1. Исходные данные	23
3.2. Аугментация данных	24
3.3. Нейросетевые модели сегментации	25
3.4. Выбор модели сегментации	26
4. Апробация	29
4.1. Метод	29
4.2. Результаты апробации	29
Заключение	32
Список литературы	33

Введение

Одной из важных клинических проблем в кардиологии является фиброз. Фиброзом называется процесс образования рубцовой ткани, вызывающий структурные изменения миокарда и сосудистой стенки. Помимо нарушений механических функций сердца, рубцевание, обусловленное фиброзом, часто инициирует желудочковые аритмии, которые, в свою очередь, ускоряют развитие событий, ведущих к сердечной недостаточности. Таким образом, диагностирование фиброза имеет важное значение для улучшения качества жизни.

Основным методом диагностирования фиброза является магнитно-резонансная томография (МРТ) с отсроченным контрастированием. МРТ позволяет исследовать состояние различных органов, основываясь на содержании молекул водорода в них и их магнитных свойств. В отличие от рентгена и компьютерной томографии, этот метод не использует вредное для человеческого организма ионизирующее излучение и дает максимально детализированную картину. В результате такой диагностики получается послойное трехмерное изображение исследуемого органа.

Принцип МРТ с отсроченным контрастированием заключается в предварительном введении специального вещества, называемого контрастом. После определенного промежутка времени выполняется МРТ-сканирование; за это время из здоровых тканей контраст выводится, а в фиброзной он задерживается, в результате здоровая ткань выглядит черной, а фиброзная — светлой, что позволяет определить местоположение фиброза на снимках.

Сегментация фиброзных тканей на снимках, полученных с помощью МРТ с отсроченным контрастированием, позволяет получить количественные измерения фиброзной ткани, а также информацию о ее местоположении, что может быть полезным для изменения способа мониторинга и лечения пациентов с данной патологией, а также помогает при планировании кардиостимуляции, так как фиброзная ткань оказывает влияние на прохождение сигнала.

Определение фиброза сердца на медицинских изображениях является сложной задачей из-за разнообразия его анатомических форм и местоположения, к тому же фиброзная ткань может содержаться не только в сердце, но и в других органах и иметь такую же интенсивность на МРТ-снимках, что и область интереса.

Большой объем данных и сложность локализации являются причиной того, что ручная сегментация занимает много времени: в среднем, врач тратит 1-1,5 часа на сегментацию 90 изображений для одного пациента. Возникает необходимость в реализации системы, позволяющей ускорить этот процесс. Разработка системы автоматической сегментации фиброзной ткани на МРТ-снимках с контрастом и является целью данной работы.

Постановка задачи

Целью данной работы является разработка системы поиска фиброзной ткани на МРТ-снимках с контрастом. Для её достижения были выделены следующие задачи:

- провести анализ существующих решений, а также методов сегментации изображений;
- разработать архитектуру системы анализа МРТ-снимков сердца с контрастом;
- реализовать модуль анализа изображений, в том числе обучить модель сегментации с целью выявления фиброзной ткани на сердце;
- провести апробацию разработанного решения.

1. Обзор

1.1. Обзор существующих решений

В настоящее время существуют инструменты, позволяющие сегментировать фиброзные ткани в интерактивном режиме. Например, компания Circle Cardiovascular Imaging [2] предоставляет программу для просмотра медицинских изображений, в которую встроен инструмент для полуавтоматической сегментации фиброзной ткани на МРТ-снимках с контрастом. Полуавтоматическая сегментация работает следующим образом: сначала появляются внешний и внутренний контуры левого желудочка сердца — области поиска фиброза, затем врач их вручную выравнивает, чтобы они совпадали с контурами на изображении, далее область делится на определенное количество регионов, для каждого из которых врач определяет примерный объем фиброзной ткани. Таким образом получается примерное представление о местоположении фиброза и его количестве.

Недостатки этого решения в том, что во-первых, результаты получаются не совсем точными, во-вторых, выравнивание контуров и определение количества фиброза в каждом сегменте для всех снимков занимает время, сопоставимое со временем ручной сегментации, к тому же это решение обладает достаточно высокой стоимостью.

Таким образом, возникает необходимость в разработке полностью автоматизированной системы сегментации фиброза на МРТ-снимках с контрастом.

1.2. Обзор методов сегментации изображений

Основу системы будет составлять сегментация, которая является одним из важных этапов процесса обработки медицинских изображений. Сегментация представляет собой разбиение набора данных на смежные регионы, элементы которых (например, пиксели или воксели) обладают общими признаками. Существует много разных подходов и методов для выделения структуры из набора изображений: от ручной сегмента-

ции до полностью автоматизированной [6]. Какой метод использовать, зависит от свойств набора данных, а также от сложности и особенностей структуры, которую необходимо сегментировать. Методы могут быть использованы независимо или в комбинации для достижения желаемого результата. В этом разделе будут описаны некоторые методы, которые применяются в анализе изображений.

1.2.1. Ручная сегментация

Этот метод представляет собой ручное выделение области интереса. Обычно это очень трудоемко, особенно для 3D данных со множеством срезов или структур с очень сложными контурами. Сложность работы требует квалифицированного специалиста. Из-за больших временных затрат и трудоемкости этот метод обходится очень дорого.

1.2.2. Сегментация на основе интенсивности

К этим методам относят алгоритмы, выполняющие сегментацию на основе значений интенсивности пикселей. Эти методы очень просты и иногда дают хорошие результаты, однако они не учитывают пространственную информацию, а также чувствительны к шуму и неоднородностям в интенсивности.

1.2.2.1. Пороговые методы

Данный класс представляют методы, разделяющие изображения на 2 и более частей, основываясь на некоторых пороговых значениях интенсивности [19]. Порог — это значение на гистограмме изображения, которое разделяет его на две части: первая часть — область интереса, все пиксели, которые имеют значение больше или равным пороговому, а вторая — фон, пиксели которой имеют значение интенсивности ниже порога. Таким образом,

$$g(x, y) = \left\{ \begin{array}{l} 1, p(x, y) \geq T \\ 0, p(x, y) < T \end{array} \right\},$$

где $p(x, y)$ — значение интенсивности в точке (x, y) , а T — пороговое значение. Для выделения нескольких объектов используется несколько пороговых значений. Также существуют методы локальных и адаптивных порогов. В методах первого типа пороговое значение для позиции (x, y) устанавливается исходя из значений соседних пикселей, а в адаптивных методах значение порога — это функция от x и y .

1.2.2.2. Разрастание регионов

Это интерактивные методы, которые требуют установку начальных точек, чтобы начать процесс разрастания региона. Изображение разделяется на области в соответствии с заранее определенным законом на основе интенсивности. Недостаток подобных методов заключается в том, что результат работы алгоритма сильно зависит от установленных стартовых точек, которые зависят от человеческого фактора [3].

1.2.3. Кластеризация

Наиболее широко используемыми методами кластеризации являются k -means и expectation-maximization (EM).

K -means итеративно пересчитывает среднюю интенсивность для каждого класса и сегментирует изображение, относя каждый пиксель к классу с ближайшим средним значением. EM-алгоритм основан на предположении, что кластеризуемые данные подчиняются линейной комбинации (смеси) гауссовых распределений с неизвестными параметрами. Параметры вычисляются по принципу максимума правдоподобия.

Хотя методы кластеризации не требуют размеченных данных, их результаты зависят от установки входных параметров. Эти методы хороши тем, что способны обобщать различные наборы данных и обычно не требуют много времени для вычислений, но они чувствительны к шуму и поэтому не всегда дают желаемый результат.

1.2.4. Нейронные сети

В последние годы методы, основанные на нейронных сетях, успешно применяются для решения многих задач, связанных с обработкой изображений, в том числе задачи сегментации. Искусственные нейронные сети — это вычислительные сети, за основу которых была взята биологической нейронная сеть человека. Они представляют собой взаимодействие нейронов, которые посылают друг другу сигналы и обучаются на их основе.

Эту технику было решено взять за основу модуля сегментации по нескольким причинам. Во-первых, исходные данные обладают различными артефактами, например, иногда появляются полосы разной интенсивности, или "ложные изображения", идущие вдоль всего снимка, возникающие из-за пульсаций. К тому же фиброз может иметь разнообразную форму и находиться в разных местах, его интенсивность также варьируется. Таким образом, для решения поставленной задачи метод сегментации должен быть устойчив к шуму и учитывать пространственную информацию; этим свойствам удовлетворяют нейронные сети при должном разнообразии входных данных.

Ниже будут рассмотрены основные разновидности нейронных сетей, применяемые для анализа изображений.

1.2.4.1. Сверточные нейронные сети

Сверточные нейронные сети — семейство нейронных сетей, предназначенных для работы с изображениями. Архитектура данных сетей содержит следующие слои: сверточные, субдискретизации, активационные, добавляющие нелинейность, а также многослойный перцептрон на выходе модели, отвечающий за классификацию. В состав сверточных сетей могут входить некоторые дополнительные, например, ответственные за нормализацию.

Основу сверточных слоев составляет операция свертки, которая представляет собой перемножение фрагментов изображения с матрицей, называемой ядром фильтра, а затем сложением получившихся результатов. Особенностью сверточных слоев является разделение параметров

(англ. parameter sharing), заключающееся в том, что каждому фильтру соответствует один набор весов, что позволяет использовать один фильтр для детекции определенных признаков в разных частях изображения. Также важным является свойство разреженности связей (англ. sparsity of connections), что подразумевает собой то, что каждое выходное значение зависит только от небольшого набора входных данных.

Слои субдискретизации (англ. subsampling, pooling) позволяют снижать размерность, что достигается заменой группы входных данных на значение определенной статистики, посчитанной по элементам группы. Помимо уменьшения размерности, слои субдискретизации обеспечивают инвариантность относительно небольших сдвигов признака во входных данных, так как при небольшом сдвиге значения выходных нейронов слоев субдискретизации практически не меняются.

Благодаря разделению параметров, разреженности связей в сверточных слоях и слоям субдискретизации, сверточные нейронные сети имеют гораздо меньшее число параметров, по сравнению с полносвязными моделями, что в свою очередь ускоряет обучение и уменьшает риск переобучения.

1.2.4.2. Полносверточные нейронные сети

Сверточные сети выполняют классификацию всего изображения, за которую отвечает последний полносвязный слой, если его убрать, то на выходе сети будут "карты" классов для изображения любого размера, то есть сегментация. Основная проблема такого подхода заключается в слишком маленьком разрешении изображения, получаемого на выходе, поэтому была представлена архитектура полносверточных сетей (англ. Fully Convolutional Neural Networks, FCNs)[26].

FCN состоит из двух частей: первая часть — уменьшение разрешения изображения с целью извлечения контекстной информации, вторая часть — увеличение разрешения для определения точного местоположения. Также, чтобы более точно восстановить детализированную пространственную информацию, потерянную в первой части при уменьшении разрешения, часто используются пропускные соединения (англ.

skip-connections), которые позволяют "обходить" хотя бы один слой, тем самым объединяя или суммируя контекстную информацию разных разрешений с локальными признаками.

1.2.4.3. Энкодеры

Распространенной практикой является замена части, отвечающей за кодирование признаков, уже обученными на больших наборах данных нейронными сетями. Самым распространенным является ImageNet[12] — набор данных, содержащий 15 миллионов изображений высокого разрешения, каждое из которых относится к одной из 22,000 категорий. С 2010 года ежегодно проводится соревнование ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), которое определяет наилучший алгоритм, решающий задачи классификации и локализации. Ниже будут рассмотрены архитектуры нейронных сетей, занявшие лидирующие места ILSVRC разных лет. Так как они показали хорошие результаты в классификации изображений, было решено их попробовать в качестве энкодеров (частей, извлекающих контекстные признаки изображения).

1.2.4.3.1. VGG

VGG[27] является улучшенной версией AlexNet[18], которая была первой сверточной нейронной сетью, достигшей наилучший результат в ILSVRC-2012(топ-5 ошибка = 15.3%)[9]. VGG отличается простотой архитектуры и глубиной: эта нейронная сеть состоит из сверточных слоев (13 и 16 для VGG-16 и VGG-19 соответственно), чередующихся со слоями субдискретизации, и 3 полносвязными слоями в конце. VGG показала, что глубокая нейронная сеть дает хорошие результаты классификации, достигнув в ILSVRC-2014 топ-1 ошибки = 7.4%[10].

Недостаток VGG в том, что, несмотря на простую архитектуру, эта модель имеет достаточно много параметров обучения (138 миллионов и 144 миллиона для VGG-16 и VGG-19 соответственно) из-за большой глубины и полносвязных нейронов.

1.2.4.3.2. Inception

Основная цель Inception[8], заключалась в достижении эффективности с точки зрения размера модели и количества необходимых вычислений. Авторы Inception заменили размер сверток на меньший, при этом на каждом уровне применяется несколько сверток разных размеров (3x3, 5x5), чтобы извлекать признаки разных масштабов. Чтобы сделать вычисления менее сложными, число входных каналов уменьшается с помощью добавления сверточного слоя размера 1x1. Также в сети отсутствуют скрытые полносвязные слои, их заменяет average-pooling. Сеть достаточно глубокая, и чтобы решить проблему "затухания" градиента[7], было добавлено два дополнительных классификатора, которые считали функцию потерь, исходя из выходов более ранних слоев, а итоговая функция потерь является взвешенной суммой функции на выходе сети и двух дополнительных.

Описанная конфигурация соответствует первой версии Inception (v1). Inception v2 и v3[24] отличаются от предыдущей тем, что свертки размером 5x5 были заменены на две размером 3x3, что дало значительный прирост в производительности, не уменьшив качество модели. Также авторы заменили операции свертки с размером ядра $n \times n$ последовательным применением $1 \times n$ и $n \times 1$, при этом такое разложение улучшило скорость вычислений. В самом конце они сделали сеть не более широкой, а более глубокой, чтобы на последнем этапе сеть обучалась лучше. Так получилась архитектура Inception v2, от которой Inception v3 отличается добавлением пакетной нормализации (англ. batch-normalization) перед добавочными классификаторами, для увеличения скорости обучения.

Из всех представленных архитектур Inception v3 показала наилучший результат на ILSVRC-2015, заняв второе место с топ-5 ошибкой = 6.7%.

1.2.4.3.3. ResNet

Обучение глубоких нейронных сетей связано с проблемой деградации, когда с ростом глубины точность сначала растет, а затем начинает стремительно падать. Благодаря остаточным блокам (англ. residual

blocks) ResNet[5], можно обучать очень глубокие нейронные сети, избегая этого. Остаточные блоки состоят из слоев с весами (например, сверточных) и слоев активации, а также соединения под названием shortcut, через которое осуществляет тождественное преобразование, то есть через него данные проходят без изменения.

В основе проблемы деградации нейронной сети (когда после достижения максимума качество модели падает) лежит факт, что, если сети было достаточно определенного количества слоев для обучения, дальнейшие слои должны возвращать тождественное преобразование, однако этого не происходит и качество начинает падать. Смысл остаточных блоков заключается в том, что, по гипотезе авторов ResNet, если тождественное преобразование является наилучшим вариантом, то проще обучить оптимизатор приравнивать веса слоев к нулю, чтобы данные проходили только по skip-connections, чем обучать его созданию тождественного преобразования.

У архитектуры ResNet бывает несколько конфигураций. ResNet-18, ResNet-34 состоят из 18 и 34 слоев соответственно, один из которых полносвязный (в конце), а остальные сверточные; остаточный блок в этих конфигурациях содержит 2 слоя. ResNet-50, ResNet-101 и ResNet-152 содержат 50, 101 и 152 слоя соответственно, а длина остаточных блоков уже равна 3.

Архитектура ResNet показала хорошие результаты: конфигурация со 152 слоями заняла 1 место в ILSVRC-2015 с топ-5 ошибкой = 3.6%[11]. При этом примечательно, что количество параметров в сети со 152 слоями равно 60,2 миллиона, что намного меньше, чем в VGG-16.

1.3. Используемые технологии

Для реализации системы выбран язык программирования Python [23], так как он получил широкое распространение, особенно среди научного сообщества, в связи с чем для него написано большое количество библиотек для глубокого обучения и работы с данными и изображениями.

Для работы с изображениями используется библиотека SimpleITK [28],

которая позволяет работать со снимками различных форматов и применять к ним различные операции для обработки. Также при реализации использовалась библиотека NumPy [21], предоставляющая интерфейс для эффективного осуществления операций над многомерными массивами.

Для языка Python существует несколько библиотек глубокого обучения, самыми распространенными являются Keras [16] и PyTorch [22]. Keras представляет собой надстройку над Tensorflow [29] (библиотека, разработанная компанией Google для создания и обучения нейронных сетей) и Theano [30] (библиотека для эффективного вычисления математических выражений, особенно матричных). Основными характеристиками Keras являются ее удобство, модульность, простота расширения, а также ее распространенность среди разработчиков и хорошая документация. Библиотека PyTorch создана на базе Torch [31] — библиотеки для языка Lua, в которой реализовано большое количество алгоритмов для глубокого обучения. PyTorch отличается от Tensorflow(и, соответственно, от надстроенного над ним Keras) динамическим графом вычислений, что позволяет легко менять низкоуровневые блоки "на лету" и дает больший простор для экспериментов в построении нейронных сетей.

Так как библиотека PyTorch начала набирать популярность сравнительно недавно, в качестве библиотеки для обучения модели сегментации была выбрана библиотека Keras, ввиду ее большей распространенности в сообществе, что гарантирует быстрое решение возникших проблем, к тому же модульной архитектура библиотеки обеспечивает быстрое создание нейронных сетей.

2. Архитектура и особенности реализации

Для обеспечения не только поиска фиброзной ткани на медицинских снимках, но также и внедрения автоматической сегментации в рабочий процесс врача, работа системы состоит из двух частей: анализ снимков и взаимодействие с пользователем.

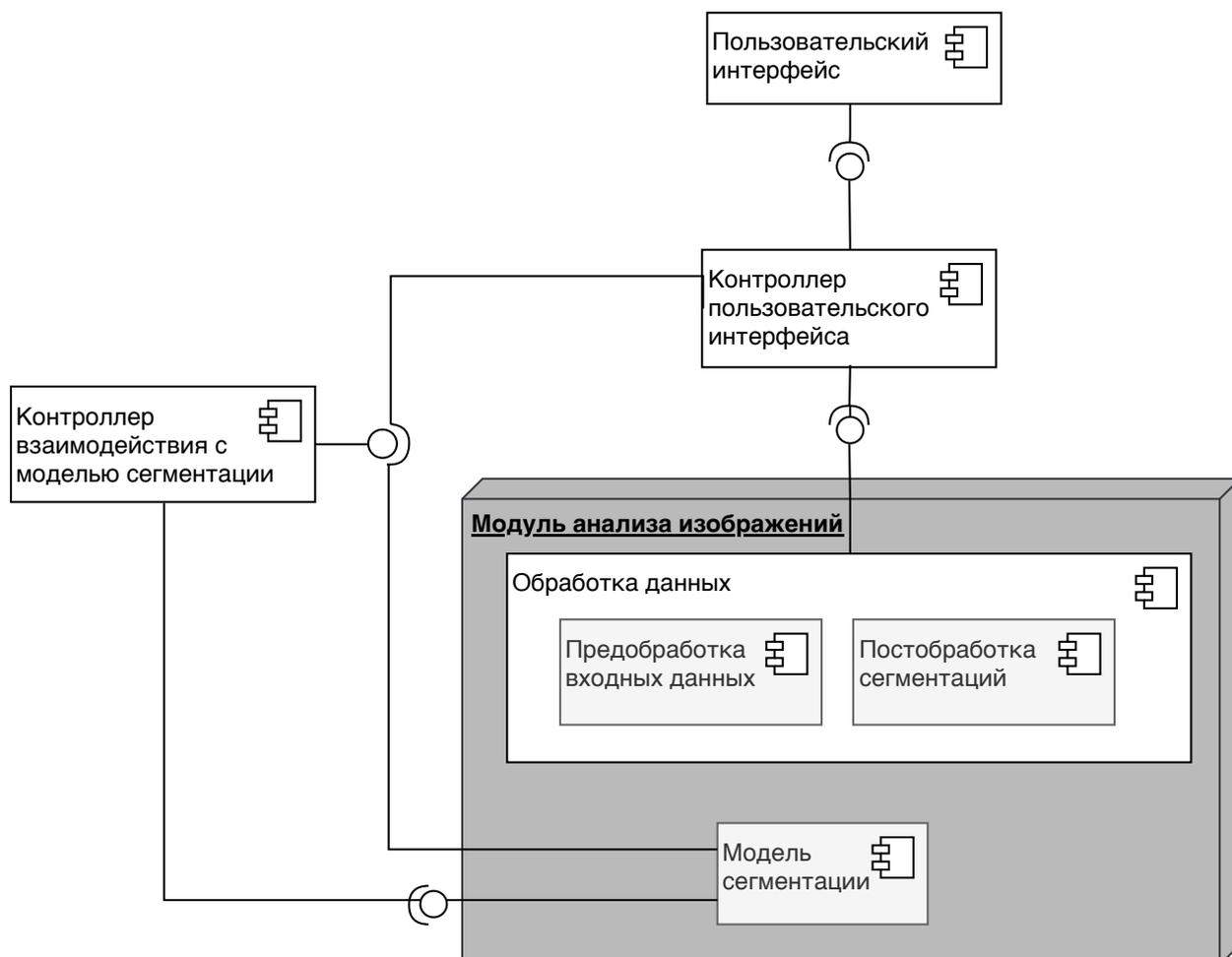


Рис. 1: Архитектура системы

На рис. 1 представлена архитектура системы. За анализ снимков отвечает модуль анализа изображений. Это основная часть системы, в которой модель сегментации выполняет главную функцию — выделение фиброзной ткани на медицинских изображениях. Данные, поступающие на вход модели, точно так же как и получающиеся в результате сегментации, имеют определенный формат и размер, поэтому перед сегментацией и после снимки обрабатываются компонентами предобработки входных данных и постобработки сегментаций соответственно.

Модель сегментации размещается удаленно, это объясняется, во-первых, сложной настройкой окружения, необходимого для запуска нейронной сети, а во-вторых, с необходимостью хранения отредактированных врачом сегментации для улучшения качества. Обработка данных может выполняться как на стороне пользователя, так и в месте размещенной системы, однако в данной реализации она выполняется на стороне пользователя, чтобы не перегружать сеть и передавать изображения меньшего разрешения. В связи с этим вызов компонентов предобработки и постобработки данных происходит через контроллер пользовательского интерфейса, который получает результат действий пользователя и отвечает за обмен данными между пользователем и модулем анализа изображений.

Для обеспечения удаленного доступа к модели сегментации реализован контроллер взаимодействия, который представляет собой веб-сервис и предоставляет API для доступа к данным и работы с ними. Контроллер позволяет запустить модель сегментации, предоставляет доступ к результатам работы модели на указанных данных для определенного пользователя, а также обеспечивает отправку отредактированной врачом автоматической сегментации для последующих тренировок модели с целью улучшения качества сегментаций.

Далее будут рассмотрены основные части системы и детали их реализации.

2.1. Модуль анализа изображений

Модуль анализа изображений состоит из двух компонентов: модели сегментации и компонента, отвечающего за обработку данных.

2.1.1. Модель сегментации

Модель сегментации изображений осуществляет поиск фиброзной ткани на изображениях, прошедших предобработку. Для сегментации используется предобученная нейронная сеть, которая принимает изображения размера $N \times M$. Результатом работы компонента являются так-

же изображения размером $N \times M$, состоящие из чисел от 0 до 1, где значение пикселя соответствует вероятности его отнесения к фиброзной ткани.

2.1.2. Обработка данных

Компонент обработки данных занимается обработкой входных изображений, а также постобработкой сегментаций, полученных в результате работы модели сегментации.

2.1.2.1. Обработка входных данных

Входные данные представлены в формате DICOM (Digital Imaging and Communications in Medicine) [4]. Это формат, предназначенный для хранения медицинских изображений, который помимо снимков позволяет хранить информацию о пациенте и данные об исследовании. Файлы объединяются в серии и представляют собой набор последовательных срезов какого-либо органа. Так как работать с отдельными файлами не очень удобно, к тому же для запуска сегментации информация о пациенте является лишней, было решено конвертировать каждую серию в формат NIfTI (Neuroimaging Informatics Technology Initiative) [20], в результате каждая серия представлена одним 3D изображением в виде трёхмерного массива.

После конвертации каждая серия изображений нормализуется, чтобы привести все входные значения к одной области изменения. Применялась линейная нормализация, которая выражается следующей формулой:

$$\tilde{x}_{ij} = \frac{x_{ij} - x_i^{min}}{x_i^{max} - x_i^{min}}, \text{ где}$$

x_{ij} — i входное значение для j пациента;

\tilde{x}_{ij} — соответствующее нормализованное значение.

Конечным этапом является уменьшение разрешения серии. В основном исходные снимки имеют разрешение 256×256 или 512×512 , что больше размера, установленного для входа модели сегментации (128×128),

поэтому необходимо уменьшить разрешение снимков, подаваемых на вход.

2.1.2.2. Обработка результатов работы модели сегментации

Полученные в результате изображения необходимо преобразовать в маски — изображения, содержащие пиксели со значением 0 или 1, где пиксели со значением 1 соответствуют фиброзной ткани. Для этого устанавливается определенный порог: пиксели со значением больше порогового становятся равными 1, а остальные 0.

Так как модель сегментации принимает изображения с уменьшенным разрешением, необходимо совершить обратную операцию и увеличить разрешение масок, чтобы корректно наложить их на исходные снимки.

2.2. Взаимодействие с пользователем

Для обеспечения удобного доступа к модулю сегментации было принято решение реализовать расширение к уже существующей программе, предназначенной для работы с медицинскими изображениями. Расширение к уже существующему решению достаточно просто решает задачу обеспечения взаимодействия пользователя с системой, так как не нужно самостоятельно реализовывать многие инструменты (например, редактирование сегментаций или просмотр изображений различного формата).

Расширение написано к программе 3D Slicer, которая содержит различные модули, предназначенные для анализа медицинских изображений. Выбор 3D Slicer объясняется тем, что, во-первых, она распространяется бесплатно с открытым исходным кодом, во-вторых, имеет популярность среди врачей. Также важным критерием при выборе программы для интеграции являлась простота встраивания: 3D Slicer позволяет писать модульные расширения на языке Python, которые можно добавить без дополнительной сборки проекта; также внутреннее окружение 3D Slicer уже содержит SimpleItk для работы с изображениями

и NumPy для выполнения операций над многомерными массивами с данными; еще одним достоинством является большое сообщество разработчиков и подробная документация.

2.2.1. Архитектура расширения

3D Slicer имеет архитектуру, соответствующую шаблону проектирования "Модель-Представление-Контроллер", поэтому и встраиваемое расширение должно иметь подобную такую структуру. На рис.2 представлена архитектура разработанного плагина. Контроллер пользовательского интерфейса реализован в виде компонента AutoSegmentation, который включает в себя классы Logic и Widget. Роль модели играют представители класса MRML node (Medical Reality Markup Language), которые являются внутренним представлением данных в 3D Slicer и хранят информацию об изображениях и параметрах визуализации.

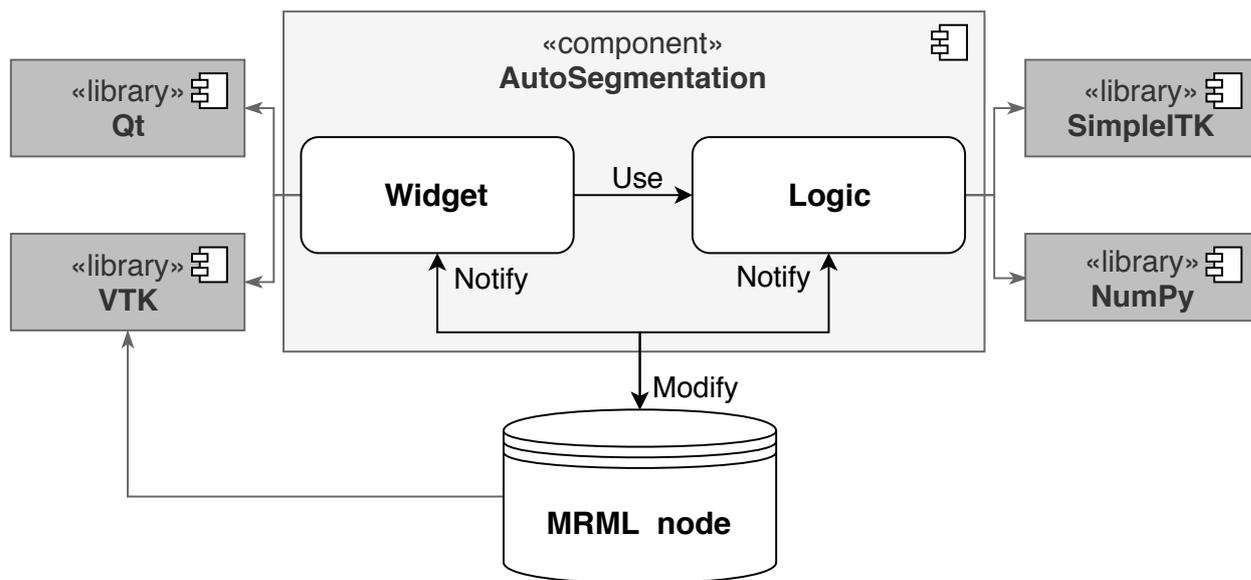


Рис. 2: Архитектура расширения к программе 3D Slicer

Для обеспечения представления был разработан класс Widget, в котором определяется пользовательский интерфейс. Этот класс при необходимости редактирует MRML объекты или их представление в окне просмотра изображений, синхронизируя таким образом пользовательский интерфейс и данные. MRML node и Widget используют библиотеку VTK[32], которая предоставляет инструменты для с изображениями,

а также содержит различные алгоритмы визуализации.

Реализованный класс `Logic` выполняет роль контроллера, в нем реализована основная логика по обработке данных. Важным моментом является отсутствие зависимости класса `Logic` от `Widget`, такое ограничение позволяет модулям использовать функциональности друг друга с помощью простого вызова методов. `Logic` использует библиотеку `SimpleITK` [28], чтобы обрабатывать трехмерные изображения пациента, и `NumPy` [21] для работы с многомерными массивами.

2.2.2. Пользовательский интерфейс

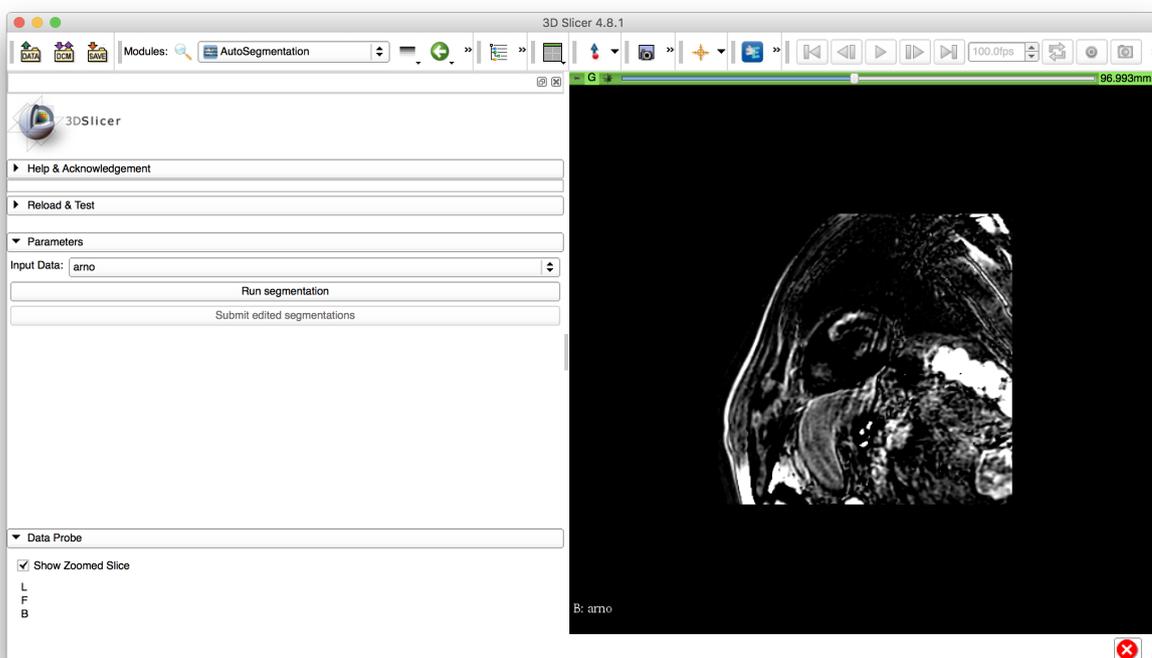


Рис. 3: Пользовательский интерфейс разработанного расширения

На рис.3 представлен пользовательский интерфейс разработанного расширения. В правой части экрана отображаются снимки пациента, а с помощью интерфейса на панели слева врач может выбирать исходные данные, запускать автоматическую сегментацию и отправлять отредактированные сегментации обратно, чтобы повышать качество модели. Редактировать полученные результаты врач может в уже встроенном в 3D Slicer модуле редактирования сегментации.

Рассмотрим подробно процесс взаимодействия пользователя с модулем анализа изображений, схема которого представлена на рис. 4. Перед отправкой на сегментацию данные предварительно обрабатываются с помощью вызова компонента предобработки данных. Далее через контроллер взаимодействия с моделью сегментации происходит запуск модели. Периодически контроллер пользовательского интерфейса отправляет на контроллер взаимодействия с моделью запрос на получение сегментаций. Как только сегментации готовы, они проходят постобработку через вызов компонента постобработки сегментаций и отправляются обратно пользователю. Пользователь может их отредактировать и отправить назад, пополняя таким образом тренировочный объём данных.

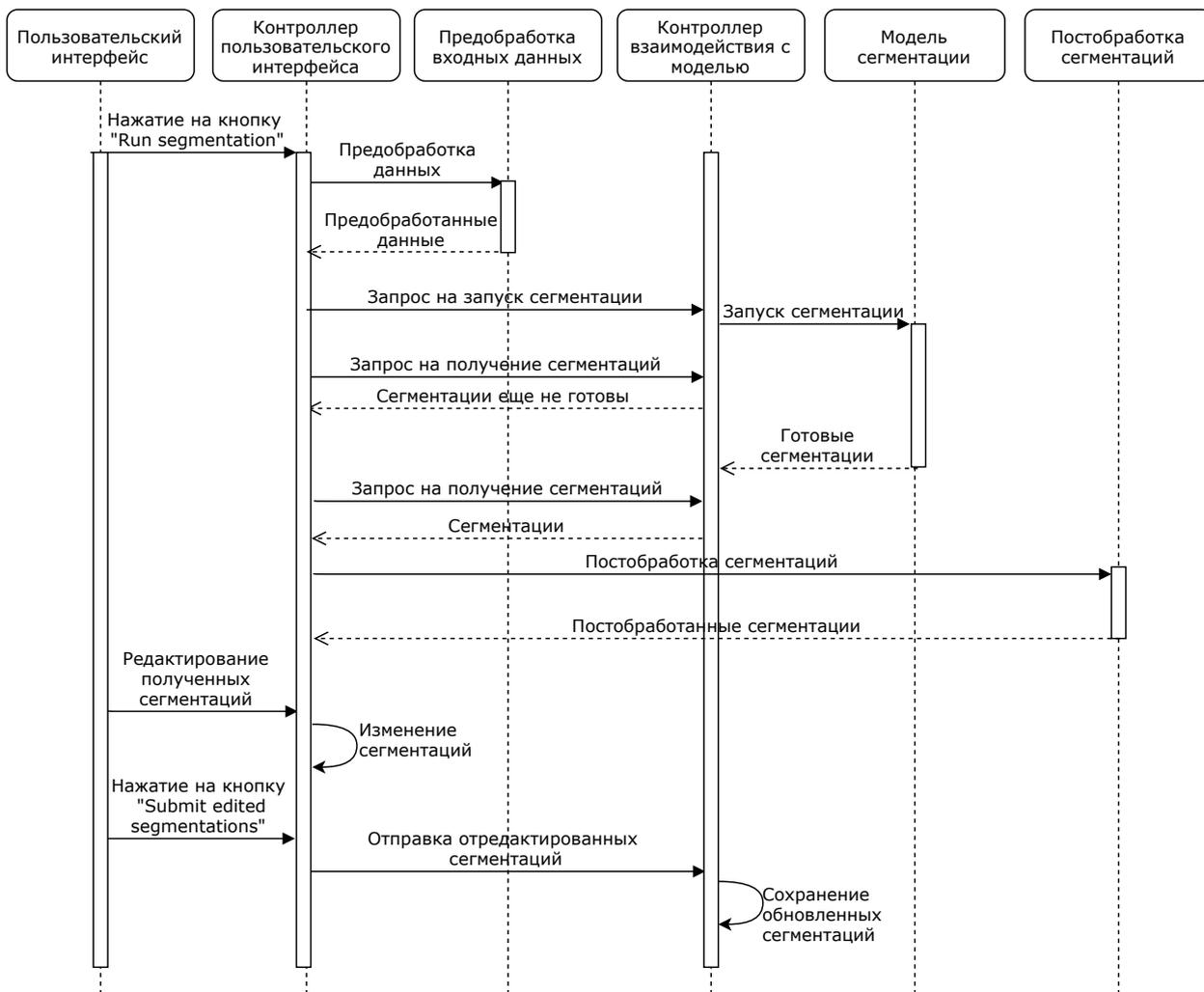


Рис. 4: Схема взаимодействия пользователя с модулем анализа изображений

3. Обучение модели сегментации

Модель сегментации выполняет основную функцию системы, а именно выделение фиброзной ткани на МРТ-снимках с контрастом, поэтому важным этапом является выбор модели и её обучение.

На рис. 5 представлена схема обучения модели. Можно выделить три этапа: подготовка данных, сегментация и оценка качества модели. Сначала входные данные проходят предобработку, описанную в 2.1.2.1.

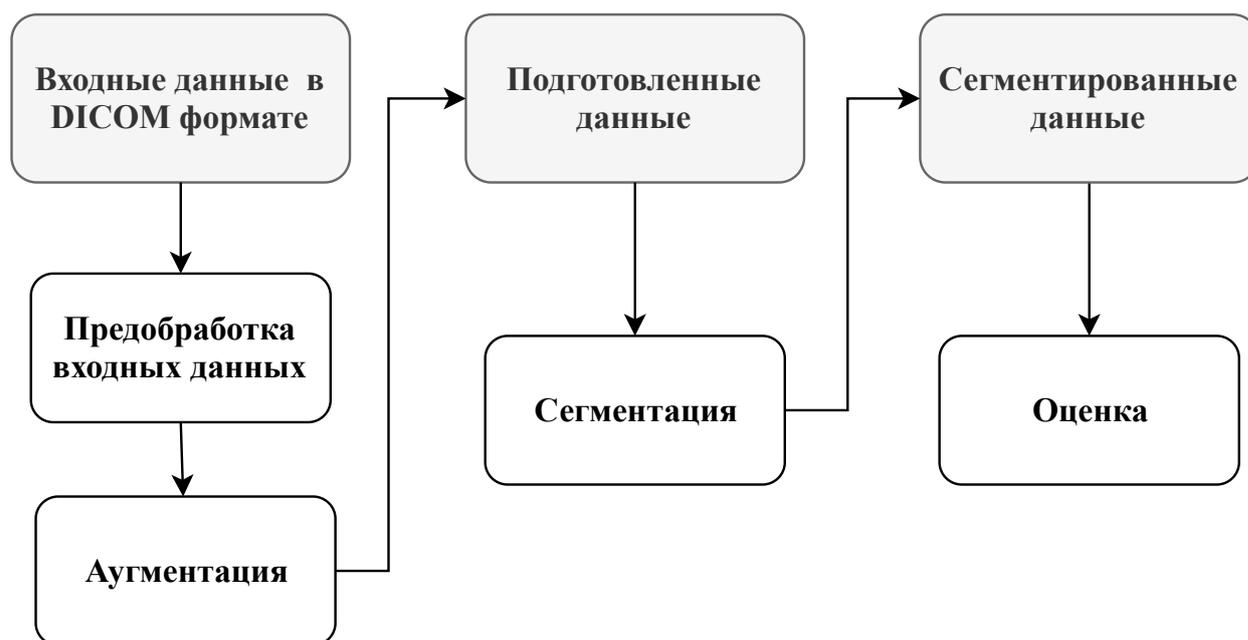


Рис. 5: Схема обучения модели сегментации

Затем для увеличения объема тренировочных данных выполняется аугментация. Далее подготовленные данные подаются на вход модели, после окончания обучения которой производится оценка качества с помощью количественных метрик и визуализации результатов. Ниже будут рассмотрены основные элементы процесса обучения.

3.1. Исходные данные

Исходные данные в формате DICOM предоставлены НМИЦ имени В.А. Алмазова после полной анонимизации. Сегментация фиброза проводилась вручную двумя специалистами. Фиброз может присутствовать не только в левом желудочке, но также в правом и обоих предсер-

диях, однако для начального этапа было решено сегментировать фиброз только в левом желудочке, так как определение местоположения в нём наиболее просто, его проще интерпретировать и к тому же фиброз именно в левом желудочке оказывает большое влияние на работу сердца, так как большая нагрузка в сердце лежит именно на нем.

Таким образом, институтом было предоставлено 960 изображений для 13 пациентов (на каждого пациента примерно 60-100 снимков) с сегментированным фиброзом в левом желудочке на каждом снимке.

3.2. Аугментация данных

В связи с достаточно маленьким количеством снимков, важным этапом являлась аугментация — процесс создания дополнительного объёма данных для обучения путем применения трансформаций к исходным данным.

Были применены различные случайные преобразования: вертикальное и горизонтальное смещение, вращение и масштабирование. Интервалы значений для преобразований подбирались опытным путем, исходя из исходных данных и изменения местонахождения сердца в снимках при применении преобразований:

- вертикальное/горизонтальное смещение — $[-0.1, 0.1)$, то есть изображение сдвигалось на число пикселей, равное случайной доле от общей высоты/ширины, взятой из приведенного полуинтервала, вниз/влево (отрицательные значения) или вверх/вправо (положительные);
- вращение — $[-15, 15)$, изображение вращалось на случайное целое число градусов из приведенного полуинтервала;
- масштабирование — $[-0.1, 0.1]$, масштаб изображения принимал случайные значения из отрезка $[0.9, 1.1]$.

Кроме расширения исходного набора данных, аугментация является одним из подходов регуляризации модели, так как помогает добиться более хорошего обобщения.

3.3. Нейросетевые модели сегментации

Для определения местоположения фиброза на снимках было решено взять за основу архитектуры U-Net[25] и LinkNet[1], которые являются модификациями полносверточных нейронных сетей. Схематичная архитектура U-Net и LinkNet представлена на рис. 6. Как и полносверточные сети, обе архитектуры состоят из двух частей: сужающейся (энкодера), отвечающей за контекстную информацию об изображении, и расширяющейся (декодера), которая служит для определения точного положения объекта.

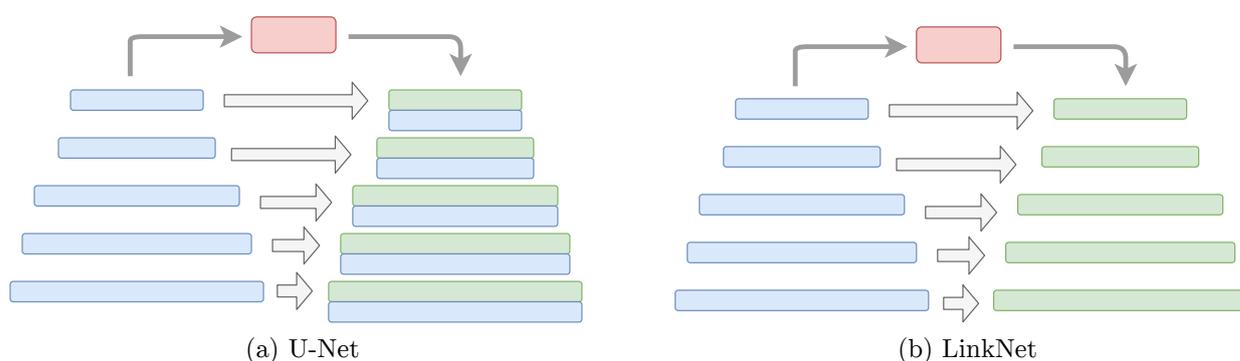


Рис. 6: Нейросетевые модели сегментации. Левая(синяя) часть архитектур — энкодер, правая(зеленая) — декодер. Светло-серые стрелки между энкодером и декодером — пропускные соединения.

Отличие этих нейронных сетей от полносверточных в том, что часть энкодера симметрична части декодера, то есть содержит множество операций увеличения разрешения (с помощью интерполяции) или транспонированных сверток (параметры ядра обучаются во время обучения модели), в то время как в FCN разрешения увеличивается всего лишь один раз.

От друг друга этих архитектуры отличаются пропускными связями: в U-Net происходит конкатенация признаков энкодера с декодером, в LinkNet — сложение. Поэтому архитектура LinkNet ”легковеснее”, так как имеет меньше параметров.

Для сегментации были выбраны именно эти архитектуры, так как пропускные соединения способствуют улучшению качества обучения глубоких нейронных сетей на медицинских данных[13], к тому же эти

архитектуры занимали первые места в различных соревнованиях по сегментации[15] [14].

3.4. Выбор модели сегментации

3.4.1. Метрика

Чтобы правильно выбрать модель для сегментации и настроить ее параметры, необходимо определиться с метрикой оценки ее качества. В качестве метрики для обучения модели и оценки результатов был взят коэффициент Дайса (Dice coefficient):

$$D = \frac{2 * |A \cap B|}{|A| + |B|}, \text{ где}$$

A — "ground truth" — правильная маска;

B — предсказанная маска.

Так как модель сегментации на выходе дает маски, в которых каждому пикселю соответствует вероятность отнесения его к фиброзной ткани, $|A \cap B|$ может быть аппроксимировано с помощью поэлементного произведения предсказанной и правильной масок, а затем сложением получившихся значений. Таким образом, пиксели, которые в действительности не соответствуют фиброзной ткани, будут равны нулю, а для оставшихся значений, пиксели с низкой вероятностью будут "штрафоваться", так как чем больше значение в числителе, тем выше значение метрики.

3.4.2. Обучение

Для преднастройки весов библиотека Keras предоставляет две опции: 78 снимков (10 пациентов) было отправлено на обучение, 182 (3 пациента) использовались в качестве валидационного множества. Так как в качестве основных моделей были выбраны LinkNet и U-Net, было решено попробовать заменить часть, соответствующую энкодеру, другими сверточными сетями, показавшими хорошее качество в соревнованиях по классификации (выделению контекстных признаков). Таким

образом, были протестированы различные сочетания энкодера и декодера: в качестве энкодеров были опробованы сверточные сети VGG, Inception и ResNet, рассмотренные в главе 1.2.4.3, а в качестве декодеров — U-Net и LinkNet.

Основные параметры обучения: оптимизационный алгоритм — Adam[17], функция потерь — перекрестная энтропия, размер батча и коэффициент скорости обучения подбирались опытным путем и составили 16 и $5e-5$ соответственно.

3.4.3. Анализ результатов обучения

Результаты представлены в таблице 1 в виде доверительных интервалов для коэффициента Дайса с уровнем доверия = 0.95. Добавленные энкодеры дали значительный прирост в качестве моделей. Так как VGG-16 показала низкие результаты по сравнению с другими энкодерами, был сделан вывод, что данная сеть слишком сложна для данной задачи, поэтому было решено не использовать VGG-19, глубина и число параметров которой еще больше. Для U-Net оптимальной оказалась ResNet-34. Возможная причина в том, что конфигурации с 50, 101 и 152 слоями имеют остаточные блоки длиной не 2, а 3, в связи с чем модель оказывается достаточно сложной при взаимодействии с U-Net, наступает переобучение. Для LinkNet оптимальным оказался энкодер ResNet-50.

Энкодер	Декодер	
	U-Net	LinkNet
—	0.22 ± 0.05	0.26 ± 0.04
VGG-16	0.31 ± 0.05	0.32 ± 0.04
ResNet-18	0.56 ± 0.04	0.45 ± 0.04
ResNet-34	0.58 ± 0.04	0.51 ± 0.04
Resnet-50	0.56 ± 0.04	0.59 ± 0.04
Resnet-101	0.49 ± 0.05	0.48 ± 0.05
Resnet-152	0.53 ± 0.04	0.55 ± 0.04
Inception v3	0.42 ± 0.05	0.43 ± 0.05

Таблица 1: Результаты обучения

Конечным решением является ансамбль U-Net и LinkNet с энкодерами ResNet-34 и ResNet-50 соответственно, результаты представлены в таблице 2. Ансамбль представляет собой простой подсчет среднего от вероятностного предсказания двух моделей, а затем, как и раньше, пиксели, значение которых меньше порогового значения (в данной реализации опытным путем выбрано значение $= 0.05$), становились равными 0, а остальные 1. Такая комбинация дала наилучший результат, что связано с тем, что архитектуры U-Net и LinkNet различаются, каждая сеть дополняет другую, улучшая общий результат.

Resnet34&U-Net + Resnet50&LinkNet
0.61±0.03

Таблица 2: Ансамбль наилучших результатов

4. Апробация

4.1. Метод

Апробация проводилась с использованием реализованного расширения к программе 3D Slicer практикующим врачом-хирургом, являющимся кандидатом медицинских наук, ассистентом кафедры рентгенологии и радиологии ФГБВОУ ВО «Военно-медицинская академия им. С.М. Кирова» МО РФ.

Процесс апробации заключался в загрузке данных пациента в DICOM формате в программу 3D Slicer, затем врач выбирал реализованное расширение и отправлял снимки на сегментацию; после получения снимков при необходимости врач редактировал результаты работы модели, а затем отправлял отредактированные сегментации обратно для расчета оценки качества модели. Качество работы системы оценивалось по двум параметрам: количественной метрике и отзыву врача.

4.2. Результаты апробации

В таблице 3 приведена количественная оценка результатов апробации. Как видно из таблицы, значение не сильно отличаются от результатов на валидационном множестве (результатов обучения модели), что позволяет сделать вывод о стабильности модели сегментации.

Результат апробации
0.58±0.04

Таблица 3: Результат апробации

На рис.7 представлены результаты апробации для разных пациентов. Зеленым цветом обозначена сегментация, сделанная моделью, а красным — исправленный врачом вариант. По полученным результатам можно сделать вывод, что модель достаточно хорошо определяет местонахождение фиброза и даже его форму, однако контуры модели шире, чем фиброз. Возможное объяснение этого недостатка состоит в

том, что тренировочный набор данных размечался с использованием достаточно широкой кисти, в результате не все контуры получались аккуратными. Так как модель обучалась на контурах, нарисованных широкой кистью, то и результат получается тоже не совсем точный. Данный недостаток можно будет исправить, дообучив модель на большем количестве отредактированных сегментаций, где контуры уже будут более аккуратными.

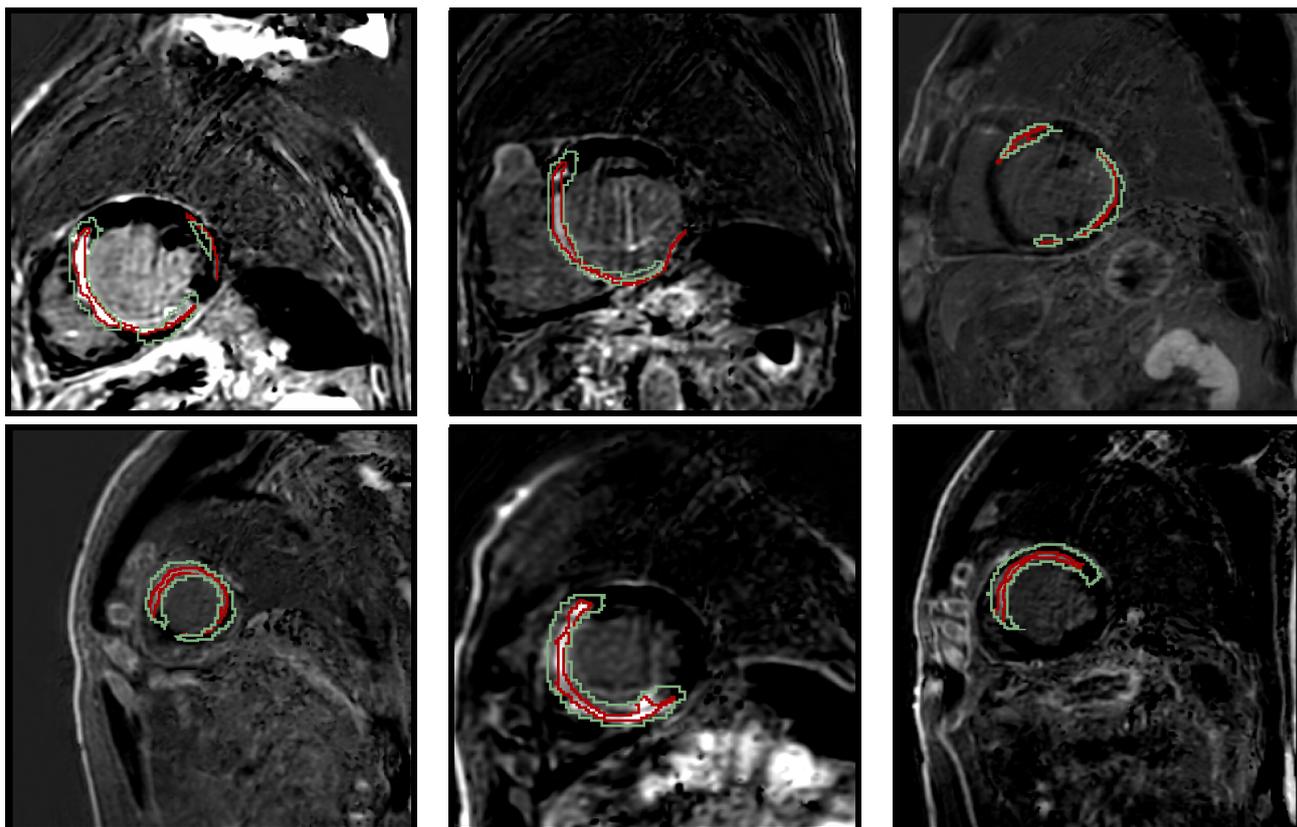


Рис. 7: Результаты апробации. Зеленый цвет — сегментация, выполненная реализованной моделью, красный — исправленный врачом вариант.

Стоит отметить, что, по сравнению с ручными методом, сегментация с помощью реализованной системы занимает гораздо меньше времени: на отправку исходных изображений, сегментацию и получение результатов ушло примерно 25 секунд, а на редактирование снимков — 20 минут, то есть в среднем 20,5 минут, в то время как ручная сегментация занимает 1-1,5 часа. В дальнейшем последний этап будет занимать меньшее время, во-первых, из-за постепенного обучения модели на все большем наборе данных и как следствие увеличение качества сегмен-

таций, а во-вторых, врач уже не будет тратить время на ознакомление с интерфейсом 3D Slicer, как это происходило в самом начале.

Несмотря на то что результаты модели немного шире контуров фиброза на снимках, система получила положительные отзывы, так как время сегментации значительно сократилось и местоположение фиброза определяется достаточно точно. Таким образом, по результатам апробации можно сделать вывод о возможности введения данной системы в рабочий процесс врача для сегментации фиброза на медицинских снимках.

Заключение

В ходе работы были получены следующие результаты:

- проведен анализ существующих решений и методов сегментации изображений; метод на основе нейронных сетей был выделен как наиболее перспективный;
- разработана архитектура системы анализа МРТ-снимков сердца с контрастом;
- реализован модуль анализа изображений; в том числе обучена модель сегментации на основе ансамбля нейронных сетей;
- проведена апробация системы с помощью написанного расширения к программе 3D Slicer, в ходе которой решение получило положительные отзывы и показало высокую точность и скорость работы.

Из полученных результатов можно сделать вывод, что реализованная система пригодна для использования во врачебной практике для выделения фиброза на медицинских снимках. На данный момент система все еще требует небольшого участия врача для редактирования сегментаций, однако в дальнейшем планируется пополнять базу пациентов для дальнейшего улучшения модели, чтобы результаты системы не требовали исправлений. Также в планах начать сегментировать фиброз не только в левом желудочке, но и в правом, а затем перейти к предсердиям, где диагностирование фиброза наиболее сложно.

Список литературы

- [1] Chaurasia Abhishek, Culurciello Eugenio. LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. — 2017. — 06.
- [2] Circle Cardiovascular Imaging // Circle Cardiovascular Imaging official page. — Access mode: <https://www.circlecvi.com/why-circle/> (online; accessed: 27.12.2018).
- [3] D. L. Pham C. Xu J. L. Prince. A survey of current methods in medical image segmentation // Annual Review of Biomedical Engineering. — 1998. — Vol. 2. — P. 315–337.
- [4] DICOM Standard // DICOM standart official page. — Access mode: <https://www.dicomstandard.org/current/> (online; accessed: 10.04.2019).
- [5] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep Residual Learning for Image Recognition. — 2015. — cite arxiv:1512.03385Comment: Tech report. Access mode: <http://arxiv.org/abs/1512.03385>.
- [6] Dilpreet Kaur Yadwinder Kaur. Various Image Segmentation Techniques: A Review // International Journal of Computer Science and Mobile Computing. — 2014. — Vol. 3. — P. 809–814.
- [7] Glorot Xavier, Bengio Yoshua. Understanding the difficulty of training deep feedforward neural networks // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics / Ed. by Yee Whye Teh, Mike Titterington. — Vol. 9 of Proceedings of Machine Learning Research. — Chia Laguna Resort, Sardinia, Italy : PMLR, 2010. — 13–15 May. — P. 249–256. — Access mode: <http://proceedings.mlr.press/v9/glorot10a.html>.
- [8] Szegedy Christian, Liu Wei, Jia Yangqing et al. Going deeper with convolutions. — 2014.

- [9] ILSVRC-2012 results // LSVRC-2012 results. — Access mode: <http://www.image-net.org/challenges/LSVRC/2012/results> (online; accessed: 10.04.2019).
- [10] ILSVRC-2014 results // LSVRC-2014 results. — Access mode: <http://www.image-net.org/challenges/LSVRC/2014/results> (online; accessed: 10.04.2019).
- [11] ILSVRC-2015 results // LSVRC-2015 results. — Access mode: <http://www.image-net.org/challenges/LSVRC/2015/results> (online; accessed: 10.04.2019).
- [12] ImageNet // ImageNet official page. — Access mode: <http://www.image-net.org> (online; accessed: 10.04.2019).
- [13] The Importance of Skip Connections in Biomedical Image Segmentation / Michal Drozdal, Eugene Vorontsov, Gabriel Chartrand et al. — 2016. — 08.
- [14] Kaggle Carvana Image Masking Challenge, 1st place solution // 1st place solution Kaggle Carvana Image Masking Challenge github source code. — Access mode: https://github.com/asanakoy/kaggle_carvana_segmentation (online; accessed: 10.04.2019).
- [15] Kaggle Dstl Satellite Imagery Competition, 1st place solution // Dstl Satellite Imagery Competition, 1st Place Winner’s Interview: Kyle Lee. — Access mode: <http://blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-1> (online; accessed: 10.04.2019).
- [16] Keras // Keras official page. — Access mode: <https://keras.io> (online; accessed: 17.12.2018).
- [17] Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization // 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9,

2015, Conference Track Proceedings. — 2015. — Access mode: <http://arxiv.org/abs/1412.6980>.

- [18] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems 25 / Ed. by F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger. — Curran Associates, Inc., 2012. — P. 1097–1105. — Access mode: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>.pdf.
- [19] M. Sonka V. Hlavac R. Boyle. Image Processing, Analysis, and Machine Vision. — 1999. — P. 122–133.
- [20] NIfTI // NIfTI format official page. — Access mode: <https://nifti.nimh.nih.gov> (online; accessed: 10.04.2019).
- [21] NumPy // NumPy official page. — Access mode: <https://www.numpy.org> (online; accessed: 07.05.2019).
- [22] PyTorch // PyTorch official page. — Access mode: <https://pytorch.org> (online; accessed: 07.05.2019).
- [23] Python 3 // Python official page. — Access mode: <https://docs.python.org/3/> (online; accessed: 17.12.2018).
- [24] Rethinking the Inception Architecture for Computer Vision / Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe et al. — 2016. — 06.
- [25] Ronneberger O., P.Fischer, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // Medical Image Computing and Computer-Assisted Intervention (MICCAI). — Vol. 9351 of LNCS. — Springer, 2015. — P. 234–241. — (available on arXiv:1505.04597 [cs.CV]). Access mode: <http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a>.

- [26] Shelhamer Evan, Long Jonathon, Darrell Trevor. Fully Convolutional Networks for Semantic Segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2016. — 05. — Vol. 39. — P. 1–1.
- [27] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // International Conference on Learning Representations. — 2015.
- [28] SimpleITK // SimpleITK official page. — Access mode: <http://www.simpleitk.org> (online; accessed: 17.12.2018).
- [29] Tensorflow // Tensorflow official page. — Access mode: <https://www.tensorflow.org> (online; accessed: 17.12.2018).
- [30] Theano // Theano official page. — Access mode: <http://deeplearning.net/software/theano/> (online; accessed: 17.12.2018).
- [31] Torch // Torch official page. — Access mode: <http://torch.ch> (online; accessed: 07.05.2019).
- [32] VTK // VTK official page. — Access mode: <https://vtk.org> (online; accessed: 07.05.2019).