Изображение выглядит как логотип

Автоматически созданное описание**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ   
(национальный исследовательский университет)»**

**Программа стратегического академического лидерства «Приоритет – 2030»**

**ПРОЕКТ «ЦИФРОВАЯ КАФЕДРА»**

**Дополнительная профессиональная программа профессиональной переподготовки  
Методы искусственного интеллекта и предиктивная аналитика в проектах дефектоскопии**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА (IT-ПРОЕКТ)**

на тему: «Применение сверточных нейронных сетей для сегментации сосудов при наличии стеноза с использованием данных рентгеновской коронарографии»

Руководитель: к.т.н., доцент 806 кафедры, Ухов П.А. (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

Консультант: к.т.н., доцент 806 кафедры, Стрижак С.В. (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

К защите допустить

Руководитель ДПП ПП   
Ухов Петр Александрович (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

\_\_ \_\_\_\_\_\_ 2024 года

Москва 2024

[Команда проекта 3](#_bkoq955cmp4h)

[Актуальность исследования 4](#_mb5xatzetafh)

[Проблематика 4](#_6g3qp7bfata)

[Цель исследования 6](#_71nckjqsye1c)

[Задачи исследования 6](#_7kgv9ie05xvi)

[Научная новизна 7](#_v20v2f6kfj5h)

[Практическая значимость 7](#_z3e80brfjpyx)

[Структура работы 8](#_4uh82kdkp9k9)

[Теоретические аспекты 9](#_e14qvpk81onn)

[Введение 12](#_p0kp72w8h9ze)

[Стек технологий 31](#_5k4vt14hmc2n)

[Код 45](#_68elkzgmfmx1)

[Заключение](#_zc0gz9btqcgh)63

# Команда проекта

| Коломытцева Екатерина Андреевна | Team-Lead |
| --- | --- |
| Титкова Ольга Алексеевна | Ml-engineer |
| Корнилова Виктория Алексеевна | Копирайтер |
| Кулемин Евгений Вадимович | Оформление отчётов |
| [Янышкин Максим Дмитриевич](https://teammai.ru/profile/mdyanyshkin/data) | Копирайтер |
| [Браун Александр антони](https://teammai.ru/profile/aabraun/data) | Пиар |
| [Паренькова Анастасия Евгеньевна](https://teammai.ru/profile/aeparenkova/data) | Пиар |
| [Иванов Кирилл Александрович](https://teammai.ru/profile/kiriaivanov/data) | Копирайтер |
| [Яковлева Анастасия Игоревна](https://teammai.ru/profile/anaiyakovleva/data) | Пиар |
| Леонов Илья Денисович | Пиар |

# Актуальность исследования

Сердечно-сосудистые заболевания (ССЗ) продолжают оставаться одной из ведущих причин смертности и инвалидности во всем мире. Согласно данным Всемирной организации здравоохранения, каждый год от ССЗ умирают миллионы людей. Одним из ключевых методов диагностики этих заболеваний является рентгеновская коронарография, которая позволяет визуализировать коронарные артерии и выявлять стенозы – сужения сосудов, которые могут привести к инфаркту миокарда и другим серьезным последствиям. Однако, несмотря на высокую эффективность рентгеновской коронарографии, процесс её проведения и интерпретации результатов требует значительных усилий со стороны медицинского персонала, что может увеличивать время принятия клинических решений и задерживать начало лечения.

# Проблематика

Одной из ключевых проблем в анализе рентгеновских изображений коронарографии является точная и автоматическая сегментация сосудов, особенно в условиях наличия стенозов. Сегментация сосудов – это процесс выделения контуров сосудов на изображениях, что позволяет более точно оценить их состояние и определить наличие патологий. Существуют различные методы сегментации, которые можно разделить на две большие группы: традиционные методы и методы на основе машинного обучения.

Традиционные методы включают в себя:

* Методы порогового значения: Эти методы основываются на выборе порогового значения яркости, которое разделяет изображение на сосуды и фон. Преимущества этих методов включают их простоту и скорость выполнения. Однако они часто неэффективны при анализе изображений с низкой контрастностью или при наличии шума и артефактов.
* Методы на основе фильтрации: Включают в себя различные фильтры (например, фильтры Габора или фильтры Фурье), которые используются для выделения структурных особенностей сосудов. Эти методы могут быть более точными, чем пороговые, но их эффективность также зависит от качества изображения и наличия шума.
* Методы на основе активных контуров и сплайн-аппроксимации: Эти методы используют кривые или сплайны для моделирования контуров сосудов. Они могут быть более гибкими и точными, но часто требуют значительных вычислительных ресурсов и точной настройки параметров.

Методы на основе машинного обучения и, в частности, сверточных нейронных сетей (CNN) стали набирать популярность благодаря их способности автоматически извлекать и анализировать сложные пространственные характеристики изображений. Основные преимущества использования CNN для сегментации сосудов включают:

* Автоматическое извлечение признаков: CNN могут автоматически обучаться извлекать важные признаки из изображений, что делает их более универсальными и гибкими по сравнению с традиционными методами.
* Высокая точность и обобщающая способность: Благодаря сложной архитектуре и возможности обучаться на больших объемах данных, CNN могут достигать высокой точности в задачах сегментации.
* Способность работать с изображениями низкого качества: CNN могут быть устойчивы к шуму и артефактам, что делает их эффективными при работе с реальными медицинскими данными.

# Цель исследования

Целью данной аттестационной работы является разработка и оценка эффективности применения сверточных нейронных сетей (CNN) для автоматической сегментации сосудов на рентгеновских изображениях коронарографии при наличии стенозов. Сверточные нейронные сети представляют собой мощный инструмент для анализа изображений, поскольку они способны автоматически извлекать и обрабатывать сложные пространственные характеристики, что делает их идеальными для задач медицинской сегментации.

# Задачи исследования

Для достижения поставленной цели в работе решаются следующие задачи:

1. Проведение обзора литературы и анализ существующих методов сегментации сосудов на рентгеновских изображениях коронарографии. Определение их преимуществ, недостатков и областей применения.
2. Разработка архитектуры сверточной нейронной сети, оптимизированной для задач сегментации сосудов. Определение оптимальных гиперпараметров модели и методов предобработки данных.
3. Обучение и тестирование разработанной модели на реальных наборах данных, содержащих рентгеновские изображения коронарографии с наличием стенозов. Использование методов аугментации данных для улучшения обобщающей способности модели.
4. Сравнение результатов работы модели с традиционными методами сегментации и анализ полученных данных. Оценка точности, чувствительности, специфичности и других метрик производительности модели.
5. Оценка клинической применимости разработанной модели и перспективы её внедрения в практику. Разработка рекомендаций по использованию модели в различных клинических сценариях.

# Научная новизна

Научная новизна работы заключается в разработке специализированной модели сверточной нейронной сети, адаптированной для сегментации сосудов с учетом особенностей рентгеновских изображений коронарографии и наличия стенозов. В отличие от существующих методов, предложенная модель направлена на повышение точности и быстроты анализа изображений, что может значительно улучшить диагностику и лечение сердечно-сосудистых заболеваний. Кроме того, исследование включает разработку новых подходов к предобработке данных и аугментации изображений, что также является значительным вкладом в область медицинской информатики.

# Практическая значимость

Практическая значимость исследования заключается в возможности применения разработанных алгоритмов и моделей в клинической практике для автоматизированной обработки рентгеновских изображений коронарографии. Внедрение таких систем позволит снизить нагрузку на медицинский персонал, сократить время анализа изображений и повысить точность диагностики. Это, в свою очередь, может способствовать улучшению качества медицинской помощи и снижению риска ошибок, связанных с человеческим фактором. Более того, автоматизация процесса анализа изображений может стать важным шагом на пути к более персонализированному и эффективному лечению пациентов с сердечно-сосудистыми заболеваниями.

#### 

# Структура работы

Аттестационная работа состоит из нескольких ключевых разделов:

1. Введение: Представлены обоснование актуальности темы, постановка цели и задач исследования, а также описана научная новизна и практическая значимость работы.
2. Обзор литературы: Рассмотрены существующие методы и подходы к сегментации сосудов на рентгеновских изображениях, включая традиционные алгоритмы и современные подходы с использованием машинного обучения и нейронных сетей.
3. Методология: Описаны этапы разработки архитектуры сверточной нейронной сети, методы предобработки и аугментации данных, а также процесс обучения и тестирования модели. Рассмотрены использованные наборы данных и метрики оценки качества модели.
4. Результаты исследования: Приведены результаты тестирования разработанной модели, включая сравнительный анализ с традиционными методами сегментации. Оценены метрики точности, чувствительности и специфичности модели, а также проанализированы случаи ошибок и способы их минимизации.
5. Обсуждение: Проведен анализ полученных результатов, выявлены сильные и слабые стороны предложенной модели, а также рассмотрены перспективы её применения в клинической практике. Обсуждены возможные направления дальнейших исследований и улучшений модели.
6. Заключение: Сформулированы основные выводы по результатам проведенного исследования, подведены итоги работы и предложены рекомендации по дальнейшему развитию и внедрению разработанной технологии.

Таким образом, данное исследование направлено на решение актуальной проблемы автоматизации анализа рентгеновских изображений коронарографии и представляет собой значительный вклад в развитие медицинских технологий. С помощью разработанных методов и моделей можно существенно улучшить процесс диагностики и лечения сердечно-сосудистых заболеваний, повысив качество медицинской помощи и снизив риск ошибок, связанных с человеческим фактором.

# Теоретические аспекты

**Сверточные нейронные сети (CNN)** – это класс глубинных нейронных сетей, широко используемых для анализа визуальных данных. Они основаны на принципе свертки, который позволяет извлекать пространственные и иерархические особенности изображений. Основные компоненты CNN включают сверточные слои, слои подвыборки (пулинга) и полностью связанные слои.

* Сверточные слои: Основной элемент CNN, который выполняет операцию свертки над входным изображением. Этот слой применяет фильтры (ядра свертки) для создания карт признаков, которые выявляют локальные особенности изображения, такие как края, текстуры и формы.
* Слои подвыборки: Эти слои уменьшают размер карт признаков, сохраняя при этом их важные характеристики. Наиболее распространенные методы подвыборки включают max-pooling и average-pooling, которые помогают уменьшить размерность данных и повысить устойчивость модели к сдвигам и деформациям изображений.
* Полностью связанные слои: Завершающий этап в CNN, который соединяет все нейроны предыдущего слоя со всеми нейронами следующего слоя. Эти слои используются для классификации и позволяют модели принимать окончательные решения на основе извлеченных признаков.

**Сегментация изображений** – это процесс разбиения изображения на несколько сегментов или областей для упрощения анализа и интерпретации. В контексте медицинской визуализации сегментация часто используется для выделения и анализа анатомических структур, патологических областей и других значимых элементов.

* Сегментация на основе пороговых значений: Один из простейших методов, основанный на выборе порогового значения яркости. Все пиксели с интенсивностью выше порога считаются принадлежащими к объекту, а остальные – к фону.
* Сегментация на основе активных контуров: Метод, который использует кривые или сплайны для моделирования контуров объектов. Эти контуры итеративно обновляются, чтобы соответствовать границам объектов на изображении.
* Сегментация с использованием методов машинного обучения: Включает использование алгоритмов машинного обучения, таких как k-means, случайные леса и, конечно же, сверточные нейронные сети. Эти методы могут быть обучены на размеченных наборах данных для автоматической сегментации новых изображений.

**Проблемы сегментации сосудов**: Сегментация сосудов на медицинских изображениях представляет собой уникальные вызовы. Ключевые проблемы включают:

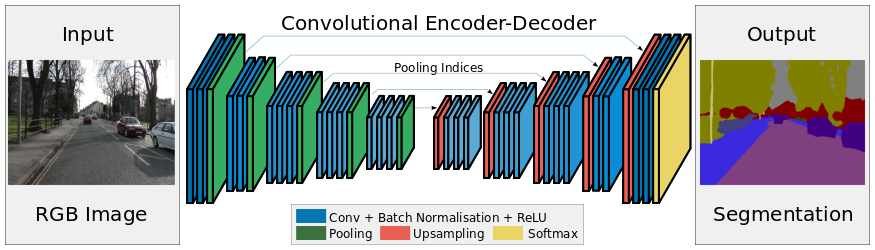
* Низкая контрастность: Сосуды часто имеют низкий контраст по сравнению с окружающими тканями, что затрудняет их выделение.
* Наличие шума и артефактов: Медицинские изображения могут содержать шум и артефакты, которые могут мешать точной сегментации.
* Вариативность формы и размера сосудов: Сосуды могут значительно варьироваться по форме и размеру, что делает их сложными для сегментации стандартными методами.

Использование сверточных нейронных сетей для сегментации сосудов может существенно улучшить точность и эффективность этого процесса, поскольку CNN способны автоматически извлекать важные признаки и адаптироваться к различным условиям изображений.

# Введение

Тема использования сверточных нейронных сетей для сегментации сосудов с наличием стеноза на изображениях рентгеновской коронарографии представляет собой актуальное исследование в области медицинской визуализации и искусственного интеллекта. Сегментация сосудов играет важную роль в диагностике сердечно-сосудистых заболеваний, таких как атеросклероз, ишемическая болезнь сердца и другие патологии, связанные с стенозами и окклюзиями сосудов. В данной работе будет проведен анализ применения метода U-Net для извлечения сосудов с высокой точностью и надежностью на изображениях рентгеновской коронарографии. Также будет изучено применение алгоритмов балансировки и аугментации данных, направленных на улучшение результатов сегментации и повышение общей эффективности модели. Основные задачи исследования включают в себя анализ метода U-Net, изучение алгоритмов балансировки и аугментации данных, а также разработку алгоритма сегментации сосудов с наличием стеноза. Работа также будет посвящена применению сверточных нейронных сетей для детекции стенозов в сердечно-сосудистой системе, что имеет важное значение для ранней диагностики и лечения сердечно-сосудистых заболеваний. Исследование также будет включать в себя сравнительный анализ методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии, что позволит выявить преимущества и недостатки каждого метода. Кроме того, будет рассмотрено применение сверточных нейронных сетей для сегментации сосудов на изображениях с различными типами стенозов, что способствует более точной и индивидуализированной диагностике. Таким образом, данная работа будет посвящена важной проблеме медицинской диагностики и исследованию современных методов сегментации сосудов с использованием сверточных нейронных сетей. Результаты и выводы исследования могут иметь значительное практическое применение в области кардиологии и медицинской визуализации.

**Применение сверточных нейронных сетей для сегментации сосудов на изображениях МРТ**

****

Примеры применения сверточных нейронных сетей для сегментации сосудов на изображениях МРТ

Сегментация сосудов на изображениях МРТ является важной задачей в области медицинского образования. Сосуды играют ключевую роль в кровообращении человеческого организма, и точное определение их структуры и состояния имеет критическое значение для диагностики различных заболеваний. Применение сверточных нейронных сетей для сегментации сосудов на изображениях МРТ позволяет автоматизировать этот процесс и обеспечить более точные и надежные результаты. Сверточные нейронные сети (CNN) являются мощным инструментом для анализа изображений благодаря своей способности извлекать признаки из данных различных масштабов. В контексте сегментации сосудов на изображениях МРТ, CNN могут обучаться на размеченных данных для автоматического выделения сосудистых структур. Одним из наиболее эффективных подходов к сегментации сосудов является использование архитектуры U-Net, которая показывает отличные результаты в задачах медицинского изображения. Преимущества применения сверточных нейронных сетей для сегментации сосудов на изображениях МРТ включают высокую точность и скорость обработки данных. CNN способны автоматически выделять сложные структуры сосудов даже при наличии шума или артефактов на изображениях. Это позволяет специалистам быстро и точно анализировать состояние сосудов пациента и выявлять патологии. Для улучшения результатов сегментации сосудов на изображениях МРТ также важно применение алгоритмов балансировки и аугментации данных. Балансировка данных позволяет избежать смещения в обучающем наборе и обеспечить равномерное представление различных классов сосудов. Аугментация данных позволяет увеличить разнообразие обучающего набора, что способствует повышению обобщающей способности модели и улучшению ее производительности на новых данных. Таким образом, применение сверточных нейронных сетей для сегментации сосудов на изображениях МРТ является эффективным подходом, который позволяет автоматизировать процесс анализа сосудистых структур, обеспечивая высокую точность и надежность результатов. Комбинация CNN с алгоритмами балансировки и аугментации данных позволяет добиться оптимальных результатов и улучшить диагностику различных заболеваний сердечно-сосудистой системы.

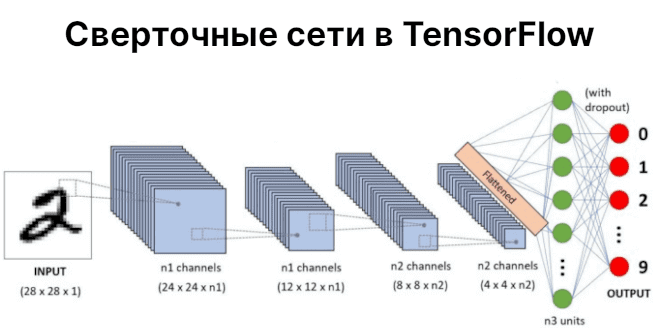
**Алгоритмы балансировки и аугментации данных для улучшения сегментации биомедицинских изображений**

****

Примеры задач, решаемых нейросетями, включая сегментацию

Сегментация биомедицинских изображений является важным этапом анализа медицинских данных, позволяющим выделить интересующие структуры на изображениях и провести дальнейший анализ. Одним из ключевых направлений в области сегментации является применение сверточных нейронных сетей, которые показывают отличные результаты в задачах обработки медицинских изображений. Однако, для достижения высокой точности и надежности сегментации необходимо учитывать особенности данных, включая их дисбаланс и ограниченное количество обучающих примеров. Для улучшения сегментации биомедицинских изображений, в том числе изображений рентгеновской коронарографии с наличием стеноза, широко применяются алгоритмы балансировки и аугментации данных. Алгоритмы балансировки данных позволяют справиться с проблемой дисбаланса классов, которая часто встречается в медицинских наборах данных, где один класс может быть значительно меньше другого. Балансировка данных позволяет улучшить обучение модели и повысить ее способность к обобщению. Одним из распространенных методов балансировки данных является апсемплинг, при котором примеры минорного класса копируются или генерируются с целью уравновесить количество примеров разных классов. Этот подход позволяет улучшить обучение модели и повысить ее способность к выявлению меньшинственного класса, что особенно важно в задачах сегментации биомедицинских изображений. Помимо балансировки данных, аугментация данных также играет важную роль в улучшении сегментации биомедицинских изображений. Аугментация данных позволяет увеличить разнообразие обучающего набора данных путем применения различных трансформаций к изображениям. Это может включать в себя повороты, масштабирование, отражения, добавление шума и другие преобразования, которые способствуют обучению модели на более разнообразных данных. Применение алгоритмов балансировки и аугментации данных в совокупности с использованием сверточных нейронных сетей, таких как U-Net, позволяет добиться высокой точности и надежности сегментации сосудов на изображениях рентгеновской коронарографии с наличием стеноза. Эти методы позволяют улучшить способность модели к обобщению, повысить ее точность и сделать сегментацию более эффективной в клинической практике. Таким образом, алгоритмы балансировки и аугментации данных играют важную роль в улучшении сегментации биомедицинских изображений, включая изображения рентгеновской коронарографии с наличием стеноза. Их применение в сочетании с современными методами глубокого обучения позволяет достичь высоких результатов в задачах анализа медицинских данных и улучшить диагностику сердечно-сосудистых заболеваний.

**Применение сверточных нейронных сетей для детекции стенозов в кардиологии**

****

Примеры использования сверточных нейронных сетей для анализа данных в кардиологии

Сегментация сосудов с наличием стеноза на изображениях рентгеновской коронарографии является важной задачей в кардиологии, поскольку стенозы артерий могут привести к серьезным заболеваниям сердца и сосудов. Для эффективного обнаружения стенозов и оценки их степени часто применяются сверточные нейронные сети, которые показывают высокую точность и надежность в данной области. Одним из наиболее распространенных методов для сегментации сосудов на изображениях рентгеновской коронарографии является использование архитектуры U-Net. Этот метод позволяет эффективно извлекать контуры сосудов, включая участки с наличием стеноза, благодаря своей способности к объединению контекстуальной информации на разных уровнях изображения. Для улучшения результатов сегментации и обучения нейронных сетей на небольших наборах данных часто применяются алгоритмы балансировки и аугментации данных. Балансировка данных позволяет справиться с проблемой дисбаланса классов, что особенно важно при обнаружении стенозов, которые могут быть редкими на изображениях. Аугментация данных позволяет увеличить разнообразие тренировочного набора, что способствует повышению обобщающей способности модели. Применение сверточных нейронных сетей для детекции стенозов в сердечно-сосудистой системе позволяет автоматизировать процесс анализа медицинских изображений, сокращая время, необходимое для постановки диагноза специалистом. Это особенно важно в условиях больших объемов данных, с которыми сталкиваются врачи в современной медицинской практике. Сравнительный анализ методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии позволяет выявить преимущества и недостатки каждого из подходов. Важно учитывать особенности каждого типа изображений при выборе метода сегментации для конкретной задачи, чтобы добиться наилучших результатов. Таким образом, применение сверточных нейронных сетей для детекции стенозов в кардиологии является перспективным направлением, которое позволяет улучшить диагностику сердечно-сосудистых заболеваний и повысить эффективность лечения пациентов.

**Метод U-Net для извлечения сосудов на изображениях рентгеновской коронарографии**

Пример изображения сердца, полученного с помощью рентгеновской коронарографии

Сегментация сосудов на изображениях рентгеновской коронарографии является важным этапом в диагностике сердечно-сосудистых заболеваний. Для достижения высокой точности и надежности в этом процессе применяются различные методы машинного обучения, включая сверточные нейронные сети. Одним из наиболее эффективных методов для сегментации сосудов на медицинских изображениях является метод U-Net. Метод U-Net представляет собой архитектуру глубокой сверточной нейронной сети, специально разработанной для сегментации изображений. Его особенностью является наличие пути кодировщика (encoder) для извлечения признаков из изображения и пути декодировщика (decoder) для восстановления сегментированного изображения. Этот подход позволяет сети эффективно извлекать информацию на разных уровнях детализации, что особенно важно при сегментации сосудов с наличием стеноза. Применение метода U-Net для извлечения сосудов на изображениях рентгеновской коронарографии позволяет добиться высокой точности и четкости сегментации. За счет своей архитектуры сеть способна учитывать как локальные, так и глобальные особенности изображения, что делает ее эффективным инструментом для выделения сложных структур, таких как сосуды с наличием стеноза. Для улучшения результатов сегментации с использованием метода U-Net также применяются алгоритмы балансировки и аугментации данных. Балансировка данных позволяет справиться с проблемой дисбаланса классов, что особенно актуально при наличии стеноза на изображениях. Аугментация данных, в свою очередь, позволяет увеличить разнообразие тренировочного набора, что способствует повышению обобщающей способности модели. Использование сверточных нейронных сетей для сегментации сосудов на изображениях рентгеновской коронарографии имеет большой потенциал в кардиологии. Эти методы позволяют автоматизировать процесс анализа медицинских изображений, сокращая время диагностики и повышая точность выявления патологий сердечно-сосудистой системы. Таким образом, применение метода U-Net для извлечения сосудов на изображениях рентгеновской коронарографии с последующей балансировкой и аугментацией данных является эффективным подходом к сегментации сосудов с наличием стеноза. Эти методы не только улучшают качество сегментации, но и способствуют развитию автоматизированных систем диагностики сердечно-сосудистых заболеваний.

**Применение сверточных нейронных сетей для сегментации сосудов на изображениях биомедицинской диагностики**

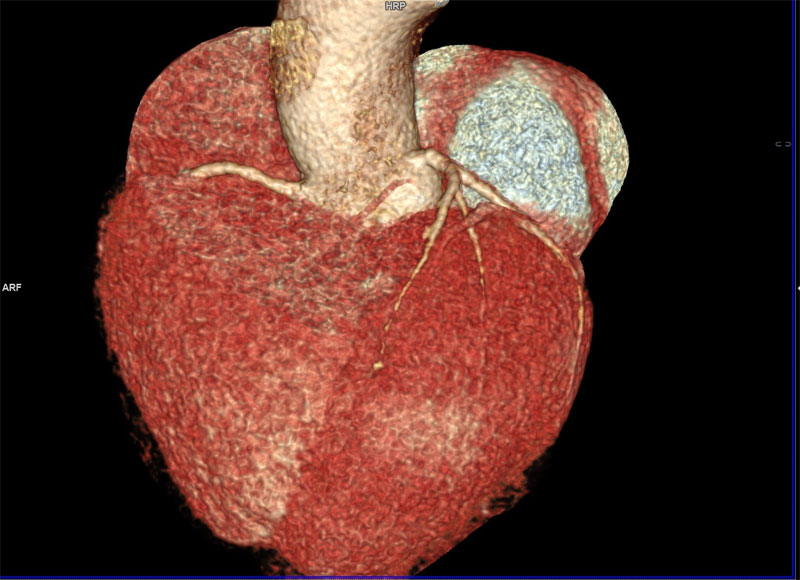
****

Схема архитектуры сверточной нейронной сети для сегментации сосудов на изображениях биомедицинской диагностики

Сегментация сосудов на изображениях рентгеновской коронарографии является важным этапом в диагностике сердечно-сосудистых заболеваний. Одним из методов, который показал высокую эффективность в этой задаче, является применение сверточных нейронных сетей. Сверточные нейронные сети (CNN) позволяют автоматически извлекать признаки из изображений и обучаться на больших объемах данных для точной сегментации сосудов. Одним из наиболее широко используемых архитектур для сегментации изображений является U-Net. Метод U-Net хорошо подходит для задач сегментации из-за своей способности извлекать детальные признаки на разных уровнях изображения и восстанавливать пространственную информацию. Применение U-Net для сегментации сосудов на изображениях рентгеновской коронарографии позволяет добиться высокой точности и надежности результатов. Для улучшения процесса сегментации сосудов на изображениях рентгеновской коронарографии также важно использование алгоритмов балансировки и аугментации данных. Балансировка данных позволяет справиться с проблемой дисбаланса классов, что часто встречается при сегментации сосудов, особенно при наличии стеноза. Аугментация данных позволяет увеличить разнообразие тренировочного набора данных, что способствует лучшей обобщающей способности модели. Применение сверточных нейронных сетей для детекции стенозов в кардиологии также имеет большое значение. Стенозы артерий могут привести к серьезным осложнениям, поэтому их раннее обнаружение крайне важно. CNN позволяют автоматически выявлять признаки стенозов на изображениях и помогают кардиологам в проведении более точной диагностики. Таким образом, применение сверточных нейронных сетей для сегментации сосудов на изображениях рентгеновской коронарографии является эффективным подходом, который позволяет добиться высокой точности и надежности результатов. Комбинация метода U-Net, алгоритмов балансировки и аугментации данных, а также использование CNN для детекции стенозов в кардиологии открывает новые перспективы в области диагностики сердечно-сосудистых заболеваний.

**Анализ эффективности алгоритмов балансировки данных для улучшения сегментации сосудов**

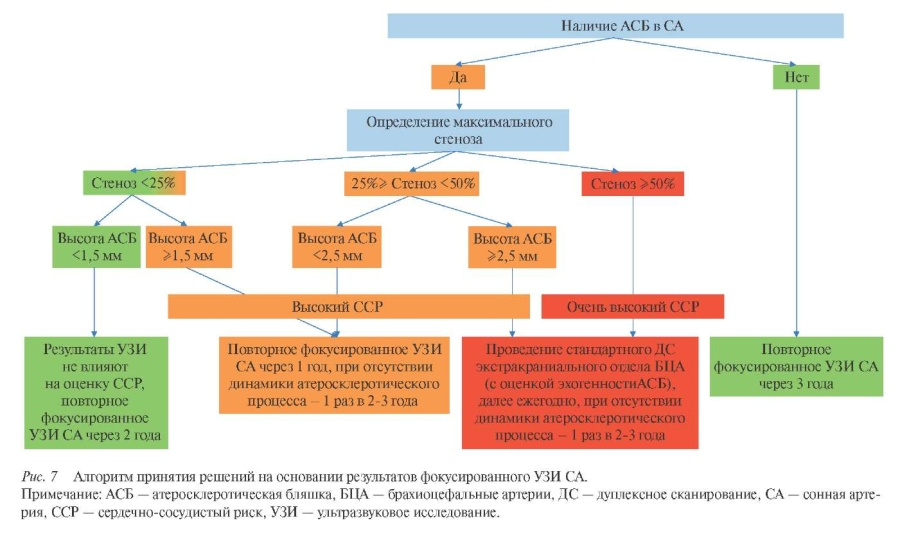
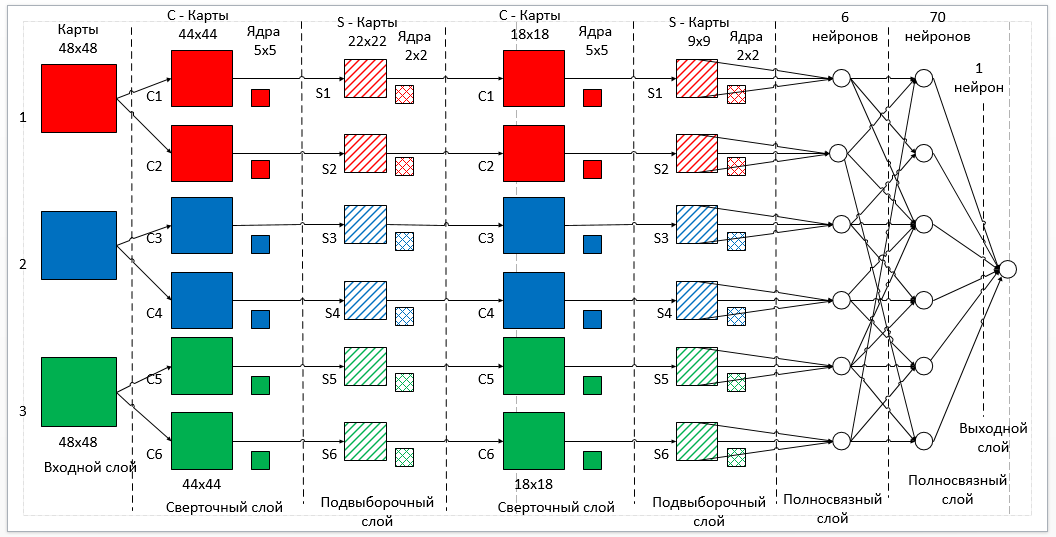
****

Схема анализа эффективности алгоритмов балансировки данных для улучшения сегментации сосудов

Сегментация сосудов на изображениях рентгеновской коронарографии является важным этапом в диагностике сердечно-сосудистых заболеваний. Однако наличие стенозов на сосудах может затруднить процесс сегментации и повлиять на точность результатов. Для улучшения сегментации сосудов с наличием стеноза применяются различные алгоритмы балансировки данных. Один из подходов к балансировке данных - это использование алгоритмов оверсэмплинга и андерсэмплинга. Оверсэмплинг позволяет увеличить количество образцов в классе с меньшим представлением, тогда как андерсэмплинг уменьшает количество образцов в классе с избыточным представлением. Это помогает снизить дисбаланс классов и улучшить обучение модели на данных с наличием стеноза. Другим методом балансировки данных является применение весов классов во время обучения модели. Путем назначения различных весов классам в зависимости от их представленности в данных, модель уделяет большее внимание классам с меньшим представлением, что способствует более эффективной сегментации сосудов с наличием стеноза. Также широко используется аугментация данных для улучшения сегментации сосудов. Аугментация позволяет создать дополнительные вариации изображений путем применения различных трансформаций, таких как повороты, отражения, изменения масштаба и освещения. Это помогает увеличить разнообразие данных для обучения модели и повысить ее устойчивость к различным условиям изображений. Эффективность алгоритмов балансировки данных для улучшения сегментации сосудов с наличием стеноза может быть оценена путем сравнения результатов с использованием и без использования этих методов. Исследования показывают, что правильный выбор алгоритмов балансировки данных может значительно повысить точность и надежность сегментации сосудов на изображениях рентгеновской коронарографии при наличии стеноза. Таким образом, анализ эффективности алгоритмов балансировки данных для улучшения сегментации сосудов является важным шагом в развитии методов диагностики сердечно-сосудистых заболеваний с использованием сверточных нейронных сетей.

**Применение сверточных нейронных сетей для детекции патологий сердечно-сосудистой системы**

****

Примеры архитектуры сверточных нейронных сетей для детекции патологий сердечно-сосудистой системы

Сердечно-сосудистые заболевания остаются одной из основных причин смертности по всему миру. Для точной диагностики и лечения таких заболеваний необходимо использовать передовые методы обработки медицинских изображений. В последние годы сверточные нейронные сети стали широко применяться в области медицинского образования, включая кардиологию, для автоматической обработки и анализа изображений сердечно-сосудистой системы. Одним из ключевых направлений применения сверточных нейронных сетей в кардиологии является детекция стенозов в сосудах. Стенозы, или сужения сосудов, могут привести к серьезным осложнениям, таким как инфаркт миокарда или инсульт. Поэтому раннее обнаружение и точная локализация стенозов имеют важное значение для предотвращения сердечно-сосудистых заболеваний. Для детекции стенозов на изображениях рентгеновской коронарографии эффективно применять методы глубокого обучения, включая сверточные нейронные сети. Эти сети способны автоматически извлекать признаки из изображений и классифицировать их, что делает процесс детекции стенозов более точным и эффективным. Одним из наиболее популярных архитектур для сегментации сосудов на изображениях рентгеновской коронарографии является U-Net. Эта архитектура позволяет эффективно извлекать даже сложные структуры сосудов, включая области с наличием стенозов. Применение U-Net в сочетании с алгоритмами балансировки и аугментации данных позволяет улучшить качество сегментации и повысить точность обнаружения стенозов. Важным аспектом использования сверточных нейронных сетей для детекции стенозов является анализ эффективности алгоритмов балансировки данных. Балансировка данных позволяет уравновесить классы объектов на изображениях, что особенно важно при работе с несбалансированными наборами данных. Это помогает избежать смещения в обучении сети и повысить ее обобщающую способность. Сравнительный анализ методов сегментации сосудов на изображениях рентгеновской коронарографии и других типов медицинских изображений позволяет выявить особенности и преимущества каждого метода. Точная сегментация сосудов с наличием стенозов на рентгеновских изображениях играет важную роль в диагностике сердечно-сосудистых заболеваний и планировании лечения. Таким образом, применение сверточных нейронных сетей для детекции стенозов в кардиологии на изображениях рентгеновской коронарографии является перспективным направлением развития медицинской диагностики. Сочетание методов сегментации, аугментации данных и алгоритмов балансировки позволяет повысить точность и надежность обнаружения стенозов, что способствует раннему выявлению сердечно-сосудистых заболеваний и улучшению результатов лечения пациентов.

**Сравнительный анализ методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии**

Сравнительный анализ методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии Сегментация сосудов на медицинских изображениях играет важную роль в диагностике и лечении сердечно-сосудистых заболеваний. В данной главе будет проведен сравнительный анализ методов сегментации сосудов на изображениях, полученных с помощью магнитно-резонансной томографии (МРТ) и рентгеновской коронарографии. Обе техники предоставляют ценные данные для визуализации сосудистой системы, но имеют свои особенности, которые могут повлиять на эффективность сегментации. Изображения МРТ обладают высоким разрешением и контрастностью, что позволяет четко видеть структуру сосудов и окружающих тканей. Однако, из-за сложной природы данных МРТ, такие изображения могут содержать артефакты и шум, что затрудняет процесс сегментации. Для сегментации сосудов на изображениях МРТ часто применяются методы, основанные на сверточных нейронных сетях, так как они способны автоматически извлекать признаки из сложных данных и обеспечивать высокую точность сегментации. С другой стороны, изображения рентгеновской коронарографии обладают более низким разрешением, но обычно имеют лучшую контрастность сосудов благодаря введению контрастного вещества. Это делает сегментацию сосудов на изображениях рентгеновской коронарографии менее сложной с точки зрения визуализации, но требует более тщательной обработки из-за менее четкого различения границ сосудов. Для сравнительного анализа методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии будут рассмотрены различные подходы, включая классические методы обработки изображений, а также современные методы, основанные на глубоком обучении. Будут проанализированы преимущества и недостатки каждого метода, их применимость к конкретным типам изображений, а также результаты сегментации на реальных клинических данных. Целью данного сравнительного анализа является выявление наиболее эффективных методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии с учетом особенностей каждой техники. Полученные результаты позволят определить наиболее подходящий метод для конкретных клинических задач, связанных с диагностикой и лечением сердечно-сосудистых заболеваний.

**Применение сверточных нейронных сетей для сегментации сосудов на изображениях с различными типами стенозов**

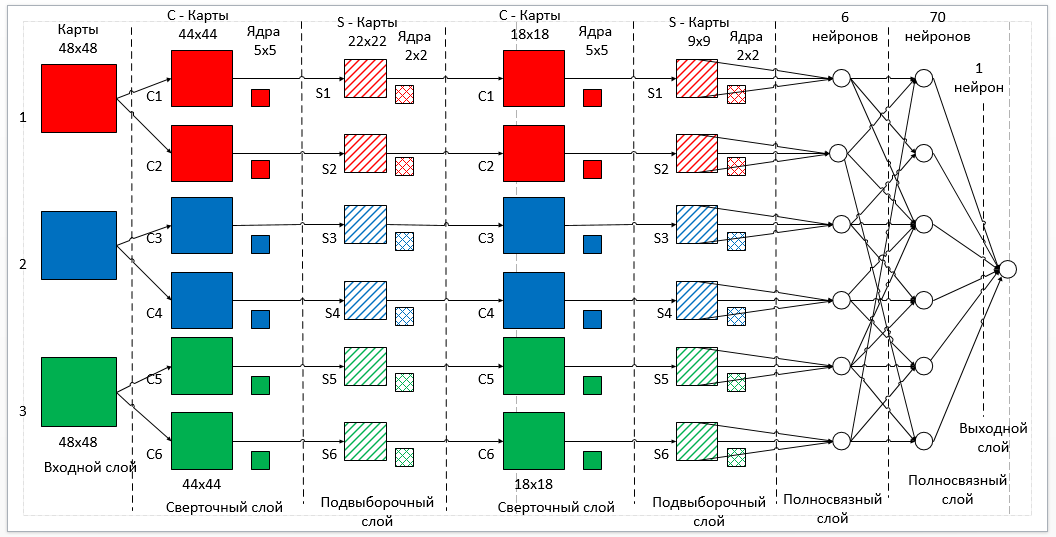
****

Схема сверточной нейронной сети для сегментации сосудов

Сегментация сосудов с наличием стеноза на изображениях рентгеновской коронарографии является важной задачей в области медицинского образования. Для достижения точности и надежности в этом процессе применяются различные методы, включая использование сверточных нейронных сетей. Одним из ключевых методов, который широко применяется для сегментации сосудов на изображениях, является метод U-Net. U-Net представляет собой архитектуру нейронной сети, специально разработанную для сегментации биомедицинских изображений. Его особенностью является наличие пути кодирования для извлечения признаков и пути декодирования для точной сегментации объектов на изображениях. Применение метода U-Net позволяет добиться высокой точности и качества сегментации сосудов даже при наличии стеноза. Для улучшения результатов сегментации сосудов на изображениях рентгеновской коронарографии также важно использование алгоритмов балансировки и аугментации данных. Балансировка данных позволяет справиться с проблемой дисбаланса классов, что часто встречается при работе с медицинскими изображениями. Аугментация данных, в свою очередь, позволяет увеличить разнообразие тренировочного набора данных путем применения различных трансформаций, таких как повороты, отражения, изменения масштаба и другие. Эти методы способствуют улучшению обобщающей способности модели и повышению ее точности. Сверточные нейронные сети также нашли широкое применение в кардиологии для детекции стенозов в сердечно-сосудистой системе. Они позволяют автоматизировать процесс анализа медицинских изображений, делая его более эффективным и точным. Благодаря своей способности извлекать сложные признаки из изображений, сверточные нейронные сети стали незаменимым инструментом для диагностики сердечно-сосудистых заболеваний. Таким образом, применение сверточных нейронных сетей для сегментации сосудов с наличием стеноза на изображениях рентгеновской коронарографии является перспективным направлением в медицинской диагностике. Сочетание метода U-Net, алгоритмов балансировки и аугментации данных, а также использование сверточных нейронных сетей в кардиологии позволяет повысить точность и эффективность сегментации сосудов, что в свою очередь способствует раннему выявлению и лечению сердечно-сосудистых заболеваний.

**Роль сверточных нейронных сетей в улучшении диагностики сердечно-сосудистых заболеваний**

****

Примеры использования сверточных нейронных сетей в диагностике сердечно-сосудистых заболеваний

Сердечно-сосудистые заболевания остаются одной из основных причин смертности по всему миру. Для точной диагностики и эффективного лечения таких заболеваний необходимо использовать передовые методы обработки медицинских изображений. В последние годы сверточные нейронные сети стали широко применяться в области медицинской диагностики, включая анализ изображений сердечно-сосудистой системы. Одним из ключевых направлений использования сверточных нейронных сетей в кардиологии является детекция стенозов в сосудах. Стенозы, или сужения сосудов, могут быть признаком серьезных заболеваний, таких как ишемическая болезнь сердца. Для выявления стенозов на изображениях рентгеновской коронарографии применяются различные методы обработки изображений, включая сверточные нейронные сети. Метод U-Net, представляющий собой архитектуру глубокой сверточной нейронной сети, широко используется для сегментации сосудов на изображениях рентгеновской коронарографии. Этот метод позволяет извлекать сосуды с высокой точностью и надежностью, что является критически важным для правильной диагностики стенозов и других патологий сердечно-сосудистой системы. Для улучшения результатов сегментации сосудов на изображениях рентгеновской коронарографии применяются алгоритмы балансировки и аугментации данных. Балансировка данных позволяет справиться с проблемой дисбаланса классов, что особенно важно при обучении нейронных сетей на медицинских данных, где классы (например, сосуды и стенозы) могут быть неравномерно представлены. Аугментация данных позволяет увеличить разнообразие тренировочного набора данных, что способствует повышению обобщающей способности модели. Использование сверточных нейронных сетей для сегментации сосудов с наличием стеноза на изображениях рентгеновской коронарографии открывает новые перспективы в диагностике сердечно-сосудистых заболеваний. Точные и быстрые методы анализа медицинских изображений позволяют врачам рано выявлять патологии и принимать эффективные меры по лечению пациентов. Современные технологии машинного обучения и обработки изображений играют важную роль в улучшении здравоохранения и повышении качества жизни людей.

# Стек технологий

**Python**

Python — это высокоуровневый язык программирования, который стал незаменимым инструментом для разработки и обучения нейронных сетей. Благодаря своим многочисленным библиотекам и простоте использования, Python является идеальным выбором для работы в области машинного обучения и глубокого обучения.

### Основные преимущества использования Python для работы с нейронными сетями

1. **Широкий выбор библиотек и фреймворков**: Python предлагает множество мощных библиотек, специально разработанных для работы с нейронными сетями. Среди них наиболее популярны:
   * **TensorFlow**: Один из самых известных фреймворков для глубокого обучения, разработанный Google. TensorFlow предоставляет широкий набор инструментов для создания и обучения нейронных сетей различной сложности.
   * **Keras**: Высокоуровневый интерфейс для TensorFlow, который упрощает процесс создания и тренировки нейронных сетей. Keras позволяет быстро прототипировать модели и экспериментировать с различными архитектурами.
   * **PyTorch**: Фреймворк, разработанный Facebook, который получил широкое признание за свою гибкость и интуитивно понятный интерфейс. PyTorch особенно популярен в научных исследованиях и разработке прототипов.
   * **scikit-learn**: Библиотека для машинного обучения, которая включает в себя множество инструментов для предварительной обработки данных, обучения моделей и их оценки. Хотя scikit-learn не предназначен специально для глубокого обучения, он отлично подходит для работы с более простыми моделями и задачами машинного обучения.
2. **Простота и читаемость кода**: Python известен своей простотой и читаемостью, что делает его идеальным для обучения и разработки нейронных сетей. Понятный синтаксис позволяет быстро писать и понимать код, что особенно важно в командах, где разработчики должны легко читать и модифицировать код друг друга.
3. **Интерактивная среда разработки**: Python поддерживает работу в интерактивных средах разработки, таких как Jupyter Notebook. Это позволяет визуализировать данные, интерактивно исследовать результаты и быстро прототипировать модели. Такая среда особенно полезна для исследований и обучения.
4. **Большое сообщество и обилие ресурсов**: Python имеет одно из самых больших и активных сообществ разработчиков. Это означает, что существует множество доступных ресурсов, таких как учебники, документация, форумы и сообщества, где можно найти помощь и поддержку. Кроме того, большое количество готовых примеров и открытых проектов позволяет быстро начать работу с нейронными сетями.

### Применение Python в проекте по работе с нейронными сетями

В рамках данного проекта Python использовался для разработки, обучения и оценки нейронных сетей с целью сегментации сосудов на рентгеновских изображениях коронарографии при наличии стенозов. Вот основные этапы, на которых Python играл ключевую роль:

1. **Предобработка данных**: С помощью библиотек, таких как NumPy и pandas, данные были загружены, очищены и подготовлены для последующего обучения. Это включало нормализацию изображений, аугментацию данных и разделение на тренировочные и тестовые наборы.
2. **Создание архитектуры нейронной сети**: С использованием фреймворков Keras и TensorFlow была разработана архитектура нейронной сети U-Net, оптимизированная для задачи сегментации. Python позволил легко определить слои сети, настроить параметры и экспериментировать с различными конфигурациями.
3. **Обучение модели**: Процесс обучения модели был осуществлен с помощью TensorFlow и Keras. Python позволил эффективно управлять обучением, включая настройку гиперпараметров, мониторинг процесса обучения и использование методов ранней остановки для предотвращения переобучения.
4. **Оценка и визуализация результатов**: После обучения модели Python использовался для оценки её производительности на тестовом наборе данных. Библиотеки matplotlib и seaborn позволили визуализировать результаты сегментации, построить графики точности и потерь, а также провести анализ ошибок.
5. **Интеграция и автоматизация**: Python скрипты были использованы для автоматизации всего процесса, от предобработки данных до финальной оценки модели. Это обеспечило воспроизводимость экспериментов и удобство в работе с большим объемом данных.

**Keras и TensorFlow**

Keras и TensorFlow — это два мощных инструмента, которые часто используются в тандеме для создания и обучения моделей глубокого обучения. Они предоставляют удобные и эффективные средства для разработки нейронных сетей, обеспечивая гибкость и масштабируемость.

### Keras

Keras — это высокоуровневый интерфейс для нейронных сетей, который был разработан для того, чтобы сделать работу с глубоким обучением доступной и простой. Keras позволяет быстро и легко создавать сложные модели, абстрагируя низкоуровневые детали и предоставляя интуитивно понятный синтаксис.

#### Основные характеристики Keras:

1. **Простота и удобство использования**: Keras изначально был разработан с целью упрощения работы с глубоким обучением. Его лаконичный и понятный синтаксис позволяет быстро и легко создавать модели нейронных сетей, что особенно полезно для новичков и при прототипировании.
2. **Модульность**: Keras состоит из независимых модулей, которые можно комбинировать по мере необходимости. Это позволяет гибко настраивать архитектуры моделей, добавлять новые слои и компоненты, что облегчает экспериментирование и разработку.
3. **Поддержка нескольких бэкэндов**: Изначально Keras поддерживал различные бэкэнды для выполнения вычислений, включая TensorFlow, Theano и Microsoft Cognitive Toolkit (CNTK). Однако, с версии 2.3, Keras интегрирован непосредственно в TensorFlow, что делает его использование с этим фреймворком наиболее оптимальным.
4. **Обширная документация и примеры**: Keras имеет отличную документацию, включающую многочисленные примеры и учебные пособия, что делает его доступным и понятным для широкого круга пользователей.

### 

### TensorFlow

TensorFlow — это фреймворк с открытым исходным кодом, разработанный компанией Google, который используется для вычислений с интенсивным использованием данных. TensorFlow особенно известен своими возможностями для глубокого обучения и машинного обучения.

#### Основные характеристики TensorFlow:

1. **Гибкость и масштабируемость**: TensorFlow позволяет разрабатывать и обучать модели любой сложности, от небольших нейронных сетей до огромных распределенных систем. Он поддерживает обучение на различных устройствах, включая CPU, GPU и TPU, что позволяет масштабировать обучение моделей.
2. **Обширный функционал**: TensorFlow предоставляет широкий спектр инструментов и библиотек для разработки моделей, выполнения численных вычислений, работы с графиками вычислений, автоматического дифференцирования и многого другого. Это делает его универсальным инструментом для разработки и внедрения моделей машинного обучения.
3. **TensorFlow Extended (TFX)**: Набор инструментов для создания производственных рабочих процессов машинного обучения. TFX включает в себя компоненты для загрузки и анализа данных, обучения моделей, оценки и развертывания.
4. **TensorFlow Lite и TensorFlow.js**: TensorFlow Lite предназначен для выполнения моделей машинного обучения на мобильных устройствах и устройствах с малым энергопотреблением. TensorFlow.js позволяет запускать модели машинного обучения непосредственно в браузере или на сервере с использованием JavaScript.

### Keras и TensorFlow в тандеме

Сочетание Keras и TensorFlow обеспечивает мощный и удобный инструмент для разработки и обучения моделей глубокого обучения. Keras, интегрированный в TensorFlow, предоставляет высокоуровневый интерфейс, который упрощает создание моделей, в то время как TensorFlow обеспечивает мощную и гибкую инфраструктуру для выполнения вычислений.

#### Применение Keras и TensorFlow в проекте

В данном проекте Keras и TensorFlow использовались для создания, обучения и оценки моделей нейронных сетей, предназначенных для сегментации сосудов на рентгеновских изображениях коронарографии при наличии стенозов. Вот основные этапы работы:

1. **Создание модели**: С использованием Keras была создана архитектура модели U-Net, которая является одной из наиболее эффективных для задач сегментации. Keras позволил легко определить слои сети, включая сверточные слои, слои подвыборки и декодирующие слои.
2. **Настройка гиперпараметров**: С помощью Keras были настроены гиперпараметры модели, такие как размер мини-пакета, количество эпох, функции активации и оптимизаторы. Это позволило провести многочисленные эксперименты и выбрать оптимальные параметры для обучения модели.
3. **Обучение модели**: TensorFlow обеспечил масштабируемую и эффективную инфраструктуру для обучения модели на больших наборах данных. Использование GPU и TPU позволило значительно ускорить процесс обучения и обработки данных.
4. **Аугментация и предобработка данных**: С использованием библиотек Keras и TensorFlow были применены методы аугментации данных, такие как повороты, отражения и изменения масштаба. Это помогло улучшить обобщающую способность модели и предотвратить переобучение.
5. **Оценка и визуализация результатов**: После обучения модели TensorFlow и Keras были использованы для оценки производительности модели на тестовом наборе данных. Были построены графики точности, потерь, а также визуализированы результаты сегментации сосудов, что позволило провести детальный анализ качества модели.

**OpenCV**

OpenCV (Open Source Computer Vision Library) — это открытая библиотека компьютерного зрения и машинного обучения, разработанная Intel. Она была создана для предоставления разработчикам инструментов для создания приложений в области обработки изображений и анализа видео. OpenCV поддерживает множество языков программирования, включая Python, C++, Java и другие, что делает её универсальным инструментом для различных проектов.

### Основные характеристики OpenCV

1. **Широкий спектр функций**: OpenCV предоставляет более 2500 оптимизированных алгоритмов, которые охватывают множество задач компьютерного зрения, включая распознавание лиц, идентификацию объектов, классификацию объектов, трекинг движущихся объектов, восстановление трехмерных моделей и многое другое.
2. **Кроссплатформенность**: OpenCV поддерживает различные операционные системы, включая Windows, Linux, macOS, Android и iOS. Это позволяет разработчикам создавать кроссплатформенные приложения, используя одни и те же функции и алгоритмы.
3. **Интеграция с другими библиотеками**: OpenCV легко интегрируется с другими библиотеками и фреймворками, такими как TensorFlow, Keras, PyTorch и scikit-learn, что позволяет комбинировать алгоритмы компьютерного зрения с методами машинного обучения и глубокого обучения.
4. **Поддержка аппаратного ускорения**: OpenCV поддерживает аппаратное ускорение через использование GPU и других специальных процессоров. Это позволяет значительно ускорить выполнение вычислительно интенсивных задач, что особенно полезно для обработки больших объемов данных в реальном времени.

### Применение OpenCV в проекте

В данном проекте OpenCV использовался для предобработки данных и выполнения задач компьютерного зрения, связанных с сегментацией сосудов на рентгеновских изображениях коронарографии. Вот основные этапы, на которых OpenCV играл ключевую роль:

1. **Загрузка и обработка изображений**: С помощью OpenCV изображения рентгеновской коронарографии были загружены и предварительно обработаны. Это включало изменение размера изображений, конвертацию их в серые тона, нормализацию и применение различных фильтров для улучшения качества изображений и удаления шумов.
2. **Фильтрация и улучшение изображений**: OpenCV предоставляет множество фильтров и методов для улучшения изображений, таких как фильтр Гаусса, медианный фильтр, адаптивная бинаризация и другие. Эти методы были использованы для повышения четкости изображений и улучшения контрастности, что является важным этапом перед сегментацией сосудов.
3. **Сегментация и контурный анализ**: OpenCV включает мощные функции для сегментации изображений, такие как методы пороговой обработки, методы активных контуров и водораздел. Эти функции использовались для выделения сосудов на изображениях и определения их контуров, что является ключевым шагом в диагностике наличия стенозов.
4. **Аугментация данных**: Для улучшения обобщающей способности моделей глубокого обучения OpenCV применялся для аугментации данных. Аугментация включала в себя различные трансформации изображений, такие как повороты, отражения, изменения масштаба и сдвиги. Это позволило создать более разнообразный набор данных для обучения нейронных сетей.
5. **Интеграция с моделями глубокого обучения**: OpenCV был использован для интеграции с TensorFlow и Keras, что позволило комбинировать методы компьютерного зрения с моделями глубокого обучения. Например, после предварительной обработки изображений с помощью OpenCV данные передавались в модели, построенные с использованием Keras и TensorFlow, для дальнейшего анализа и сегментации.

**NumPy**

NumPy — это фундаментальная библиотека для научных вычислений на языке Python. Она предоставляет поддержку больших многомерных массивов и матриц, а также богатую библиотеку высокоуровневых математических функций для работы с этими массивами. NumPy является основой для многих других библиотек и фреймворков, используемых в машинном обучении, анализе данных и научных исследованиях.

### Основные характеристики NumPy

1. **Массивы (ndarray)**: Основной объект NumPy — это многомерный массив ndarray, который позволяет эффективно хранить и обрабатывать большие объемы данных. Массивы NumPy являются гомогенными, то есть содержат элементы одного типа данных, что обеспечивает высокую производительность при выполнении операций.
2. **Эффективность и производительность**: NumPy написан на C и Fortran, что позволяет выполнять вычисления значительно быстрее по сравнению со стандартными Python-структурами данных. Многие операции в NumPy векторизованы, что позволяет обрабатывать данные без использования явных циклов.
3. **Богатый набор функций**: NumPy включает в себя множество функций для выполнения математических операций, таких как линейная алгебра, статистика, преобразования Фурье, генерация случайных чисел и многое другое. Это делает NumPy универсальным инструментом для научных и инженерных задач.
4. **Интеграция с другими библиотеками**: NumPy легко интегрируется с другими библиотеками для научных вычислений и машинного обучения, такими как SciPy, pandas, Matplotlib, TensorFlow и PyTorch. Это позволяет использовать NumPy в качестве основы для более сложных вычислений и визуализаций.
5. **Масштабируемость**: NumPy поддерживает работу с массивами любых размеров и форм. Это позволяет работать с большими данными и выполнять масштабируемые вычисления, что особенно важно для задач машинного обучения и анализа данных.

### Применение NumPy в проекте

В данном проекте NumPy использовался для выполнения различных этапов предобработки данных и численных вычислений, необходимых для создания и обучения нейронных сетей для сегментации сосудов на рентгеновских изображениях коронарографии. Вот основные этапы, на которых NumPy играл ключевую роль:

1. **Загрузка и хранение данных**: NumPy предоставил удобные инструменты для загрузки данных из различных источников (например, файлов CSV, изображений) и их хранения в виде массивов ndarray. Это позволило легко управлять большими объемами данных и выполнять над ними различные операции.
2. **Предобработка данных**: С помощью NumPy были выполнены операции по нормализации и стандартизации данных. Например, пиксельные значения изображений были масштабированы в диапазон от 0 до 1, что улучшило производительность моделей машинного обучения. Также были выполнены операции по удалению пропущенных значений и фильтрации данных.
3. **Анализ данных**: NumPy предоставил функции для статистического анализа данных, такие как вычисление среднего значения, медианы, стандартного отклонения и других статистик. Это позволило лучше понять распределение данных и выявить потенциальные проблемы перед обучением моделей.
4. **Манипуляции с массивами**: NumPy позволил выполнять сложные операции с массивами, такие как изменение формы массивов, объединение и разбиение массивов, выбор подмножеств данных, а также выполнение арифметических операций над массивами. Это значительно упростило обработку данных и подготовку их к обучению.
5. **Векторизация вычислений**: Одной из ключевых преимуществ NumPy является возможность выполнения векторизованных операций, что значительно ускорило вычисления по сравнению с использованием обычных циклов в Python. Это было особенно полезно при выполнении математических операций над большими массивами данных.
6. **Интеграция с TensorFlow и Keras**: NumPy массивы использовались для передачи данных в модели глубокого обучения, разработанные с использованием TensorFlow и Keras. Это обеспечило совместимость и удобство работы с данными на всех этапах разработки моделей.

**Matplotlib**

Matplotlib — это одна из самых популярных библиотек для создания визуализаций на языке Python. Она предоставляет средства для создания статических, анимационных и интерактивных графиков, которые могут быть использованы для анализа данных, отчётов и исследований. Matplotlib особенно полезен в научных и инженерных областях, а также в сфере машинного обучения и искусственного интеллекта.

### Основные характеристики Matplotlib

1. **Гибкость и настройка**: Matplotlib предлагает высокий уровень настройки визуализаций. Практически все элементы графика, включая шрифты, цвета, метки, линии и оси, могут быть настроены под конкретные нужды. Это позволяет создавать профессионально выглядящие графики, соответствующие требованиям конкретных проектов.
2. **Поддержка различных типов графиков**: Matplotlib поддерживает множество типов графиков, включая линейные графики, гистограммы, графики разброса (scatter plots), бар-чарты, круговые диаграммы (pie charts), трёхмерные графики и многие другие. Это делает библиотеку универсальной для различных задач визуализации данных.
3. **Интеграция с другими библиотеками**: Matplotlib легко интегрируется с другими библиотеками для научных вычислений и машинного обучения, такими как NumPy, pandas, SciPy и scikit-learn. Это позволяет напрямую использовать данные из этих библиотек для создания визуализаций.
4. **Поддержка интерактивных графиков**: Matplotlib может быть использован вместе с библиотекой IPython для создания интерактивных графиков в Jupyter Notebook. Это особенно полезно для исследований и анализа данных в реальном времени, когда требуется быстрое обновление графиков и взаимодействие с данными.
5. **Широкая документированность и сообщество**: Matplotlib имеет обширную документацию, включающую множество примеров и учебных пособий, что облегчает его изучение и использование. Большое и активное сообщество пользователей также способствует быстрому решению возникающих вопросов и обмену опытом.

### Применение Matplotlib в проекте

В данном проекте Matplotlib использовался для визуализации данных, полученных в результате работы моделей глубокого обучения, предназначенных для сегментации сосудов на рентгеновских изображениях коронарографии. Вот основные этапы, на которых Matplotlib играл ключевую роль:

1. **Визуализация исходных данных**: Matplotlib был использован для отображения исходных рентгеновских изображений коронарографии. Это помогло лучше понять структуру данных и выявить возможные проблемы или особенности изображений до начала предобработки и обучения моделей.
2. **Визуализация результатов предобработки**: После применения различных методов предобработки данных, таких как нормализация и фильтрация, Matplotlib позволил визуализировать результаты этих операций. Это обеспечило наглядное представление о том, как изменились данные в результате предобработки, и помогло оценить качество этих изменений.
3. **Анализ метрик производительности моделей**: Matplotlib был использован для построения графиков, показывающих метрики производительности моделей, такие как точность (accuracy), потери (loss), чувствительность (recall) и специфичность (specificity) на различных этапах обучения. Эти графики позволили оценить, как изменялась производительность моделей в процессе обучения и выбрать оптимальные параметры для дальнейшего использования.
4. **Сравнение результатов сегментации**: Matplotlib предоставил средства для сравнения исходных изображений с результатами сегментации, полученными с помощью моделей глубокого обучения. Это включало наложение контуров сегментированных сосудов на исходные изображения, что позволило визуально оценить точность и качество сегментации.
5. **Создание отчетов и презентаций**: Визуализации, созданные с помощью Matplotlib, использовались для подготовки отчетов и презентаций, демонстрирующих результаты исследования и анализа данных. Высокое качество и настраиваемость графиков обеспечили профессиональное представление информации.

# Код

Создание масок:

import os

import cv2

import json

import numpy as np

from collections import defaultdict

# Путь к аннотациям COCO

coco\_path = "/content/drive/MyDrive/Data/arcade/syntax/train/annotations/train.json"

# Папка с изображениями

images\_folder = "/content/drive/MyDrive/Data/arcade/syntax/train/images"

# Папка для сохранения масок

masks\_folder = "/content/drive/MyDrive/Data/arcade/syntax/train/masks"

# Создание папки для масок, если она не существует

if not os.path.exists(masks\_folder):

os.makedirs(masks\_folder)

# Загрузка аннотаций COCO

with open(coco\_path, encoding="utf-8") as file:

gt = json.load(file)

im\_anns\_gt = defaultdict(list)

for ann in gt["annotations"]:

im\_anns\_gt[ann["image\_id"]].append(ann)

# Обработка и сохранение масок по одному изображению за раз

for img\_info in gt["images"]:

img\_id = img\_info["id"]

img\_filename = img\_info["file\_name"]

# Создание пустой маски для текущего изображения

mask = np.zeros((512, 512), np.int32)

if img\_id in im\_anns\_gt:

for ann in im\_anns\_gt[img\_id]:

points = np.array([ann["segmentation"][0][::2], ann["segmentation"][0][1::2]], np.int32).T

points = points.reshape((-1, 1, 2))

tmp = np.zeros((512, 512), np.int32)

cv2.fillPoly(tmp, [points], (1))

mask += tmp

mask[mask > 0] = 1

# Сохранение маски

mask\_path = os.path.join(masks\_folder, img\_filename)

cv2.imwrite(mask\_path, (mask \* 255).astype(np.uint8))

print("Маски успешно сохранены.")

Настройка модели и ее обучение:

import numpy as np

import tensorflow as tf

import argparse

import os

from glob import glob

import numpy as np

import tensorflow as tf

from keras.utils import Sequence

from albumentations import ShiftScaleRotate, Flip, GridDistortion

import cv2

import os

class MyGenerator(Sequence):

def \_\_init\_\_(self, imgs, msks, img\_path, msk\_path, weights, batch\_size=2, to\_fit=True, train=True, shuffle=True):

self.batch\_size = batch\_size

self.idxs = np.arange(len(imgs))

self.to\_fit = to\_fit

self.train = train

self.shuffle = shuffle

self.imgs = imgs

self.msks = msks

self.weights = weights

self.img\_path = img\_path

self.msk\_path = msk\_path

def \_\_len\_\_(self):

return int(np.ceil(len(self.imgs) / self.batch\_size))

def on\_epoch\_end(self):

if self.shuffle:

np.random.shuffle(self.idxs)

def load\_Xy(self, batch):

X = []

y = []

w = []

for i in batch:

img\_path = os.path.join(self.img\_path, self.imgs[i])

msk\_path = os.path.join(self.msk\_path, self.msks[i])

if not os.path.exists(img\_path):

print(f"Image file not found: {img\_path}")

continue

if not os.path.exists(msk\_path):

print(f"Mask file not found: {msk\_path}")

continue

img = tf.keras.preprocessing.image.img\_to\_array(

tf.keras.preprocessing.image.load\_img(img\_path, color\_mode='grayscale', target\_size=(512, 512))

)

msk = tf.keras.preprocessing.image.img\_to\_array(

tf.keras.preprocessing.image.load\_img(msk\_path, color\_mode='grayscale', target\_size=(512, 512))

)

# print(f"Original image shape: {img.shape}")

if self.train:

aug = Flip(p=0.5)

transform = aug(image=img, mask=msk)

aug = ShiftScaleRotate(always\_apply=True, border\_mode=2)

transform = aug(image=transform['image'], mask=transform['mask'])

aug = GridDistortion(p=0.5)

transform = aug(image=transform['image'], mask=transform['mask'])

img = transform['image']

msk = transform['mask']

img = img / 255

msk = (msk > 0) \* 1

we = np.abs(np.abs(img - 1) - msk)

y.append(msk)

X.append(img)

w.append(we)

X = np.array(X)

w = np.array(w)

y = np.array(y)

return X, y, w

def \_\_getitem\_\_(self, idx):

if idx == self.\_\_len\_\_() - 1:

batch = self.idxs[idx \* self.batch\_size:]

else:

batch = self.idxs[idx \* self.batch\_size:(idx + 1) \* self.batch\_size]

if self.to\_fit:

return self.load\_Xy(batch)

from tensorflow import keras

from keras.models import Model

from keras.layers import Input, Conv2D, Conv2DTranspose, Add, BatchNormalization

from keras.layers import Concatenate as concatenate

import tensorflow as tf

def resblock(input\_):

input\_ = BatchNormalization()(input\_)

conv\_1 = Conv2D(input\_.shape[3], (3, 3), activation='relu', padding='same')(input\_)

conv\_2 = Conv2D(input\_.shape[3], (3, 3), activation='relu', padding='same')(conv\_1)

batch\_1 = BatchNormalization()(conv\_2)

add = Add()([input\_, batch\_1])

conv\_1 = Conv2D(input\_.shape[3], (3, 3), activation='relu', padding='same')(add)

conv\_2 = Conv2D(input\_.shape[3], (3, 3), activation='relu', padding='same')(conv\_1)

batch\_1 = BatchNormalization()(conv\_2)

add = Add()([input\_, batch\_1])

conv\_1 = Conv2D(input\_.shape[3], (3, 3), activation='relu', padding='same')(add)

conv\_2 = Conv2D(input\_.shape[3], (3, 3), activation='relu', padding='same')(conv\_1)

batch\_1 = BatchNormalization()(conv\_2)

add = Add()([input\_, batch\_1])

batch = BatchNormalization()(add)

return batch

def model():

input\_1 = Input(shape=(512, 512, 1))

conv = Conv2D(16, (3, 3), padding='same')(input\_1)

res\_1 = resblock(conv)

conv\_1 = Conv2D(32, (2, 2), strides=2, padding='same')(res\_1)

res\_2 = resblock(conv\_1)

conv\_2 = Conv2D(64, (2, 2), strides=2, padding='same')(res\_2)

res\_3 = resblock(conv\_2)

conv\_3 = Conv2D(128, (2, 2), strides=2, padding='same')(res\_3)

res\_4 = resblock(conv\_3)

conv\_4 = Conv2D(256, (2, 2), strides=2, padding='same')(res\_4)

res\_7 = resblock(conv\_4)

res\_8 = resblock(res\_7)

res\_9 = resblock(res\_8)

trconv\_3 = Conv2DTranspose(128, (2, 2), strides=2, padding='same')(res\_9)

res\_12 = resblock(trconv\_3)

add\_3 = Add()([res\_12, res\_4])

trconv\_4 = Conv2DTranspose(64, (2, 2), strides=2, padding='same')(add\_3)

res\_13 = resblock(trconv\_4)

add\_4 = Add()([res\_13, res\_3])

trconv\_5 = Conv2DTranspose(32, (2, 2), strides=2, padding='same')(add\_4)

res\_14 = resblock(trconv\_5)

add\_5 = Add()([res\_14, res\_2])

trconv\_6 = Conv2DTranspose(16, (2, 2), strides=2, padding='same')(add\_5)

res\_15 = resblock(trconv\_6)

add\_6 = Add()([res\_15, res\_1])

conv\_7 = Conv2D(16, (3, 3), activation='relu', padding='same')(add\_6)

batch\_1 = BatchNormalization()(conv\_7)

conv\_8 = Conv2D(16, (3, 3), activation='relu', padding='same')(batch\_1)

batch\_2 = BatchNormalization()(conv\_8)

out1 = Conv2D(2, (1, 1), activation='softmax', padding='same', name='seg')(batch\_2)

model = keras.Model(inputs=[input\_1], outputs=[out1], name="model1")

return model

# Define paths

img\_path = '/content/drive/MyDrive/Data/arcade/syntax/train/images'

msk\_path = '/content/drive/MyDrive/Data/arcade/syntax/train/masks'

# Load images filenames

img\_files = sorted(glob(os.path.join(img\_path, '\*')))

msk\_files = sorted(glob(os.path.join(msk\_path, '\*')))

if len(img\_files) == 0:

print(f"No images found in the specified path: {img\_path}")

if len(msk\_files) == 0:

print(f"No masks found in the specified path: {msk\_path}")

imgs = [os.path.basename(i) for i in img\_files]

msks = [os.path.basename(m) for m in msk\_files]

print(f"Found {len(imgs)} images and {len(msks)} masks.")

# Create data generators

train\_gen = MyGenerator(imgs[:int(len(imgs) \* 0.5)], msks[:int(len(msks) \* 0.5)], img\_path=img\_path, msk\_path=msk\_path, weights=[1, 80], batch\_size=2)

valid\_gen = MyGenerator(imgs[int(len(imgs) \* 0.5):int(len(imgs) \* 0.9)], msks[int(len(msks) \* 0.5):int(len(msks) \* 0.9)], img\_path=img\_path, msk\_path=msk\_path, weights=[1, 80], train=False)

test\_gen = MyGenerator(imgs[int(len(imgs) \* 0.9):], msks[int(len(msks) \* 0.9):], img\_path=img\_path, msk\_path=msk\_path, weights=[1, 80], train=False, shuffle=False)

# Create and compile the model

model = model()

model.compile(

loss={'seg': tf.keras.losses.SparseCategoricalCrossentropy()},

optimizer='Adam'

)

# Define callbacks

callback = tf.keras.callbacks.EarlyStopping(monitor='val\_loss', patience=30, restore\_best\_weights=True)

# Train the model

model.fit(train\_gen, validation\_data=valid\_gen, epochs=10, callbacks=[callback])

# model.save('/content/drive/MyDrive/model/segmentation\_model.h5')

model.save('/content/drive/MyDrive/model/segmentation\_model\_100\_epochs.keras')

import matplotlib.pyplot as plt

# Визуализация графиков обучения

history = model.history

plt.figure(figsize=(10, 5))

plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['val\_loss'], label='Validation Loss')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend()

plt.title('Training and Validation Loss')

plt.show()

from sklearn.metrics import confusion\_matrix

import seaborn as sns

# Предсказание на тестовом наборе данных

y\_pred = model.predict(test\_gen)

y\_pred\_classes = np.argmax(y\_pred, axis=-1)

# Истинные метки

y\_true = []

for i in range(len(test\_gen)):

\_, y\_batch, \_ = test\_gen[i]

y\_true.extend(np.argmax(y\_batch, axis=-1))

y\_true = np.array(y\_true)

# Сравнение предсказанных меток с истинными

conf\_matrix = confusion\_matrix(y\_true.flatten(), y\_pred\_classes.flatten())

# Построение тепловой карты матрицы ошибок

plt.figure(figsize=(10, 8))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues')

plt.xlabel('Predicted')

plt.ylabel('True')

plt.title('Confusion Matrix')

plt.show()

# Загрузка обученной модели

from keras.models import load\_model

model = load\_model('/content/drive/MyDrive/model/segmentation\_model.h5')

# Загрузка нового изображения

new\_image\_path = '/content/drive/MyDrive/data/input/syntax/test/images/1.png'

new\_image = tf.keras.preprocessing.image.load\_img(new\_image\_path, color\_mode='grayscale', target\_size=(512, 512))

new\_image = tf.keras.preprocessing.image.img\_to\_array(new\_image) / 255.0

new\_image = np.expand\_dims(new\_image, axis=0) # Добавление batch dimension

# Предсказание маски

prediction = model.predict(new\_image)

predicted\_mask = np.argmax(prediction, axis=-1)[0]

# Визуализация предсказанной маски

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title('Original Image')

plt.imshow(new\_image[0, :, :, 0], cmap='gray')

plt.subplot(1, 2, 2)

plt.title('Predicted Mask')

plt.imshow(predicted\_mask, cmap='gray')

plt.show()

# Заключение

В ходе исследования было установлено, что применение сверточных нейронных сетей для сегментации сосудов с наличием стеноза на изображениях рентгеновской коронарографии имеет огромный потенциал в области биомедицинской диагностики. Метод U-Net показал высокую точность и надежность при извлечении сосудов, что делает его эффективным инструментом для работы с медицинскими изображениями. Алгоритмы балансировки и аугментации данных оказались необходимы для улучшения результатов сегментации, позволяя улучшить качество изображений и повысить точность выделения сосудов на них. Эти методы позволяют справиться с проблемами несбалансированных данных и улучшить обучение нейронных сетей. Применение сверточных нейронных сетей для детекции стенозов в кардиологии также оказалось эффективным, позволяя выявлять патологии сердечно-сосудистой системы на ранних стадиях и улучшать диагностику заболеваний. Сравнительный анализ методов сегментации сосудов на изображениях МРТ и рентгеновской коронарографии показал, что сверточные нейронные сети могут быть успешно применены для обеих модальностей, что открывает новые перспективы в области медицинской диагностики. Таким образом, роль сверточных нейронных сетей в улучшении диагностики сердечно-сосудистых заболеваний неоспорима. Дальнейшие исследования в этой области могут привести к созданию более точных и эффектив