

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ «ШКОЛА № 626 ИМ. Н.И.
САЦ»

ПРОЕКТНАЯ РАБОТА
по теме
AI ассистент врача при операциях - «Health-AI»

Работу выполнил:
Ученик «10Б» класса
Скопцов Василий Олегович
Научный руководитель:
Учитель физики
ГБОУ Школа № 626
Калинина Алена Игоревна

Москва
2026

ОГЛАВЛЕНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	3
ВВЕДЕНИЕ	5
Обзор существующих систем мониторинга состояния пациента	7
Пульсоксиметры (рис. 1.4).....	7
Мониторы пациента (рис. 1.5).....	8
Почему существующие системы являются однопараметрическими	10
Почему существующие системы реактивные, а не предиктивные	11
Почему предложенный мой подход является гибридным	11
ОБОСНОВАНИЕ	13
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ	14
ЭТАПЫ РАБОТЫ НАД ПРОЕКТОМ	15
РАЗРАБОТКА МОДЕЛИ КЛАССИФИКАЦИИ ВИЗУАЛЬНЫХ ПРИЗНАКОВ	15
РАЗРАБОТКА И АНАЛИЗ СИГНАЛОВ ЭЛЕКТРОКАРДИОГРАММЫ (ЭКГ).....	29
ОПИСАНИЕ СИСТЕМЫ НА ОСНОВЕ ПРАВИЛ ДЛЯ КЛАССИФИКАЦИИ ОСЛОЖНЕНИЙ ПО ДАНЫМ ИНТРАОПЕРАЦИОННЫХ ДАТЧИКОВ	46
ОБЩАЯ РАБОТА ПРОГРАММЫ.....	51
Используемые библиотеки	51
Алгоритм программы	53
РЕЗУЛЬТАТ И ТЕСТИРОВАНИЕ	58
ВЫВОД.....	64
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	65

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

AB – Атриовентрикулярная блокада.

ВОЗ – Всемирная Организация здравоохранения.

ИИ – (AI — Artificial Intelligence): Искусственный интеллект.

ИМ – Инфаркт миокарда.

МЛ – (ML — Machine Learning): Машинное обучение.

РЯВ – (Расширенная Яремная Вена).

ФП – Фибрилляция Предсердий.

ЧСС – Частота Сердечных Сокращений.

ЭКГ – Запись электрической активности сердца, позволяющая оценить его ритм и состояние миокарда.

AV – Атриовентрикулярный (относящийся к предсердиям и желудочкам).

CNN – (Convolutional Neural Network): Сверточная нейронная сеть. Тип нейросети для распознавания образов и графиков.

EtCO₂ – (End-Tidal CO₂): Капнография. Концентрация углекислого газа в конце выдоха.

F-волны – «Пилообразные» волны при трепетании предсердий (f-волны — при фибрилляции).

HR – (Heart Rate): ЧСС.

JVD – (Jugular Venous Distension): Набухание яремных вен.

LA – (Left Arm): Левая рука (электрод).

Lr – (Learning Rate): Скорость обучения нейросети.

MAP – (Mean Arterial Pressure): СрАД (среднее артериальное давление).

P – (Зубец P): Отражает сокращение предсердий.

PI – (Perfusion Index): Индекс перфузии (интенсивность кровотока).

QRS – (Комплекс): Отражает сокращение желудочков.

RA – (Right Arm): Правая рука (электрод).

ResNet – (Residual Network): Резидуальная (остаточная) нейронная сеть.

RL – Правое ребро / Правая нога (заземляющий электрод).

SpO2 – (Saturation of Peripheral Oxygen): Сатурация (кислород в крови).

T – (Зубец T): Отражает восстановление желудочков.

Urine – Диурез (моча). Контроль объема за время.

V~ – (V): Напряжение (Вольт)

ВВЕДЕНИЕ

Врачи, подобно всем людям, подвержены влиянию человеческого фактора. При проведении длительных хирургических вмешательств (свыше 4 часов) у них может возникать усталость, что приводит к снижению концентрации внимания, особенно в ключевых этапах операции. В результате этого осложнения, проявляющиеся в процессе, могут не быть замечены своевременно. Некоторые из них требуют немедленного реагирования, поскольку в противном случае чреваты послеоперационными осложнениями, необходимостью повторной операции или летальным исходом. По данным ВОЗ (Всемирная организация здравоохранения) (рис. 1.1) около 3 миллионов человек в мире ежегодно умирает ежегодно вследствие небрежности или по недосмотру медицинских работников. Определенный вред наносится каждому десятому пациенту, при этом более чем половине случаев он мог бы быть предотвращен.

Наибольшее бремя ложится на страны со средним и низким уровнем дохода, где по этой причине умирает каждый 4-й из 100 пациентов. По данным экспертов, лидирует по количеству смертей Южная Азия, а также страны Центральной и Западной Африки.

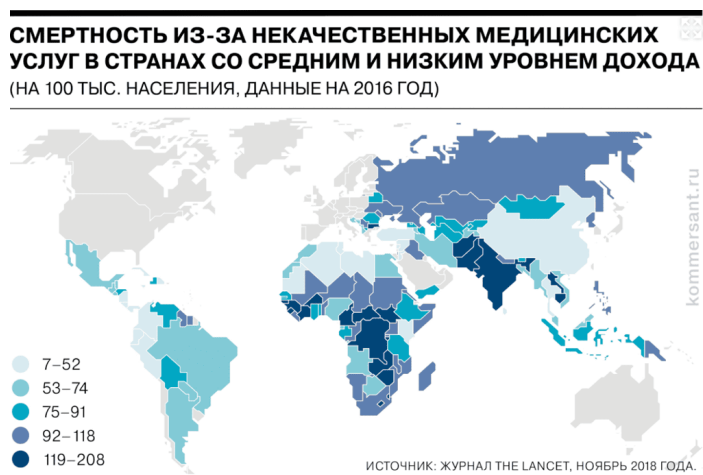


Рис. 1.1 смертность из-за ошибок врачей в мире по данным ВОЗ

В России ежегодно из-за врачебных ошибок во время операций умирает каждый 100 человек на 100_000 человек. В России проживает 148_000_000 человек и следовательно что в год умирает приблизительно 148_000 человек. Тут наибольшей причиной смертности (рис. 1.2) при операциях является неверная диагностика какого-либо осложнения перед операцией – 70% и неверное принятие решений во время операций хирургом при незапланированном результате какого-либо действия или обычная халатность врача – 10%

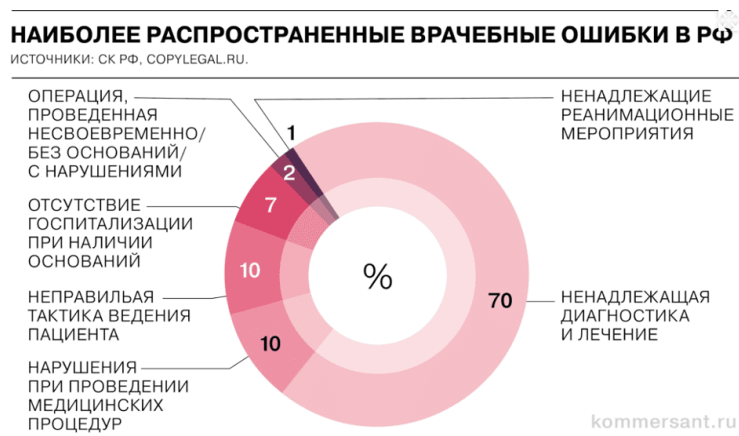


Рис. 1.2 распространенные врачебные ошибки в РФ

А самой частой причиной смерти в мире (рис. 1.3) являются (на 2017-2021 год) Терапия – 28,5%, Диагностика осложнения – 26%, хирургия – 24,1%. Отсюда видно, что преимущественно следствие летального исхода пациента зависит от неверного принятия решение врачом в дооперационный момент и в операционный период.

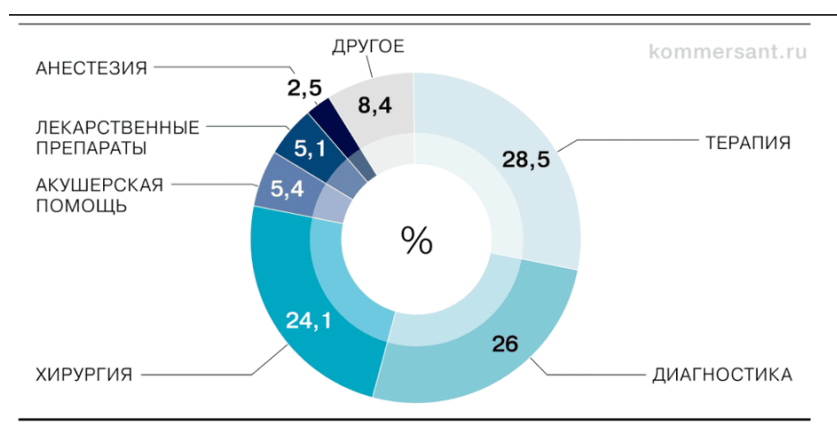


Рис. 1.3 распространенные врачебные ошибки в мире

По итогу видно, что наиболее частой ошибкой можно считать неверная диагностика осложнения или несвоевременная. И из-за чего вследствие пациент умирает на операционном столе.

Сейчас невероятно быстро прогрессирует и развивается технологии машинного обучения. И многие пытаются внедрить эту технологию во все сферы жизнедеятельности человека. И медицина — это не исключение, его внедрение помогает избежать все вышеперечисленные причины летального исхода во время операции. Во многих странах ИИ заменяет рутинные задачи в поликлиниках и больницах, так как у нее гигантские вычислительные возможности: она может анализировать большие объемы данных, анализировать

многие данные и диагностировать осложнения также, как и опытные врачи. Например, на ранних стадиях выявлять онкологические заболевания. Но не смотря на все это методы искусственного интеллекта маловероятно будут внедрены в операционные и также как и врачи полноценно участвовать в операции так как ИИ не заменяет врача полностью, а только лишь страхует его, чтобы тот не совершил какую-либо врачебную ошибку или не пропустил какое-либо осложнение. Это все, потому что немаловажную роль играет клиническое мышление врача, которое формируется ещё в студенческие годы. Мышление доктора основано не только на знаниях о механизмах болезней, но и на опыте различных клинических случаев. ИИ выдаёт только то, что в него загрузили. А человек может не только механически определиться с диагнозом и лечением пациента, но и учесть тонкие детали и взаимосвязи, особенно в сложных случаях.

Чтобы решить проблему с большой смертностью во время операций на данный момент интенсивно внедряют автономные системы которые контролируют состояние пациента во время операции и если врачи не заметят какое-либо проявляющее себя осложнение то они сигнализируют об этом им. Что исключит случай несвоевременной диагностики и возможных плохих последствий.

Обзор существующих систем мониторинга состояния пациента

Пульсоксиметры (рис. 1.4)

Пульсоксиметры являются одними из наиболее распространённых устройств для мониторинга состояния пациента в клинической практике. Основной принцип их работы основан на оптическом измерении степени насыщения гемоглобина кислородом (SpO_2) и частоты сердечных сокращений (ЧСС). Устройство использует два светодиода (красный и инфракрасный диапазоны), излучение которых по-разному поглощается оксигенированным и дезоксигенированным гемоглобином, что позволяет оценить уровень кислорода в крови.

К преимуществам пульсоксиметров относятся:

- неинвазивность;
- простота использования;
- возможность непрерывного мониторинга;
- компактность и низкая стоимость.

Однако пульсоксиметры обладают рядом существенных ограничений. В первую очередь, они предоставляют информацию **только об одном физиологическом параметре**, что не позволяет сформировать целостную картину состояния пациента. Кроме того, изменения SpO_2 зачастую проявляются **на поздних стадиях осложнений**, когда патологический процесс уже значительно развился. Например, при нарушениях сердечного ритма или острой сердечной недостаточности сатурация может оставаться в пределах нормы до критического момента.

Таким образом, пульсоксиметры являются полезным, но ограниченным инструментом, пригодным скорее для базового контроля, чем для раннего выявления осложнений.



Рис. 1.4 Пульсоксиметр

Мониторы пациента (рис. 1.5)

Мониторы пациента представляют собой более комплексные системы, широко используемые в операционных и отделениях интенсивной терапии. Они объединяют данные с нескольких датчиков, таких как:

- электрокардиограмма (ЭКГ);
- сатурация кислорода (SpO_2);
- артериальное давление;
- частота дыхания;
- температура тела.

Несмотря на многопараметричность, данные системы в большинстве случаев выполняют **функцию отображения информации**, а не интеллектуального анализа. Монитор пациента фиксирует текущие значения показателей и

сравнивает их с заданными порогами. При выходе параметра за допустимый диапазон система подаёт тревожный сигнал.

Ключевым недостатком таких мониторов является **отсутствие глубокого анализа взаимосвязей между параметрами**. Каждый сигнал обрабатывается изолированно, без учёта комплексного контекста. В результате система может либо:

- пропустить ранние признаки осложнений;
- либо генерировать ложные тревоги при кратковременных отклонениях.

Кроме того, большинство мониторов не используют методы машинного обучения и не адаптируются к индивидуальным особенностям пациента.



Рис.1.5 Монитор пациента

Closed-loop системы (рис. 1.6)

Closed-loop системы (системы замкнутого цикла) представляют собой более продвинутый класс медицинских технологий. Они не только анализируют входные данные, но и автоматически воздействуют на пациента, изменяя параметры терапии. Примерами таких систем являются:

- автоматические инфузионные насосы;
- системы управления анестезией;
- инсулиновые помпы.

Основной принцип closed-loop систем заключается в следующем:

1. измерение физиологического параметра;
2. сравнение с целевым значением;
3. автоматическая коррекция воздействия.

Несмотря на высокий уровень автоматизации, такие системы, как правило, **ориентированы на один доминирующий параметр** (например, уровень глюкозы или глубину анестезии). Это связано с требованиями безопасности: чем меньше параметров учитывает система, тем проще её сертифицировать и контролировать.

Следовательно, closed-loop системы обладают высокой точностью в узкой задаче, но не предназначены для комплексного мониторинга состояния пациента и выявления разнообразных осложнений.

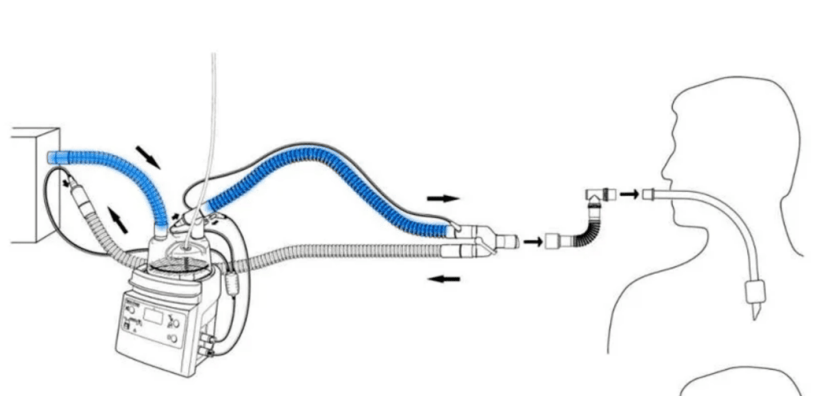


Рис. 1.6 Closed-loop система

Почему существующие системы являются однопараметрическими

Однопараметричность большинства медицинских устройств обусловлена несколькими факторами:

1. **Регуляторные ограничения**

Медицинские устройства проходят строгую сертификацию. Чем сложнее алгоритм и больше параметров он использует, тем выше риск отказа системы и сложнее доказать её безопасность.

2. **Интерпретируемость**

Простые системы легче объяснить врачу. Врач видит конкретный параметр и его значение, не полагаясь на «чёрный ящик».

3. **Технические ограничения**

Исторически вычислительные ресурсы медицинских приборов были ограничены, что не позволяло реализовать сложные модели анализа данных в реальном времени.

В результате большинство систем ориентированы на контроль отдельных показателей, а не на анализ их совокупности.

Почему существующие системы реактивные, а не предиктивные

Реактивный характер медицинских систем заключается в том, что они **реагируют на уже произошедшее отклонение**, а не прогнозируют его появление. Причины этого следующие:

- традиционные алгоритмы основаны на пороговых значениях;
- отсутствует анализ динамики сигналов;
- не учитываются скрытые закономерности и паттерны;
- отсутствует обучение на больших объёмах данных.

Например, при анализе ЭКГ большинство систем подают сигнал тревоги только при выраженной аритмии, игнорируя ранние изменения формы зубцов или нестабильность ритма, которые могут предшествовать осложнению.

Предиктивный подход требует применения методов машинного обучения, анализа временных рядов и комплексного учета нескольких источников данных, что до недавнего времени было трудно реализуемо в клинической практике.

И я предлагаю сделать новую автономную систему Health-AI. Которая может контролировать состояние пациента по трем параметрам: кардиограмма ЭКГ первого или второго отведения, данным с 9 интраоперационным датчикам, визуальные признаки с камеры направленной на лицо пациента. Моя программа умеет классифицировать 9 различных осложнений: Инфаркт миокарда (ИМ), фибрилляция предсердий (ФП), Атриовентрикулярная блокада (АВ), цианоз, вздутие яремной вены, гипоксия, аллергическая реакция, гемодинамический шок, гипокаугуляция.

Почему предложенный мой подход является гибридным

Разрабатываемая в рамках данной работы система Health-AI использует **гибридный подход**, объединяющий преимущества различных методов анализа данных:

1. Нейронные сети

Используются для анализа сложных, высокоразмерных данных:

- визуальные признаки осложнений;

- формы и паттерны ЭКГ-сигналов.
Это позволяет выявлять скрытые закономерности, недоступные классическим алгоритмам.

2. Алгоритм на основе правил

Применяется для интерпретируемого анализа данных датчиков. Такой подход обеспечивает:

- прозрачность логики принятия решений;
- соответствие клиническим нормам;
- дополнительную проверку предсказаний нейронных сетей.

3. Мультимодальная интеграция

Система объединяет данные из трёх независимых источников: ЭКГ, видеопотока и физиологических датчиков. Это снижает вероятность ложных срабатываний и повышает надёжность итогового вердикта.

Таким образом, гибридный подход позволяет совместить **предиктивные возможности искусственного интеллекта с надёжностью и интерпретируемостью классических медицинских алгоритмов**, что делает систему более устойчивой и перспективной для дальнейшего развития.

ОБОСНОВАНИЕ

Актуальность:

Актуальность проекта обусловлена несколькими факторами:

1. **Повышение качества медицинской помощи:** Современные технологии позволяют существенно повысить точность диагностики и скорость реагирования на критические состояния пациентов. Мониторинг в реальном времени позволяет выявлять проблемы на ранних стадиях, предотвращая развитие серьезных осложнений.
2. **Обеспечение непрерывного контроля:** Использование многоканальной регистрации данных (ЭКГ, датчики, видеоконтроль лица) обеспечивает комплексный мониторинг здоровья пациента, позволяя оперативно реагировать на изменения в состоянии даже в условиях ограниченного медицинского персонала.
3. **Оптимизация ресурсов здравоохранения:** Автоматизированная система снижает нагрузку на медицинский персонал, обеспечивая раннее выявление проблем и минимизацию ошибок в диагностике, что особенно важно в условиях дефицита кадров и увеличения нагрузки на медицинские учреждения.
4. **Расширение возможностей телемедицины:** Проект способствует развитию удаленных форм мониторинга и оказания медицинских услуг, делая медицинскую помощь доступной для жителей отдалённых регионов и малоподвижных групп населения.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель: создание автономной программы, контролирующая состояние пациента

Задачи:

1. Создать нейронные сети на основе Pytorch
2. Обучить нейронные сети распознавать визуальные осложнения.
3. Обучить нейронные сети распознавать осложнений на кардиограммах ЭКГ.
4. Создать алгоритм классификации осложнений по датчикам и проверку данных
5. Создание алгоритма взаимодействия моделей и алгоритмов
6. Создание интерфейса.

ЭТАПЫ РАБОТЫ НАД ПРОЕКТОМ

РАЗРАБОТКА МОДЕЛИ КЛАССИФИКАЦИИ ВИЗУАЛЬНЫХ ПРИЗНАКОВ

Характеристики визуальных данных

Входные данные для модели представляют собой видеокадры, получаемые с видеокамеры, направленной на лицо пациента. Разрешение изображений составляет 64×64 пикселей, формат — RGB, что соответствует трём каналам тензора для обработки цветовой информации.

Разработка и оптимизация модели

На начальном этапе была разработана бинарная нейронная сеть. Предполагалось, что ограниченный объём данных (датасет объёмом около 1000 ± 200 изображений на класс) в сочетании с методами аугментации, увеличивающими объём данных в пять раз (до примерно 5000 изображений на класс), обеспечит высокую точность классификации. Однако в ходе обучения была выявлена ключевая проблема: ввиду ограниченного доступа к специализированным датасетам осложнений данные собирались самостоятельно. Это привело к наличию схожих визуальных признаков у некоторых осложнений, что снижало надёжность бинарной модели, поскольку она не учитывала возможность других классов осложнений и могла генерировать ложноположительные предсказания.

В связи с этим было принято решение о переходе к много классовой модели. Хотя это привело к снижению общей точности модели ввиду усложнения задачи классификации, много классовый подход позволил повысить достоверность предсказаний за счёт учёта большего числа возможных состояний. В результате разработанная модель была успешно интегрирована в программное обеспечение проекта.

Разработанная нейронная сеть предназначена для выполнения следующих функций:

1. Дифференциации различных клинических осложнений на основе визуальных данных.
2. Функционирования в режиме реального времени для оперативного анализа видеопотока.

3. Обеспечения устойчивости к вариациям освещения, теням и бликам, характерным для операционной среды.

Классификация и выбор целевых состояний

В рамках проекта определены пять основных классов для классификации: цианоз, вздутие яремной вены, аллергическая реакция, норма (здоровое состояние) и общий класс для нераспознанных осложнений. Введение общего класса позволило избежать некорректной классификации неизвестных осложнений как нормального состояния, что повысило надёжность модели.

Выбор классов осложнений основывался на их клинической значимости: отобраны состояния, встречающиеся с умеренной частотой, но потенциально ведущие к серьёзным последствиям, включая летальный исход, при несвоевременном выявлении. Указанные осложнения изначально диагностируются визуально, а их подтверждение с помощью датчиков возможно лишь спустя некоторое время, что подчёркивает важность автоматизированного визуального анализа в реальном времени.

1) Цианоз (рис. 2.1):

Цианоз (посинение отдельных участков кожи, губ или кончиков пальцев) — симптом, который указывает на то, что ткани получают недостаточно кислорода. Такое состояние может говорить о сердечно-сосудистых, лёгочных или аутоиммунных патологиях, а также указывать на передозировку некоторых лекарств.

При некоторых патологиях даже артериальная — поступающая от лёгких — кровь становится менее насыщенной кислородом и меняет цвет на более тёмный. Кожа при этом теряет розоватый оттенок и приобретает серый, синеватый или фиолетовый. Это говорит о том, что кислорода в крови недостаточно.

Также избыток бедной кислородом крови в клетках возникает вследствие нарушения оттока венозной крови.

Такие состояния могут быть вызваны целым рядом патологий. Как правило, это отклонения работы дыхательной или сердечно-сосудистой системы. К примеру это:

- респираторные инфекционные заболевания;
- миокардит — воспалительное поражение клеток сердечной мышцы;
- аритмии — нарушения сердечного ритма;
- нарушение работы клапанов сердца;
- синдром Рейно — спазм сосудов кистей, стоп или носа;

Цианоз указывает на серьёзные нарушения в работе жизненно важных органов, главным образом лёгких или сердца. Если своевременно не диагностировать такие патологии и не начать лечение, они могут привести к опасным осложнениям, например некроз тканей



Рис 2.1 цианоз

2) Вздутие яремной вены (рис. 2.2):

Расширение яремных вен (РЯВ) относится к видимому выпячиванию яремных вен на шее, которое может возникнуть при повышенном давлении в правой части сердца. РЯВ часто является признаком проблем с сердцем или кровообращением, особенно когда сердце не может эффективно перекачивать кровь.

Обычно вызывается состояниями это сердечная недостаточность легочная гипертензия или Заболевания трехстворчатого клапана.

Вздутие Яремной вены может привести к инсульту или образованию тромб или накоплению в легких жидкости или повреждению различных органов



Рис. 2.2 вздутая яремная вена

3) Аллергическая реакция (рис. 2.3):

Аллергическая реакция (аллергия) — это чрезмерная реакция организма на определённые вещества, которые в норме не оказывают патологического действия, — **аллергены**. В аллергических реакциях участвует иммунная система: она ошибочно принимает безвредное вещество за опасное и атакует его, чтобы защитить организм.

Бывают несколько видов аллергических реакций: анафилаксия, крапивница, отек Квинке.

Но все эти аллергические реакции приводят к одному и тому же результату: к примеру остановки сердца или дыхания человека.

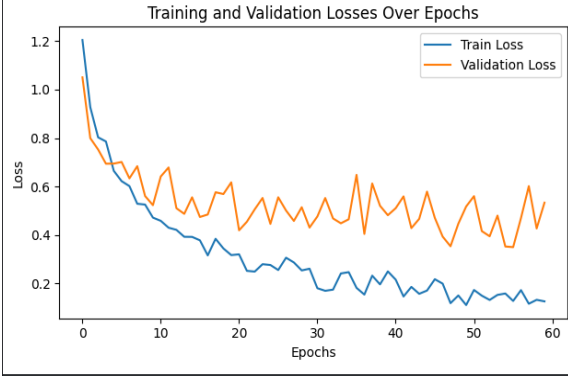
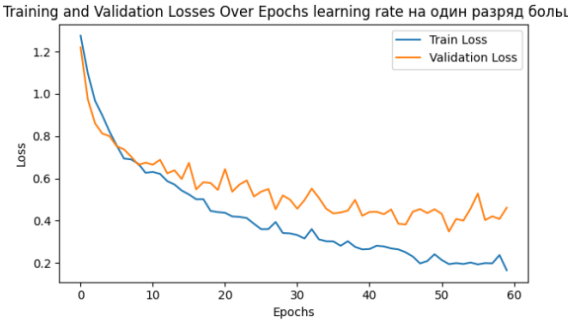


Рис. 2.3 аллергическая реакция

Выбор архитектуры модели

В ходе экспериментов были протестированы несколько архитектур нейронных сетей: CNN, U-Net и ResNet50 - предобученная. Однако уже на первых тестах было понятно, что архитектура U-NET в моем случае не подходит, так как на нем шло переобучение уже на первых 2 - 3 эпохах и также она обучалась довольно долго, а именно 60 эпох занимают приблизительно половины недели обучения и следовательно время обучения не оправдывает ожидания. Это следовало все от туда, что это довольно сильно тяжелая модель и мощная. И там располагается не маленькое количество нейронов и она дает глубокое обучение. И тогда остаются только две модели CNN и

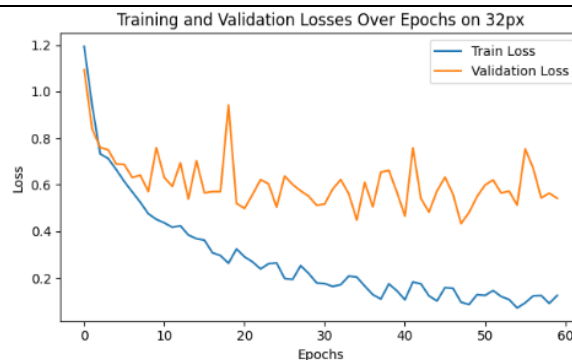
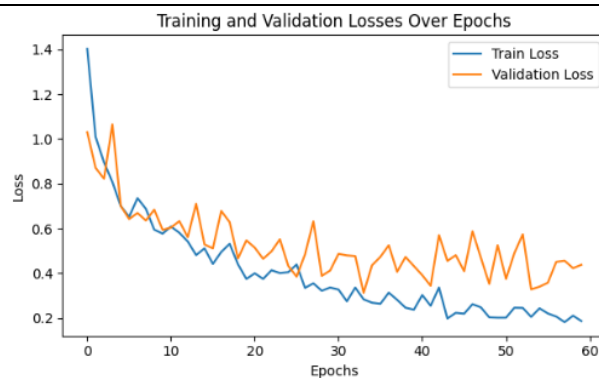
ResNet50. И тут я начал обучать множество моделей на данной архитектуре CNN изменяя разные параметры: кол-во линейных слоев, аргументация, размер изображений, кол-во нейронов. И на базе этого эксперимента у меня получилась данная таблица (табл. 1.1)

Вывод	Фото
Данная модель является с базовыми часто используемыми мной параметрами. Learning rate – 0,0001; линейные слои - 3; размер картинки – 64 пикселя. По данному графику видно, что лучшая ошибка составляет 0.45 на тестовой выборке. Ошибка интенсивно падала до 20 эпохи, а далее очень сильно замедлила свой темп, так как значение тестовой выборки стала выравниваться и в конце стала параллельна обучающей выборки, что нам говорит о том, что в конце модель переобучилась.	 The graph shows training and validation losses over 60 epochs for a learning rate of 0.0001. The training loss (blue line) decreases steadily from approximately 1.1 to 0.15. The validation loss (orange line) decreases from approximately 1.0 to 0.45 by epoch 20, then fluctuates between 0.4 and 0.6 for the remainder of the training process.
При уменьшении Learning Rate (шаг обучения) до 0.00001 предполагалось, что у модели будет больше шансов лучше обучиться так как она делает маленький шаг и с большей вероятностью попадет в локальный минимум. И по итогу модель начинает переобучаться уже на 15 эпохе и на данной эпохе ошибка приблизительно равна 0.65. Благодаря уменьшению шага обучения у меня градиентный спуск нашел довольно быстро и неплохой локальный минимум функции, но на этом все. Ошибка в тот момент перестала падать так стремительно	 The graph shows training and validation losses over 60 epochs for a learning rate of 0.00001. The training loss (blue line) decreases from approximately 1.1 to 0.2. The validation loss (orange line) decreases from approximately 1.0 to 0.65 by epoch 15, then fluctuates between 0.4 and 0.6 for the remainder of the training process.

как в начале, но и перестала и расти и резко изменяться по амплитуде в конце обучения (в периоде деградирования). И по итогу можно было не много дать ей переобучиться до 20 эпохе, где ошибка составляла приблизительно 0.6, для дальнейшего использования.

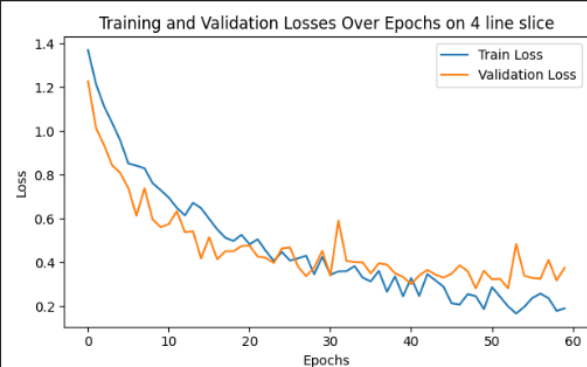
При увеличении размера изображений до 128 пикселей и количества входных нейронов модель получила более хорошую информацию об изображениях (больше мелких деталей). И следовательно модель стала обучается довольно стабильно. Процесса деградирования модели почти нет даже на 60 эпохе. Но все же ошибка не опустилась ниже отметки 0.4. Данные ошибки часто пересекаются друг с другом, что является хорошим маркером. Процесс переобучения приблизительно зарождается на 32 эпохе где далее тестовая ошибка выравнивается и идет параллельно обучающей ошибки. Но при увеличении размера изображения можно увидеть что амплитуда колебания ошибка из выборки тестовой довольно большая особенно в конце.

При уменьшении размера картинки до 32 пикселей и уменьшая следовательно значение количества входных нейронов мы даем меньше информации модели пренебрегая маленькими деталями. И тем самым давай модели меньше нейронов даем ей меньше шансов на переобучение. Но по итогу обу-



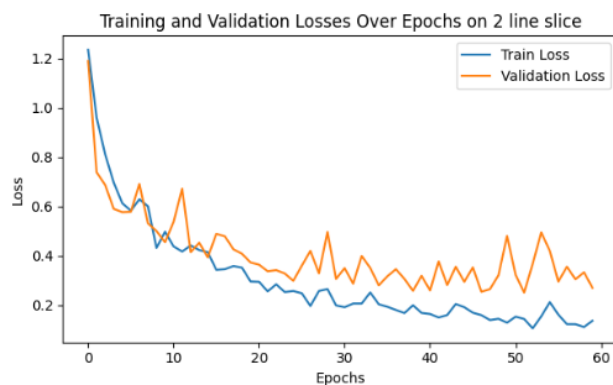
чение пошло довольно плохо. Уже на 10 эпохе пошла деградация модели, а в тот момент ошибка составляет приблизительно 0.6. а обучающую выборку модель просто запомнила и при этом очень хорошо. Также мы видим что амплитуда колебания тестовой выборки ошибки очень велико – в целую единицу. Это следует из того, что у модели слишком мало информации об изображениях и она не может найти основные черты каждого класса и просто в слепую пытается их найти передвигаясь большими шагами по производной ошибки.

При увеличении количества линейных слоев до 4, делая модель более глубокой, ошибка в первой трети обучения падает довольно интенсивно. И на 20 эпохе ошибка составляет уже 0.45 но далее тестовая выборка ошибки выравнивается и идет параллельно оси X. А обучающая выборка продолжает падать что нам говорит что модель переобучилась. Также мы видим что самая первая ошибка при обучении была большая Приблизительно 1.3. следовательно увеличение линейных слоев имеет место быть лишь только при увеличении кол-во данных в датасете. Также видим что обучающая выборка ошибки тоже колеблется. А это следует из того что модель ее не заучивала просто так как в других случаях а пытается улучшить свое

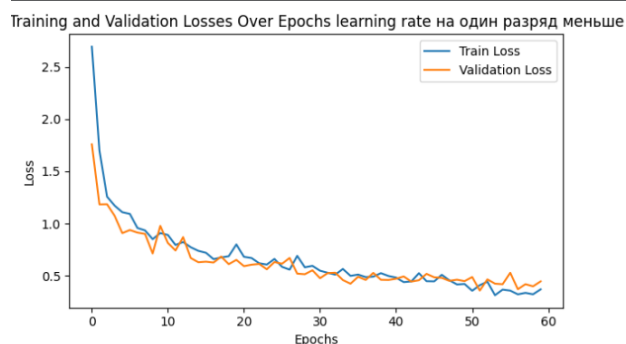


положение.

При уменьшении линейных слоев до 2, делая модель более простой, чтобы она не запоминала данные. И в первые 10 эпох ошибка стала довольно быстро падать с 1.2 до 0.5. но далее ошибка стала медленно падать, но без явного переобучения. Хотя и с некоторыми колебаниями ошибки тестовой выборки. И лучший ее результат ошибки около 0.4 - 0.3. также видим что тестовая выборка на протяжении всего обучения имела большую амплитуду колебания. Что следует из того что она находила локальный минимум и сразу из него выходила так как они все были не очень глубокие.

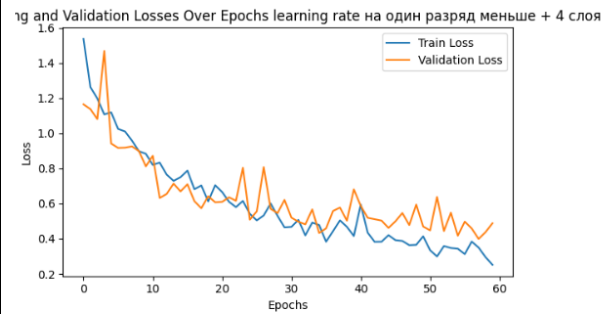


При увеличении Learning rate (шаг обучения) до 0.001 изначальная ошибка была очень большая приблизительно 2.6 и за одну эпоху она ее сбila до 1.3. но потом модель стала довольно медленно обучаться. Но самое главное то что модель не переобучалась и не деградировала. Ошибка обучающей выборки с тестовой выборки довольно часто переплетались и при этом высота амплитуды и ее частота были довольно малы. Но не смотря на это все минимальная ошибка составила 0.5. дальше продолжать обучение было бессмысленно так как ошибка падала гиперболическим образом и при увеличении даже количество эпох в 2 раза результат бы изменился со-

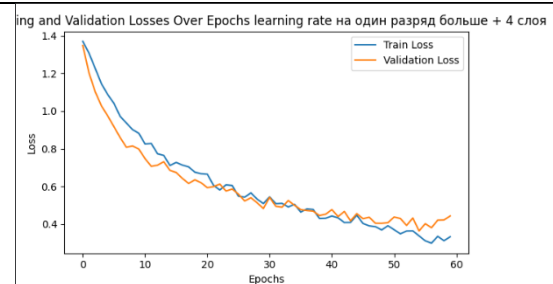


всем бы на чуть-чуть.

При совокупности 4 линейных слоев и увеличения шага обучения смоли добиться ошибки до 0.4. Здесь ошибки двух выборок довольно часто пересекаются, что нам говорит, что модель довольно хорошо понимает классы, заложенные в нее. Также данная модель имеет наибольшую точность по сравнению с другими мною обученными моделями: 87%. И при этом на протяжении всех эпох не было замечено процесса деградации или переобучения модели. Также видим что начальная ошибка была 1.6 что не мало. Также видим что направление обучения обучающей выборки очень хорошая, во первых, модель не запоминает ее так как две ошибки выборки близко расположены друг к другу и при этом скорость изменения ошибки на последних эпохах не уменьшается как в других случаях обучения.



При совокупности 4 линейных слоев и уменьшении шага обучения мы видим следующий результат. Модель имеет ошибку всего 0.4. Точность данной модели составляет 83%. Данная модель тоже имеет большое количество пересечения между двумя выборками ошибок. Также отличительной особенностью этой модель то, что модель эта понимала лучше тестовую выборку, нежели обучающую до 20 эпохи. Но обучение этой модели.



Остановил алгоритм EarlyStopping из-за переобучения в конце так как тестовая ошибка выборки пошла вверх. А обучающая выборка продолжала падать. ы	
---	--

Табл. 1.1 эксперимент по CNN

И при обучении предобученной модели на архитектуре ResNet50 получились следующие результаты (рис. 2.4) (рис. 2.5). Ошибка модели обучения составляет приблизительно 0.3. А точность 90%. Это очень сильно хороший результат учитывая размер моего датасета. Но при обучении видно что уже на 15 эпохе модель дала свой максимум. Далее обучаясь модель просто колебалась но не улучшая точность и не переобучаясь. Она шла параллельно оси X немного просто изменяя ошибку. Но это значение пренебрежимо мало. Но по итогу при тесте модели она показала крайне плохие результаты. Она не могла определить осложнения поступавшие в модель с видео потока с камеры. Скорее всего это было вызвано тем, что модель предобучена на общих обывденных данных на подобии человека, кошек, машин... И давая ей мои медицинские данные специфичные и при этом довольно малом размере датасета, она начала путаться и по факту не обучилась должным образом Так как у нее другая специфика.

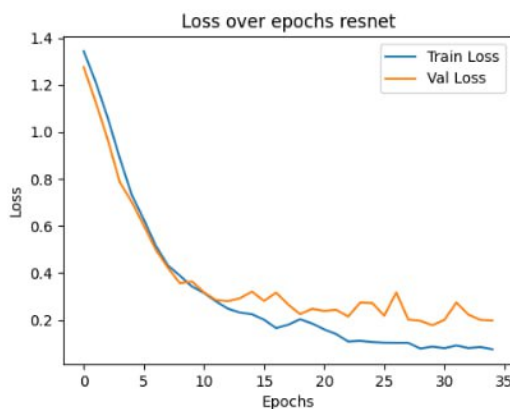


Рис. 2.4 ошибка обучения

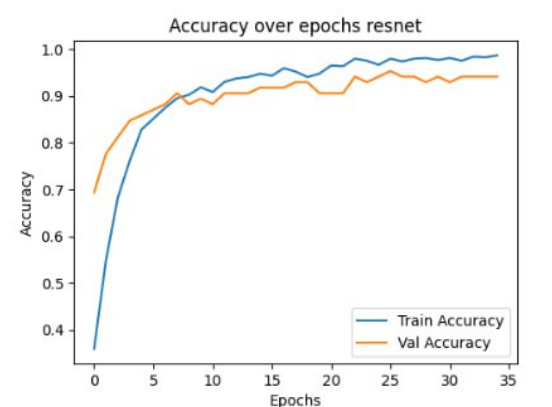


Рис. 2.5 точность обучения

Обучение модели

1. Подготовка данных для обучения

- Датасет загружается через `torchvision.datasets.ImageFolder`, структура папок соответствует стандартной: `root/class_name/.jpg`.
- Применяется аугментация + нормализация, плюс обязательный `Resize` до 64×64 и `Normalize` с `mean/std = 0.5`.
- Датасет случайным образом делится на тренировочную (70%) и валидационную (30%) части.
- Создаются `DataLoader`-ы с `batch_size=16` и `shuffle=True` для тренировочной выборки.

2. Архитектура модели

Предложена относительно глубокая кастомная CNN с последовательностью сверточных блоков:

Feature extractor (backbone):

- `Conv2d(3 → 64) → BatchNorm → ReLU → MaxPool(2)`
- `Conv2d(64 → 128) → BatchNorm → ReLU → MaxPool(2)`
- `Conv2d(128 → 256) → BatchNorm → ReLU → MaxPool(2)`
- `Conv2d(256 → 512) → BatchNorm → ReLU`

После трех `MaxPool` с `stride=2` размерность feature map уменьшается в 8 раз:
 $64 / 8 = 8 \rightarrow 8 \times 8 \times 512 = 32768$ признаков.

Classifier head:

- `Flatten`
- `Linear(32768 → 512) → ReLU → Dropout(0.4)`
- `Linear(512 → 256) → ReLU → Dropout(0.2)`
- `Linear(256 → 64) → ReLU → Dropout(0.2)`
- `Linear(64 → 4)`

Модель использует `Batch Normalization` для стабилизации обучения и `Dropout` для регуляризации, что особенно важно при ограниченном объеме медицинских данных.

3. Обучение

- Устройство: CUDA при наличии, иначе CPU.
- Loss: CrossEntropyLoss (стандарт для многоклассовой классификации).
- Оптимизатор: Adam с learning rate = $1e-3$.
- Количество эпох: 60.
- Реализован механизм сохранения лучшей модели по минимальной валидационной потере.
- Применен Early Stopping (закомментирован, но готов к использованию) с patience=8 и min_delta= $1e-4$.

В цикле обучения:

- На каждой эпохе вычисляется средняя тренировочная потеря.
- Проводится валидация без градиентов: вычисляется валидационная потеря и ассурасу на тестовом подмножестве.
- Выводятся метрики: Train Loss, Val Loss, Validation Accuracy.

Ограничения

Данная модель по итогу с довольно хорошим качеством может определять классы. Его датасет я попытался собрать со всевозможными исключениями: когда голова повернута так как во время операции возможны разные сценарии развития событий и при некоторых осложнениях возможно нужно будет повернуть голову пациенту или что-то произойдет с камерой и она сдвинется со своей оси, при разном освещении, так как возможны разные операционные с разным освещением (сделал за счет аргументации при добавлении фильтров), разных полов людей, так как чтобы модель выучила осложнение по параметрам таким к примеру перед ним женщина или мужчина, разный поворот камеры, так как камера может быть смещена или быть какая либо особенность операционной (за счет аргументации при добавлении фильтров), отдельные части тела для более лучшего понимания ключевых параметров (к примеру при цианозе иногда кроме всего лица я добавлял только посиневшие губы, пальцы и нос). Но каждый класс не смотря на все исключения, которые я учел иногда модель тоже может путать классы между собой по индивидуальным причинам классов (табл. 1.2) и по общим: к примеру при некачественной графике (при шуме камеры), я специально не стал собирать данные с зашумленными кадрами или применять различные фильтра, так как при маленьком размере моего датасета модель бы просто на просто запуталась бы и не обучилась бы должным образом; если не получится сегментировать ли-

цо (хоть крайне маловероятно, но все же при некоторых обстоятельствах это возможно) то модель с большей вероятностью перепутает классы если рядом с человеком будет находится много вещей и менее приблизительно чем 60% кадра будет занимать лицо пациента так как модель может найти что-либо другое для себя, что будет ей напоминать данный класс; и так же если лицо перекрывает что-либо то нейронная моя сеть тоже может не правильно классифицировать осложнение (при перекрытии примерно 50% лица), это ясное дело, так как уже шансы сегментации лица уменьшаются и так же модель обучена на данных где ни чем не перекрыты основные параметры (исключение это вздутие яремной вены так как в датасете модели этого класса были также положены фотографии еще и обычных вздутых вен и артерий на ногах, руках, сонной артерии).

норма	Вздутие яремной вены	цианоз	Аллергическая реакция
Распознается практически всегда при любом исходе событий (так как лучше промолчать об осложнении чем навести врача на ложный след) но ошибется только при доминировании довольно сильно другого какого-либо класса.	Если будет видно лишь половина шеи на кадре (без предплечий) или будет перекрывать что-либо шея, также если по шее будет проходить какая-либо тонкая трубка, то модель ее может спутать с яремной веной вздутой	Если падает специфический синий цвет на лицо пациента или губы чем-либо перекрываются или пациент лежит на боку и видно лишь часть губ или он имеет не белого цвета кожу, а более темную.	Распознается с большей вероятностью правильно. Ошибка возможна только при доминировании другого класса или наличие у человека петехии (полопавшиеся капилляры на теле) на лице или на лице есть красные пятна от чего, ибо другого (к примеру крови)

Табл. 1.2 уязвимость каждого класса

Но, чтобы уменьшить шанс выбора ненужного класса, я реализовал систему перепроверки данных по датчикам: датчики дополнительно контролируют каждое осложнение визуальное. Если это осложнение которая выбрала

модель не подтвердило совокупность датчиков (алгоритм классификации осложнения по датчикам описан ниже), то программа считает осложнение, выбранное моделью ложным.

Но также данная модель ни как не контролирует ход развития осложнения: она не просматривает временные ряды осложнения, от чего возможно принятие ошибочного класса из-за того, что модель не видит логическую взаимосвязь того что было и того что есть: к примеру долгое время наблюдая цианоз и модель резко сказала, что сейчас у пациента вздутая яремная вена, да, конечно, вероятность такого событие не равно нулю, но оно крайне маловероятно. Следовательно, данный алгоритм может говорить о состоянии пациента, только именно в данный момент без истории в прошлом, что с ним было пять минут назад (как черный ящик).

Также данная модель является по виду классификации *Multiclass* – то есть она имеет просто больше двух классов для классификации, но она не является *Multilabel classification*, она не может отнести объект сразу к двум и более классом – то есть она может отнести данный кадр только к одному какому-либо осложнению. То есть она не может сразу классифицировать два и более осложнений одновременно. Модель будет отдавать большее предпочтение на тот класс, который будет являться более выраженным в данном случае, нежели другие (исключение «Норма», так как в модель изначально заложено, что если она не уверена и из вариантов классов есть норма, то она возьмет ее как приоритетный, это сделано специально так как модель лучше промолчит нежели ошибется и скажет неверное осложнение врачу во время операции).

РАЗРАБОТКА И АНАЛИЗ СИГНАЛОВ ЭЛЕКТРОКАРДИОГРАММЫ (ЭКГ)

Характеристики входных данных

Входной сигнал для модели представляет собой запись ЭКГ второго отвода с частотой дискретизации 500 Гц и длительностью 5000 отсчётов (10 секунд). Такой временной диапазон обеспечивает возможность обнаружения как единичных аномалий (например, экстрасистолий), так и устойчивых патологических паттернов (фибрилляция предсердий, изменения ST).

Для обработки сигнала используется двухканальный тензор:

1. Первый канал — исходный сигнал ЭКГ (амплитудные значения).
2. Второй канал — производная сигнала, отражающая локальные изменения амплитуды.

Применение производной акцентирует резкие изменения сигнала (например, фронты комплекса QRS), повышая чувствительность модели к форме зубцов.

Решение контролировать в ЭКГ именно по второму отведению (рис. 3.1) (электроны - правая рука левая нога) из двенадцати следует из того, что именно на этом отводе хорошо видны комплекс QRS, зубец Р. Направление данного отведения идеально совпадает с направлением распространением основной массы электрического импульса по желудочкам.

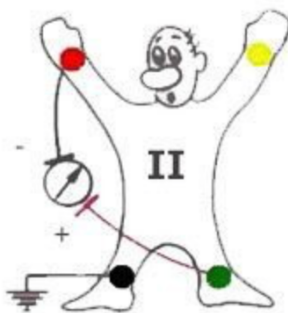


рис. 3.1 второе отведение ЭКГ

Выбор подхода к обработке данных

Изначально рассматривался подход на основе компьютерного зрения для анализа изображений ЭКГ. Однако из-за присутствия посторонних дета-

лей (сетка, аннотации) модель испытывала трудности в обучении. Переход к обработке сигнала в виде точечного массива минимизировал влияние лишних признаков, обеспечив более эффективное обучение.

Нейронная сеть разработана для выполнения следующих функций:

1. Дифференциация клинически значимых состояний на основе 10-секундного фрагмента ЭКГ.
2. Работа в режиме реального времени.
3. Устойчивость к вариациям амплитуды и фоновым помехам в операционной среде.

Классификация и выбор целевых состояний

Выделены три класса отражающих критические аритмии и состояния:

1. Фибрилляция предсердий (ФП) (рис. 3.2)

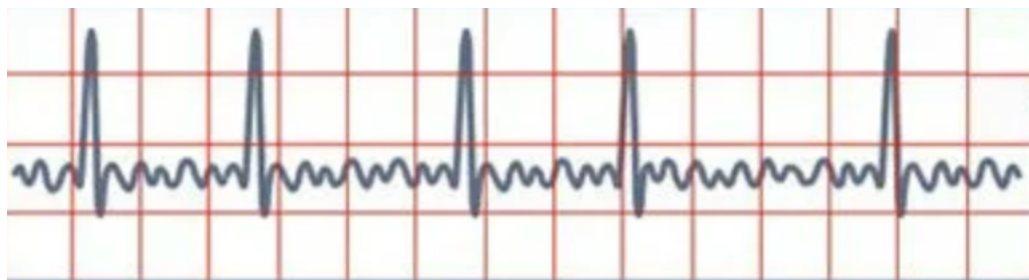


Рис. 3.2 фибрилляция предсердий

Фибрилляция предсердий (мерцательная аритмия): это вид аритмий при которых предсердия (правое и левое) начинают в прямом смысле дребезжать. Обычно служит причиной этого колеблющийся импульс, который образуется после прохождения синусового узла (рис. 3.3) и уже по ножкам пучка Гиса (правым и левым) доходят до клеток миокарда (продолжая колебаться) и те уже принимая его начинают сокращаться и в связи с этим и появляются лишние дребезжание сердца. Чаще всего это может происходить если в предсердия появляются различные зубцы(как правило от гипертонии) и из-за этого импульс не может пройти на прямую через него и начинает дробиться на сотни других маленьких волн или это может возникать когда в устьях легочных вен появляются клетки который начинают генерировать импульс быстрее чем синусовый узел. Тогда этот «ненормальный» им-

пульс сталкивается с «нормальным» и образуются такие мелкие волны в результате.

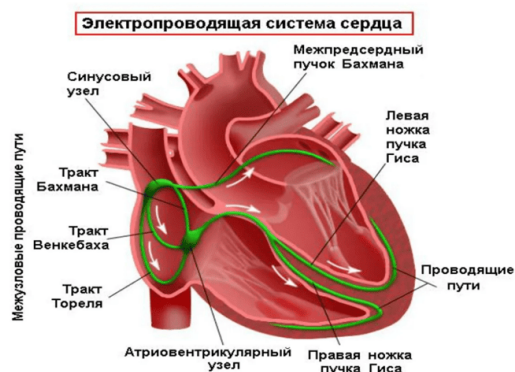


Рис. 3.3 схема сердца

Фибрилляция предсердий опасна тем, что она может перейти в фибрилляцию желудочек, которая очень сильно опасна для жизни, при которой человек в среднем живет минуты две или три.

На кардиограмме ЭКГ это осложнение видно по характерным F-волнам между зубцами QRS, а так же с различным интервалом QRS. F- волны могут быть высокочастотными и низкочастотными и так же ЧСС в этот период может быть в различном диапазоне от 60 ударов в минуту до 120-140 ударов в минуту.

2. Инфаркт миокарда (ИМ) (рис. 3.4)

Инфаркт Миокарда – это необратимое состояние пациента, при котором происходит некроз клеток миокарда в сердце. Как правило это происходит из-за гипоксии сердца (голодание кислородом сердца), в большинстве случаев это происходит из-за частичной или полного тромбоза коронарной артерии или вследствие ишемии миокарда (тоже самое что и инфаркт миокарда но тут состояние пациента обратимое). При ИМ часть клеток миокарда перестают сокращаться и тогда сердце более хуже начинать качать кровь.

Инфаркт Миокарда опасен тем, что он может перейти в Фибрилляцию желудочек (тогда сердце перестанет качать кровь, будет оно просто часто колебаться и все) или может произойти разрыв сердца, кардиологический шок, отек легких.

На кардиограмме ЭКГ ИМ можно увидеть по характерному деградации сегмента ST (он начинает располагаться выше изолинии или ниже соответственно), также ярки признак ИМ – это зазубренность в сегменте PQ или округленный зубец R.



Рис. 3.4 инфаркт миокарда

3. Атриовентрикулярная (AV) блокада (рис. 3.5)

Атриовентрикулярная блокада – это нарушение или полное прекращение проведения электрического импульса от **предсердий** к **желудочкам** вследствие поражения проводящей системы сердца (узла, пучка Гиса или его ножек) (рис 3.3).

1. I степень (Замедление проводимости)

Импульс доходит до желудочков, но с аномальной задержкой.

- **ЭКГ-признак:** Постоянное удлинение интервала **PQ** > 0,20 сек.
- **Статус:** обычно бессимптомна, лечения не требует, но требует наблюдения.

2. II степень (Неполная блокада)

Часть импульсов проводится, а часть — блокируется. Делится на два типа:

- **Мобитц 1 (с периодами Самойлова-Венкебаха):** происходит прогрессирующее удлинение интервала **PQ** до тех пор, пока очередной комплекс **QRS** не выпадает совсем. После паузы цикл повторяется.
- **Мобитц 2:** Интервал **PQ** стабилен, но выпадение **QRS** происходит внезапно и без предвестников. Это

критическое состояние, высокий риск перехода в полную блокаду.

3. III степень (Полная блокада)

Полная диссоциация: импульсы из предсердий вообще не доходят до желудочков.

- **Механизм:** Предсердия сокращаются в своем ритме (от синусового узла), а желудочки — в своем собственном, очень медленном темпе (идиовентрикулярный ритм, обычно 20–40 уд/мин).
- **Опасность:** Критическое падение сердечного выброса, приступы Морганьи-Адамса-Стокса (потеря сознания из-за гипоксии мозга), риск остановки сердца.

На кардиограмме ЭКГ это видно пропаданием в некоторых местах сегмента QRS и появление зубцов Р и Т с разной частотой относительно сегмента QRS, то есть предсердия бьются в своем темпе, а желудочки в своем, вследствие чего сердце бьется, но оно не качает кровь.



Рис. 3.5 АВ блокады (I, II, III)

Изначально планировалось сделать четвертый отдельный класс: Ишемия миокарда, но из-за пересечения признаков с инфарктом миокарда и довольно небольшим датасетом была объединена в класс ИМ для снижения межклассовой путаницы и ошибки функции потерь и ложных ответов.

Обоснование таково действия:

Ишемия миокарда (рис. 3.6) — это состояние, при котором миокард (сердечная мышца) получает недостаточно кислорода из-за сниженного кровотока, часто из-за сужения коронарных артерий. Это обратимое состояние, но также которое может прогрессировать в инфаркт, а Инфаркт миокарда (ИМ) (рис. 3.7) — это необратимая некроз (гибель) клеток миокарда из-за длительного отсутствия кровоснабжения, часто как следствие ишемии. И по

[illegible]

Рис. 3.7 Ишемия миокарда

В самом начале планировалось сделать вместо нейронной сети алгоритм классификации кардиограмм. И для этого нужно было узнать максимально все строения кардиограмм по данным осложнениям (рис. 3.6, 3.7). И по итогу мой алгоритм шагая по массиву точек отсматривал их амплитуды, производные, частоту изменения векторов, их период, их положение в кардиограмме. Из начально алгоритм распознает основные точки и зубцы Q, R, S, P, T (рис. 3.8). И потом основываясь на их координаты пытается классифицировать осложнение: на фибрилляцию предсердий алгоритм ищет многочисленные колебания, на ИМ ищет зазубренность на сегменте PQ или округленную вершину R или искаженную от нормы зубец T (он опущен или сильно поднят относительно изолинии и прилег к сегменту QRS), на АВ блокаду он ищет большие интервалы не равномерные между сегментами QRS и так же если не одно осложнение или норма не подходит то программа сразу выбирает это

осложнение (так как его алгоритмом тяжело классифицировать первую или третью стадию, а вторую довольно просто). Но в скором времени оказалось, что кардиограммы бывают очень сильно зашумленные, что алгоритм моментально ломался и при этом против этого не помогали никакие фильтрации кардиограмм. Вследствие чего пришлось от этой идеи отказаться.



Рис. 3.8 нахождение всех точек

Далее использовалась сверточная нейронная сеть (CNN) для выявления локальных паттернов. Однако точность составила лишь 70% (рис. 3.9), что недостаточно для медицинских задач. Основные проблемы:

1. Градиентное затухание и деградация: Углубление CNN ухудшало оптимизацию и вызывало переобучение.
2. Чувствительность к шуму: CNN была уязвима к артефактам ЭКГ. Лёгкие фильтры устраняли явный шум, но агрессивные фильтры удаляли ключевые паттерны (например, зазубренность Q-зубца).

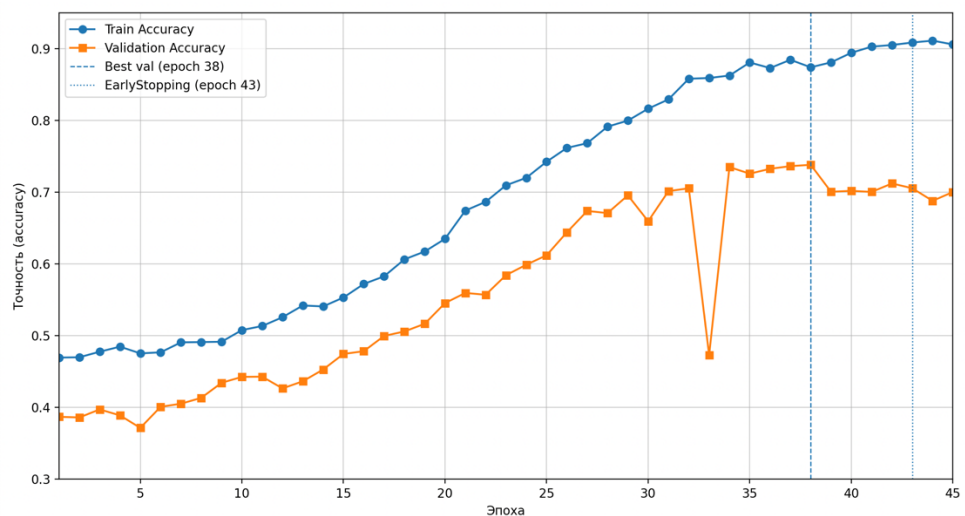


Рис. 3.9 точность обучения

Эксперименты с увеличением слоёв, размеров фильтров, числа нейронов, функций активации и потерь не привели к никакому улучшению. И, в связи с этим был осуществлен переход к другой архитектуре ResNet с остаточными связями, позволяющими устранить проблемы градиентного затухания и повысить качество обучения на зашумлённых клинических данных.

Архитектура нейронной сети

Входные данные

Модель принимает двухканальный тензор размером $[2, L]$, где $L=5000$ соответствует 10-секундному сигналу ЭКГ с частотой дискретизации 500 Гц. Два канала включают:

1. Исходный сигнал ЭКГ: Последовательность амплитуд, отражающая электрическую активность сердца.
2. Производная сигнала: вычисляется с использованием метода `np.diff` с предзаполнением первого значения ($x[0]$) для сохранения длины сигнала. Производная подчёркивает резкие изменения, такие как фронты комплекса QRS, что улучшает способность модели выделять ключевые морфологические особенности зубцов.

Математически производная определяется как:

$$derivative[i] = x[i] - x[i - 1]$$

Структура модели

Архитектура ResNet1D состоит из следующих компонентов, обеспечивающих глубокое извлечение признаков и точную классификацию:

1. Начальный сверточный блок (Stem):

Сверточный слой (`nn.Conv1d`) преобразует входной сигнал ($C_{in}=2$) в 64-канальное представление с ядром размера 9 и отступом (`padding`) 4, сохраняющим длину сигнала.

Слой пакетной нормализации (`nn.BatchNorm1d`) нормализует активации для ускорения и стабилизации обучения.

Функция активации ReLU

$$f(x) = \max(0, x)$$

вводит нелинейность, необходимую для моделирования сложных зависимостей.

2. Остаточные блоки (ResBlock):

Модель включает три остаточных блока, каждый из которых состоит из двух последовательных сверточных слоёв (`nn.Conv1d`) с ядром размера 9, нормализацией (`nn.BatchNorm1d`) и активацией ReLU.

Остаточные связи (`skip connections`) добавляют входной сигнал к выходу каждого блока:

$$y = x + F(x, \{W_i\})$$

где $F(x, \{W_i\})$ — преобразование, выполняемое сверточными слоями, а y — выход блока после активации ReLU.

Для захвата паттернов на разных временных масштабах применяются коэффициенты расширения (`dilation`) 1, 2 и 4, что позволяет модели анализировать как локальные, так и более протяжённые особенности сигнала.

3. Глобальный пулинг:

Слой адаптивного среднего пулинга (`nn.AdaptiveAvgPool1d`) преобразует выходное представление размером $[B, 64, L]$ в вектор $[B, 64]$, усредняя признаки по временной оси. Это обеспечивает инвариантность к длине сигнала и снижает вычислительную нагрузку.

4. Классификационная голова:

Слой `nn.Flatten` преобразует выход пулинга в одномерный вектор.

Два полносвязных слоя (`nn.Linear`) с промежуточной активацией ReLU и выбросом (`dropout, p=0.25`) выполняют финальную классификацию на C классов (в данном случае $C=4$).

Dropout случайным образом отключает 25% нейронов на этапе обучения, предотвращая переобучение.

Выходной слой

Выход модели представляет собой логиты для каждого класса, которые обрабатываются функцией потерь CrossEntropyLoss с весами классов и сглаживанием меток ($\epsilon=0.05$) для повышения устойчивости к переобучению.

Процесс обучения

Подготовка данных

Загрузка данных: Данные считываются из CSV-файла, содержащего метки классов в первом столбце и 5000 амплитудных значений для каждого образца. Для ускорения загрузки используется кэширование в формате .prz, что значительно сокращает время подготовки данных.

Разделение выборки: Применяется стратифицированное разделение (StratifiedShuffleSplit) с соотношением 80% для обучения и 20% для валидации, обеспечивая равномерное распределение классов.

Взвешивание классов: Для компенсации дисбаланса классов вычисляются веса:

$$w_i = \frac{N}{C \cdot n_i}$$

где N — общее количество обучающих примеров, C — число классов, n_i — количество примеров класса i . Эти веса интегрируются в функцию потерь для повышения внимания к менее представленным классам.

Оптимизация

Функция потерь: Используется CrossEntropyLoss с весами классов и сглаживанием меток ($\epsilon=0.05$) для смягчения влияния шума в метках.

Оптимизатор: применяется Adam с начальной скоростью обучения

$lr=10^{-4}$ и регуляризацией весов ($weight_decay=10^{-5}$ $weight_decay=10^{-5}$) для предотвращения переобучения.

L1-регуляризация: добавляется с коэффициентом $\lambda=10^{-6}$, чтобы ограничить сложность модели путём минимизации L1-нормы весов.

Ограничение градиентов: Метод `torch.nn.utils.clip_grad_norm_` с максимальной нормой 1.0 предотвращает взрывной рост градиентов, обеспечивая стабильность обучения.

Планировщик скорости обучения: `ReduceLROnPlateau` уменьшает скорость обучения в 2 раза, если валидационная ошибка не снижается в течение 2 эпох, с минимальным значением $lr=10^{-6}$.

Раннее прекращение

Механизм раннего прекращения (`EarlyStopping`) останавливает обучение, если валидационная ошибка не улучшается на величину $\delta=10^{-4}$ в течение 8 эпох. Это позволяет избежать переобучения и оптимизировать использование вычислительных ресурсов.

Цикл обучения

Обучение проводится в течение максимум 60 эпох с размером батча 16.

На каждой эпохе вычисляется средняя обучающая ошибка, а также валидационная ошибка и точность с помощью функции `evaluate`, которая выполняет проход по валидационной выборке в режиме без градиентов (`torch.no_grad`).

Модель с наименьшей валидационной ошибкой сохраняется в файл `best_resnet1d_two_channel.pth` для последующего использования.

И по итогу точность данной модели составляет 81%, на довольно зашумленных данных, снятых в реальных медицинских условиях. И в итоге получилась такая тенденция обучения (рис. 3.8), (рис. 3.9). На данном графике синяя линия – это обучающая выборка, а красная – это тестовая выборка. На схеме видно, что самый лучший результат зафиксирован на 50 эпохе обучения, прерванной `EarlyStopping` из-за не изменения качества обучения длительный промежуток времени. А ошибка обучения равна около 0.7. Что является довольно хорошим результатом для модели имеющая 4 класса.

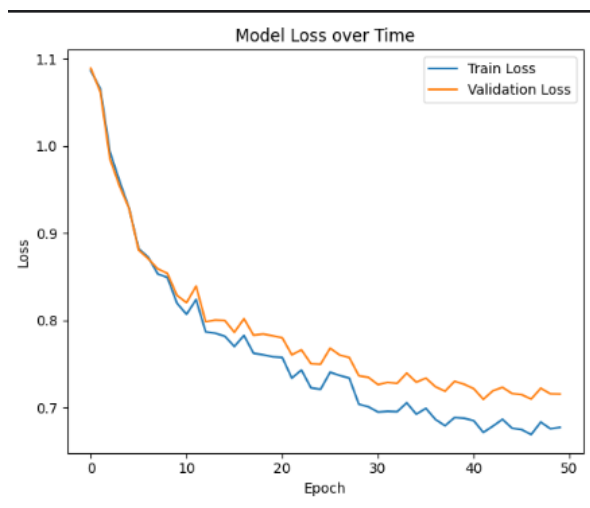


Рис. 3.8 ошибка обучения

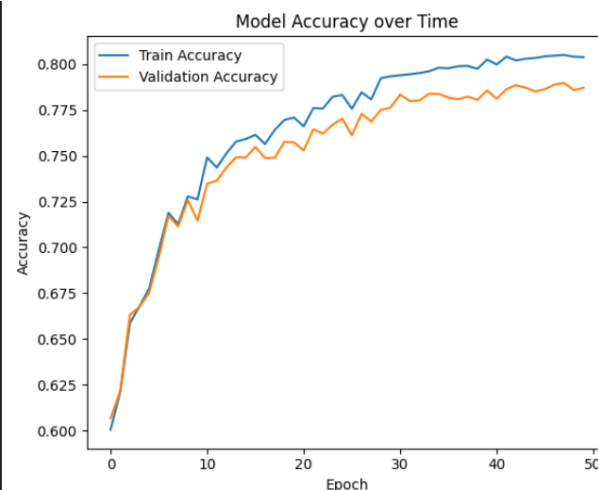


Рис. 3.9 точность обучения

Проблема с датасетом и ее решение:

Датасет с кардиограммами ЭКГ изначально имел одну большую проблему его данные были довольно сильно зашумленные и многие из них были сняты, когда пациент(наблюдаемый) делал какие-либо физические упражнения или при учащенном и не ровном дыхании. И по итогу получались кардиограммы как показано на рисунке ниже (рис. 3.10). Да, с одной точки зрения это даже и хорошо так как модель сможет лучше понять некоторые артефакты и более лучше обучить и на реальных тестах, когда человек находится под анестезией будет с ровным дыханием будет давать еще большую точность. Но уровень зашумленность приблизительно был 10% кардиограмм ЭКГ(при общем размере датасета в 50_000 кардиограмм, то это равно 5_000 кардиограмм) на столько высок, что даже визуально самому мне классифицировать кардиограмм было довольно сильно тяжело а в некоторых случаях даже и не понятно от чего именно такой вид аритмии.

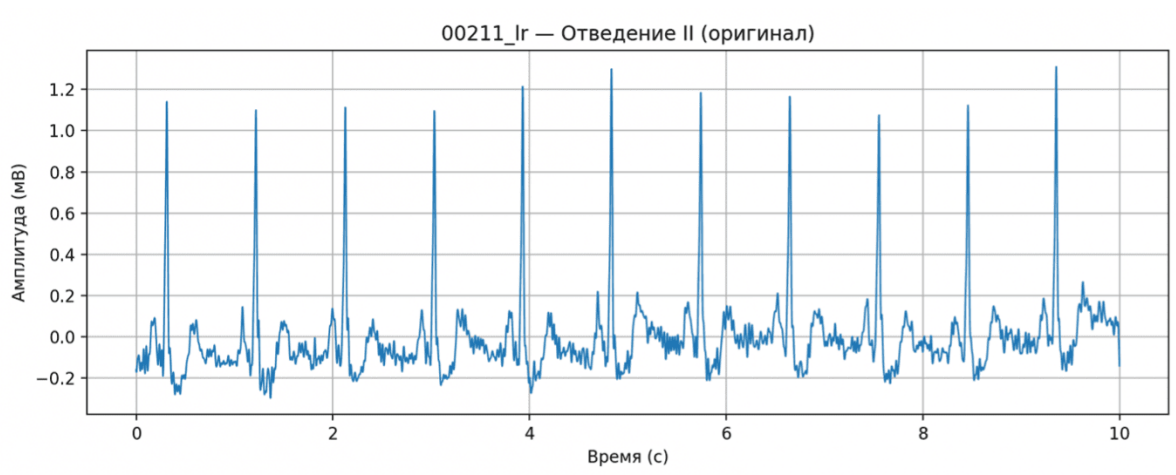


Рис. 3.10 зашумленная кардиограмма

И, в связи с этим пришлось фильтровать и изменять кардиограммы. И был реализован код для предотвращения этой проблемы:

И я реализовал две функции которые реализуют полосовую фильтрацию сигнала с использованием фильтра Баттерворта. Это классический метод цифровой обработки сигналов, который полезен для выделения определённого диапазона частот в ЭКГ-сигналах (от 0.3 Гц до 10 Гц, чтобы убрать низкочастотный шум от дыхания или высокочастотный от мышечных артефактов, сохраняя ключевые компоненты на подобии QRS-комплекса).

Аналоговый прототип фильтра Баттерворта для полосового фильтра основан на низкочастотном (lowpass) прототипе с передаточной функцией:

$$H(s) = \frac{1}{\sqrt{1 + \left(\frac{s}{\omega}\right)^{2n}}}$$

Где ω - частота среза а n – порядок(order).

И по итогу кардиограмма ЭКГ стали иметь отфильтрованный вид (рис. 3.11) без лишних шумов. И специально я оставил небольшой шум по нескольким причинам:

- 1) Для более глубокого обучения чтобы программа знала не только идеальные случаи, но, и чтобы при шумах могла тоже давать истинные предсказания
- 2) Из-за того, что данный фильтр при установке меньшей частоты некоторые главные мелкие детали исчезают. К примеру, было замечено, что

при использовании более сильной фильтрации был полностью сглажено зазубрение у подножья с левой стороны у комплекса QRS, что являлся главным признаком инфаркта миокарда.

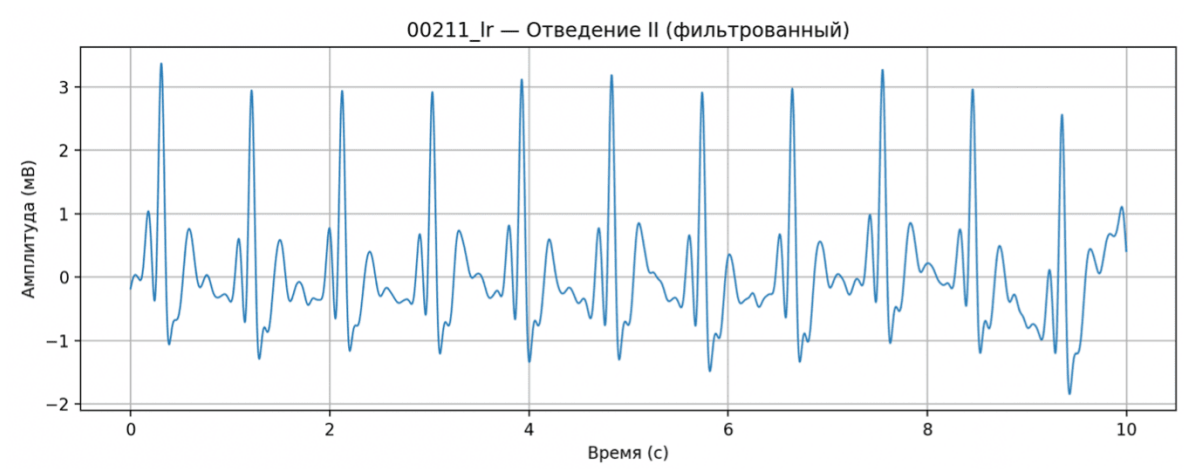


Рис. 3.11 отфильтрованная кардиограмма

Качество обучения модели.

На фотографии (рис. 3.12) изображено confusion matrix по данной модели. Где 0, 1, 2, 3 означают классы АВ блокады, фибрилляцию предсердий, инфаркт миокарда соответственно. И по данной матрице можно анализировать какой класс на сколько хорошо обучен, сколько дает правильных ответов, с какими классами путает данный. И если проанализировать ее, то получим следующее: модель распознает АВ блокады с точностью 87% относительно других классов. Больше всего этот класс путается с фибрилляцией предсердий. А Фибрилляция предсердий распознается с точностью 81%. Данный класс чаще всего теряется в АВ блокаде. И единственный класс, который вообще не ошибается в Норме. Инфаркт Миокарда распознается с точностью 70%. Он чаще всего путается с АВ блокадой. А норма обучилась с точностью 87%. Более всех путается с классом Инфаркт Миокарда. И отсюда можно сделать вывод: что лучше всех работает класс Норма. А это не мало важно, а даже очень хорошо так как лучше пусть программа ошибется с классом осложнения нежели перепутает его с Нормой и промолчит при каком-либо осложнении или наоборот, программа не будет срабатывать как осложнение на Норме. Также можно заметить, что два класса больше всего путаются на АВ блокаде: инфаркт миокарда и фибрилляция предсердий, а также третий класс Норма на 4% относительно самого класса этого. И этому есть логическое объяснение: АВ блокады, во-первых, имеют три степени:

- 1) Степень 1 – временной интервал увеличивается прохождения импульса через атриовентрикулярный узел. На кардиограммах это удлинение промежутка PQ.
- 2) Степень 2 – В любой момент времени с определенной частотой может не проходить возбуждение желудочек (деполяризация). Другими словами, может выпадать главный удар сердца. На кардиограммах это выпадение комплекса QRS.
- 3) Степень 3 – Это полностью сбитый ритм работы предсердий. Они совершенно идут вразнобой по отношению к желудочкам и при этом они возбуждаются с совершенно с разными частотами 45 ударов/минуту и 60 ударов/минуту соответственно. На кардиограммах это видно, что зубец P идет не в такт всей системе и в каждый момент времени появляется в разных местах и может накладываться на другие возбуждения (QRS, T). И это опасно тем, что это осложнение может перейти в фибрилляцию желудочек, а там если пропустить это осложнение более чем на 2 минуты спасти человек даже при реанимации шансы равны не более 15%.

И все эти три степени АВ блокад объединены в один общий класс АВ блокады. Это сделано по той причине, что кардиограмм с таким осложнением довольно мало. Буквально несколько тысяч. И при этом данный класс имеет и так является самым маленьким классом во всем датасете моем. Во-вторых, у меня в классах не равномерное количество данных и при этом это реализовано специально. Дело в том, что по-хорошему мне надо было все классы подводить к единой размерности равному самой маленькой размеру класса. И это самое действие могло привести к тому, что у меня нейронная сеть либо переобучится из-за маленького количества данных, либо недоучится по итогу. И было принято решение классы оставить исходной размерности (± 500 кардиограмм) и просто каждому классу дать начальный вес. И так как у меня АВ имеет наименьшую размерность и к тому же данный имеет наибольшую значимость я к формуле расчёта начального именно к этому классу добавил Bias (дополнительная переменная исходящая из формулы $y = k * x + b$, и являющаяся b).

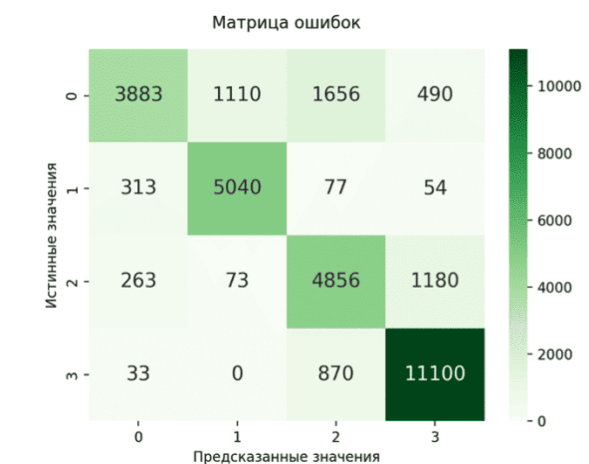


Рис. 3.12 Тепловая матрица

Ограничения

Данная модель довольно хорошо справляется со своей задачей классификацией осложнений по кардиограммам ЭКГ. При сборке данных были учтены многие исключения: на артефакты кариограмм при движении человека или его дыхании, данные которые я нашел в интернете уже имели некоторые кардиограммы с данными дефектами, также я полностью заменил данный с класса «норма» на свою запись ЭКГ, где специально каждые 5 минут записи совершал какое-либо движение руками, ногами или вставал со стула и изменял интенсивность дыхания, специально, чтобы модель знала не только идеальный вид кариограммы, но и реально который возможен случится (рис. 3.13), то есть, к примеру когда кардиограмма находится выше изолинии или ниже или много колебаний лишних, с которыми даже не справляется мой фильтр с которым я прохожу не помогает и остаются лишние волны или когда имеется какой-либо дефект с прибором или электродами или когда электроды не плотно прижаты к телу; с многими шумами у меня справляется фильтр, который сглаживает кардиограмму если шум умеренного характера.

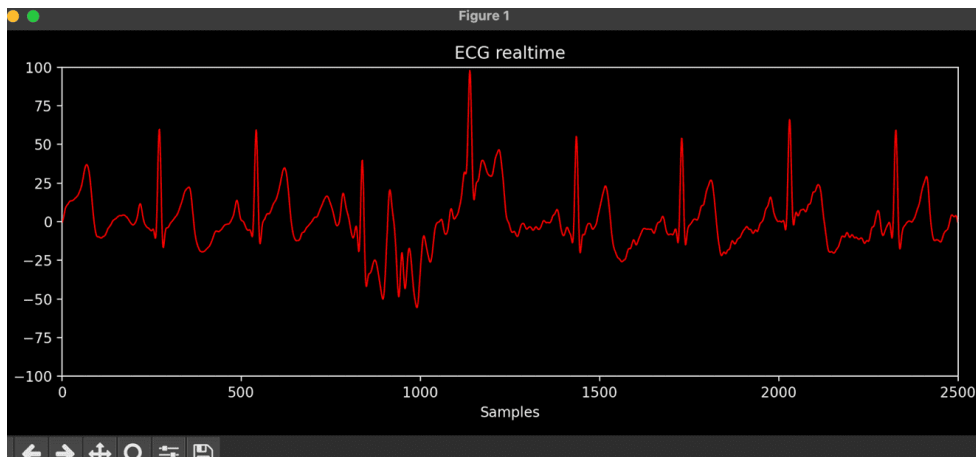


Рис. 3.13 Запись ЭКГ с шумом

Но моя модель может давать ложный ответ если на кардиограмме будет осложнение которое не знает моя модель, в этом случае модель выберет класс которые более схож с данным осложнением, к примеру если поступит на вход кардиограмма с аритмией тахикардия, то модель скажет что здесь норма, так как она довольно с ней схожа относительно других 3 классов; так же моя модель тут тоже не отслеживает временные ряды и не может видеть зависимость развития и тенденций изменения данных осложнений, я пробовал в датасетах учесть данный момент и давать модели на обучение кардиограммы с разных этапах развития данных аритмий, но так как у меня не такой большой датасет относительно (приблизительно 27_000 кардиограмм) , то в большей части модель принимала их как норму и также именно из-за этого модель может часто менять свой выбор классов при анализировании кардиограмм; также моя модель обучена для работы только со вторым отведением кардиограмм, но не смотря на это моя модель может корректно принимать на вход и анализировать первое отведение кардиограмм, так как оно практически не отличается от второго, но данная модель совершенно не умеет анализировать другие виды отведений, хотя во время операций чаще всего смотрят 12 отведений, я специально это не стал делать модель на все 12 отводов (хотя возможность была, мой датасет имел все 12 отводов, но я его отфильтровал и оставил только 2 и 1) так как мне пришлось бы добавлять еще один признак очень важный как “отведение” то модель бы не обучилась должным образом. Так как на некоторые осложнения у меня было не, так и много кардиограмм и при этом на некоторые осложнения на некоторые отводы x вообще не было. И все вновь упирается на размер данных. Я бы суммарно увеличил бы данные для модели, но ее бы сложность вместе с ней бы тоже. И по итогу у меня бы модель не обучилась бы или некоторые осложнения при некоторых отведения очень часто и сильно путала.

ОПИСАНИЕ СИСТЕМЫ НА ОСНОВЕ ПРАВИЛ ДЛЯ КЛАССИФИКАЦИИ ОСЛОЖНЕНИЙ ПО ДАННЫМ ИНТРАОПЕРАЦИОННЫХ ДАТЧИКОВ

Для выявления клинически значимых осложнений в условиях операционной среды разработана система на основе правил, реализованная в скрипте `rules_monitoring.py`. Эта система анализирует данные с интраоперационных мониторов и классифицирует состояния пациента, основываясь на заранее заданных пороговых значениях для различных физиологических параметров. Система является прототипом и предназначена для поддержки принятия решений в реальном времени.

Данная необычная вариация классификации и выявления осложнения выбрана не спроста. Изначально планировалось сделать ИИ, но в связи довольно редких признаков и информации в виде датасетов, имеющаяся только в закрытом доступе, было принято решение пойти многоуровневой алгоритмической программой.

Ниже представлено подробное описание её структуры, логики работы и ключевых компонентов.

Входные данные

Система принимает словарь с восемью физиологическими параметрами, измеряемыми в реальном времени (табл. 2.1)

SpO2 (%)	Сатурация кислорода в крови
MAP (мм рт. ст.)	Среднее артериальное давление
HR (уд/мин)	Частота сердечных сокращений
EtCO2 (мм рт. ст.)	Концентрация углекислого газа в конце выдоха
CVP (мм рт. ст.)	Центральное венозное давление
Urine (мл/ч)	Диурез (объём мочи в час).
Temp (°C)	Температура тела
PI (безразмерный)	Индекс перфузии

Табл. 2.1 датчики и их применение

Каждый параметр представлен числовым значением с плавающей точкой, что позволяет проводить сравнения с заданными пороговыми значениями.

Архитектура системы

Система основана на базе правил (rule-based approach), где каждое осложнение связано с набором вариантов (вариаций), представляющих комбинации условий для различных датчиков. Логика работы включает проверку условий, разрешение конфликтов и формирование объяснений.

База правил

База правил (RULES) представляет собой словарь, где ключи — это названия осложнений, а значения — списки вариантов. Каждый вариант — это список из восьми условий, каждое из которых задаётся тройкой:

(sensor_name, operation, threshold), (sensor_name, operation, threshold),

где:

sensor_name	название датчика (например, "SpO2").
operation	операция сравнения ("<", "<=", ">", ">=", "==").
threshold	пороговое значение для данного датчика

Пример условия для тяжёлой гипоксии:

("SpO2", "<=", 85).

Это означает, что значение SpO2 должно быть меньше или равно 85% для выполнения условия.

База правил охватывает 11 состояний:

1) Тяжёлая гипоксия

Клиническая суть: критичная нехватка кислорода в крови тканях ведущая к повреждению органов.

Признаки: сатурация кислорода стойко ниже 90%.

Критичность: высокая, прямая угроза остановки сердца и необратимого повреждения мозга.

2) Гемодинамический шок

Клиническая суть: неспособность системы кровообращения обеспечить достаточный приток крови к организму.

Признаки: резкое падение артериального давления и учащение пульса.

Критичность: высокий риск летального исхода.

3) Гипокоагуляция/массивное кровотечение

Клиническая суть: нарушение свертываемости крови, приводящим к опасным кровотечениям.

Признаки: патологические значения коагулограммы.

Критичность: высокий риск смерти от неконтролируемой кровопотери или внутримозгового кровоизлияния.

4) Вздутие яремной вены (JVD)

5) Анафилаксия/аллергическая реакция

6) Цианоз (изолированный)

7) Инфаркт миокарда (на основе косвенных признаков)

8) Фибрилляция предсердий (на основе косвенных признаков)

9) Атриовентрикулярная блокада I, II и III степени

10) Норма

Каждое состояние включает до 10 вариантов, описывающих различные комбинации показателей датчиков, характерные для данного осложнения. Пороговые значения выбраны на основе клинических правдоподобных диапазонов для интраоперационного мониторинга.

Приоритетная шкала

Для разрешения конфликтов, когда данные соответствуют нескольким осложнениям, используется список приоритетов (PRIORITY). Осложнения упорядочены по степени клинической тяжести:

PRIORI-

TY=["Hemodynamic shock","Anaphylaxis / Allergic reaction",...,"Normal"].

Система выбирает осложнение с наивысшим приоритетом в случае множественных совпадений.

Логика обработки

Основная функция `evaluate_reading` выполняет следующие шаги:

1. **Проверка условий:** Для каждого осложнения и его вариантов вызывается функция `eval_variant`, которая проверяет, выполняются ли все условия варианта: $\text{eval_variant}(\text{reading}, \text{variant}) = \bigwedge_{i=1}^8 \text{check_cond}(\text{reading}[\text{sensor}_i], \text{opi}, \text{threshold}_i). \text{eval}$. Здесь $\bigwedge$ обозначает логическое "И", а функция `check_cond` реализует сравнение значения датчика с порогом (например, `value < threshold` `value < threshold`).
2. **Сбор совпадений:** Все подходящие осложнения и их варианты сохраняются в списки `matched` и `matched_details`.
3. **Разрешение конфликтов:** Если найдено несколько совпадений, выбирается осложнение с наивысшим приоритетом из списка `PRIORITY`.
4. **Обработка отсутствия совпадений:** Если ни один вариант не подходит, система проверяет, соответствуют ли данные нормальному состоянию. В противном случае возвращается "No rule matched" с рекомендацией провести более детальный анализ.
5. **Формирование объяснения:** Функция `_build_explanation` создаёт читаемое описание, выделяя три датчика с наибольшим отклонением от нормы. Отклонение вычисляется как: $\text{deviation} = |\text{value} - \text{midpoint}| \text{std_dev}$, $\text{deviation} = \text{std_dev} |\text{value} - \text{midpoint}|$, где `midpoint` — среднее нормальное значение, а `std_dev` — стандартное отклонение для данного датчика.

Выходные данные

Функция `evaluate_reading` возвращает словарь с полями:

- `diagnosis`: Название выявленного осложнения (или "Normal", или "No rule matched").
- `matched_variant`: Список индексов подходящих вариантов.
- `explanation`: Краткое описание ключевых датчиков, способствующих диагнозу.
- `conflicts`: Список конфликтующих осложнений (если есть).

Также данная программа является проверкой предсказания нейронных сетей. Если какое-либо предсказание близко с другим по вероятности, то данный алгоритм смотрит и говорит окончательные вердикт. Это грубо говоря вторичная проверка чтобы не давать врачам ложных показаний. Также это работает и в обратную сторону. Если алгоритм нашел какое-либо осложнение, которое должно предсказывать нейронная сеть то он ей говорит чтобы она на определенное осложнение понизило пороговое значение определенного осложнения для предсказания.

ОБЩАЯ РАБОТА ПРОГРАММЫ

Используемые библиотеки

1. Collections - Используется для создания кольцевого буфера (deque) — хранит последние значения ЭКГ-сигнала с ограниченным размером.
2. cv2 (opencv-python) - Захват видео с веб-камеры, детекция лиц (Haar Cascade), обработка кадров, отрисовка текста, наложение изображений и графиков на экран монитора.
3. matplotlib.pyplot - Построение и сохранение графика ЭКГ-сигнала в файл PNG
4. numpy (np) - Основная работа с массивами: обработка сигналов ЭКГ, нормализация, фильтрация, математические операции, приведение типов данных.
5. os - работа с файловой системой: создание и удаление папок (card_for_code), установка переменных окружения (kmp_duplicate_lib_ok, omp_num_threads).
6. PIL (Pillow — Image) - Преобразование массива OpenCV в объект PIL Image для дальнейшей обработки через torchvision.transforms.
7. platform - для кросс-платформенной совместимости со звуком и портами.
8. psutil - мониторинг памяти и процессора для предотвращения переполнения оперативной памяти и классификации ошибок при их возникновении.
9. random – в случае непредвиденной ошибки с Arduino или моделями, чтобы можно было перейти на симуляцию программы или данных.
10. scipy.signal - Фильтрация ЭКГ-сигнала (полосовой фильтр Баттерворта: butter, filtfilt), ресемплинг сигнала (resample).
11. serial (и serial.tools.list_ports) - Подключение к порту, чтение данных с устройства (Arduino), поиск доступных COM-портов.
12. shutil - Полное удаление папки card_for_code перед созданием новых файлов (shutil.rmtree).
13. sklearn – Используется для обучения на линейной регрессии мини моделей (несколько признаков 3-5 для классификации 3-4 классов на ограниченных данных).
14. sounddevice (sd) - Воспроизведение тревожного звукового сигнала (медицинский "beep") в кросс-платформенном режиме.
15. Struct - Распаковка бинарных данных из последовательного порта (2 байта → uint16 значение ЭКГ-сигнала).

- 16.threading - Запуск фоновых потоков: чтение данных из serial-порта, воспроизведение звука тревоги.
- 17.time - Задержки (sleep), измерение времени, таймеры для периодического выполнения диагностики.
- 18.Torch - Основной фреймворк для запуска нейронных сетей (модели для ЭКГ и анализа лица).
- 19.torch.nn (nn) - Определение архитектуры нейросетей: свёрточные слои, BatchNorm, ReLU, Linear, Dropout, Sequential и др.
- 20.torchvision.transforms - Подготовка изображений для CNN: изменение размера (Resize), преобразование в тензор (ToTensor).

Алгоритм программы

Программа работает следующим образом: В начале идет загрузка всех моделей в программу, далее по вторичному потоку считывается кардиограмма ЭКГ с платы Arduino, где порт нужный находится автоматически. Она считывается постоянно бесконечным циклом, так как она не привязана к программе. Далее эта кардиограмма заполняет очередь и отправляется на фильтрацию, чтобы убрать весь шум и считывается производная каждой точки сигнала. И за тем я в программе делаю визуализацию через график данной кардиограммы и сохраняю как изображение с расширением .png и затем отправляю данные массив точек в мою модель. После этого камера с нейросетью сегментирует мое лицо обрезая фон и отправляет ее тоже модель. С датчиков я не могу брать данные так как многие из них требуют медицинских условий для их эксплуатации и, следовательно, в программе они сами генерируют свои значение рандомным образом и все эти значения отправляются в мой алгоритм классификации.

После получения всех данных и проход их через ML модели и алгоритмы я получаю данные о состоянии пациента по каждому параметру здоровья в виде вероятностей каждого осложнения. И далее я все эти полученные результаты отправляю в следующий алгоритм, который будет просматривать все данные и решать какое по итогу осложнение у человека есть на данный момент (Здесь я использую именно алгоритм, а не обученную ML модель так как для обучения нужны данные которых нет для обучения, так как очень редкий набор признаков. Была мысль собрать самому датасет, но в связи с тем что модель имеет вид мультиклассной классификации и имеет 9 классов и 14 числовых признаков (от 0.00 до 1.00 – диапазон 100 единиц), данных требуется много и так как я не имею медицинского образования такое количество данных самому сделать было бы ошибкой так как из-за этого довольно сильно может упасть качество предсказания). Изначально алгоритм ищет одно осложнение, которое с большей вероятностью есть у каждого параметра. Но кардиограммы имеют дополнительную проверку, если разница вероятностей двух наиболее возможных осложнений больше десяти единиц, то берется с максимальной вероятностью, а если нет, то данные две вероятности отправляются в дополнительную функцию состоящая из трех этапах, где однозначно выявляется осложнение одно, главное.

Изначально эти две вероятности проходят через метод, основанный на медианах треугольника (рис. 4.1). Три вершины треугольника имеют X Y X имеют значения осложнений взаимосвязанных соответственно. К примеру,

гипоксия и цианоз, и инфаркт миокарда образуют треугольник. Они могут перепроверяться взаимно. И строится медиана треугольника. И в зависимости от осложнения, которое я выявляю умножается на определенный коэффициент. И далее я нахожу разницу двух значений длин медианы. И если у меня это значение больше 10%, то я беру осложнение за основное с меньшей медианой и вывожу его как значения как осложнение более вероятное. Если нет, то перехожу на второй алгоритм перепроверки, основанный на линейной регрессии.

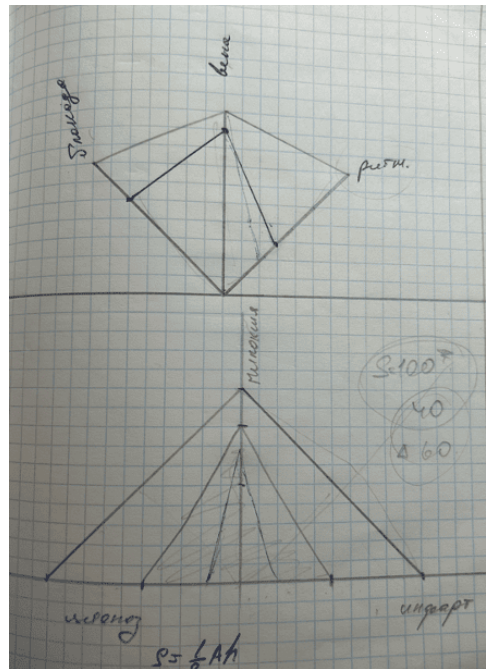


Рис. 4.1 метод треугольника

Во втором алгоритме я обучаю 4 линейных регрессий для каждого класса осложнения ЭКГ, где признаками являются значения других параметров контроля осложнений и значения некоторых датчиков. Все комбинации зависят от определенного класса осложнения ЭКГ: класс Инфаркт миокарда имеет признаки вероятность ИМ от первоначальной модели, значение артериального давления и ЧСС и вероятность цианоза и вздутие яремной вены тоже от CNN модели; ФП имеет признаки вероятность ФП от первоначальной модели, значения нерегулярности пульса и уровень сатурации уровня кислорода в крови (SpO_2) и вероятность цианоза и вздутие яремной вены тоже от CNN модели; АВ блокада имеет признаки вероятность АВ блокады от первоначальной модели, значение ЧСС и артериального давления и вероятность цианоза и вздутие яремной вены тоже от CNN модели. И далее я просто сравниваю значение мини моделей этих и если дельта этих вероятностей

больше 0.1, то я возвращаю с максимальной вероятностью осложнение, если нет, то перехожу к третьей ступени.

Третий алгоритм перепроверки довольно прост и однозначен. Я просто возвращаю осложнение с максимальной вероятностью и все.

Далее продолжает работать основной алгоритм. Он берет значение алгоритма классификации осложнений по датчикам и сравнивает его с значением осложнений из списка эталонного при данном осложнении ЭКГ которые могут сопутствовать и вследствие чего у меня увеличивается или уменьшается вероятность данного осложнения. И далее, если осложнение с самой большой вероятностью больше 50%, то я его кладу в список найденных осложнений.

Значение результатов модели классификации осложнений по визуальным признакам я беру два с максимальным значением. И далее смотрю, если они подтвердились и по датчикам, то я их оставляю еже ли нет, то программа считает, что это ошиблась нейронная сеть и тогда я просто тем самым отсеиваю список возможных осложнений. Если в результате все осложнения не прошли эту проверку или они с вероятностью меньше 50%, то я считаю, что их нет, еже ли остался один вариант, то я кладу его в массив найденных осложнений, но если осталось два варианта, то я смотрю осложнения или значения датчиков которые могут сопутствовать при данном визуальном осложнении и из-за чего у меня вероятность осложнения увеличивается или уменьшается соответственно, и потом я беру осложнение просто в максимальной вероятностью

Датчики никак не перепроверяются и не фильтруются по результату классификации, так как здесь работает алгоритм, а не ML модель. И программа просто берет все найденные осложнения по датчикам и дополняет массив осложнений.

Если массив осложнений, найденных по окончании итерации моего перепроверяющего алгоритма и вывода итогового результат пуст, то программа считает, что пациент здоров и не имеет каких-либо осложнений.

Если программа считает, что пациент здоров и не имеет каких-либо осложнений (массив осложнений пуст), то интерфейс выглядит следующим образом: экран имеет зеленый цвет, с вероятностью 0.0 и пустым списком осложнений (рис. 4.2)

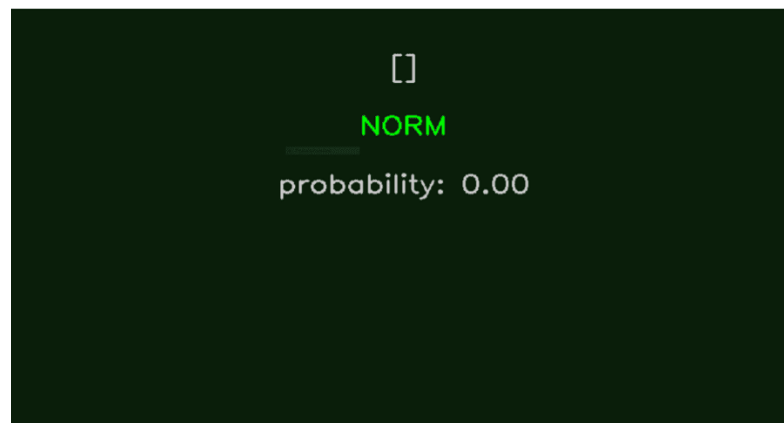


Рис. 4.2 Норма

Если общая максимальная вероятность больше 0.35, но меньше 0.6, то интерфейсное окно меняется: она меняет свой цвет на темно-зеленый, надпись меняется на “Warning”, в скобках появляются осложнения, в строке probability пишется максимальная вероятность осложнения, и внизу появляются в зависимости от типа осложнения кардиограмма ЭКГ и/или топ-3 датчика играющие наибольшую роль в этом осложнении и численное значение на сколько значение датчика больше/меньше нормы (рис. 4.3).

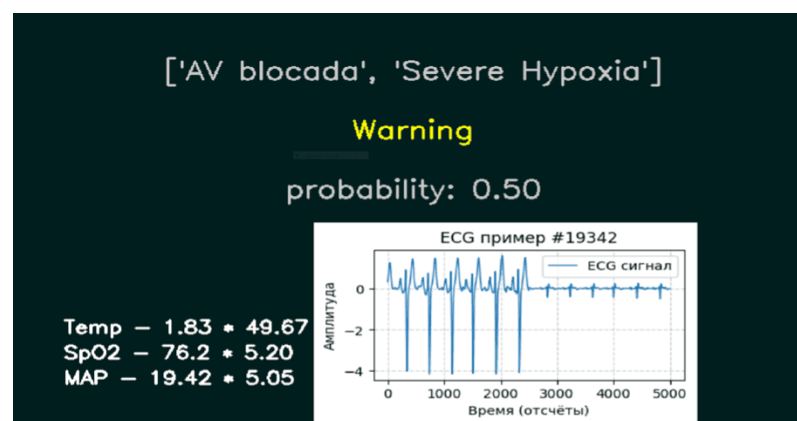


Рис. 4.3 Спорная ситуация

Но если общая вероятность больше 0.6, то фон перекрашивается в красный цвет, и он начинает плавно мигать, перебирая все оттенки темно-красного цвета от самого темного до самого яркого. Также это все сопровождается звуковым сопровождением сигнала тревоги. И меняется надпись на Dangerous красного цвета. А все остальное по алгоритму схоже с вышеописанным интерфейсом Warning. (рис. 4.4)

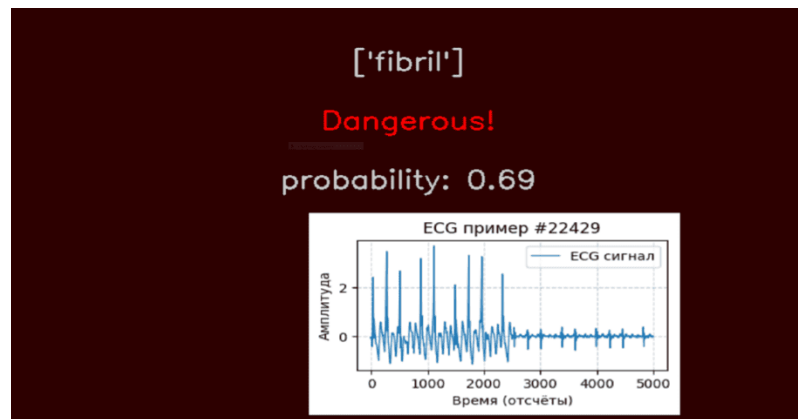


Рис. 4.4 Точно определено осложнение

Вероятность итоговая для программы – это максимальная вероятность из всего массива осложнений. (вероятность датчиков при успешном прохождении алгоритма классификации имеет значение константное равное 0.5)

РЕЗУЛЬТАТ И ТЕСТИРОВАНИЕ

После реализации моего алгоритма, мне предстояло провести проверку ее работы не на данных при обучении или синтезированных, а на реальных: то есть протестировать ее на себе.

Для этого я взял данные с датчиков Pulse Sensor на базе Arduino, ECG тоже на базе Arduino. Видео кадры я взял с видео с моим лицом. SpO₂ я связал с smart часов. А все остальные датчики были синтезированы.

Датчик Pulse Sensor подключил к плате Arduino (рис. 5.1) к пинам 0; 5V; GND. Далее был реализован код на C++ для считывания данных с этого датчика и фильтрации значений, так как они оказались довольно сильно зашумленные. Далее, когда получил на выходе массив значений мне нужно было отобразить его на графике для лучшего восприятия и заключения вывода. По ходу реализации оказалось, что в Arduino IDE использование встроенного окна для отображения графика, очень неудобным. И из-за чего был вынужден перейти на Python и там через библиотеку matplotlib отобразить этот график.

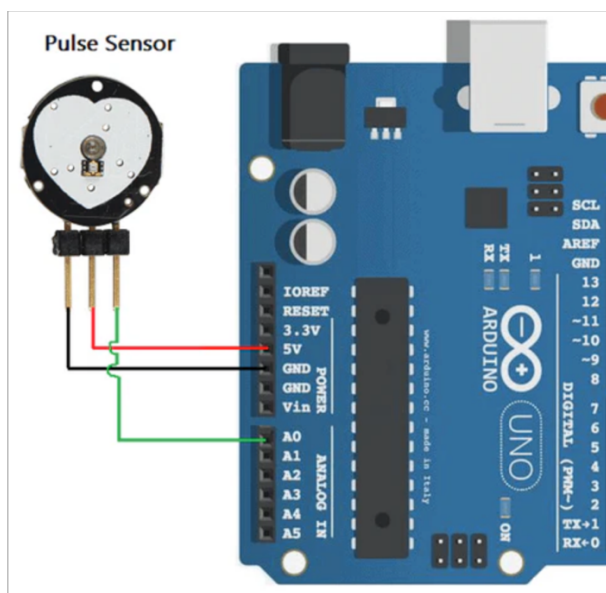


Рис. 5.1 Схема с датчиком пульса

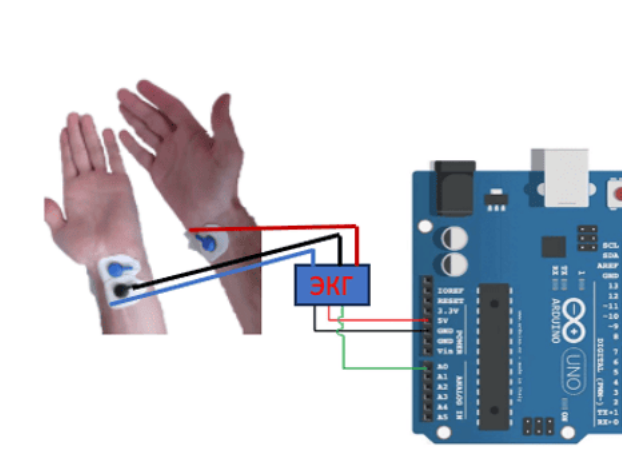


Рис. 5.2 Схема с датчиком ЭКГ

Итоговая подпрограмма работает следующим образом: я беру датчик, к его сенсору прижимаю указательный палец, и далее с определенным периодом Arduino берет данные с него и выводит. Далее я передаю его на Python и тот выводит итоговый график пульса (рис. 5.3)

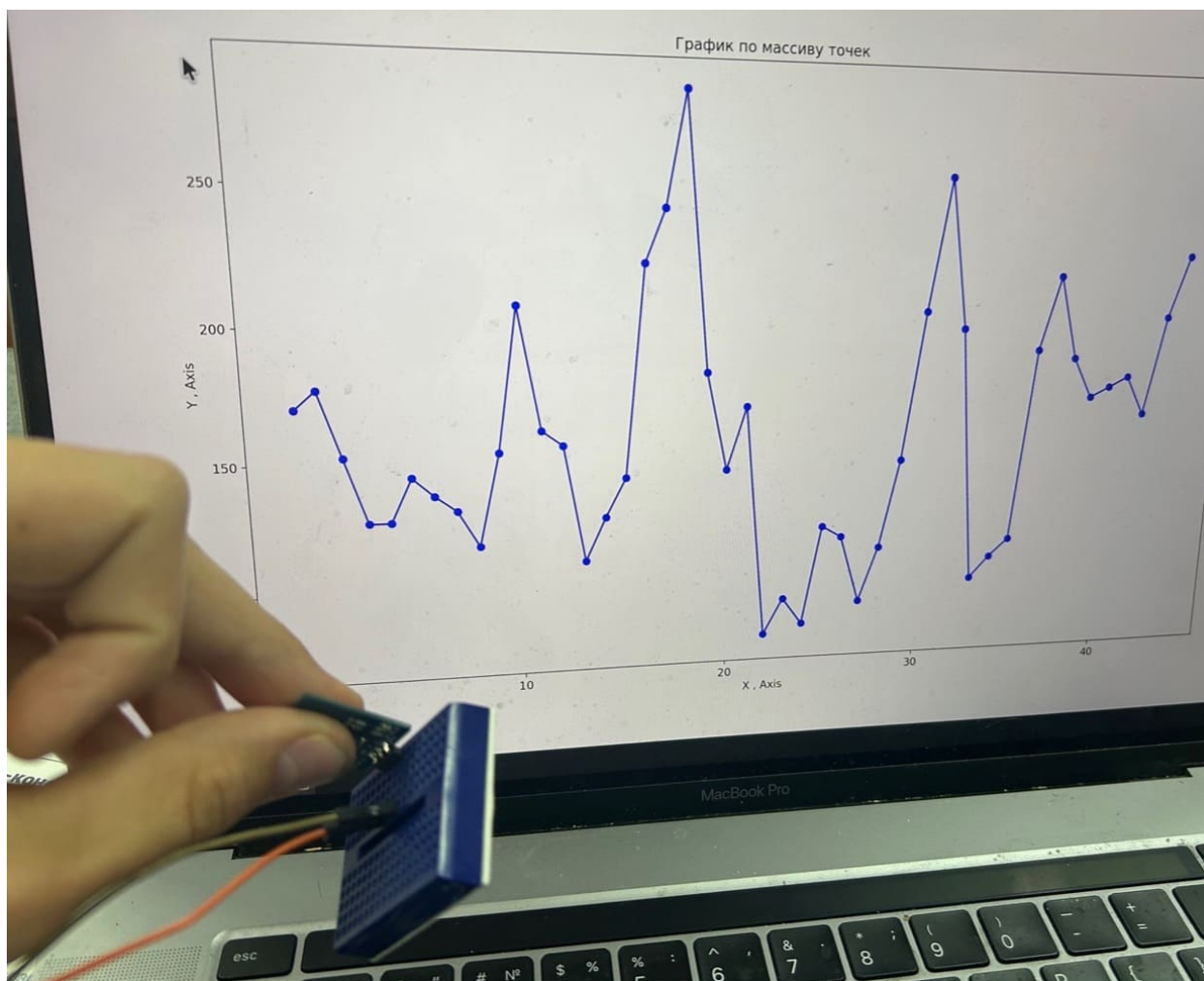


Рис. 5.3 Вывод данных

Далее ECG датчик подключил к плате Arduino (рис. 5.2). Далее был реализован код на C++ для считывания данных с этого датчика (в этом случае не реализовывал фильтрацию данных так как все данные передаются основной программе, где и был реализован алгоритм фильтрации Баттерворта (см. выше)).

Электроды я прикрепил как первое отведение (правая и левая рука), так как оказалось, что провода у электродов довольно короткие, и они не такие мощные, чтобы дотянуться и зафиксировать нужной амплитуды электрический импульс на ноге для второго отведения (левая рука, правая нога). Так же я выбрал именно этот отвод как альтернативный, потому что визуально при норме и некоторых других случаях он более всего похож на 2 отвод. (рис. 5.4). Так же здесь я использовал не два электрода(синие), но и еще один черный. Это сделано для фильтрации электрический импульсов принимаемые датчиком ЭКГ. Так как без него появляются очень большие шумы, в которых очень сложно разглядеть даже сам комплекс QRS.

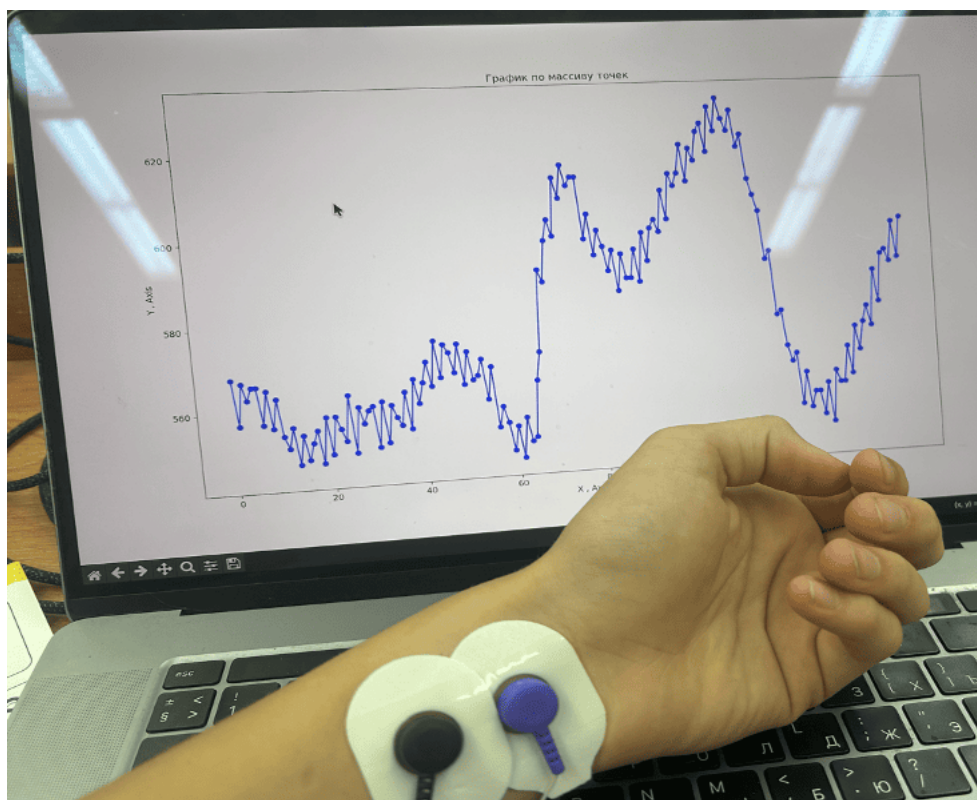


Рис. 5.4 Снятие ЭКГ

Но это сильно не спасло ситуацию: электроды очень плохо читали электрический импульс и могли вследствие чего возникать некоторые дефекты, к примеру подниматься точка Q или наоборот очень слабый сигнал идти, который вместо синусового ритма больше похож на нитевидный или он очень сильно чувствительный был к любому моему движению тела или какой-либо конечности. И тогда было принято решение переместить электроды ближе к сердцу на ключицы правую и левую (рис. 5.6) как показано на рисунке ниже на точки RA и LA соответственно а черный поставить на нижнее правое ребро в RL, это для того чтобы электрический импульс был более мощным и при этом это места где минимальное количество мышц расположено которые мешают прохождению сигнала и при их движении создают лишние шумы как на руке где ранее располагались электроды. Также для улучшения сигнала я стал протирать место крепления электрода спиртом/антисептиком так как он более лучше проводит электрический импульс смачивая кожу и обезжиривая ее.

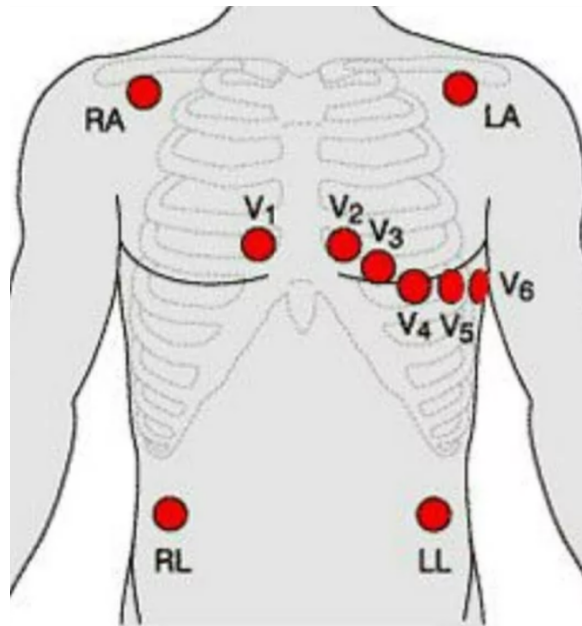


Рис. 5.6 схема распределение электродов

И по итогу у меня довольно сильно улучшилась кардиограмма (рис. 5.7): хорошо виден комплекс QRS и зубцы Т и Р и U. Минимальное количество шумов без применения сильных фильтров а лишь легких который удаляют на кардиограмме легкий шум от дыхания и колебания мышц. На кардиограмме это изображено в виде высокочастотных колебаний с минимальной амплитудой. Также для улучшения модели распознавать кардиограммы ЭКГ было принято решение дописать в Датасет 1_000 кардиограмм, снятые с меня, в размерности 2_500 точек, в них входит приблизительно 8-10 сжатий сердца то есть 8-10 комплексов QRS. Именно это решение даст более обширную картину для нейронной сети так как когда всего лишь 3 импульса сердца то можно упустить какую-либо патологию или наоборот принять шум или какое-либо другое явление за патологию ошибочно а это решение не даст этому случиться. Также специально при записи на датасет кардиограммы (всего было отснято 6_800 кардиограмм длинной по 2_500 точек) с некоторыми дефектами, к примеру при интенсивном дыхании или при “случайном” движении микросхемы (из-за этого кардиограмма меняется она подымается или опускает относительно изолинии по оси Y которая равна 0)

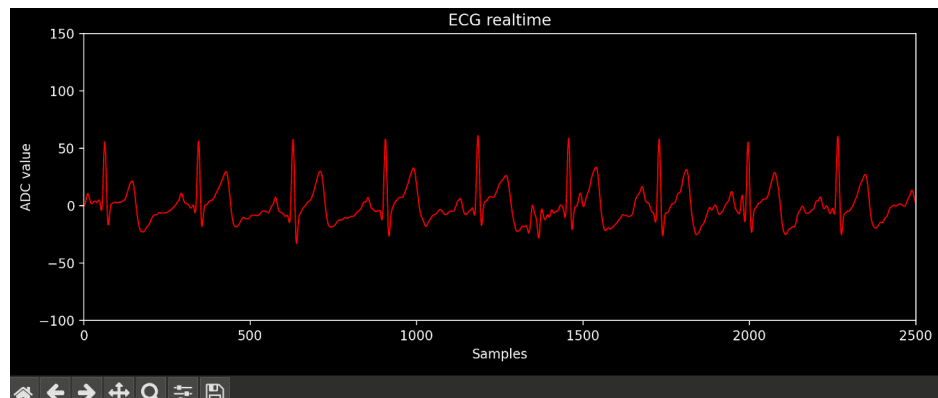


Рис. 5.7 ЭКГ кардиограмма

Все остальные данные с датчиков я взял как фиксированные данные при нормальном состоянии человека которые хранятся в массиве в некотором диапазоне и которые выбираются совершенно случайным способом. Далее я все эти данные положил в свою систему и запустил ее. По итогу моя программа по мере всей ее работы выводила мое состояние как нормальное.

И по итогу эксперимента, когда я загрузил все эти мои данные программе, полученный результат от программы моей подтвердил отсутствие каких-либо осложнений выдав класс «Норму» и это свидетельствует о нормальном моем состоянии здоровья и, следовательно, о корректной работе моей автономной системы (рис. 5.8).



рис. 5.8 итоговый вывод программы

Все кода в репозитории от моего проекта в GitHub, который доступен по данной ссылке:

Скопцов В. Health-AI // github.com. URL: <https://github.com/skoptsovvasilii/Health-AI>

В нем содержатся следующие файлы:

- 1) отдельные алгоритмы
- 2) обучающие программы модели
- 3) вспомогательные алгоритмы
- 4) тестовые алгоритмы
- 5) прототипные алгоритмы
- 6) алгоритмы сбора информации с датчиков
- 7) кода архитектур модель
- 8) описание
- 9) итоговый код

ВЫВОД

В рамках проектной работы была разработана автономная мультимодальная система Health-AI для мониторинга состояния пациентов в реальном времени на основе данных видеопотока, электрокардиограммы (ЭКГ) второго отвода и показаний восьми датчиков. Система включает нейронные сети для анализа визуальных осложнений (цианоз, вздутие яремной вены, аллергическая реакция и др.) и аномалий ЭКГ (фибрилляция предсердий, инфаркт миокарда, атриовентрикулярная блокада), а также алгоритм на основе правил для классификации состояний по данным датчиков.

Значимость проекта заключается в создании прототипа, который объединяет три источника для данных комплексного мониторинга, чего не обеспечивают большинство существующих устройств, работающих с одиночными датчиками. Такой подход позволяет оперативно выявлять осложнения, требующие немедленного вмешательства, что особенно важно при длительных операциях. Разработанный интерфейс, сигнализирующий о критических состояниях (например, мигание экрана красным), делает систему удобной для использования в реальных условиях.

Таким образом, разработанный прототип демонстрирует значительный потенциал для облегчения работы врачей, повышения качества медицинской помощи и минимизации врачебных ошибок. Проект открывает перспективы для дальнейших исследований в области применения искусственного интеллекта в операционной медицине, что делает его актуальным вкладом в развитие здравоохранения.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. **Nayak S., Mallick S.** Transfer Learning For Pytorch Image Classification. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with code, & tutorials. May 21, 2019. URL: <https://learnopencv.com/image-classification-using-transfer-learning-in-pytorch/>.
2. **Bentrevett.** GitHub - bentrevett/pytorch-image-classification: Tutorials on how to implement a few key architectures for image classification using PyTorch and TorchVision. July 21, 2018. URL: <https://github.com/bentrevett/pytorch-image-classification>.
3. **Propropogs.ru.** Остаточные нейронные сети (Residual Networks — ResNet). January 1, 2025. URL: https://proproprogs.ru/nn_pytorch/pytorch-ostatochnye-neyronnye-seti-residual-networks--resnet.
4. **ChengChengChen1.** Skin-Disease-Classification-CNN/resnet50.ipynb at main. December 5, 2024. URL: <https://github.com/ChengChengChen1/Skin-Disease-Classification-CNN/blob/main/resnet50.ipynb>.
5. **Devavratatripathy.** ECG dataset. March 24, 2021. URL: <https://www.kaggle.com/datasets/devavratatripathy/ecg-dataset>.
6. **Invitro.ru.** Электрокардиография – основные и дополнительные отведения. August 25, 2025. URL: <https://www.invitro.ru/library/instrumentalnaya-diagnostika/57450/>.
7. **Жданова И.В., Зуева Т.В., Коряков А.И. и др.** Электрофизиологические основы электрокардиографии. Екатеринбург: УГМУ, 2019. 37 с. URL: https://elib.usma.ru/bitstream/usma/1578/1/UMK_2019_027.pdf.
8. **Тимофеев Е.В., Булавко Я.Э., Реева С.В. и др.** Электрокардиографические синдромы в клинике внутренних болезней. Часть I. Основы ЭКГ-диагностики. Признаки гипертрофии камер сердца. СПбГПМУ, 2024. URL: <https://gpmu.org/userfiles/file/кафедра%20пропедевтики%20внутр%20болезней/metodichky/ЭКГ%201%20chast.pdf?ysclid=mgr2tj0m5z261599796>.

9. **Зафираки В.К.** Аритмия — симптомы и лечение. May 16, 2019. URL: <https://probolezny.ru/aritmia/>.
10. **CMRT.ru.** Синюшность кожи причины, способы диагностики и лечения. March 24, 2021. URL: <https://cmrt.ru/simptomy/sinyushnost-kozhi/>.
11. **Сердце под присмотром алгоритмов :** Искусственный интеллект помогает диагностировать аритмию и другие заболевания — Текст : электронный // Коммерсантъ : [сайт]. — 2023. — 13 сентября. — URL: <https://www.kommersant.ru/doc/6211993>.
12. **Почему искусственный интеллект никогда не заменит врачей** — Текст : электронный // SberMedAI : [сайт]. — URL: <https://sbermed.ai/pochemu-iskusstvennyj-intellekt-nikogda-ne-zamenit-vrachej-2#почему-ии-не-заменит-врача>.
13. **Набухание яремных вен: симптомы, причины и диагностика** — Текст : электронный // Apollo Hospitals : [сайт]. — URL: <https://www.apollohospitals.com/ru/symptoms/jugular-vein-distention>
14. **Симптоматика, диагностика и профилактика лейкемии** — Текст : электронный // Propedevtika WordPress : [сайт]. — URL: <https://propedevtika.wordpress.com/wp-content/uploads/2019/10/d0a1d0b8d0bcd0bfd182d0bed0bcd0b0d182d0bed0bbd0bed0b3d0b8d18f-d0add09ad093-d0b4d0b8d0b0d0b3d0bdd0bed181d182d0b8d0bad0b0-d0b8-d0bfd180d0b8d0bdd186d0b8d0bfd18b-d0bbd0b5d187d0b5d0bdd0b8d18f.pdf>
15. **Учебное пособие Н.И. Волков и др., ЭКГ** — Текст : электронный // ASMU.RU : [сайт]. — URL: https://do.asmu.ru/pluginfile.php/1145330/mod_folder/content/0/%D0%AD%D0%9A%D0%93.%20%D1%83%D1%87%D0%B5%D0%B1-%D0%BD%D0%BE%D0%B5%20%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B8%D0%B5%20%D0%9D.%D0%98.%20%D0%92%D0%BE%D0%BB%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%B8%20%D0%B4%D1%80.%20%282022%29.pdf?ysclid=mlpa6lsxkx32674577

16. **Документация Python Standard Library** — Текст : электронный // docs.python.org : [сайт]. — URL: <https://docs.python.org/3/library/index.html>
17. **Уроки Arduino Reference на сайте AlexGyver** — Текст : электронный // AlexGyver.ru : [сайт]. — URL: <https://alexgyver.ru/lessons/arduino-reference/?ysclid=mlpa8yzol3441523815>
18. **Принцип последовательного сокращения кардиомиоцитов** — Текст : электронный // ПИРОГОВ-ЦЕНТР : [сайт]. — URL: <https://www.pirogov-center.ru/specialist/recommended/princip-poocherednogo-sokraschenia-kardiomiocitov/?ysclid=mlpaasyurm975592833>