

Санкт–Петербургский государственный университет

*ХОХЛОВА Снежана Романовна*

Выпускная квалификационная работа

*Построение алгоритма обнаружения пневмонии  
на рентгенограмме органов грудной клетки*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2016 «Прикладная математика, фундаментальная информатика и программирование»

Профиль «Процессы управления и высокопроизводительные  
вычислительные системы»

Научный руководитель:

доцент кафедры теории систем управления электрофизической  
аппаратурой, к.ф. - м.н. Головкина Анна Геннадьевна

Рецензент:

профессор кафедры теории управления,  
д.ф. - м.н. Котина Елена Дмитриевна

Санкт-Петербург

2020 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Постановка задачи</b> . . . . .	5
<b>Глава 1. Обзор литературы</b> . . . . .	6
1.1. Основные определения . . . . .	6
1.2. Обзор существующих решений задачи классификации . . . . .	12
1.3. Обзор существующих решений задачи детекции . . . . .	15
1.4. Доступные программные средства . . . . .	18
<b>Глава 2. Архитектура и особенности реализации</b> . . . . .	19
2.1. Обработка данных . . . . .	20
2.2. Архитектура модели классификации . . . . .	23
2.3. Архитектура модели детекции . . . . .	26
<b>Глава 3. Полученные результаты</b> . . . . .	28
3.1. Модель классификации . . . . .	28
3.2. Модель детекции . . . . .	32
<b>Заключение</b> . . . . .	35
<b>Список литературы</b> . . . . .	36

## Введение

В настоящее время основным методом исследования и диагностики заболеваний лёгких является рентгенограмма грудной клетки, по которой опытный специалист может выделить различить около десятка патологий. В настоящей работе проводится исследование возможности автоматической диагностики наиболее важной из них – пневмонии. Пневмония является очень опасным заболеванием. В год диагноз пневмония ставится более 17 млн человек. Особенную опасность она представляет для детей и пожилых людей. Она является причиной смертности 18 % детей до 5 лет во всем мире [1]. Также на сегодняшний день пневмония, возникшая на фоне коронавирусной инфекции (COVID-19), унесла жизни более 300 тысяч людей по всему миру. [2]

В настоящее время анализ рентгенограмм чаще всего выполняется высококвалифицированными специалистами вручную. В связи с этим, нагрузка на рентгенологов существенно возрастает за счёт того, что им приходится иметь дело с огромным потоком снимков, который возникает в силу массовости процедуры, особенно в период эпидемий. Для повышения точности постановки диагноза и уменьшения влияния человеческого фактора одни и те же снимки проверяются двумя специалистами или одним с интервалом в один день, что увеличивает нагрузку вдвое. Даже очень опытным специалистам бывает сложно проанализировать более 100 снимков в день. Чем позднее обнаружена пневмония, тем выше риск летального исхода. Таким образом, задача создания алгоритма для автоматического обнаружения пневмонии на рентгенограмме грудной клетки является *актуальной*. Например, существуют различные соревнования RSNA Pneumonia Detection Challenge по данной тематике с денежным призовым фондом.

Задача диагностики пневмонии по рентгенограмме фактически сводится к задаче детектирования областей и распознавания изображений, которая в настоящее время в основном решается с использованием методов глубокого обучения [3]. Наилучшим подходом на данный момент является применение свёрточных нейронных сетей, которые и будут использовать-

ся в данной работе. Существует множество архитектур нейронных сетей, которые имеют различную степень предрасположенности к обучению. При обучении нейросетей, предназначенных для обработки изображений, большое значение имеет «чистота» выбранных данных. В данном случае подразумевается отсутствие различных шумов, например, в контексте рентгенограмм требуется иметь не засвеченные снимки и снимки без наличия посторонних предметов на них. Следовательно, необходимым этапом является предварительная обработка изображений. В данной работе используются классические методы по предобработке изображений, приведенные на официальной странице соревнования на сайте [kaggle.com](https://www.kaggle.com) [4].

В ходе данной работы был построен алгоритм обнаружения пневмонии на рентгенограммах органов грудной клетки и проведён анализ полученных результатов.

Уникальность данной работы заключается в том, что часть построенного алгоритма является модификацией существующей архитектуры [5], имея при этом существенно меньшее число параметров, но сравнимую точность.

## Постановка задачи

**Целью данной работы** является построение алгоритма обнаружения пневмонии на рентгенограмме грудной клетки и выделение границ поражённой области с помощью методов глубокого машинного обучения.

Для достижения данной цели необходимо решить следующие **задачи**:

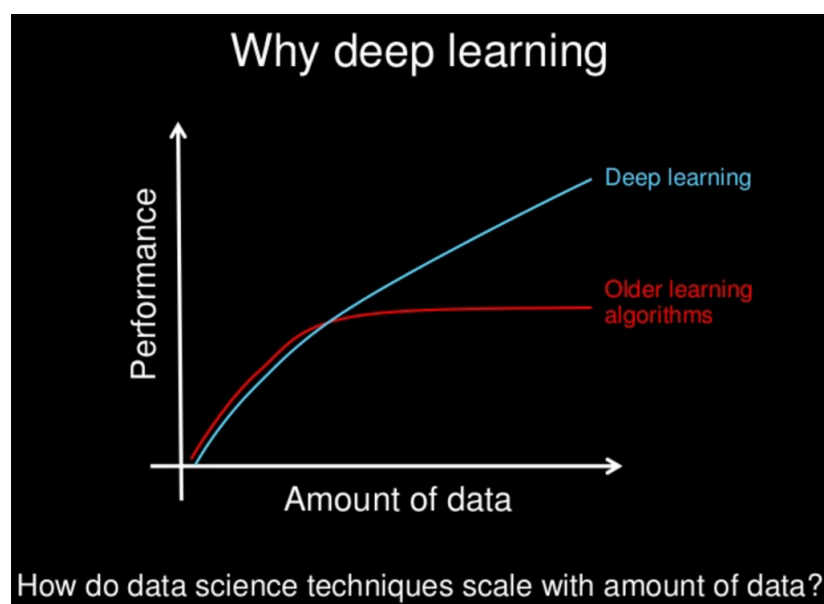
- 1) провести изучение предметной области и анализ существующих решений поставленной задачи;
- 2) рассмотреть методы предобработки медицинских изображений и применить их к изображениям из имеющегося датасета;
- 3) разработать архитектуру системы, позволяющей эффективно осуществлять анализ рентгеновских снимков;
- 4) подобрать подходящие гиперпараметры и произвести обучение реализованной модели;
- 5) провести тестирование реализованного решения, измерить его производительность и сделать выводы относительно полученных результатов.

# Глава 1. Обзор литературы

## 1.1 Основные определения

Рассмотрим некоторые понятия из теории машинного обучения, которые используются в работе.

**Машинное обучение** (МО) [6] является подвидом искусственного интеллекта, занимающимся построением алгоритмов, способных анализировать данные, обучаться на этих данных, а затем применять накопленные «знания» для принятия обоснованных решений. В отличие от традиционного программирования, которое выглядит как набор правил, обеспечивающих обработку данных и вывод результата, МО – это попытка научить компьютер самостоятельно создавать правила, основываясь на данных и ответах.



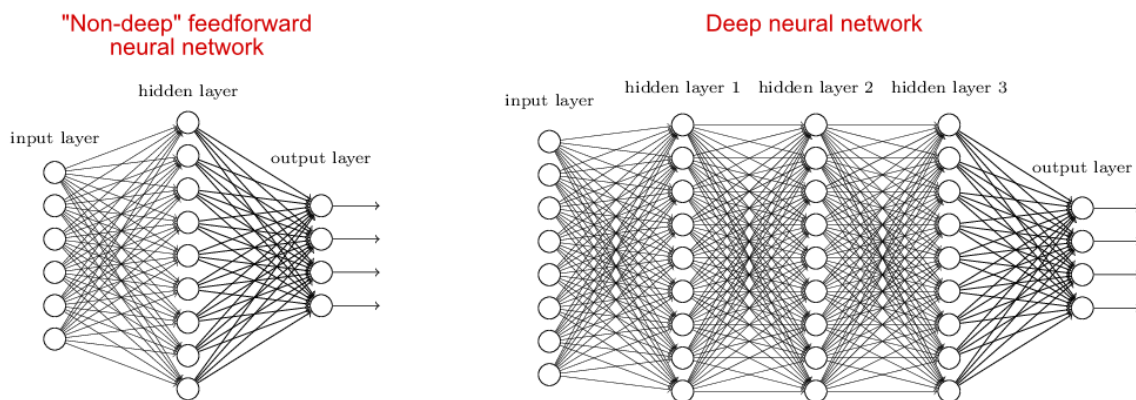
**Рис. 1:** График зависимости производительности от объема используемых данных

**Глубокое обучение** (ГО) [7] – одна из форм машинного обучения, основанная на моделировании структуры и работы мозга. Технически глубокое обучение является машинным обучением и функционирует аналогичным образом, единственное отличие заключается в количестве скрытых слоёв, если их больше одного, то говорят о глубоком обучении. ГО в основном

предназначено для использования с очень сложными и глубокими (по количеству слоёв) сетями. Под слоями подразумеваются нелинейные фильтры, используемые для извлечения признаков, при этом каждый новый слой осуществляет комбинацию информации, извлечённой на предшествующих ему слоях. *Скрытыми* называются все слои расположенные между входным и выходным слоем. Такое качество глубокого обучения, как способность улучшать верность предсказаний (увеличивать производительность) с ростом и усложнением нейронной сети, а также увеличением количества используемых данных, является одним из основных его отличий от других видов МО. Во многом это возможно благодаря мощностям современных компьютеров и наличию достаточного объёма подходящих для обучения данных. Высокая точность привела к тому, что в некоторых областях, таких как распознавание речи и обработка изображений, ГО полностью заменило условно классические алгоритмы. Глубокое обучение также характеризуется переходом представления данных от низкого уровня абстракции на более высокие уровни. При этом признаки с низким уровнем абстракции являются основой для получения более высокоуровневых признаков. Ещё одним отличием методов ГО от МО является самостоятельное создание необходимых для работы слоёв-функций, основываясь на данных, полученных во время обучения, в то время как для других методов МО требуется задание функций перед началом обучения.

Алгоритмы глубокого обучения делятся на обучение с учителем, без учителя, а также их комбинацию. Наибольший интерес представляет обучение с учителем. Методы ГО чаще всего применяются для работ с аналоговыми входными данными, такими как пиксельные изображения, аудио-файлы и текстовые данные. Самый простой пример модели ГО – это многослойный персептрон, который является многослойной полносвязной сетью прямого распространения.

**Свёрточные нейронные сети (CNN)** [6] – это нейросети прямого распространения, имеющие особую структуру. Построение свёрточной сети базируется на знании о том, что данные, поступающие на вход – это изображения, что позволяет существенно снизить число параметров по сравнению с



**Рис. 2:** Пример глубокой и неглубокой нейронной сети

полносвязными сетями. В свою очередь, это упрощает обучение CNN и позволяет снизить необходимое количество обучающих данных. Свёрточные сети способны выполнять три типа задач:

- *Классификация* – нейросеть определяет к какому классу принадлежат объекты на изображении или само изображение;
- *Детекция* – помимо выполнения задачи классификации выявляет положение искомого объекта на изображении, заключая его в ограничивающие параллелепипеды;
- *Сегментация* – наиболее сложная задача. Выполняет задачу классификации для каждого пикселя изображения, позволяет находить пиксели, принадлежащие искомым объектам.

CNN работает по принципу перехода от базовых признаков (градиенты, линии) изображения к более абстрактным деталям, генерируя более высокоуровневые понятия на основе признаков нижних уровней. В процессе обучения сеть самостоятельно выстраивает иерархию признаков, определяя какие из них наиболее значимы, а какими можно пренебречь. Слои свёрточных сетей состоят из нейронов, которые расположены в трёх измерениях (высота, ширина и глубина). В основе организации свёрточных сетей лежат слои трёх видов:



- *Слой свёртки (convolutional layer)* со встроенной функцией активации представляет собой логическое объединение слоя свёртки и слоя активации. Каждый нейрон в свёрточном слое представляет собой плоскость (срез его объема). Нейроны реализуют операцию свёртки и фактически являются небольшими матричными (тензорными в общем случае) фильтрами. Матрица называется ядром фильтра. Значения элементов матрицы – это веса нейронов, определяются в процессе обучения. Подобные нейроны выполняют функцию поиска локальных признаков в видимой части изображения. При этом один фильтр применяется для поиска одного и того же признака на разных частях изображения. Это свойство называется разделением параметров. Оно позволяет существенно сократить число весов в свёрточном слое по сравнению с полносвязным слоем. Построение активационной карты признаков осуществляется прогоном одного фильтра по всему изображению, выполняя операцию свёртки. После сложения всех карт признаков по глубине формируется выходной объём. Пространственный размер (ширина и высота) выходного объёма может быть вычислен по формуле:

$$w_{out} = \frac{w_{in} - n + 2p}{s} + 1,$$

где  $w_{out}$  – высота (ширина) выходного объёма,  $w_{in}$  – высота (ширина) входного объёма,  $n$  – размер фильтра,  $p$  – паддинг, количество строк (столбцов) нулевого дополнения,  $s$  – шаг свёртки.

Принцип операции свёртки заключается в наложении фильтра на небольшой фрагмент исходного изображения, поэлементного умножения коэффициентов матрицы на значения исходного изображения под соответствующими коэффициентами, затем суммировании получившихся произведений и записи этой суммы в результирующий слой на место центра фильтра. Из него следует свойство свёрточных сетей под названием разряженность связей, которое означает, что на любое выходное значение влияет только небольшое число входных значений. Во время свёртки невозможно вычислить выходные данные для

$(n - 1)/2$  строк и столбцов изображения с каждой стороны, так как пиксели, попадающие в них, не имеют окрестности  $n*n$ . Для управления пространственным размером выходного объема применяется нулевое дополнение  $p$  путём добавления строк и столбцов, состоящих из нулей, по краям изображения. Также на выходной пространственный размер влияет шаг свёртки  $s$ . Перемещение фильтра на  $s$  пикселей за раз позволяет уменьшить размер во время свёртки. Для сохранения входного пространственного размера необходимо установить для этих параметров следующие значения:  $p = (n - 1)/2$  и  $s = 1$ . Слой активации применяет поэлементно нелинейную функцию активации к результату свёртки. Чаще всего используется функция линейной ректификации ReLU (rectified linear unit), так как она упрощает вычисления и позволяет сократить время обучения. [8]

- *Слой субдискретизации (пулинга)* – уменьшают пространственный размер изображения (высоту и ширину), за счёт уплотнения карты признаков, при этом глубина изображения остаётся прежней за счёт того, что работа данного слоя осуществляется независимо на каждом срезе глубины входного слоя. Это позволяет получить инвариантность к сдвигам, а также снизить вероятность переобучения и уменьшить вычислительную мощность, необходимую для обработки данных. Чаще всего операция субдискретизации осуществляется путём деления входного изображения на небольшие непересекающиеся прямоугольные площадки и дальнейшей замене каждой из таких площадок на максимальное (Max Pooling) или среднее значение (Average Pooling) входящих в неё элементов. Max Pooling имеет существенное преимущество перед Average Pooling за счёт дополнительного подавления шумов на изображении.
- *Полносвязные слои* используются на выходе модели и отвечают за классификацию. По сути они организованы как обычный персептрон (каждый нейрон полносвязного слоя соединён со всеми выходами нейронов предыдущего слоя). Кроме того, они имеют меньшую размерность по сравнению с размерностью исходного изображения и

осуществляют преобразования входных карт признаков в вектор или скаляр.

**Расчёт количества параметров свёрточной сети.** Рассмотрим алгоритм вычисления количества параметров для каждого отдельного типа слоя. Общее число параметров сети равняется сумме параметров всех слоёв.

- Входной слой: не имеет параметров, он лишь считывает изображение, поступающее в сеть.
- Слой свёртки: пространственная размерность свёрточного слоя с размером фильтра (нейрона)  $n * m$ , принимающего  $l$  карт признаков на вход и выдающего  $k$  карт признаков на выход, будет равна  $n * m * l$ , а общее количество весов –  $n * m * l * k$  соответственно. Но из-за наличия смещения для каждой карты признаков итоговая формула для вычисления будет выглядеть следующим образом:

$$N_{conv} = (nml + 1)k.$$

- Слои субдискретизации: так же, как и входной слой не имеет параметров.
- Полносвязный слой: для полносвязного слоя, принимающего на вход слой с количеством нейронов  $n$  и имеющего размерность выхода  $m$ , число весов равно  $n * m$ , так как все нейроны слоя связаны со всеми выходами нейронов предыдущего слоя и каждый нейрон соответствует одному выходному узлу. Однако из-за наличия смещения выходных узлов, итоговая формула имеет вид:

$$N_{dense} = (n + 1)m.$$

## 1.2 Обзор существующих решений задачи классификации

Задача классификации [9] состоит в определении класса, к которому принадлежит рассматриваемый объект, на основе имеющихся у него признаков. Более продвинутой задачей классификации является в нахождении степени принадлежности объекта к определённому классу или категории. Задача подразделяется на два вида: бинарную, если классов всего два, и мультиклассовую, если количество классов больше двух. Рассмотрим наиболее простые методы, позволяющие решить данную задачу:

- *Метод ближайших соседей.* Принцип заключается в попиксельном сравнении поступающего на классификацию изображения с изображениями из обучающей выборки и нахождении среди этих изображений  $k$  ближайших соседних изображений для тестового изображения путём вычисления расстояния между картинками:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|.$$

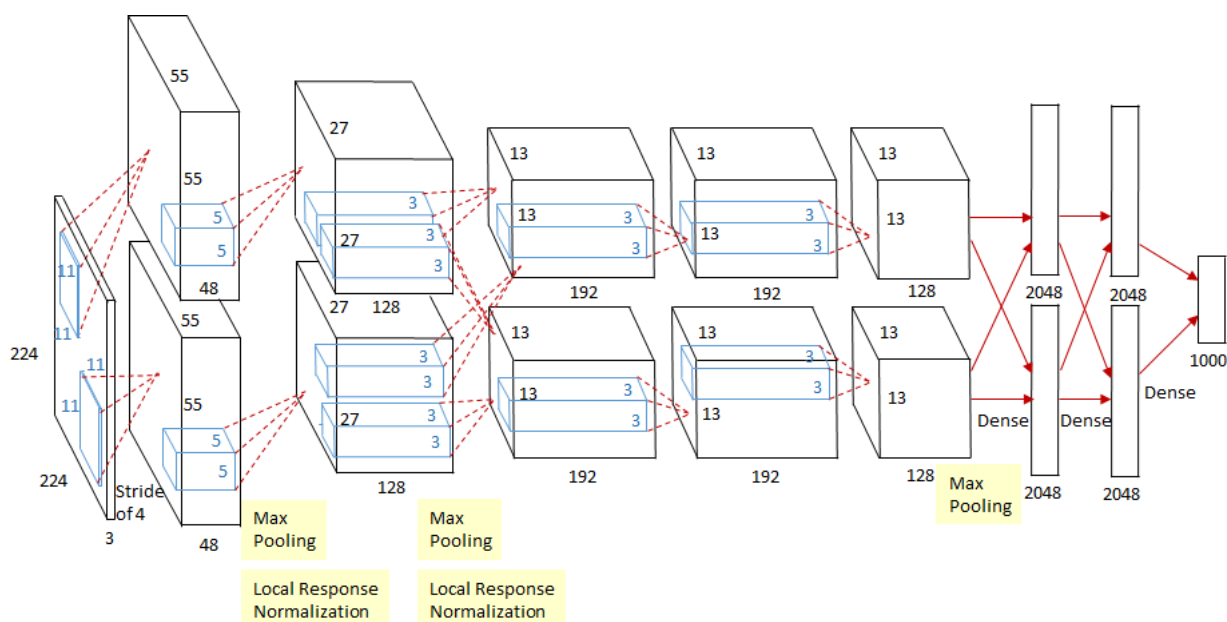
В формуле выше  $I_1, I_2$  - два изображения, вычисление расстояния производится по метрике  $L_1$ . Результат работы классификатора вычисляется как наиболее часто встречаемый класс среди всех соседних. Для более точной работы осуществляется искажение входного изображения путём случайного сдвига его пикселей. Подобный алгоритм быстро обучается (так как обучение заключается в простом запоминании тренировочных объектов), но медленно выдаёт решения для тестовых изображений. К другим минусам можно отнести неустойчивость к изменениям изображения (сдвигу, изменению ракурса, цвета и другим).

- *Метод опорных векторов SVM (Support Vector Machine)* [10] относится к методам машинного обучения с учителем. Его суть заключается в поиске оптимальной гиперплоскости (пространства, размерность которого на единицу меньше рассматриваемого), разделяющей

пространство исходных значений на два класса. Оптимальной является гиперплоскость, расстояние от которой до каждого класса максимальное. Эта плоскость называется линейным классификатором. Если исходные данные являются линейно неразделимыми, необходимо осуществить перевод векторов признаков в пространство большей размерности.

**Решения основанные на глубоких нейросетях** всё чаще применяются в последние годы для решения задачи классификации, они превосходят другие методы по точности распознавания и устойчивы к случайным помехам входных данных.

В свёрточных сетях получить решение задачи классификации позволяет последний полносвязный слой. Количество выходов этого слоя соответствует количеству классификационных классов. Выход, имеющий наибольшее значение, кодирует класс, к которому относится рассматриваемое изображение. На сегодняшний день существуют сотни различных архитектур нейронных сетей способных решить задачу классификации. Но всего десять лет назад интерес учёных к применению глубокого обучения в зада-



**Рис. 3:** Архитектура сети AlexNet

чах классификации только начинал зарождаться. В этом году был создан конкурс ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [11]. Соревнование происходило между новыми архитектурами нейронных сетей. Задача, поставленная перед участниками, заключалась в классификации изображений, относящихся к одному из 1000 классов.

Первой сетью, совершившей прорыв на этом конкурсе в 2012, стала AlexNet [12]. Она получила результат в 15,3 % топ-5 ошибок (вероятность ненахождения истинной метки среди первых пяти прогнозов сети), а следующий за ней результат остался далеко позади (26,2%). Архитектура сети состоит из 5 свёрточных и трёх полносвязных слоёв. В качестве функции активации после каждого из этих слоёв использовалась линейная ректификация (ReLU), слой субдискретизации использовал алгоритм Max Pooling. Самым простым методом улучшения результата предсказаний является использование ансамбля сетей. К каждому тестовому изображению применяется несколько обученных сетей, а результат вычисляется с помощью усреднения.

Со временем число слоёв в нейронных сетях значительно возросло. Увеличение слоёв приводит к улучшению показателей точности, однако подобный рост имеет предел, после которого точность резко падает. Это происходит по причине затухания градиента, сигнал, изменяющий веса, во время прохождения по сети в обратном направлении становится ничтожно

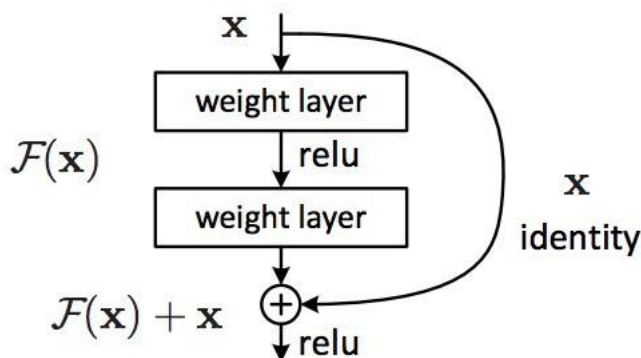


Рис. 4: Соединение быстрого доступа ResNet

малым, не доходя до начальных слоёв. Однако обучению для достижения максимальной точности может быть достаточно меньшего числа слоёв. Для сохранения результата необходимо на оставшихся слоях произвести тождественное преобразование. Ещё одним открытием в истории свёрточных сетей является ResNet (Residual Network) [13], занявшая первое место в ILSVRC-2015 с результатом 3.6% топ-5 ошибки. Её архитектура позволила сохранить рост точности с увеличением глубины сети, благодаря наличию соединений быстрого доступа (short-cut), пропускающих по два слоя и выполняющих функцию сопоставления идентификаторов. Данные проходят по этому пути только тогда, когда веса приравнены оптимизатором к нулю. По сути, этот переход эквивалентен тождественному преобразованию, однако реализован намного проще. Также в данной архитектуре применяется блок bottleneck, который позволяет уменьшить количество весов и увеличить скорость обучения. Существует несколько вариаций сети ResNet с разным количеством слоёв: 18, 34, 50, 101 и 152.

### 1.3 Обзор существующих решений задачи детекции

К наиболее простым способам детекции объектов на изображении относятся такие алгоритмы, как каскадные классификаторы с признаками Хаара и каскадные классификаторы на локальных бинарных шаблонах.

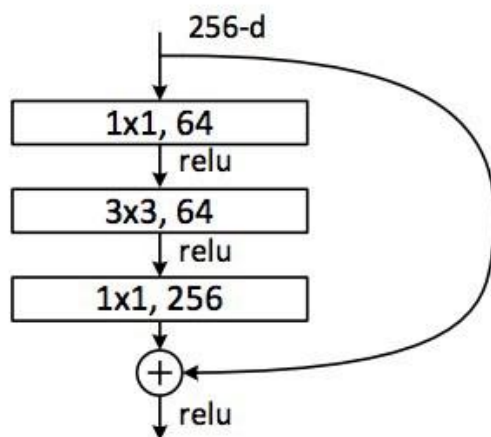


Рис. 5: Блок bottleneck сети ResNet

В основе обоих методов лежит применение каскадного классификатора, принцип работы которого заключается в прохождении по всему изображению окном разного размера. Зоны, в которых окно не нашло искомый объект, моментально отбрасываются. При попадании объекта интереса в окно выдаётся результат «истина».

Признаки Хаара – признаки изображения, представляющие собой графические примитивы в виде черно-белых прямоугольных масок. Подразделяются на признаки края, центра и линейные признаки. Поиск искомого объекта осуществляется на основе всевозможных комбинаций данных примитивов. Преимуществами алгоритма является высокая скорость при точности сравнимой с решениями на основе глубоких сетей.

Локальные бинарные шаблоны LBP (Local Binary Patterns). [15] Суть действия оператора LBP заключается в построении восьмисвязного шаблона окрестности пикселя, к которому применён этот оператор. Шаблон строится с использованием значения центрального пикселя в качестве порогового значения, места всех пикселей, значения которых меньше этого порога, заполняются нулями, остальные – единицами. LBP – это двоичный код, полученный круговым обходом вокруг шаблона. Бинарная интерполяция значений интенсивности пикселей позволяет получить шаблоны с определённым числом шагов вокруг и заданным радиусом. Равномерные LBP определяют важные локальные особенности (углы, линии и т.д.). Для них строятся гистограммы, по одному столбцу за один равномерный LBP. Детекция происходит на основе сравнения расстояния между двумя гистограммами.

### **Решения основанные на глубоких нейросетях:**

- *R-CNN* – подход на основе классификационной нейронной сети и скользящего окна. Окно перемещается по изображению, изменяя свой размер и положение. Для каждого положения и размера окна применяется предобученная классификационная нейронная сеть. Предобучение проводится на тех классах, объекты которых необходимо детектировать. При попадании искомого объекта в окно детектора,



классификационная сеть сообщает о его обнаружении, после этого найденное изображение заключается в соответствующую ограничивающую рамку. Недостаток данного метода заключается в том, что для прохождения скользящим окном по всему изображению и в разных масштабах необходимо многократно применить классификационную сеть, что приводит к высокой временной затратности данного алгоритма.

- *Selective search R-CNN*. Улучшение предыдущего подхода возможно, заменой классификационной сети на более лёгкий классификатор, позволяющий находить области, в которых искомого объекта точно нет. При этом на оставшемся пространстве запускается классификационная сеть для обнаружения искомого объекта.
- *Faster R-CNN* – это улучшение предыдущего варианта посредством использования нейронной сети вместо классификатора. В этом решении нейронная сеть используется не только для классификации, но и для предсказания: есть ли объект в рассматриваемом окне. Она также позволяет определить расположение искомого объекта внутри окна с помощью ограничивающих рамок. Идея объединить все действия в одной сети, оказалась наиболее производительным решением задачи детекции.
- *YOLO* [14] использует аналогичный принцип объединения всех задач в одной сети. Его суть состоит в разделении изображения на несколько ячеек и применении классификатора в каждой из ячеек. На основе выходных данных сети строится предсказание окна, в котором находится объект и местоположения ограничивающего прямоугольника. Этот метод отличается высокой скоростью, что позволяет применять его для детекции изображений в реальном времени. При этом потери в качестве по сравнению с другими методами минимальная.
- Задача детекции сводится к задаче сегментации, что позволяет использовать для её решения сегментационные сети. Из анализа методов детекции выявлено, что наилучшими являются решения, осно-

ванные на глубоких нейронных сетях и выполняющие одновременно и задачу классификации, и построения ограничивающего параллелепипеда. Принцип работы сегментационных сетей позволяет с лёгкостью соединить в себе оба этих действия. В наши дни задача сегментации является наиболее актуальной, а сегментационные нейросети имеют большую распространённость и универсальность применения. Из этого следует широкий выбор разнообразных архитектур и их модификаций, специально разработанных для сегментации определённых категорий изображений.

## 1.4 Доступные программные средства

Для решения задач, поставленных в рамках данной работы, был использован язык программирования Python 3.7 [16], так как он наиболее часто используется для реализации алгоритмов машинного обучения и имеет высокую производительность при обработке данных. Также Python имеет достаточное количество библиотек для работы с сетями глубокого обучения, данными и изображениями, что позволяет существенно ускорить процесс написания кода.

Для обработки многомерных массивов использовалась библиотека NumPy [17], а для машинного обучения Keras [18] и Tensorflow [19]. Все вычисления производились при помощи облачного сервиса Google Colab, который является интерфейсом Jupyter Notebook и предоставляет доступ к одному GPU (NVidia Tesla K80). Это позволило существенно уменьшить время обучения, но также привело к ряду ограничений. Первым ограничением является время работы в одной сессии, которое составляет 12 часов. Вторым – ограниченные объёмы оперативной памяти (RAM) и жёсткого диска (Disk). Экспериментальным путём было установлено, что для обучения с использованием Google Colab подходят свёрточные сети, не превышающие по числу 20млн параметров. В случае несоблюдения этого условия обучение прерывается.

## Глава 2. Архитектура и особенности реализации

На основе изученных материалов для решения поставленной задачи был разработан алгоритм, состоящий из следующих частей:

- Предобработка данных
- Классификация изображений
- Детекция областей интереса
- Постобработка результатов детекции

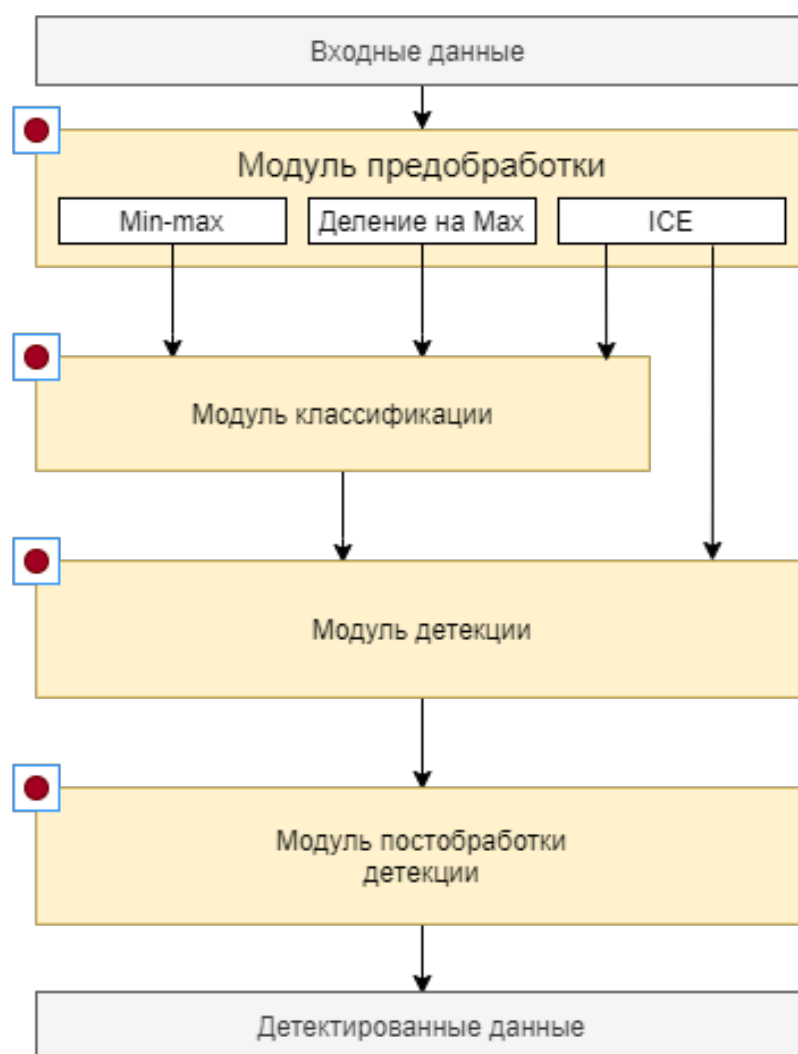


Рис. 6: Схема алгоритма

Также была поставлена цель произвести сравнение производительности двух вариантов решения исходной задачи: 1. С предварительной классификацией изображений и последующим обучением модели детекции исключительно на снимках с выявленной пневмонией. 2. Без предварительной классификации, обучение модели детекции с использованием всех входных данных. Так как появилась гипотеза о том, что предварительная классификация сможет облегчить задачу обучения модели детекции, ускорить время вычислений, а также улучшить точность полученных результатов.

## 2.1 Обработка данных

### Обработка входных данных

Входные данные представлены в виде датасета изображений в формате DICOM (Digital Imaging and Communications in Medicine) [20], который непосредственно предназначен для медицинских сканирований, с разрешением  $1024 * 1024$  пикселя. Этот формат позволяет хранить в каждом изображении дополнительные метаданные о снимке и пациенте. Входные данные содержат также csv файл, в котором хранится подробная информация о наличии заболевания у пациента, которому принадлежит данный снимок, а также о расположении ограничивающих рамок (мест локализации непрозрачности) для снимков, на которых обнаружена пневмония. Всего в датасете 26684 снимка, из которых 32 % имеют патологию в виде непрозрачности участков лёгких, характерную для пневмонии, 39 % здоровых, а остальные снимки не относятся к первым двум категориям и имеют какие-либо другие отклонения.

	patientId	x	y	width	height	Target
0	0004cfab-14fd-4e49-80ba-63a80b6bdd6	NaN	NaN	NaN	NaN	0
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	NaN	NaN	NaN	NaN	0
2	00322d4d-1c29-4943-afc9-b6754be640eb	NaN	NaN	NaN	NaN	0
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	NaN	NaN	NaN	NaN	0
4	00436515-870c-4b36-a041-de91049b9ab4	264.0	152.0	213.0	379.0	1
5	00436515-870c-4b36-a041-de91049b9ab4	562.0	152.0	256.0	453.0	1

Рис. 7: Пример содержимого csv файла

Для повышения точности работы классификационной нейронной сети были рассмотрены три способа предварительной обработки изображений. Первый – минимаксная линейная нормализация. Суть её работы заключается в приведении всех входных значений к одному диапазону изменения  $[0, 1]$  или  $[-1, 1]$ . В нашем случае был выбран интервал от 0 до 1, так как он позволяет корректно отображать изображения. Этот способ позволяет упростить подбор параметров, а также повышает скорость сходимости градиентного спуска во время обучения. Новые значения пикселей изображений были вычислены по следующей формуле:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}},$$

где  $x$  – входное значение интенсивности пикселя,  $x'$  – значение интенсивности после нормализации, а  $x_{min}$ ,  $x_{max}$  – минимальная и максимальная интенсивность.

Второй способ – нормализация путём деления всех значений пикселей на максимально возможное. Так как мы имеем дело с изображениями в оттенках серого, мы можем гарантировать, что интенсивность пикселей всех входных снимков изменяется в диапазоне от 0 до 255 и относительные масштабы пикселей примерно одинаковы. Эти знания позволяют произвести нормализацию более простым путём:

$$x' = \frac{x}{max},$$

где  $max = 255$  – верхняя граница интервала изменения интенсивности пикселей входных изображений.

Третий способ – метод локального усиления контраста (lce), который позволит распределить значения интенсивности пикселей вдоль всего диапазона изменений, что позволит выровнять его контраст и сделать некоторые признаки изображения более заметными для нейронной сети.

$$x' = \frac{x - \tilde{x}}{\sqrt{x^2 - (\tilde{x})^2}},$$

где  $\tilde{x}$  – значение интенсивности пикселя изображения, полученного из исходного путём применения фильтра размытия Гаусса.

Для моделей классификации после обработки любым из способов производилось уменьшение разрешения исходных изображений до размера  $128 * 128$  пикселей. А для модели детекции производилась обработка исключительно методом усиления локальных контрастов, и на вход модели подавались изображения  $256 * 256$  пикселей, так как выбранная архитектура позволяет использовать большее разрешение и время обучения значительно превышает время обучения модели классификации.

Из файла с расширением csv извлекался вектор классификации снимков, содержащий метки 0 и 1, где 1 обозначает наличие пневмонии у пациента, которому принадлежит данный снимок, а 0 – отсутствие пневмонии. Также были извлечены данные о положении ограничивающих окон. Все окна, принадлежащие одному пациенту, были объединены на одной маске, соответствующей снимку данного пациента, где белым (нулями) заполнялись зоны, принадлежащие ограничивающим боксам, а чёрным (единицами) все

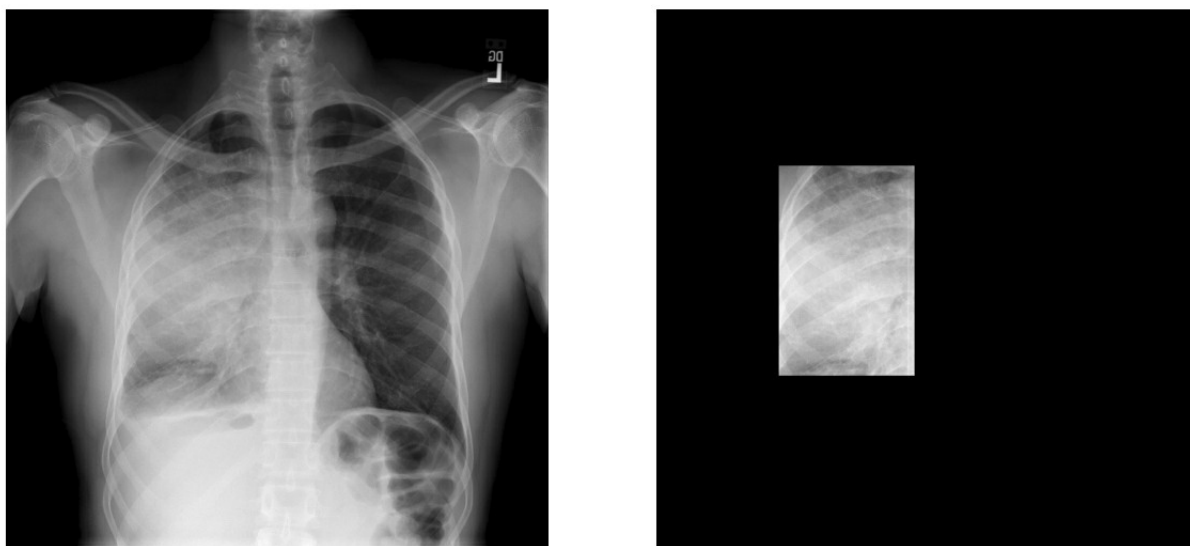
```
(0008, 0005) Specific Character Set          CS: 'ISO_IR 100'
(0008, 0016) SOP Class UID                  UI: Secondary Capture Image Storage
(0008, 0018) SOP Instance UID              UI: 1.2.276.0.7230010.3.1.4.8323329.15474.1517874386.25548
(0008, 0020) Study Date                    DA: '19010101'
(0008, 0030) Study Time                    TM: '000000.00'
(0008, 0050) Accession Number              SH: ''
(0008, 0060) Modality                      CS: 'CR'
(0008, 0064) Conversion Type               CS: 'WSD'
(0008, 0090) Referring Physician's Name    PN: ''
(0008, 103e) Series Description             LO: 'view: AP'
(0010, 0010) Patient's Name                PN: 'dc2a32fa-d49c-4c49-bcdc-3ae6c394fbc9'
(0010, 0020) Patient ID                   LO: 'dc2a32fa-d49c-4c49-bcdc-3ae6c394fbc9'
(0010, 0030) Patient's Birth Date          DA: ''
(0010, 0040) Patient's Sex                 CS: 'M'
(0010, 1010) Patient's Age                 AS: '31'
(0018, 0015) Body Part Examined            CS: 'CHEST'
(0018, 5101) View Position                 CS: 'AP'
(0020, 000d) Study Instance UID            UI: 1.2.276.0.7230010.3.1.2.8323329.15474.1517874386.25547
(0020, 000e) Series Instance UID          UI: 1.2.276.0.7230010.3.1.3.8323329.15474.1517874386.25546
(0020, 0010) Study ID                     SH: ''
(0020, 0011) Series Number                 IS: "1"
(0020, 0013) Instance Number               IS: "1"
(0020, 0020) Patient Orientation           CS: ''
(0028, 0002) Samples per Pixel             US: 1
(0028, 0004) Photometric Interpretation     CS: 'MONOCHROME2'
(0028, 0010) Rows                          US: 1024
(0028, 0011) Columns                       US: 1024
(0028, 0030) Pixel Spacing                 DS: [0.139, 0.139]
(0028, 0100) Bits Allocated                US: 8
(0028, 0101) Bits Stored                   US: 8
(0028, 0102) High Bit                      US: 7
(0028, 0103) Pixel Representation          US: 0
(0028, 2110) Lossy Image Compression        CS: '01'
(0028, 2114) Lossy Image Compression Method CS: 'ISO_10918_1'
(7fe0, 0010) Pixel Data                    OB: Array of 140306 elements
```

Рис. 8: Метаданные случайного снимка

значения во вне. Для всех масок производилось уменьшение разрешения до  $256 * 256$  пикселей. Маски в дальнейшем использовались для обучения модели детекции вместе со снимками из датасета.

### **Обработка результатов работы модели детекции**

Выходные изображения модели детекции имеют разрешение  $256 * 256$  пикселей. На них алгоритм выделил предполагаемые области непрозрачности. Решение задачи детекции заключается в построении на тестовых изображениях с выявленной пневмонией ограничивающей рамки для каждой из непрозрачностей. Для этого необходимо увеличить выходные изображения до исходного разрешения. Затем убрать с них области, меньшие некоторого порогового размера. На оставшихся областях найти минимальные и максимальные значения по ширине и высоте изображения, пары этих значений будут вершинами ограничивающих прямоугольников.



**Рис. 9:** Пример наложения маски на снимок

## **2.2 Архитектура модели классификации**

Для построения модели бинарной классификации было решено взять за основу архитектуру свёрточной сети VGG. Она достаточно проста в построении и модификации, а также достигает точность 8.1% — топ-5 при тестировании на соревновании ILSVRC.

**Классическая архитектура VGG** создана на основе сети AlexNet. Главный акцент в её разработке сделан на глубину. На вход подаётся изображение с разрешением  $224 * 224$  пикселей. Далее оно проходит через блоки слоёв свёртки, использующих очень маленькие рецептивные поля  $3 * 3$  (минимальный размер, позволяющий получить информацию влево/вправо и вверх/вниз). Также для некоторых конфигураций используют свёрточные фильтры  $1 * 1$ , выполняющие функцию линейного преобразования входа. Шаг свёртки принимается равным 1 пикселю, а дополнение (padding) входа слоёв свёртки подбирается так, чтобы на выходе пространственное разрешение изображения не изменялось. Свёрточные слои группируются в блоки по 2-3, между ними находятся слои пулинга, всего используется пять таких слоёв. Операция субдискретизации осуществляется с помощью окна  $2 * 2$  пикселя и фиксированным шагом свёртки равным двум. В конце всегда идут три полносвязных слоя. Два из них имеют 4096 каналов, а последний 1000 по числу выходных классов. Данная сеть имеет два значительных недостатка: медленная скорость обучения и очень большое количество параметров. В связи с тем, что ресурсы на обучение при использовании Google Colab ограничены, была произведена модификация данной архитектуры, позволяющая уменьшить число весов.

**Разработанная архитектура.** Главной целью при проектировании новой архитектуры являлось сокращение числа параметров до значения, соответствующего техническим ограничениям при максимальном сохранении производительности модели. Из принципа вычисления количества параметров свёрточной сети следует, что достижение цели возможно при уменьшении количества карт признаков слоёв свёртки, увеличении размера их ядра и шага свёртки, а также поскольку значительное количество параметров сети приходится на полносвязные слои, уменьшение количества выходов, подающихся на полносвязные слои, и количества нейронов в слоях приведёт к существенному сокращению параметров во всей сети. Экспериментальным путём была составлена сеть, основа которой состоит из чередования слоёв пулинга со стандартным (для VGGnet) шагом свёртки и групп из двух свёрточных слоёв со стандартными ядрами фильтра.



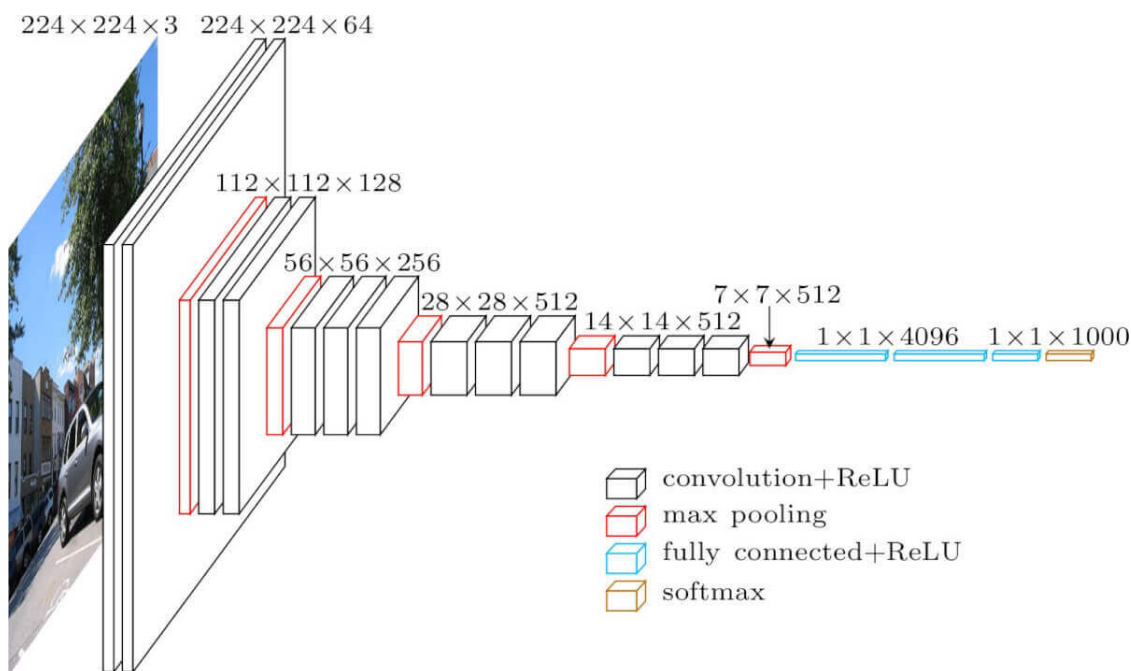


Рис. 10: Архитектура VGG16

Количество полносвязных слоёв решено оставить равным трём. Так как входной объём имеет размерность  $128 \times 128 \times 1$ , а на вход слоя уплотнения желательно подать слой с пространственным размером  $2 \times 2$ , то есть уменьшить высоту и ширину в шесть раз, было решено использовать 6 слоёв пулинга, а не пять, как в стандартной сети. Так же первые четыре группы свёрточных слоёв увеличивают свою глубину в два раза при переходе от блока к блоку (аналогично классической VGG), но при достижении глубины в 512 пикселей оставшиеся два блока сохраняют установленную глубину. Таким образом, первый полносвязный слой принимает на вход со слоя уплотнения 2048 входных каналов и подаёт на выход 1024 канала. Второй полносвязный слой имеет такое же количество выходных каналов, а третий всего два - по числу классов классификации. Для уменьшения вероятности переобучения сети перед вторым и третьим полносвязным слоем решено поставить слои дропаут, которые позволяют исключить долю нейронов из участия в процессе выполнения алгоритма обратного распространения ошибки. Итоговая архитектура свёрточной сети имеет чуть больше 17 млн параметров, это в несколько раз меньше, чем в классической архитектуре VGG-16 (138 млн).

Для контроля производительности классификации было решено произвести сравнение результатов работы модели собственной архитектуры и модели свёрточной сети ResNet-18. Эта архитектура занимает более низкое место в рейтинге наименьших ошибок на конкурсе ILSVRC, по сравнению с оригинальной VGG-16, а так же подходит под технические ограничения, так как имеет 11,5млн. параметров. Архитектуры ResNet с большим количеством слоёв имеют более близкие результаты к результатам производительности VGG-16, нежели ResNet-18, однако число параметров в них сильно возрастает, что не позволяет использовать их для сравнения.

## 2.3 Архитектура модели детекции

Модель детекции должна выполнять одновременно и классификацию объекта, и его обнаружение на изображении. Для решения данной задачи использовалась усовершенствованная архитектура U-Net [22], изначально созданная для сегментации медицинских снимков. На вход модели для обучения поступают исходные снимки и маски для них, для тестирования – только тестовые снимки. На выходе модель выдаёт маску, наложение которой на тестовый снимок выделяет очаг локализации пневмонии. Из этой маски на этапе постобработки извлекается искомая ограничивающая рамка. Данная модель состоит из двух частей:

- Энкодер – сужающая часть, соответствует обычной классификационной свёрточной сети с заменой всех полносвязных слоёв на свёрточные. Состоит из повторяющихся блоков по два слоя свёртки  $3 \times 3$  с активационными функциями ReL. На каждом шаге понижает размерность входного слоя в два раза, используя слои макспулинга  $2 \times 2$  с шагом 2. Пространственная размерность выхода получается пропорциональной входу, при этом выход содержит очень большое количество информации.
- Декодер – расширяющая часть, по сути представляет собой развёрнутый энкодер с некоторыми изменениями. Состоит из тех же блоков свёртки, но на местах слоёв макспулинга расположены слои деконво-

люции  $2 \times 2$ , использующие интерполяцию для повышения размерности и уменьшения числа каналов вдвое. Декодер отвечает за определение положения искомого объекта.

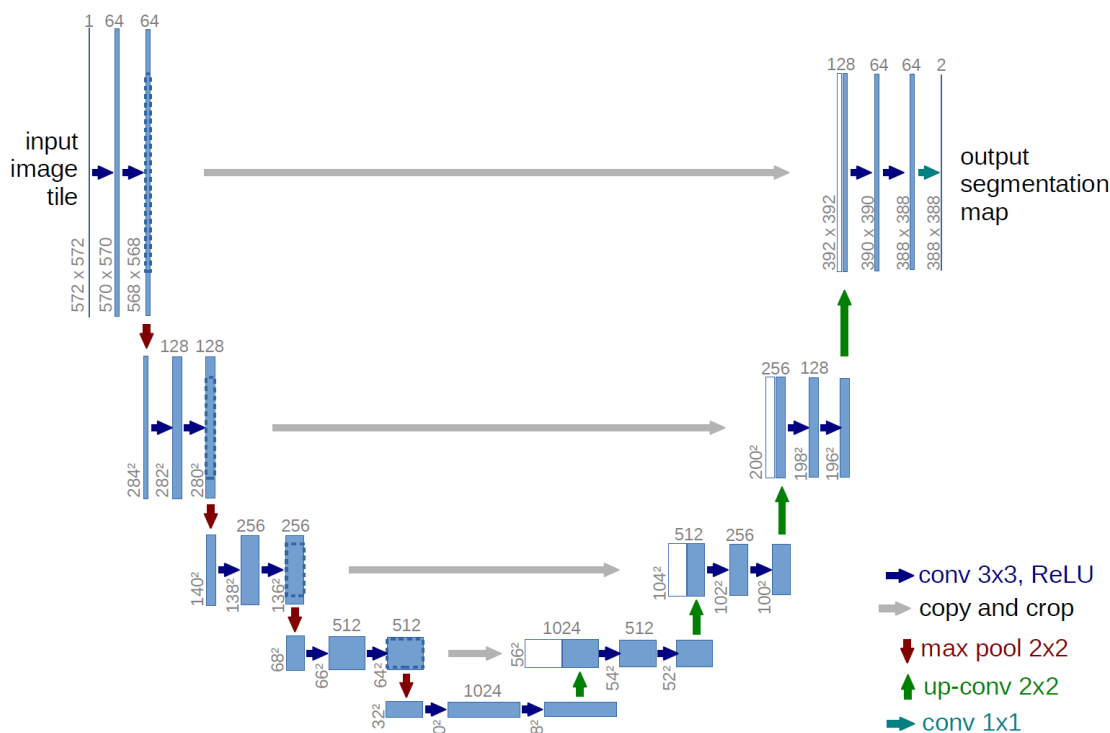


Рис. 11: Архитектура U-Net

Для того, чтобы получить на выходе всей сети маску того же разрешения и качества, что и входные данные, для построения второй части сети - декодера - необходимо использовать промежуточные выходы энкодера. Таким образом, в части декодера происходит конкатенация обрезанной карты признаков сжимающей части и карты признаков разжимающей части. Обрезка необходима, так как операция свёртки происходит без нулевого дополнения и исходный пространственный размер слоёв постепенно уменьшается.

Для увеличения точности работы сети произведена замена блоков свёртки U-Net на residual блоки с соединением быстрого доступа (аналогичные используются в сети ResNet). Полученная сеть имеет чуть больше 1,2 млн. параметров, что подходит под наши ограничения.

## Глава 3. Полученные результаты

### 3.1 Модель классификации

**Метрика.** Для оценки качества работы модели необходимо выбрать метрику, по которой будет вычисляться её производительность. Для оценки качества классификации использовалась наиболее распространенная метрика – верность (Ассигасу), она позволяет оценить общее число верно предсказанных результатов и вычисляется как отношение количества элементов, для которых алгоритм вынес верное решение  $P$ , к общему количеству элементов  $N$  в обучающей выборке:

$$Acc = \frac{P}{N}.$$

**Обучение** происходило методом стохастического градиентного спуска на основе снимков из датасета и соответствующих им метках, содержащих информацию о наличии заболевания в виде значения 0 или 1. Каждый из трёх массивов предобработанных изображений (по количеству способов предобработки) предварительно разделили на три части. Первая часть состояла из 3000 отделённых снимков из начала массивов и предназначалась для тестирования предварительно обученной модели, вторая и третья части – это тренировочные и валидационные данные соответственно разделялись из оставшихся изображений в массивах в соотношении 8:2. Подобный вариант разделения корректен, так как имеется достаточно большое количество данных, они являются независимыми друг от друга и расположены в случайном порядке.

В качестве алгоритма оптимизации взят алгоритм Adam (adaptive moment estimation), сочетающий в себе и идею накопления движения, и идею более слабого обновления весов для типичных признаков. Он позволяет получить более высокие результаты обучения, но замедляет его скорость. Целевой функцией потерь выбрана перекрестная энтропия. Параметры обучения подбирались экспериментальным путём:

- Количество эпох обучения 30 – значение данного коэффициента за-

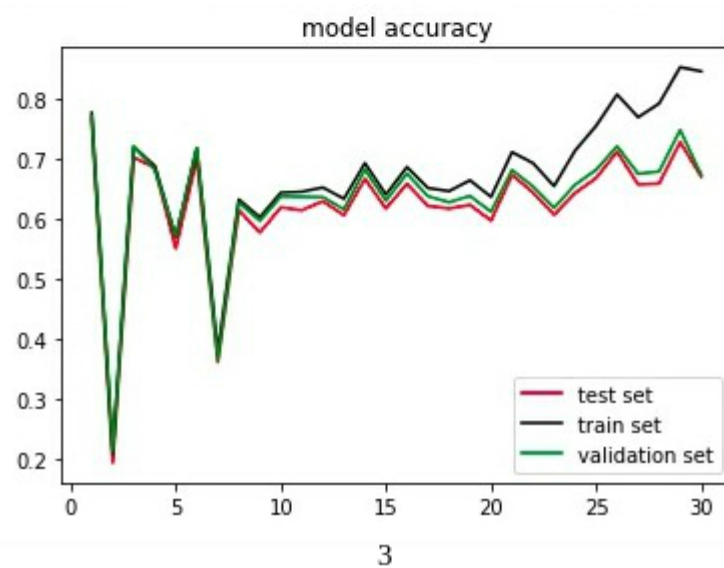
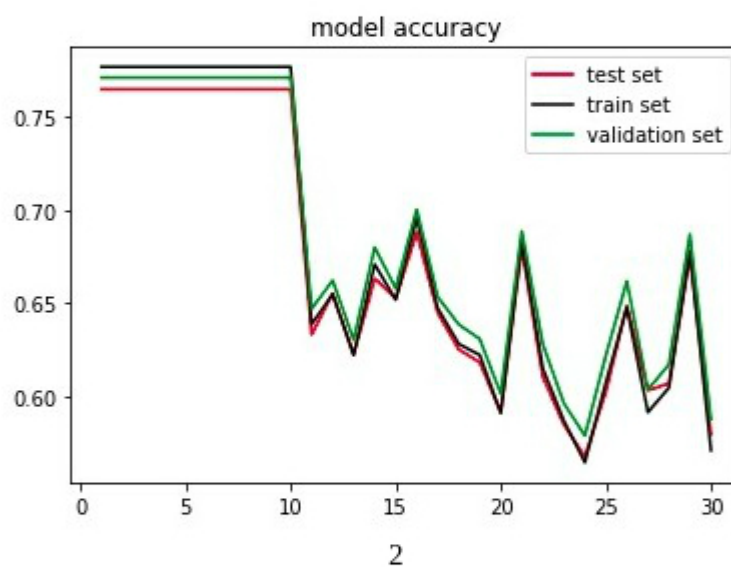
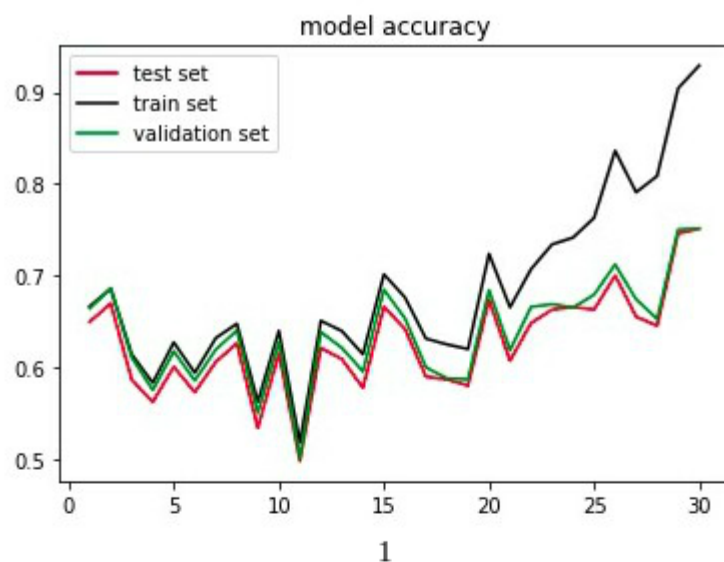
ключается в количестве полных прохождений тренировочных данных через нейросеть в прямом и в обратном направлении. За одну эпоху происходит несколько итераций, их количество определяется числом входных снимков и размером батча.

- Размер батча 256 – количество снимков подаваемое модели на обучение за одну итерацию.
- Коэффициент скорости обучения 0,0001 – параметр, определяющий скорость схождения алгоритма градиентного спуска.
- Коэффициент дропаут 0,3 – доля нейронов, отсеянных от участия в алгоритме обратного распространения ошибки. Работает по принципу обнуления части выходов предыдущего слоя как один из способов регуляризации.

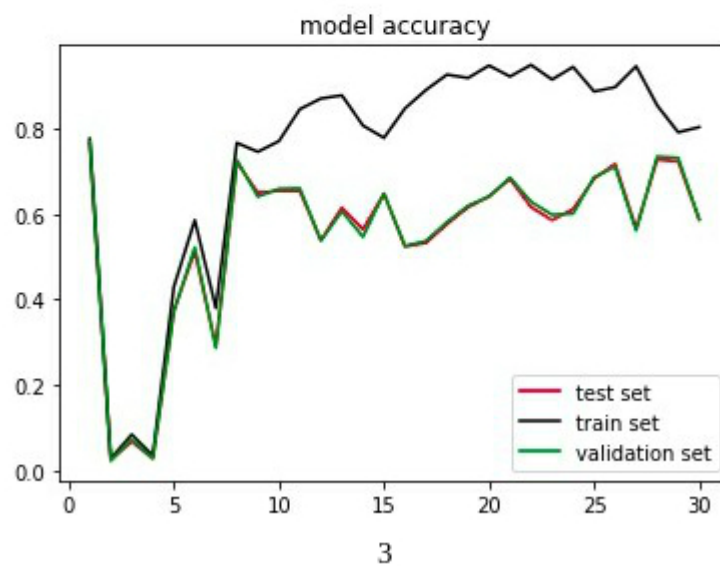
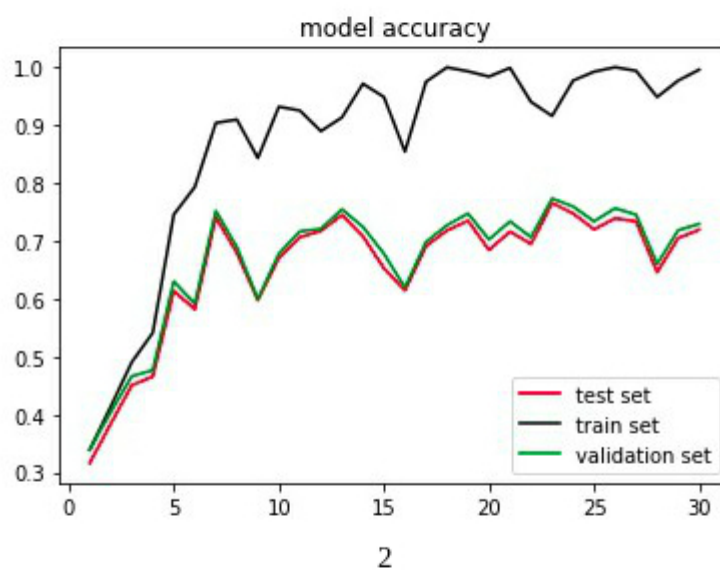
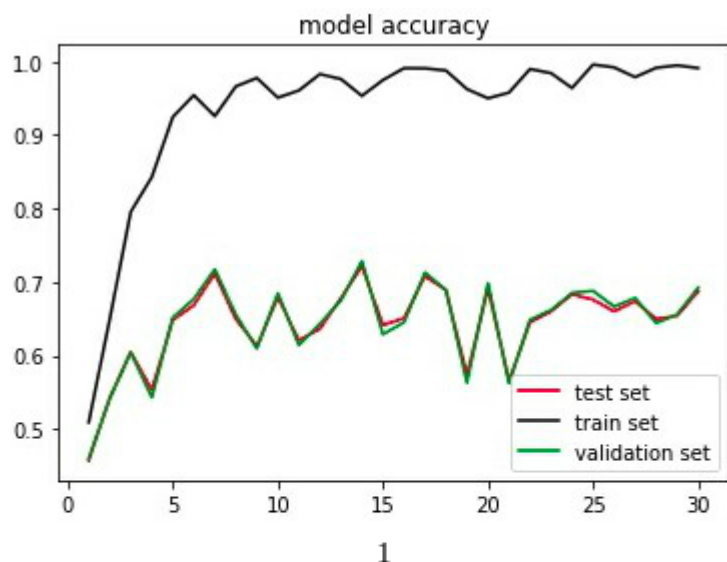
**Анализ полученных результатов** Результаты обучения для каждой модели представлены в виде трёх графиков зависимости верности от количества эпох обучения и таблицы, в которой записаны значения максимальной верности и эпохи, на которой она достигается для трёх вариантов предобработки входных данных. После обучения были отброшены те предсказания, значения которых для обоих классов были ниже коэффициента уверенности (confidence) равного 0.8.

**Таблица 1:** Результаты модели на основе VGG

	верность	эпоха
Минмаксная нормализация	0.7510	30
Нормализация делением на максимум	0.7650	1
Локальное усиление контраста	0.7650	1



**Рис. 12:** Зависимость верности от эпох модификации VGG. (Сверху вниз: минимаксная нормализация, нормализация делением на максимум, локальное усиление контраста.)



**Рис. 13:** Зависимость верности от эпох ResNet-18. (Сверху вниз: минимаксная нормализация, нормализация делением на максимум, локальное усиление контраста.)

**Таблица 2:** Результаты модели ResNet-18

	верность	эпоха
Минмаксная нормализация	0.7216	14
Нормализация делением на максимум	0.7646	23
Локальное усиление контраста	0.7650	1

Наилучший результат показало обучение на данных, к которым предварительно было применено локальное усиление контраста, но разница с результатами обучения на данных с нормализацией делением на максимум является несущественной. Хуже всего себя проявила предобработка в виде минмаксной нормализации. Возможно это связано с тем, что при такой обработке уменьшался общий контраст изображения.

ResNet-18 и модель аналог VGG показали одинаковые результаты на данных с предобработкой в виде локального усиления контраста, однако на данных с двумя другими вариантами обработки ResNet-18 уступила.

## 3.2 Модель детекции

**Метрика.** Оценка производительности модели детекции осуществлялась при помощи метрики IoU (Intersection over Union). Её суть заключается в определении процента перекрытия между объектами. Значение :

$$IoU = \frac{|P \cap Q|}{|P \cup Q|}.$$

В нашем случае первым объектом является ограничительная рамка  $P$ , построенная алгоритмом, а вторым – зона реальной области нахождения ограничительной рамки  $Q$  вокруг зоны непрозрачности лёгких (ground truth). Для одновременной проверки точности классификации изображений установлен порог  $Th = 0.5$ . Если значение  $IoU > Th$ , то изображение считается классифицированным верно, в противном случае считается, что объект не обнаружен и дан неверный ответ для задачи классификации.



**Обучение** осуществлялось на основе снимков из датасета и соответствующих изображений, содержащих маски ограничивающих рамок. К снимкам на этапе предобработки применялось усиление локального контраста, так как этот способ показал наилучшие результаты во время классификации. Для детекции изображений без предварительной классификации использовалось разделение снимков на тренировочное, валидационное и тестовое множество аналогичное разделению для модели классификации. В случае детекции на снимках с выявленной пневмонией разделение происходило следующим образом: 1000 изображений оставалось для тестирования, а остальные – примерно 7,5 тысяч – делились в отношении 9:1 на тренировочные и валидационные данные, так как количество снимков для обучения в этом случае существенно меньше. В качестве алгоритма оптимизации остаётся алгоритм Adam, а функцией потерь – кросс-энтропия. Параметры обучения также подбирались экспериментальным путём:

- Количество эпох обучения 2 и 4 для методов без предварительной классификации и с ней соответственно. Значения существенно меньше, чем значение этого же параметра для сети классификации, так как продолжительность одной сессии всего 12 часов, и время на обучение одной эпохи составляло примерно 5.5 часов и 2.5 часа соответственно.
- Размер батча 4.
- Коэффициент скорости обучения 0.001 увеличен по сравнению с моделью классификации для ускорения сходимости градиентного спуска.

**Анализ полученных результатов.** Результаты обучения на всём датасете и предварительно классифицированных данных представлены в виде таблицы. Было принято решение не строить графики, так как число эпох слишком мало.

**Таблица 3:** Результаты модели детекции

---

	IoU	эпоха
Без предварительной классификации	0.592	2
С предварительной классификацией	0.675	4

Результаты детекции после обучения на классифицированных данных значительно (на 0.083 IoU) превосходят результаты без классификации. Однако ограничения, накладываемые на время обучения, не позволяют раскрыть весь потенциал работы алгоритма, так как получить более точные результаты возможно только после гораздо более долгого обучения.

## Заключение

В ходе работы были получены следующие результаты:

1. Проведено изучение предметной области, анализ существующих решений поставленной задачи и рассмотрены технические возможности для их реализации. В результате было принято решение использовать комбинированный алгоритм, первой частью которого является классификация имеющихся данных, а второй – детекция пневмонии с помощью сегментационной свёрточной сети.
2. Для подготовки данных было рассмотрено три метода предобработки медицинских изображений. Было проанализировано влияние всех трёх методов на верность предсказания классификационных моделей. Лучшие результаты получены при предобработке с помощью метода усиления локального контраста.
3. Разработана архитектура системы, позволяющая эффективно осуществлять детекцию пневмонии на снимках и вместе с этим сравнить эффективность двух разных подходов к её решению. Принято решение сравнить детекцию поражённых областей с предварительной классификацией изображений и без предварительной классификации.
4. Опытным путём были подобраны гиперпараметры, позволяющие получить максимально высокие показатели точности предсказаний обученной модели на тестовых выборках.
5. Произведен анализ полученных результатов, следуя которому можно сделать выводы о том, что детекция с предварительной классификацией снимков дает намного более точные результаты.

В дальнейшем планируется доработать существующее решение с использованием более мощных технических средств, достаточных для проведения нескольких эпох обучения, с целью улучшения показателей его производительности. А также апробировать предварительную классификацию изображений на три класса.

## Список литературы

- [1] Wikipedia // Pneumonia — Access mode:<https://en.wikipedia.org/wiki/Pneumonia> (online; accessed: 28.04.2020)
- [2] World Health Organization // WHO Coronavirus Disease (COVID-19) Dashboard — Access mode:<https://covid19.who.int/> (online; accessed: 14.05.2020)
- [3] Гудфеллоу Я., Бенджио И., Курвилль А. «Глубокое обучение». // Москва: ДМК-Пресс, 2017
- [4] Kaggle // RSNA Pneumonia Detection Challenge — Access mode:<https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/overview> (online; accessed: 14.02.2020)
- [5] Karen Simonyan, Andrew Zisserman // Very Deep Convolutional Networks for Large-Scale Image Recognition // 2014 — Access mode:<https://arxiv.org/pdf/1409.1556.pdf> (online; accessed: 15.04.2020)
- [6] Антонио Джулли, Суджит Пал «БИБЛИОТЕКА KERAS – ИНСТРУМЕНТ ГЛУБОКОГО ОБУЧЕНИЯ. РЕАЛИЗАЦИЯ НЕЙРОННЫХ СЕТЕЙ С ПОМОЩЬЮ БИБЛИОТЕК THEANO И TENSORFLOW». 2018, pp. 14, 79-109
- [7] Саймон Хайкин «Нейронные сети: полный курс, 2-е издание». 2008
- [8] Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua. «Deep sparse rectifier neural networks. In International Conference on Artificial Intelligence and Statistics». 2011, pp. 315–323.
- [9] Орельен Жерон «Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем.». // пер. с англ. – СПб.: ООО «Альфа-книга», 2018, pp. 121–153.

- [10] Воронцов К.В. Машинное обучение. Курс лекций // 2004 — Access mode:<http://www.machinelearning.ru/wiki/index.php?title=Mo> (online; accessed: 11.04.2020)
- [11] ImageNet // Large-Scale Visual Recognition Challenge — Access mode:<http://www.image-net.org/challenges/LSVRC/> (online; accessed: 01.04.2020)
- [12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton // ImageNet Classification with Deep Convolutional Neural Networks // 2012 — Access mode:<https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf> (online; accessed: 15.04.2020)
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // Deep Residual Learning for Image Recognition // 2015 — Access mode:<https://arxiv.org/pdf/1512.03385.pdf> (online; accessed: 15.04.2020)
- [14] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi // You Only Look Once: Unified, Real-Time Object Detection — Access mode:<https://arxiv.org/pdf/1506.02640.pdf> (online; accessed: 08.04.2020)
- [15] Ojala T., M. Pietikäinen, T. Mäenpää «Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns». // Machine Vision and Media Processing Unit, 2002.
- [16] Python 3 // Python official page. — Access mode: <https://docs.python.org/3/> (online; accessed: 09.05.2020)
- [17] NumPy // NumPy official page. — Access mode: <https://www.numpy.org> (online; accessed: 09.05.2020)
- [18] Keras // Keras official page. — Access mode: <https://keras.io/> (online; accessed: 09.05.2020)
- [19] TensorFlow // TensorFlow official page. — Access mode: <https://www.tensorflow.org/> (online; accessed: 09.05.2020)

- [20] DICOM Standard // DICOM standart official page. — Access mode: <https://www.dicomstandard.org/current/> (online; accessed: 05.04.2020)
- [21] Image Classification on ImageNet. — Access mode: <https://paperswithcode.com/sota/image-classification-on-imagenet> (online; accessed: 01.05.2020)
- [22] Olaf Ronneberger, Philipp Fischer, Thomas Brox — Access mode: U-Net:Convolutional Networks for Biomedical Image Segmentation — Access mode:<https://arxiv.org/pdf/1506.02640.pdf> (online; accessed: 02.04.2020)