

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**КЛАССИФИКАЦИЯ ЭНДОСКОПИЧЕСКИХ СНИМКОВ ЖЕЛУДКА С  
ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ И СВЕРТОЧНЫХ  
НЕЙРОННЫХ СЕТЕЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы  
направления 09.03.04 — Программная инженерия  
факультета КНиИТ  
Железко Алексея Дмитриевича

Научный руководитель  
зав. каф. техн. прогр,  
к. ф.-м. н., доцент

---

И. А. Батраева

Заведующий кафедрой  
к. ф.-м. н., доцент

---

А. С. Иванов

Саратов 2020

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	3
1 Теоретические аспекты машинного обучения	4
1.1 Общие сведения о машинном обучении	4
1.2 Модели машинного обучения для решения задачи классификации	4
1.2.1 Линейная регрессия	5
1.2.2 Логистическая регрессия	5
1.3 Основные понятия теории нейронных сетей	6
1.3.1 Строение нейронных сетей	6
1.3.2 Обучение нейронных сетей	7
1.4 Использование transfer learning на нейронных сетях для решения задачи классификации изображений	7
2 Применение моделей машинного обучения для классификации изображений	9
2.1 Описание работы	9
2.2 Модели, которые используют данные, полученные из изображений с помощью алгоритмов компьютерного зрения	9
2.2.1 Извлечение данных из изображений с помощью алгоритмов компьютерного зрения	9
2.2.2 Применение логистической регрессии	10
2.2.3 Применение нейронной сети вида «многослойный персепtron»	12
2.3 Модель, которая работает непосредственно с изображениями	13
2.3.1 Применение предобученной многослойной нейронной сети с последующим ее дообучением	13
<b>ЗАКЛЮЧЕНИЕ</b>	15
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	16

## ВВЕДЕНИЕ

Одной из наиболее актуальных проблем в медицине является проблема диагностики и лечения гастродуodenальной язвы, а также осложненных форм заболевания. Показатели смертности при осложнениях язв в форме кровотечения крайне высоки, но наибольшая летальность достигается среди пациентов с продолжающимся и, особенно, рецидивным кровотечением. Решить данную проблему сложно, поскольку зачастую процесс кровотечения протекает очень медленно и незаметно для больного, вследствие чего врач не подозревает об осложнении, пока оно не переходит в острую стадию. Одним из возможных решений данной проблемы являются технологии машинного обучения, позволяющие предсказывать наличие крови на эндоскопических снимках желудков пациентов. Данные для обучения моделей предоставлены сотрудниками отделения хирургии государственного учреждения здравоохранения «Саратовская городская клиническая больница №6 имени академика В.Н. Кошелева», которые предложили интегрировать технологию нейронных сетей в методику распознавания наличия крови в желудках пациентов.

Целью настоящей работы является построение моделей машинного обучения для решения задачи классификации снимков желудка по признаку наличия на них крови. В рамках указанной цели были поставлены следующие задачи:

- изучение существующих технологий машинного обучения, в том числе многослойных нейронных сетей и трансферного обучения;
- разработка трех различных типов моделей машинного обучения, способных диагностировать наличие крови в желудке;
- определение модели, наиболее успешно справляющейся с задачей диагностики наличия крови;
- разработка алгоритма извлечения данных о содержащихся на изображении цветовых диапазонах.

# 1 Теоретические аспекты машинного обучения

## 1.1 Общие сведения о машинном обучении

Машинное обучение — обширный подраздел науки под названием Искусственный интеллект, изучающий компьютерные алгоритмы, которые автоматически улучшаются с накапливающимся «опытом». Алгоритмы машинного обучения создают математическую модель на основе данных, взятых за образец, которые называются «обучающие данные». Этот процесс совершается для возможности составлять прогнозы и принимать решения, не будучи явно запрограммированным для этого. Алгоритмы машинного обучения используются в основе различных технологий, таких как компьютерное зрение.

Несмотря на то, что машинное обучение начало развиваться не так давно — 1990-х годах, оно быстро стало самым популярным и успешным направлением ИИ. Особенно сильный толчок развития данная дисциплина получила в первом десятилетии XXI века, что обусловлено тремя причинами [1]:

- появление большого количества накопленных данных;
- снижение стоимости параллельных вычислений;
- появление алгоритмов глубокого обучения.

Глубокое обучение — это одно из направлений машинного обучения, в котором упор делается на последовательном изучении идущих друг за другом слоев (Layers) все более значимых представлений. Современное глубокое обучение часто включает в себя десятки или даже сотни последовательных слоев представлений — все они обучаются автоматически под воздействием входных данных, в то время как другие подходы к машинному обучению часто направлены на изучение только одного или двух уровней представления данных.

## 1.2 Модели машинного обучения для решения задачи классификации

Классификация в машинном обучении является одной из множества задач, которые решаются с помощью обучения «с учителем», в рамках которого задаются классы, к которым принадлежат элементы данных [2]. Модели для классификации используют датасеты, предназначенные для обучения, и выделяют закономерности, чтобы наилучшим образом сопоставить входные данные с соответствующими им метками классов, после чего итоговая точность

моделей определяется с помощью тестовых данных, которые не участвовали в процессе обучения. Для успешного решения этой задачи, используемый тренировочный набор данных должен быть достаточно репрезентативным и иметь много экземпляров каждого класса. Применение машинного обучения для решения задачи классификации показывает наибольшую эффективность, когда выходные данные имеют конечные и дискретные значения. Далее рассмотрены некоторые модели для решения данной задачи.

### 1.2.1 Линейная регрессия

Линейная регрессия используется для нахождения линейной взаимосвязи между входными и выходными данными. Она относится одновременной к области статистики и машинного обучения. Линейная регрессия подходит для нахождения отношений между двумя непрерывными переменными. Одна из них является предиктором или независимой переменной, а другая — ответной или зависимой переменной. Линейная регрессия находит статистические закономерности, но не детерминированные отношения. Отношение между двумя переменными называется детерминированным, если одна переменная может быть точно выражена через другую. Например, на основе температуры в градусах Цельсия можно точно предсказать температуру по Фаренгейту. Статистическая закономерность не является точной при определении взаимосвязи между двумя переменными. В качестве примера можно привести связь между ростом и весом.

Основная идея заключается в том, чтобы построить линию или плоскость, которая наиболее точным образом описывает данные. Наилучшей в таком случае будет являться линия, для которой суммарная ошибка прогноза настолько мала, насколько это возможно. Ошибка — это расстояние между точкой и линией регрессии.

С помощью тренировочных данных, предназначенных для обучения, строится линия регрессии, дающая минимальное значение ошибки для большинства точек. Составляется линейное уравнение  $Y(pred) = b_0 + b_1 * x$ , впоследствии оно используется для любых новых данных.

### 1.2.2 Логистическая регрессия

Логистическая регрессия — это линейный классификатор, в котором используется линейная функция  $f(x) = b_0 + b_1\chi_1 + \dots + b_r\chi_r$  где  $b_0, b_1, \dots, b_r$  —

это оценки коэффициентов регрессии.

Функция логистической регрессии  $p(x)$  представляет собой сигмоиду  $f(x) : p(x) = \frac{1}{1+\exp(-f(x))}$ , то есть обычно она стремится к 0 или 1. Под функцией  $p(x)$  подразумевается вероятность предсказания того, что результат выражения равен 1 при заданном  $x$ . Следовательно,  $1 - p(x)$  — это вероятность того, что результат выражения равен 0.

Задача состоит в том, чтобы подобрать такие коэффициенты  $b_0, b_1, \dots, b_r$ , при которых результат вычисления функции  $p(x)$  будет максимально приближенным ко всем фактическим ответам  $y_i, i = 1, \dots, n$ , где  $n$  — количество наблюдений. Как правило, для получения наиболее точных коэффициентов максимизируется функция логарифмического правдоподобия (log-likelihood function, LLF) для всех наблюдений  $i = 1, \dots, n$ . Этот метод называется оценкой максимального правдоподобия и представлен уравнением  $LLF = \sum_{i=1}^n (y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i)))$ . Если  $y_i = 0$ , LLF соответствующего наблюдения становится равна  $\log(1 - p(x_i))$ .

## 1.3 Основные понятия теории нейронных сетей

### 1.3.1 Строение нейронных сетей

Нейронная сеть [3] — это набор алгоритмов, смоделированных по принципу нейронов, соединенных между собой синапсами, которые предназначены для распознавания определенных паттернов. Структура нейронной сети пришла в мир программирования из биологии. Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. Распознающиеся паттерны представляют собой числа, содержащиеся в векторах, в которые переводятся различные данные, такие как изображения, звук или текст. С помощью нейронных сетей неопределенные данные группируются в соответствии с общими чертами примеров, подающихся на вход, а уже определенный набор данных для обучения классифицируется. Технология такова, что на вход нейрона поступает некоторое множество сигналов, которые зачастую являются выходами других нейронов. Каждый вход умножается на соответствующий вес, и все произведения суммируются, определяя уровень активации нейрона. Несмотря на то, что архитектура сетей может различаться, в основе почти всех лежит данная конфигурация.

Различные типы нейронных сетей используют разные принципы при

установке своих норм. Существует много типов нейронных сетей, каждый из которых имеет свои преимущества и недостатки.

### 1.3.2 Обучение нейронных сетей

Процесс обучения нейронной сети строится на том, что на вход подаются тренировочные данные, получается ответ для них, после чего высчитывается ошибка — процентная величина, отражающая расхождение между ожидаемым и полученным ответом. Для этого используется функция потери (loss function). Это метод оценки того, насколько хорошо конкретный алгоритм моделирует данные. Постепенно, с помощью функций оптимизации, функция потерь учится уменьшать значение ошибки предсказания.

Одним из используемых в функциях оптимизации методов является метод градиентного спуска. Градиентный спуск — это алгоритм оптимизации, идея которого заключается в нахождении локального экстремума функции с помощью движения вдоль градиента. Градиентный спуск лучше всего использовать, когда параметры не могут быть рассчитаны аналитически (например, с использованием линейной алгебры), и их необходимо искать с помощью алгоритма оптимизации. В контексте обучения нейронной сети, его используют для минимизации функции потерь. Именно на этом, например, строится «алгоритм обратного распространения ошибки».

Алгоритм обратного распространения ошибки запускается со случайными весами [4], и цель состоит в том, чтобы скорректировать их, чтобы уменьшить эту ошибку, пока искусственная нейронная сеть не изучит тренировочные данные. Комбинация весов, которая минимизирует функцию ошибки, считается решением проблемы обучения. Алгоритм обратного распространения ошибки требует дифференцируемой функции активации, и наиболее часто используемыми являются тангенциальные и логарифмические сигмоиды и, иногда, линейные функции.

## 1.4 Использование *transfer learning* на нейронных сетях для решения задачи классификации изображений

Трансферное обучение (Transfer learning) — это метод, позволяющий использовать предобученную модель (то есть накопленный при решении одной задачи опыт) для решения другой, аналогичной проблемы. Нейросеть сначала обучается на большом объеме данных, затем — на целевом наборе. Как

правило, из-за вычислительной стоимости обучения таких моделей, обычно импортируются и используются уже готовые модели.

Большинство типов предобученных моделей, используемых в трансферном обучении, основаны на сверточных нейронных сетях (CNN), в которые входит очень большое количество слоев [5]. Благодаря высокой производительности и легкости в обучении, в данный момент CNN преуспевает в широком спектре задач компьютерного зрения. При адаптации предобученной модели под конкретную задачу, сначала удаляется исходный классификатор, затем добавляется новый классификатор, соответствующий текущей задаче, и после этого модель меняется в соответствии с одной из трех стратегий:

1. Обучение всей модели. Используется структура предобученной модели, а обучение происходит по имеющемуся датасету. В этом случае, модель будет обучаться с нуля, поэтому при соблюдении данной стратегии требуется большой датасет, а также высокая вычислительная мощность.
2. Обучение нескольких слоев, в то время как другие остаются «замороженными». В случае, если имеется небольшой датасет и большое количество параметров, существует возможность оставить больше слоев замороженными, чтобы избежать переобучения. В противном случае, если датасет большой, а параметров немного, также можно улучшить модель посредством увеличения обучения слоев, поскольку в данной ситуации переобучения не возникнет.
3. Заморозка большинства слоев сверточной сети. Данная стратегия представляет собой компромисс между обучением и заморозкой. Основная идея заключается в том, чтобы сохранить слои CNN в первоначальном виде, а затем использовать выходные данные для подачи в классификатор. Предобученная модель используется в качестве механизма извлечения постоянных особенностей, что может быть полезно в случае, когда недостаточно вычислительных мощностей, маленький датасет и/или предобученная модель уже умеет решать похожую задачу.

## **2 Применение моделей машинного обучения для классификации изображений**

### **2.1 Описание работы**

В данной работе будет реализовано конфигурирование и обучение нескольких моделей машинного обучения, которые будут решать задачу бинарной классификации для определения наличия крови на эндоскопических снимках желудка. Модели будут разделены на 2 группы:

- использующие данные, полученные в результате обработки с помощью специального алгоритма анализа цветов на изображении;
- использующие непосредственно изображения, подвергнутые некоторой предварительной обработке.

В качестве рабочей среды будет использоваться бесплатный облачный сервис на основе Jupyter Notebook — Google Colab. В окружении, которое предоставляет данный сервис, использовался язык программирования Python и библиотеки: Numpy, OpenCV, Scikit-learn, Matplotlib и Pytorch. Также данный сервис предоставляет доступ к одному из GPU Nvidia: K80s, T4s, P4s или P100s [6], который будет применяться для обучения модели на основе многослойной нейронной сети.

### **2.2 Модели, которые используют данные, полученные из изображений с помощью алгоритмов компьютерного зрения**

В данном разделе будут рассмотрены модели машинного обучения, которые не требуют большого количества ресурсов, таких как производительный GPU, для реализации обучения и последующего применения для генерации предсказаний. Чтобы построить такие модели, вместо непосредственно изображений в качестве данных выступает информация о соотношении количества определенных значений цветов на изображении, что значительно упрощает формат входных данных моделей. Благодаря этому удалось применить относительно простые модели, способные решать поставленную задачу с приемлемой точностью.

#### **2.2.1 Извлечение данных из изображений с помощью алгоритмов компьютерного зрения**

Для получения данных о соотношении количества определенных значений цветов на изображении была разработана функция, на вход которой

подается изображение, представленное в цветовом формате HSV. Для работы с изображением используется библиотека OpenCV [7], именно с помощью нее производятся предварительное считывание и перевод изображения в нужный формат. Сначала в функции определяется размер картинки, а затем происходит извлечение массива пикселей и переформатирование его в набор троек значений Hue, Saturation, Value и итерирование по ним с целью подсчета количества пикселей каждого цветового значения, которое встречается на изображении (в библиотеке OpenCV параметр Hue принимает значение в диапазоне от 0 до 179). Результатом работы функции является массив с соотношением количества пикселей для каждого цветового значения к размеру изображения.

Также было разработано 2 альтернативных варианта этой функции: в первой — `color_value_distribution_v2` производилось игнорирование пикселей с малыми значениями параметров Saturation и Value (данные параметры в библиотеке OpenCV принимают значения в диапазоне от 0 до 255), так как в таком случае независимо от значения Hue итоговый цвет пикселя слишком незначительно отличается от абсолютно черного или белого. Во второй — `color_value_distribution_v3` — 180 вариантов цветовых значений были сгруппированы таким образом, чтобы получилось только 60.

### 2.2.2 Применение логистической регрессии

В данном разделе будут рассмотрены: подготовка датасета, обучение модели логистической регрессии на данных, полученных в результате применения разных вариантов функции, описанной в предыдущем разделе, к исходным изображениям и подбор наиболее оптимальных параметров обучения.

Сначала производится считывание изображений и приведение их в цветовой формат HSV с помощью библиотеки OpenCV. Затем происходит разбиение данных на тренировочные и тестовые в соотношении 1 к 4 с помощью метода `train_test_split` из библиотеки scikit-learn [8], с сохранением разделения изображений по признаку наличия на них крови.

Стоит отметить, что набор данных не является сбалансированным — изображений с кровью в нем меньше, чем без нее, но из-за относительно малого размера датасета (менее 1000 изображений), исключать из него что-либо в целях баланса не является целесообразным.

Для построения моделей логистической регрессии будет использоваться библиотека scikit-learn. Сначала будут рассмотрены модели с параметрами,

установленными по умолчанию. Для удобства была создана функция, с помощью которой можно инициализировать и обучить все модели сразу.

Для измерения точности обученных моделей была разработана функция, в которой происходит вычисление точности предсказаний на тренировочных данных, отдельно на тестовых, где присутствует и отсутствует кровь, а также суммарной точности на тестовых данных. Стоит отметить, что изначально изображений с кровью было меньше, поэтому при подсчете общей точности необходимо учитывать количество экземпляров каждого класса.

Далее было произведено обучение моделей на тех же данных, но с разными значениями параметра  $C$  (он отвечает за регуляризацию модели). Также было увеличен параметр `max_iter` до 1000, по умолчанию он установлен на 100, и из-за этого происходило преждевременное прекращение обучения моделей. Помимо этого параметр `class_weight` был изменен на «balanced» (он отвечает за то, чтобы автоматически делать поправку значений весов в соответствии с количеством экземпляров класса, поданных модели на вход в процессе обучения).

Итоговая точность моделей с подобранными значениями параметров для достижения наибольшей эффективности продемонстрирована на рисунке 1.

Результаты работы модели, использующей данные, полученные с помощью функции `color_value_distribution_v1`

Точность на данных для тренировки: 0.8072

Точность на тестовых данных с изображениями, где присутствует кровь: 0.7429

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.6984

Точность на всех тестовых данных: 0.7143

Результаты работы модели, использующей данные, полученные с помощью функции `color_value_distribution_v2`

Точность на данных для тренировки: 0.7943

Точность на тестовых данных с изображениями, где присутствует кровь: 0.6857

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.7460

Точность на всех тестовых данных: 0.7245

Результаты работы модели, использующей данные, полученные с помощью функции `color_value_distribution_v3`

Точность на данных для тренировки: 0.7506

Точность на тестовых данных с изображениями, где присутствует кровь: 0.7143

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.6667

Точность на всех тестовых данных: 0.6837

Рисунок 1 – Итоговая точность моделей логистической регрессии

Из полученных данных следует, что именно модель, которая использует исходную, не модифицированную версию функции для извлечения информации о соотношении количества определенных значений цветов на изображении, показывает наибольшую точность в классификации изображений с кровью, а модель, использующая альтернативную версию функции с пороговыми значениями для параметров Saturation и Value, показывает большую точность при классификации изображений без крови.

### 2.2.3 Применение нейронной сети вида «многослойный персептрон»

В данном разделе будет рассмотрено конфигурирование и обучение нейронных сетей, которые принадлежат виду «многослойный персептрон», они будут содержать относительно небольшое количество скрытых слоев — не более 5, такое ограничение обусловлено простотой входных данных (извлечение которых было описано ранее) и относительно небольшим датасетом — менее 1000 экземпляров. Также будет рассмотрено улучшение точности с помощью подбора наиболее оптимальных параметров и архитектуры моделей.

В качестве основы для построение нейронных сетей будет использоваться модель `MLPClassifier` из библиотеки `scikit-learn`. Сначала будут рассмотрены модели с параметрами по умолчанию, в качестве функции активации будет применяться «relu», а для обучения будет использоваться оптимизатор «adam». Архитектура сети также будет оставлена неизменной — между входным и выходным слоем будет всего 1 скрытый на 100 нейронов.

Далее будет рассмотрен подбор оптимальных параметров обучения для вышеописанных моделей нейронных сетей. В рамках него будет подбираться количество скрытых слоев и число нейронов на каждом из них (параметр `hidden_layer_sizes`), а также количество циклов обучения (параметр `max_iter`). Для реализации подбора будет использоваться вспомогательный инструмент из библиотеки `scikit-learn` — `GridSearchCV`, он перебирает все возможные комбинации параметров, высчитывает точность, используя кросс-валидацию, и выдает наиболее оптимальный набор.

В результате были получены оптимальные конфигурации для трех моделей, использующих разные данные для обучения. Для более детального рассмотрения полученных результатов и визуализации обучения моделей были построены графики зависимости значений точности от количества итераций обучения (параметр `max_iter`). После подбора наиболее оптимальных парам-

метров для всех моделей были посчитаны итоговые характеристики точности, полученные результаты проиллюстрированы на рисунке 2.

Результаты работы модели, использующей данные, полученные с помощью функции `color_value_distribution_v1`

Точность на данных для тренировки: 0.9871

Точность на тестовых данных с изображениями, где присутствует кровь: 0.8286

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.9365

Точность на всех тестовых данных: 0.8980

Результаты работы модели, использующей данные, полученные с помощью функции `color_value_distribution_v2`

Точность на данных для тренировки: 0.9820

Точность на тестовых данных с изображениями, где присутствует кровь: 0.6857

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.9048

Точность на всех тестовых данных: 0.8265

Результаты работы модели, использующей данные, полученные с помощью функции `color_value_distribution_v3`

Точность на данных для тренировки: 0.9152

Точность на тестовых данных с изображениями, где присутствует кровь: 0.7714

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.8889

Точность на всех тестовых данных: 0.8469

Рисунок 2 – Итоговая точность нейронных сетей

## 2.3 Модель, которая работает непосредственно с изображениями

В данном разделе будет рассмотрена модель машинного обучения, которая требует относительно много ресурсов для реализации обучения и последующего применения для генерации предсказаний. Так как датасет содержит малое количество изображений (менее 1000) будет применяться технология трансферного обучения, потому что для полноценного обучения сверточной нейронной сети с нуля требуется гораздо больше данных.

### 2.3.1 Применение предобученной многослойной нейронной сети с последующим ее дообучением

В данном разделе будет рассмотрено использование нейронной сети ResNet34 [9], предобученной на датасете ImageNet, для классификации снимков желудков по признаку наличия на них крови. Для этого будет реализовано дообучение нескольких последних слоев сети на имеющихся изображениях желудков.

Была произведена инициализация модели и конфигурирование слоев таким образом, чтобы при обучении обновлялись только последние 15 наборов

параметров сети (всего их в данном случае 110). Далее было реализовано переопределение полносвязного слоя (расположенного на выходе нейронной сети) таким образом, чтобы он соответствовал ожидаемым выходным значениям (в данном случае имел бы 2 выхода) и преобразование модели, чтобы ее можно было обучать с помощью GPU, а также инициализация loss функции и оптимизатора.

Для подсчета точности модели в процессе обучения была создана функция, в которой сначала происходит перевод модели в режим работы (оценки точности) и помещение кода в область, где не будет высчитываться градиент. Затем происходит проход по группам тестовых данных, получение предсказаний от нейронной сети, преобразование их в питону `array` и высчитывание точности на основе сопоставления полученных результатов и ожидаемых методов для каждого из изображений. Модель затем обратно переводится в режим обучения, а на выходе возвращается итоговая точность на тестовых данных.

Обучение происходит в течение 30 итераций (уже после получения результатов их количество можно сократить, чтобы избежать переобучения). Модель переводится в режим обучения, а затем для каждой группы данных из сформированного ранее `dataloader` происходит их преобразование, чтобы можно было использовать эти данные при вычислениях на GPU. Затем происходит обнуление градиента, получение предсказаний сети, подсчет loss функции и вычисления градиента, а потом обновление параметров модели на основе полученных данных. В конце каждой итерации обучения подсчитывается тестовая точность модели, чтобы затем можно было проанализировать то, как обучалась нейронная сеть.

В итоге после подбора оптимального количества итераций обучения модель, основанная на дообученной сверточной нейронной сети ResNet34, показала очень высокую точность, результаты продемонстрированы на рисунке 3.

Точность на данных для тренировки: 0.9948

Точность на тестовых данных с изображениями, где присутствует кровь: 0.9688

Точность на тестовых данных с изображениями, где отсутствует кровь: 0.9394

Точность на всех тестовых данных: 0.9490

Рисунок 3 – Итоговая точность предсказаний модели, полученной в результате применения трансферного обучения

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной работы были изучены существующие технологии машинного обучения, в том числе многослойные нейронные сети и трансферное обучение. Были разработаны три различных вида моделей машинного обучения, способных диагностировать наличие крови в желудке, и была определена модель, наиболее успешно справляющейся с поставленной задачей. Также был разработан алгоритм извлечения данных о содержащихся на изображении цветовых диапазонах.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 История глубинного машинного обучения [Электронный ресурс]. — URL: <https://www.computerworld.ru/articles/Istoriya-glubinnogo-mashinnogo-obucheniya> (Дата обращения: 07.05.2020). Загл. с экрана. Яз. рус.
- 2 Classification - Machine Learning [Электронный ресурс]. — URL: <https://www.simplilearn.com/classification-machine-learning-tutorial> (Дата обращения: 12.05.2020). Загл. с экрана. Яз. англ.
- 3 Guide to Neural Networks and Deep Learning [Электронный ресурс]. — URL: <https://pathmind.com/neural-network> (Дата обращения: 19.05.2020). Загл. с экрана. Яз. англ.
- 4 Backpropagation Algorithm [Электронный ресурс]. — URL: <https://www.sciencedirect.com/topics/engineering/backpropagation-algorithm> (Дата обращения: 23.05.2020). Загл. с экрана. Яз. англ.
- 5 Transfer learning from pre-trained models [Электронный ресурс]. — URL: <https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751#:~:text=In%20computer%20vision%2C%20transfer%20learning,that%20we%20want%20to%20solve> (Дата обращения: 22.05.2020). Загл. с экрана. Яз. англ.
- 6 Colaboratory. Frequently Asked Questions [Электронный ресурс]. — URL: <https://research.google.com/colaboratory/faq.html#resource-limits> (Дата обращения: 13.05.2020). Загл. с экрана. Яз. англ.
- 7 OpenCV API Reference [Электронный ресурс]. — URL: <https://docs.opencv.org/2.4/modules/refman.html> (Дата обращения: 16.05.2020). Загл. с экрана. Яз. англ.
- 8 Жерон, О. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем / О. Жерон. — Вильямс, 2019.
- 9 ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks [Электронный ресурс]. — URL: <https://neurohive.io/en/popular-networks/resnet/> (Дата обращения: 23.05.2020). Загл. с экрана. Яз. англ.