



Министерство образования и науки Российской Федерации  
федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«Иркутский государственный университет»  
(ФГБОУ ВО «ИГУ»)  
Факультет бизнес-коммуникаций и информатики

**КУРСОВАЯ РАБОТА**  
**Б1.В.25 Курсовая работа по профилю**  
**по направлению 09.03.03 «Прикладная информатика»**  
**(прикладной бакалавриат)**  
**профиль «Прикладная информатика (Разработка программного**  
**обеспечения)»**

Приложение для идентификации пола и возраста  
по рентгеновским снимкам зубов

Студент 3 курса очной формы  
обучения, группа 14321-ДБ  
\_\_\_\_\_ Е.В. Кугот

Руководитель: к.ф.-м.н., доцент  
\_\_\_\_\_ О.И. Бернгардт

Работа защищена:  
« \_\_\_\_\_ » \_\_\_\_\_ 2024 г.  
С оценкой \_\_\_\_\_

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 ТЕОРИЯ ПО РАБОТЕ С НЕЙРОННЫМИ СЕТЯМИ И МЕДИЦИНСКИМИ ИЗОБРАЖЕНИЯМИ.....	5
1.1 Существующие аналоги .....	5
1.2 Особенности рентгеновских снимков .....	5
1.3 Выбор метода обучения моделей.....	6
1.4 Определение задач обучения моделей .....	7
1.5 Выбор архитектуры моделей.....	8
1.6 Выбор предварительно обученной модели.....	10
1.7 Используемые инструменты.....	13
Выводы по главе .....	15
2 ПРАКТИКА ПО СОЗДАНИЮ И ОБУЧЕНИЮ МОДЕЛЕЙ, НАПИСАНИЮ ПРИЛОЖЕНИЯ.....	16
2.1 Загрузка и подготовка данных.....	16
2.2 VGG19 и извлечение признаков из снимков .....	17
2.3 Создание моделей .....	17
2.4 Выбор оптимизаторов, функций потерь и метрик .....	20
2.5 Процесс обучения моделей.....	22
2.6 Результаты обучения моделей.....	23
2.7 Написание приложения для определения пола и возраста.....	25
Выводы по главе .....	26
ЗАКЛЮЧЕНИЕ .....	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	28
ПРИЛОЖЕНИЕ.....	30
Приложение А Архитектура модели VGG19.....	30

## ВВЕДЕНИЕ

В настоящее время искусственный интеллект активно применяется в медицине. Всё больше разрабатывается и обучается нейронных сетей для идентификации заболеваний, личности пациента, его антропометрических показателей, таких как пол, возраст, рост, вес и других. Пока что в таких серьёзных вопросах, как рак, нейросети могут быть лишь вторым мнением, но в случаях определения пола и возраста умерших или пропавших без вести лиц в правоохранительной и судебно-медицинской деятельности они являются ценным инструментом. Кроме того, подобные сети могут быть использованы для научных исследований в области антропологии и эволюционной биологии. По всем вышеперечисленным причинам данная тема является **актуальной**.

**Целью** работы являлось создание приложения для определения пола и возраста пациента по рентгеновским снимкам зубов для медицинской организации.

**Объект исследования:** рентгеновские снимки зубов и построение нейросетевой модели их зависимости от пола и возраста пациента. Нейросетевая модель включает в себя сбор и подготовку данных, выбор архитектуры нейронной сети, ее обучение и оценка итоговой точности модели на ранее неизвестных ей данных.

**Предмет исследования:** особенности рентгеновских снимков зубов, которые могут быть полезными для определения пола и возраста. Они могут включать в себя форму зубов, структуру челюсти, различия в развитии зубов у мужчин и женщин, а также изменения, связанных с возрастом.

### **Задачи:**

- 1) подготовка датасета изображений;
- 2) выбор архитектуры нейронной сети и метода обучения;
- 3) выбор предварительно обученной нейронной (backbone) сети;
- 4) построение архитектуры моделей для определения пола и возраста;
- 5) выбор метрик качества моделей;
- 6) обучение моделей для определения пола и возраста;

- 7) оценка итогового качества моделей на неизвестных ранее данных;
- 8) написание приложения для отображения результатов моделей.

Многие из существующих проектов являются публично недоступными, поэтому эта работа может стать отличным помощником в современной судебной медицине.

# **1 ТЕОРИЯ ПО РАБОТЕ С НЕЙРОННЫМИ СЕТЯМИ И МЕДИЦИНСКИМИ ИЗОБРАЖЕНИЯМИ**

## **1.1 Существующие аналоги**

Одним из примеров нейронной сети, которая определяет пол человека по его рентгеновскому снимку зубов, является работа Б. Виджаякумари, С. Видья и Дж. Саранья из инженерного колледжа Мерсо Schlenk в Индии. Их модель имеет точность в 94% и состоит из трёх компонент: предварительная обработка изображения, сегментация методом GBRT (Gradient Boosted Regression Trees) и классификация [1].

Ещё одним примером является DENSEN – сверточная нейронная сеть для оценки хронологического возраста пациента по панорамным рентгенограммам. Она разработана на основе SSR-Net (Soft Stagewise Regression Network) для оценки хронологического возраста как подростков, так и пожилых людей на основе их ортопантомограмм. Модель оценили по средней абсолютной ошибке (MAE) между предполагаемым возрастом и фактическими данными. Для разных возрастных групп: 3–11 (дети), 12–18 (подростки), 19–25 (молодые) и 25+ (взрослые) DENSEN определил MAE как 0,6885, 0,7615, 1,3502 и 2,8770 соответственно, что является неплохим результатом. Эта модель с открытым исходным кодом [2].

Некоторая информация, полученная при изучении этих моделей, также может пригодиться при решении поставленной задачи. В частности, использование метрики MAE (Mean Absolute Error), как метрику качества для оценки возраста.

## **1.2 Особенности рентгеновских снимков**

Рентгеновские снимки, или рентгенограммы, представляют собой изображения, полученные с помощью рентгеновского излучения. Они используются в медицинской диагностике для визуализации внутренних структур организма, таких как кости, суставы, органы и ткани.

Рентгеновские снимки обладают своими особенностями.

1. Рентгеновские снимки обычно хранятся в формате DICOM (Digital Imaging and Communications in Medicine). Это стандарт в медицинской области. Формат DICOM включает в себя сам снимок и медицинские метаданные, такие как информация о пациенте, устройство, которое произвело изображение, параметры съемки и т. д. [3].

2. Рентгеновские снимки обычно являются черно-белыми изображениями с высоким контрастом (в градациях серого).

3. Снимки позволяют визуализировать структуры организма в зависимости от их плотности. Например, кости и металлические имплантаты обычно выглядят более яркими и плотными, чем мягкие ткани, такие как мышцы или органы [4].

### **1.3 Выбор метода обучения моделей**

Вначале необходимо выбрать метод обучения моделей. Всего их существует три.

1. Обучение с учителем (supervised learning). Метод заключается в том, что на вход модели подаются исходные данные и их «разметка». В результате обучения нейронная сеть подбирает свои параметры так, чтобы ее прогнозные результаты совпадали с исходной разметкой.

2. Обучение без учителя (unsupervised learning). Метод предполагает самостоятельное обучение модели при помощи поиска закономерностей во входных данных. Подобный метод требует большого количества времени и информации, для того чтобы модель научилась выдавать верные выводы.

3. Обучение с подкреплением (reinforcement learning). На вход модели подаются неразмеченные данные, она обрабатывает их и получает положительные или отрицательный ответ на результат своей работы. Важно пройти много предварительных тренировок, для того чтобы обучение с подкреплением достигло нужного эффекта [5].

Для решения данной задачи медицинской организацией был предоставлен датасет рентгеновских снимков в формате DICOM, который содержал

обезличенные данные о пациентах, включающие, однако его пол и возраст. Таким образом, для достижения поставленной цели метод обучения с учителем подойдет лучше всего.

Для этого необходимо построить одну или две (для пола и возраста отдельно) нейронных сети. Для обучения сетей медицинской организацией был предоставлен обезличенный датасет в количестве 14 тысяч снимков в формате DICOM.

Поскольку данные представляют собой изображения, для решения поставленной задачи возможно использовать метод переноса знаний методом извлечения признаков. Этого количества данных, по-видимому, достаточно, так как для извлечения признаков из фотографий возможно использовать предварительно обученную на большом количестве изображений модель, например VGG19.

#### **1.4 Определение задач обучения моделей**

Было решено создать две отдельные модели, одна из которых вычисляла бы возраст, а другая пол. Для каждой из них необходимо сформулировать задачу.

Поскольку пол пациента представляет собой категориальные данные и имеет всего два значения: мужчина (М) и женщина (F), то единственно верным решением будет использовать бинарную классификацию. Классификация — это задача машинного обучения, целью которой является присвоение категории или метки набору входных данных на основе их характеристик. В бинарной классификации таких меток всего две.

Для определения возраста человека будем использовать линейную регрессию, которая предназначена для прогнозирования количественных переменных. Регрессия — это задача машинного обучения, целью которой является прогнозирование числового значения метки по набору исходных признаков изображения. Метка принимает любое числовое значение, а не выбирается из конечного набора, как в задачах классификации, что отлично подойдет для определения возраста [6].

## 1.5 Выбор архитектуры моделей

В машинном обучении наиболее популярны рекуррентные и сверточные нейронные сети, которые применяются для анализа последовательных и пространственных данных соответственно.

Рекуррентная нейронная сеть (RNN) — вид нейронных сетей, где связи между элементами образуют направленную последовательность. Отличительной особенностью RNN является их способность сохранять информацию о предыдущих шагах обработки, что позволяет им учитывать контекст и временные зависимости в последовательностях данных (Рис. 1). У рекуррентных сетей достаточно областей применения:

- работа с текстом: автоматический перевод, оценка грамматической и семантическую корректность, генерация нового текста;
- работа с аудиозаписями: автоматическое распознавание речи;
- работа с видеорядом: распознавание эмоций, определение следующего кадра, на основе предыдущего, выявление значимых кадров;
- работа с изображениями: генерация описания изображения, сегментация, определение следующего пикселя на основе окружения [7].

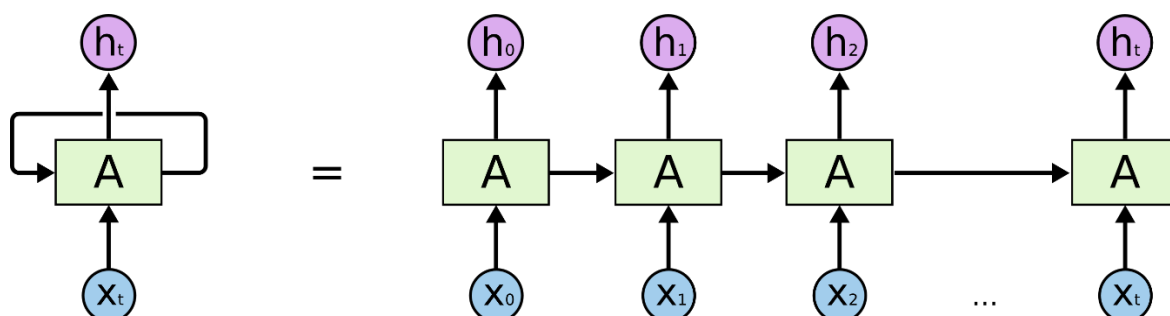


Рисунок 1 – RNN и ее развернутое представление

Пусть данная архитектура сети и может работать с изображениями, но обучение таких сетей достаточно длительный процесс. Более эффективной для решения этой задачи будет сверточная нейронная сеть.

Сверточная нейронная сеть (CNN) – вид нейронных сетей, который специализируется на обработке изображений и видео. Такие нейросети хорошо работают с непрерывной информацией, которая расположена близко друг к другу. Например, пиксели — близко расположенные части изображения, которые



содержат визуальные данные: яркость и цвет. Поэтому если на одном пикселе на фотографии нейронная сеть определяет собаку, значит, и в соседних пикселях тоже [8]. Визуализация архитектуры CNN отображена на рисунке 2. Основные компоненты сверточной нейросети представлены ниже.

1. Входной слой (Input layer): на вход подается изображение, которое представляет собой матрицу данных, где каждый элемент – это пиксель с цветом и интенсивностью.

2. Слой свертки (Convolutional Layer): этот слой выполняет операцию свертки над входным изображением с набором фильтров. Каждый фильтр обнаруживает определенные признаки в изображении, например, границы, углы или текстуры. Результатом операции свертки является карта признаков (feature map).

3. Слой пулинга (Pooling layer): этот слой уменьшает размер карты признаков, сохраняя самые важные информационные характеристики. Наиболее распространенной операцией является MaxPooling, где для каждого участка карты признаков выбирается максимальное значение.

4. Выпрямляющий слой (Flatten Layer): на этом слое данные подготавливаются к дальнейшей классификации. Карты признаков преобразовываются в одномерный вектор.

5. Полносвязные слои (Fully Connected Layers): принимают выпрямленные выходы предыдущего слоя. В этих слоях каждый нейрон связан со всеми выходами предыдущего слоя, что позволяет комбинировать все высокоуровневые признаки для выполнения специфических задач, таких как классификация.

6. Выходной слой (Output Layer): Этот слой определяет формат выходных данных нейронной сети в зависимости от задачи, например, для задачи классификации это может быть слой полносвязный слой с активацией Softmax, который выдает вероятности принадлежности к каждому классу.

7. Слои регуляризации: в некоторых случаях в сверточные нейронные сети могут быть добавлены слои регуляризации, такие как Dropout, для предотвращения переобучения [9].

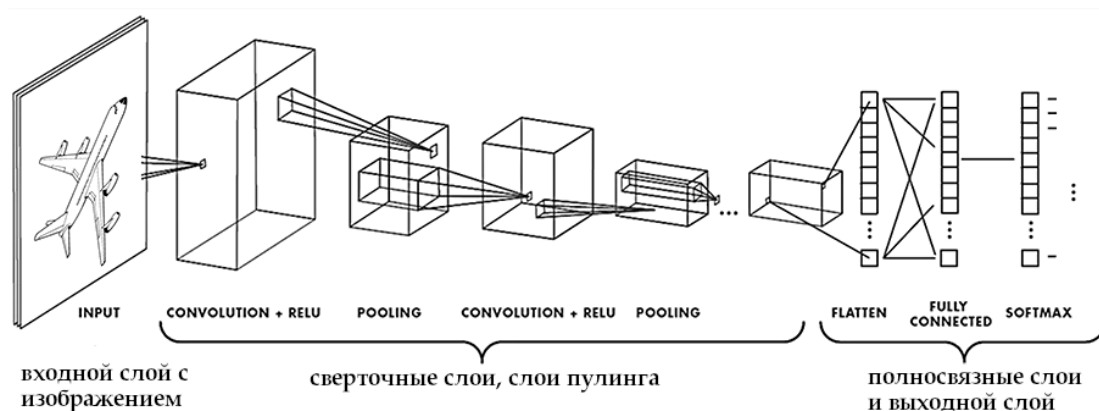


Рисунок 2 – Архитектура сверточной нейронной сети

Сейчас сверточные нейросети применяют повсеместно в системах видеонаблюдения, анализе видео и другом. Например, роботы-доставщики или беспилотные автомобили оценивают обстановку и строят безопасный маршрут с помощью камеры и сверточной нейросети.

Также сверточные сети активно используются в генерации изображений по описанию. Обработкой текста занимаются различные алгоритмы, а созданием картинки — нейросеть. Самые популярные модели для генерации изображений — Midjourney, Stable Diffusion, Dream, DALL-E 2 и ruDALL-E.

### 1.6 Выбор предварительно обученной модели

Для сокращения времени обучения и экономии ресурсов было решено использовать transfer learning. Предварительно обученная нейросеть будет извлекать признаки из изображений, они будут сохраняться в массив и передаваться в регрессионную и классификационную модели. Этот метод менее затратен по времени обучения и ресурсам, нежели если сверточная основа была бы включена внутрь моделей.

Для принятия выбора в сторону какой-либо предварительно обученной модели необходимо подробно изучить их плюсы и минусы. Хотелось бы рассмотреть три модели: InceptionV3, ResNet50 и VGG19.

Inceptionv3 — это сверточная нейронная сеть на основе модели глубокого обучения, разработанная командой Google. Модель InceptionV3 состоит из 42 слоев и имеет архитектуру, представленную на рисунке 3.

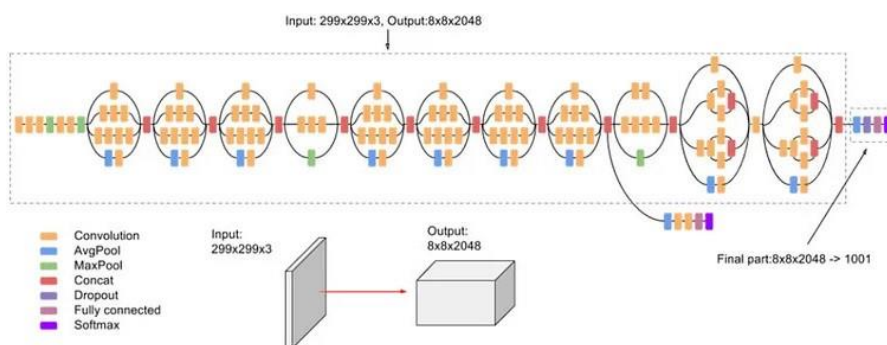


Рисунок 3 – Архитектура InceptionV3

Плюсы модели Inceptionv3:

- использует блоки инцепции, которые позволяют извлекать различные уровни абстракций из изображений;
- предварительно обучена на большом наборе данных изображений ImageNet, что позволяет ей извлекать общие признаки из различных типов изображений.

Минусом данной модели является то, что она имеет относительно большое количество параметров и глубокую архитектуру, что может привести к высокому объему вычислений и требованиям к вычислительным ресурсам при обучении и использовании модели.

VGG-19 — сверточная нейронная сеть, которая имеет 19 главных слоев (16 сверточных, 3 полносвязных) а также 5 слоев MaxPool и 1 слой SoftMax. Архитектура этой сети представлена на рисунке 4.

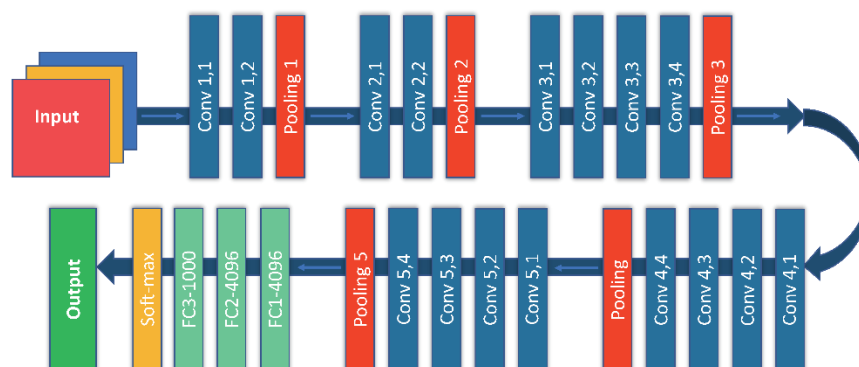


Рисунок 4 – Архитектура VGG19

### Плюсы модели VGG-19:

- имеет относительно простую архитектуру, состоящую из последовательности сверточных и пулинговых слоев, что делает ее легкой в понимании и использовании;
- обучалась на большом наборе данных изображений ImageNet, что означает, что она способна извлекать общие признаки из различных типов изображений;
- несмотря на свою простоту, все еще способна извлекать высокоуровневые признаки из изображений.

### Минусы модели VGG-19:

- поскольку VGG19 применяет свертки на всем изображении без учета его структуры, она может игнорировать локальные особенности и контекст;
- это глубокая модель с большим количеством параметров, что может привести к высокому объему вычислений и требованиям к вычислительным ресурсам, особенно при работе с большими наборами данных, но в данной задаче большой набор данных не будет использоваться.

ResNet50 — это сверточная нейронная сеть, которая имеет 50 основных слоев (сверточные + полносвязные). Сеть ResNet50 была разработана в Microsoft для решения задачи распознавания изображений. Архитектура данной нейросети представлена на рисунке 5.

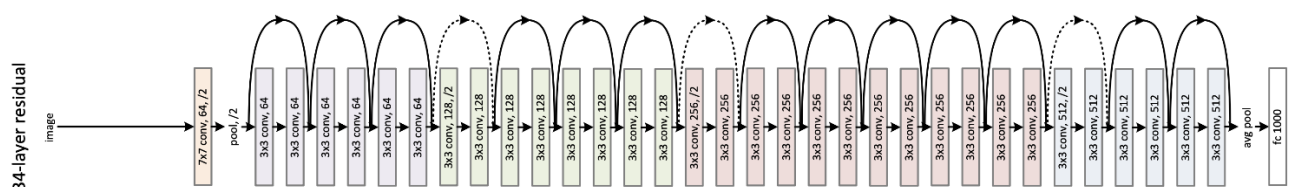


Рисунок 5 – Архитектура ResNet50

### Плюсы модели ResNet50:

- имеет глубокую архитектуру с использованием остаточных (Residual) блоков, которые помогают избежать проблемы затухания градиента при обучении глубоких нейронных сетей;

- предварительно обучен на большом наборе данных изображений ImageNet, что позволяет ему извлекать общие признаки из различных типов изображений.

Минусы модели ResNet50:

- модель имеет те же минусы, что и InceptionV3;
- ResNet50 может игнорировать локальные особенности и контекст в изображениях [10].

Поскольку для InceptionV3 и ResNet50 требуются большие объемы данных для обучения и высокие требования к вычислительным ресурсам, то для решения данной задачи будет использоваться VGG19. Кроме того, она имеет относительно простую архитектуру.

### **1.7 Используемые инструменты**

Для решения поставленной задачи использовался язык программирования Python и его библиотеки. В качестве среды разработки был выбран Google Colab (Colaboratory), предоставляющий облачное окружение для выполнения кода в Jupyter Notebook через браузер и имеющий широкий набор популярных библиотек и пакетов Python. Он предоставляет доступ к вычислительным ресурсам Google, включая графические процессоры (GPU) и тензорные процессоры (TPU), что делает его особенно полезным для обучения нейронных сетей. Кроме того, можно легко импортировать данные из Google Drive и сохранять результаты обучения на диске, что обеспечивает удобство и доступность файлов. В случае исчерпывания ОЗУ на Google Colab, работа будет продолжаться в Jupyter Notebook.

Для работы с рентгеновскими снимками зубов необходимо использовать библиотеки PIL и pydicom, которые хорошо интегрируются с другими библиотеками. pydicom позволяет легко читать и записывать DICOM-снимки, обрабатывает различные типы DICOM-файлов, включая одиночные изображения и серии изображений. Самое главное, что библиотека позволяет извлекать метаданные из снимков, что позволит легко получить пол и возраст пациента.

Также для работы с изображениями, а точнее их аугментацией, будет использоваться библиотека `imgaug`. Она позволит поворачивать и отражать изображения, менять их контрастность, что сделает модель более устойчивой к разным снимкам.

Чтобы отображать графики результатов, например график для сравнения предсказанных и настоящих возрастов, будет использоваться библиотека `Matplotlib`. Она предоставляет простой и интуитивно понятный интерфейс для создания различных типов графиков и визуализаций.

Для быстрой работы с массивами необходимо использовать `NumPy`. `Ndarray` позволяет эффективно хранить и обрабатывать многомерные данные, такие как изображения, и имеет обширный набор математических функций. В данном проекте `NumPy` – это незаменимый инструмент для быстрой работы.

Также необходимо упомянуть о `Keras` – высокоуровневом API для создания и обучения нейронных сетей, который является частью библиотеки машинного обучения `TensorFlow`. Он предоставляет простой и интуитивно понятный интерфейс для создания различных типов нейронных сетей, включая сверточные нейронные сети (CNN). И что крайне важно, `Keras` содержит в своем составе `VGG19`, которая необходима для извлечения признаков из изображений.

Ещё одной необходимой библиотекой является `scikit-learn`, часто называемой `sklearn`. Это библиотека машинного обучения для Python, которая предоставляет простой и эффективный инструментарий для анализа данных и построения моделей машинного обучения. В поставленной задаче из `sklearn` нам понадобится функция `train_test_split` для разделения на обучающую, валидационную и тестовую выборки. А также `confusion_matrix`, для построения матрицы ошибок модели по определению пола.

Всё для построения и обучения моделей есть, поэтому необходимо определить библиотеку для написания интерфейса приложения. Самым популярным и широко используемым вариантом является `tkinter`. Это отличный вариант для написания простого приложения. Приложение должно представлять из себя окно с возможностью загрузки рентгеновского снимка формата DICOM,

отображения результата работы сети (предсказанный пол и возраст) и возможность заново загрузить изображение. Tkinter будет отличным решением для этой задачи.

### **Выводы по главе**

В данной главе были рассмотрены важные теоретические аспекты работы. Найденные аналоги помогли понять, какими могут быть модели для определения пола и возраста человека по рентгеновским снимкам его зубов. В связи с особенностью формата DICOM-изображений и предоставленным обезличенным датасетом был выбран метод обучения с учителем. Были определены задачи обучения моделей: пол – бинарная классификация, возраст – регрессия, а также решено строить две отдельные нейронные сети, а не одну общую. В результате рассмотрения предварительно обученных моделей для извлечения признаков из изображений была выбрана VGG19 с использованием transfer learning для сокращения времени обучения и экономии ресурсов. Сверточную основу было решено не включать внутрь моделей, а передавать в них массив уже извлеченных признаков. Было обосновано применение различных Python библиотек, главной из которых выступает TensorFlow с высокоуровневым API Keras для создания и построения моделей.

## 2 ПРАКТИКА ПО СОЗДАНИЮ И ОБУЧЕНИЮ МОДЕЛЕЙ, НАПИСАНИЮ ПРИЛОЖЕНИЯ

Разобравшись с теоретической частью работы, определив задачи, методы и архитектуру моделей, обосновав необходимые инструменты, приступаем к практической части проекта.

### 2.1 Загрузка и подготовка данных

Обезличенный датасет из рентгеновских снимков зубов был предоставлен медицинской организацией, и для обучения будет использоваться именно он. Для начала необходимо подгрузить 2000 тысячи изображения из Google Drive, куда они предварительно были загружены в формате zip-архива.

На первом этапе проходимся по всем изображениям и устраняем дисбаланс классов для решения задачи определения пола. Для этого считываем каждое изображение, извлекаем значения из PatientSex и заносим каждый пол в свой массив. Если по итогу количество мужчин и женщин разнится, то удаляем лишние снимки для балансировки датасета.

С помощью библиотеки `imgaug` на каждую фотографию создается по 7 копий с аугментациями:

- `iaa.Fliplr(0.5)` – отражение фотографии по горизонтали с вероятностью 50%;
- `iaa.Affine(rotate=(-10, 10))` – поворот изображения вправо или влево до 10 градусов;
- `iaa.GammaContrast((0.3, 2.0))` – изменение контрастности изображения по формуле (1), где  $v$  — значение пикселя, а гамма выбирается равномерно из интервала  $[0.3, 2.0]$ .

$$\text{contrast} = 255 * \left(\frac{v}{255}\right)^{\text{gamma}} \quad (1)$$

Благодаря подобным изменениям модели будут более стабильны к снимкам, сделанным на разном оборудовании.

В конце формируем три numpy-массива:  $X$  для изображений,  $y\_gender$  для пола,  $y\_age$  для возраста. Перед добавлением изображения в массив  $X$



происходит обработка рентгеновского изображения и преобразование его в формат RGB. Массив `y_gender` содержит в себе числовые значения пола, где 1 – мужчины, 0 – женщины.

## **2.2 VGG19 и извлечение признаков из снимков**

Создаем модель на основе предварительно обученной сети VGG19. Указываем параметр `weights='imagenet'`, который говорит о том, что будут загружены веса, обученные на наборе данных ImageNet. Добавляем `include_top=False`, показывающий, что верхний (полносвязный) слой классификатора не будет включен в модель. Таким образом, мы загружаем только архитектуру сверточной базовой сети VGG19 без верхних полносвязных слоев. После создаем новую модель, используя входные данные базовой модели `base_model.input` и ее выходные данные `base_model.output`. Это позволяет нам работать с выходом сверточной части VGG19 как с новой моделью. Архитектура модели представлена в Приложении А.

Теперь необходимо извлечь признаки из рентгеновских снимков зубов. Для этого вызываем метод `predict()` от ранее созданной модели и передаем в качестве аргумента массив `X`.

Для обучения нейронной сети датасет необходимо разбить на три выборки – тренировочную, тестовую и валидационную. Для этого используется метод `train_test_split()` дважды. Вначале датасет делится на обучающий и тестовый в пропорции 7:3, приобретенные тренировочные данные разбиваются пополам для валидационного датасета.

## **2.3 Создание моделей**

Датасет полностью готов и теперь можно приступать к созданию моделей. Для пола разрабатываем классификационную модель, для возраста – регрессионную.

Пишем нейронную сеть для определения пола с использованием библиотеки TensorFlow и ее модуля Keras.

Для этого с помощью Sequential() создается последовательная модель, где слои добавляются один за другим.

1. Слой Flatten() с параметром input\_shape=(7, 7, 512) для преобразования данных из входного размера в плоский вектор.

2. Dense(128, activation='relu') – полносвязный слой с 128 нейронами и функцией активации ReLU (Rectified Linear Activation) (2), означающая, что если входное значение  $x$  положительное, функция возвращает это значение, а если  $x$  отрицательное, функция возвращает 0. Каждый нейрон этого слоя будет связан со всеми нейронами предыдущего слоя (плоского вектора).

$$\text{ReLU}(x) = \max(0, x) \quad (2)$$

3. Dense(1, activation='sigmoid') – выходной слой с одним нейроном и функцией активации sigmoid. Этот слой используется для бинарной классификации, когда модель должна предсказать один выход (в данном случае, вероятность принадлежности к одному из двух классов). Функция активации sigmoid обычно используется для преобразования выхода в диапазон вероятностей от 0 до 1 [11].

Вся архитектура модели, предсказывающей пол по рентгеновским снимкам зубов представлена на рисунке 6.

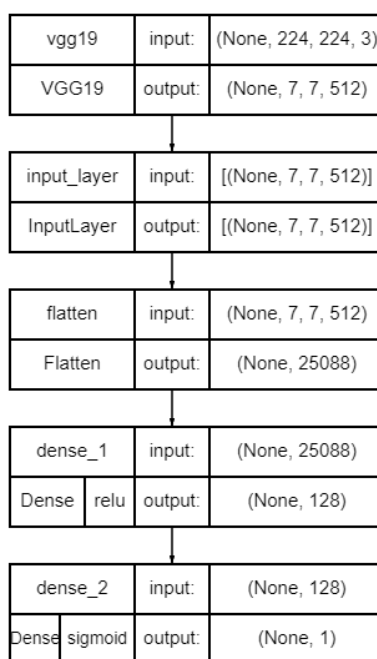


Рисунок 6 – Архитектура модели пола

Для модели по определению возраста также используется Sequential() и поочередно добавляются слои.

1. Также, как и в модели пола, добавляется слой Flatten() с параметром input\_shape=(7, 7, 512) для преобразования данных из формы (7, 7, 512) в плоский вектор.

2. Dense(128, activation='relu') – первый полносвязный слой с 128 нейронами и функцией активации ReLU. Этот слой помогает извлекать важные признаки из данных.

3. Dense(128, activation='relu') – второй полносвязный слой снова с 128 нейронами и функцией активации ReLU. Увеличение числа нейронов и слоев поможет модели изучить более сложные зависимости в данных.

4. Dense(1) – выходной слой с одним нейроном без активации. Модель предсказывает одно числовое значение, которое представляет собой возраст. Отсутствие функции активации на выходном слое означает, что будут выдаваться непрерывные значения, а не вероятности как в случае с классификацией.

Вся архитектура модели, предсказывающей возраст по рентгеновским снимкам зубов представлена на рисунке 7.

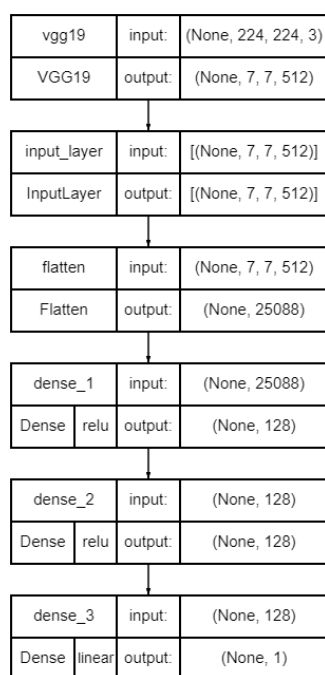


Рисунок 7 – Архитектура модели возраста

## 2.4 Выбор оптимизаторов, функций потерь и метрик

Теперь для каждой модели необходимо выбрать оптимизатор, функцию потерь и метрики. Вначале разберемся, что это.

Функция потерь — это функция, которая оценивает, насколько хорошо модель справляется с задачей во время обучения, сравнивая предсказанные значения с истинными метками данных и выдавая числовую оценку. В процессе обучения нужно минимизировать функцию потерь.

Оптимизатор — это алгоритм, который используется для обновления параметров модели (весов), чтобы минимизировать функцию потерь во время обучения [12].

Для обеих моделей в качестве оптимизатора хорошо подойдет Adam (Adaptive Moment Estimation) из `tensorflow.keras.optimizers`. Это эффективный алгоритм оптимизации, который обновляет веса сети, учитывая как градиенты, так и средние квадраты градиентов. Это позволяет Adam эффективно работать в различных задачах обучения нейронных сетей, включая задачи классификации и регрессии. Он также обычно имеет быструю сходимость и хорошее поведение на практике.

Для модели по определению пола хорошей функцией потерь станет `binary_crossentropy` (3), которая часто используется в задачах бинарной классификации. Потери на кросс-энтропии увеличиваются по мере того, как прогнозируемая вероятность отклоняется от фактической метки.

$$Loss = -\frac{1}{N} \sum_{i=1}^N [y_{true} \log(y_{pred}) + (1 - y_{true}) \log(1 - y_{pred})] \quad (3)$$

- N: это общее количество образцов в наборе данных;
- `y_true`: это фактическая метка для каждого образца (0 или 1);
- `y_pred`: это прогнозируемая вероятность того, что образец принадлежит положительному классу [13].

Для задачи регрессии подойдет функция потерь `mean_absolute_error` (4). Она вычисляет среднее абсолютное значение разности между истинными

значениями и предсказанными значениями. Минимизация MAE приводит к тому, что модель делает наименьшие возможные ошибки в предсказаниях возраста.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_{true} - y_{pred}| \quad (4)$$

- N: количество примеров;
- $y_{true}$ : истинные значения;
- $y_{pred}$ : предсказанные значения [14].

И последнее, что осталось выбрать – это метрики. Метрики — это показатели, используемые для оценки качества работы модели после обучения. Они представляют собой числовые меры, которые помогают понять, насколько хорошо модель работает на новых данных. Для задачи бинарной классификации могут подойти метрики accuracy (5) и AUC-PR (Area Under the Precision-Recall Curve).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Ассигуру измеряет долю правильных предсказаний по отношению ко всем предсказаниям. Для поставленной задачи она дает хорошее представление о том, насколько хорошо модель справляется с классификацией. Значения данной формулы (5) хорошо понятны на рисунке 8.

## Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Рисунок 8 – Матрица ошибок

AUC-PR особенно полезна в случаях, когда данные несбалансированы, потому что она не зависит от доли классов, а сконцентрирована на качестве предсказаний положительного класса. Но поскольку данные заранее были сбалансированы по количеству мужчин и женщин, то данная метрика не понадобится.

Для регрессионной модели хорошей метрикой может стать MAE (4). Она предоставляет информацию о среднем абсолютном отклонении предсказаний модели от истинных значений. Использование MAE в качестве метрики позволяет оценить, насколько хорошо модель предсказывает возраст на новых данных [15].

## **2.5 Процесс обучения моделей**

Прежде чем начать обучение, необходимо определить callbacks – объекты, которые передаются в метод `fit()` и позволяют выполнить определенные действия во время обучения на различных этапах. Для каждой модели понадобится по два callback.

1. `EarlyStopping` – прекращает обучение, если заданное условие не удовлетворяется в течение определенного числа эпох. В модели по идентификации пола остановка будет происходить, когда значение метрики `accuracy` на валидационном датасете (`val_accuracy`) перестанет увеличиваться. В модели по определению возраста будет рассматриваться значение метрики `mae` на валидационном датасете (`val_mae`), его минимизацию. Также задается параметр `patience=5`, который показывает, через какое количество эпох необходимо остановить обучение, если не происходит улучшений.

2. `ModelCheckpoint` – позволяет сохранять веса модели в процессе обучения, например, после каждой эпохи или при достижении наилучшего значения метрики на валидационном наборе данных. В этом callback будут рассматриваться те же значения, что и в предыдущем, и сохраняться лишь лучшая модель (параметр `save_best_only=True`).

Обучение моделей. Вызываем метод `fit()` и в качестве первых двух параметров передаем обучающие данные, в `validation_data` – валидационный

датасет, количество эпох равное 100, batch\_size=32, чтобы в эпохах модель обновлялась только после обработки каждой группы из 32 обучающих примеров, и ранее заданные callbacks.

Обучение модели по идентификации пола длилось суммарно 17 эпох и максимального значения val\_accuracy в 0.92352 достигла на 11 эпохе. После 17 эпохи значения перестали улучшаться.

Обучение модели по определению возраста заняло 20 эпох, притом минимальное значение в val\_mae в 6.95175 было достигнуто на 15 эпохе.

Поскольку мы передали моделям уже извлеченные признаки из VGG19, а не сами изображения, то их обучение прошло быстро.

## 2.6 Результаты обучения моделей

Необходимо оценить модели, вызвав метод evaluate() и передав в него тестовые датасеты. Для модели по определению пола по рентгеновским снимкам зубов были получены следующие значения (Рис. 9):

- Accuracy: 0.93

```
results = model_gender.evaluate(X_test, y_gender_test)

print(f'Loss: {results[0]}')
print(f'Accuracy: {results[1]}')
```

49/49 ————— 0s 3ms/step - accuracy: 0.9356 - loss: 4.3787  
Loss: 3.9511303901672363  
Accuracy: 0.9306358098983765

Рисунок 9 – Результаты модели пола

Для модели по идентификации возраста по рентгеновским снимкам зубов было получено следующие значение метрики mae (Рис. 10):

- MAE: 6.94

```
result = model_age.evaluate(X_test, y_age_test)
print("Age Loss:", result[0])
print("Age MAE:", result[1])
```

49/49 ————— 0s 3ms/step - loss: 6.8877 - mae: 6.8877  
Age Loss: 6.944816589355469  
Age MAE: 6.944816589355469

Рисунок 10 – Результаты модели возраста

Для точной оценки определения пола необходимо построить матрицу ошибок при помощи метода `confusion_matrix()`. На рисунке 11 хорошо видно, что классы сбалансированы. Обозначения в матрице ошибок были представлены ранее на рисунке 8.

```
print(confusion_matrix(gt, gp))  
[[749  54]  
 [ 54 700]]
```

Рисунок 11 – Матрица ошибок для модели пола

Чтобы наглядно увидеть, насколько разнятся истинные и предсказанные значения в модели по идентификации возраста, был построен точечный график с линией тренда (Рис. 12). Можно заметить, что расхождение значений растет по мере увеличения возраста.

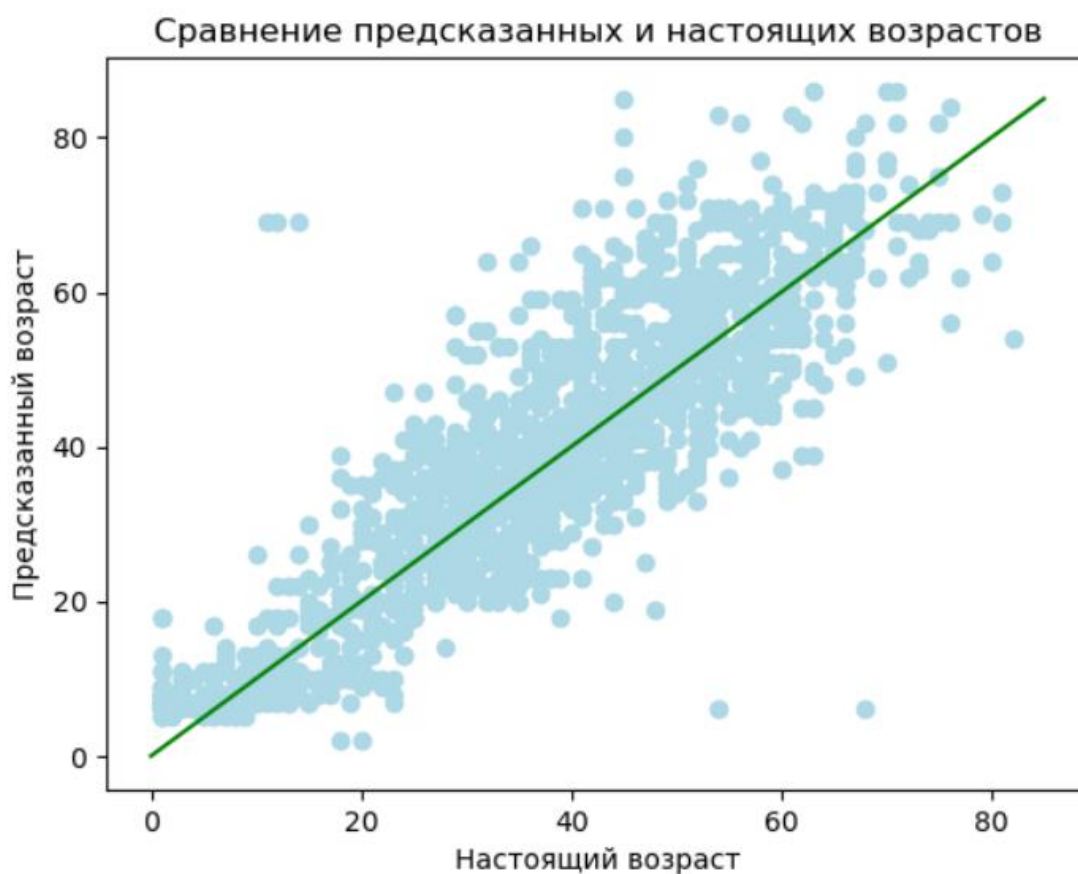


Рисунок 12 – График истинных и предсказанных возрастов



## 2.7 Написание приложения для определения пола и возраста

Для разработки интерфейса была использована библиотека tkinter. А также создано два класса.

1. TeethApp, отвечающий за визуализацию данных. При вызове передается экземпляр класса Models, который сохраняется в атрибут self.models, далее создает первое окно приложения размером 1200x700, отвечающее за загрузку файлов (Рис. 13). После нажатия на кнопку «Load Photo» срабатывает метод load\_photo(), открывающий окно выбора снимка. Если путь к выбранному снимку, и правда, существует, то срабатывает метод display\_value(), отвечающий за отображения рентгеновского снимка зубов и результатов предсказаний моделей с помощью метода predict\_value() (Рис. 14). Для соблюдения конфиденциальности снимок зубов был размыт.

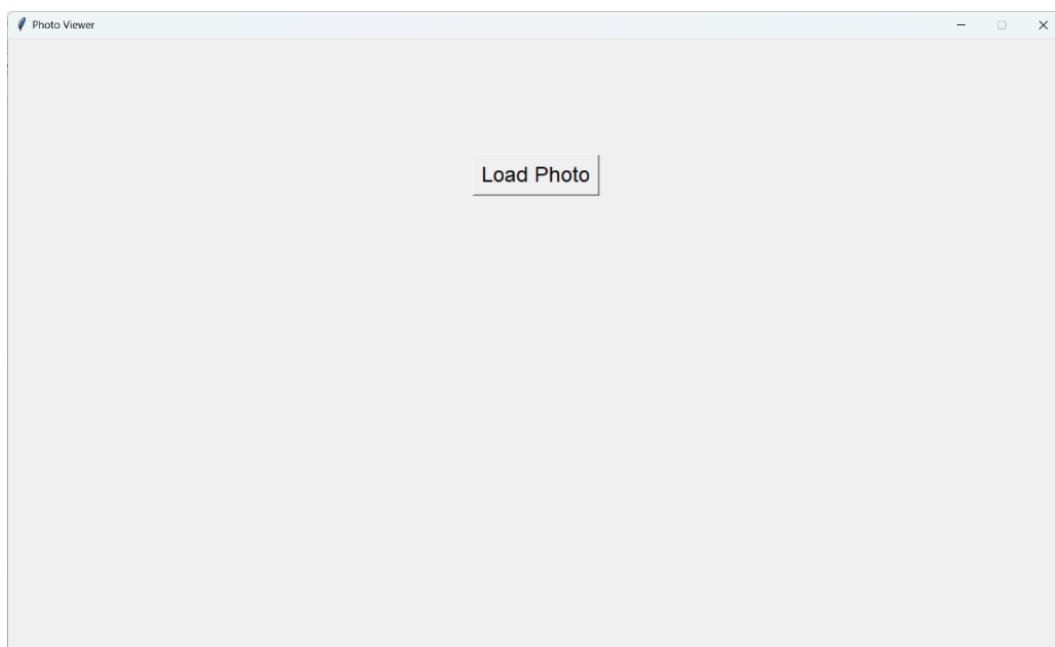


Рисунок 13 – Первый экран приложения

2. Models, отвечающий за загрузки моделей и вызова метода predict() у каждой из них. В качестве аргумента передается путь до папки, в которой лежат модели, вызывается метод load\_models(), который загружает их и сохраняет в атрибуты класса. Во время срабатывания метода predict\_value() у класса TeethApp, происходит извлечения признаков с помощью функции класса Models predict\_features(), а после predict\_age() и predict\_gender() вычисляют возраст и пол.

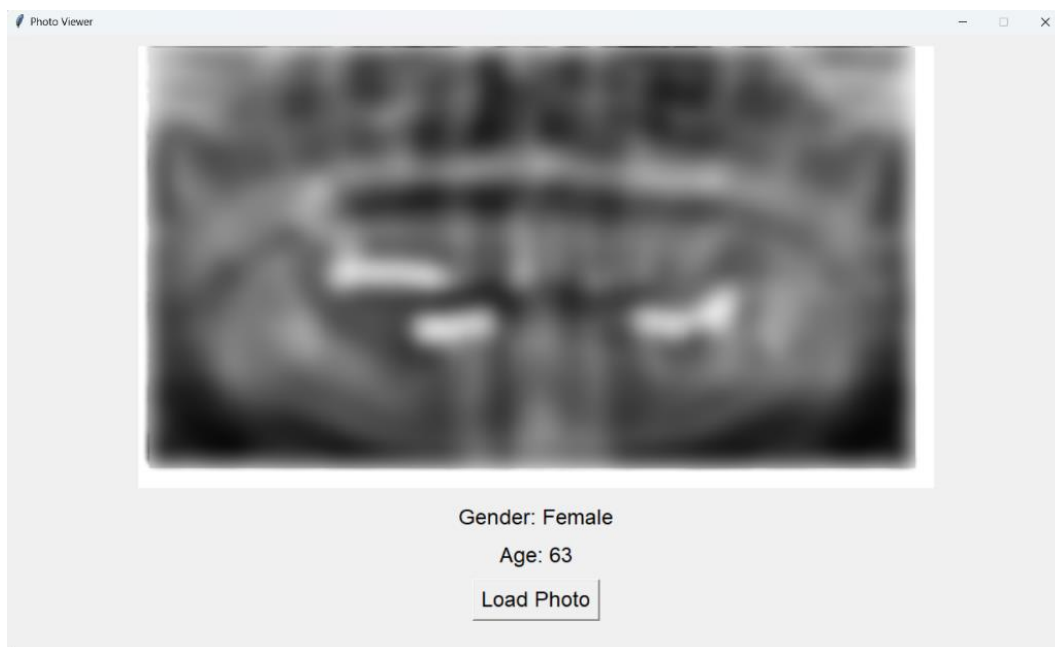


Рисунок 14 – Второй экран приложения

Для загрузки доступны изображения лишь формата DICOM, так как это стандарт в медицинской области. Фото можно загрузить повторно. Само приложение было написано при использовании виртуального окружения среды Anaconda. А также использован менеджер зависимостей poetry.

### **Выводы по главе**

В данной главе были описаны этапы создания и обучения моделей для определения пола и возраста пациента по рентгеновским снимкам зубов. Датасет для обучения был предварительно сбалансирован по классам, а изображения аугментированы для устойчивости моделей. Был рассмотрен процесс работы с моделью VGG19 для извлечения признаков из снимков для дальнейшего разбиения датасета на обучающий, валидационный и тестовый. Выборы оптимизатора, функций потерь и метрик были обоснованы, а результаты моделей оказались достаточно хороши. Был показан и расписан процесс работы приложения.

## ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен весь процесс создания приложения для идентификации пола и возраста пациента по его рентгеновским снимкам зубов. Теоретическая часть помогла разобраться в особенностях рентгеновских снимков, в том, как и какие модели нужно построить. Уже на практике данные были тщательно подготовлены: проведена аугментация фотографии и разбиение на выборки. Была использована сверточная основа VGG19 для извлечения признаков из фотографий для дальнейшей передачи. Модель бинарной классификации показала достижение метрики Ассигасу до 0.93 на тестовом датасете. Регрессионная модель (возраст) достигла значения 6.94 (года) по метрике MAE. Для комфортной работы с нейронными сетями было разработано простое приложение для загрузки фото и вывода результата.

Данное решение может быть усовершенствовано для повышения качества работы, в частности увеличить обучающий датасет для получения лучших значений метрик, а также рассмотреть другие методы аугментации изображений для повышения стабильности модели к снимкам, сделанным на разном оборудовании.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. AI identifies biological sex using dental X-rays David Bradley, Inderscience [Электронный ресурс]. – Режим доступа: <https://medicalxpress.com/news/2023-06-ai-biological-sex-dental-x-rays.html> (дата обращения: 4.04.2024).
2. DENSEN: a convolutional neural network for estimating chronological ages from panoramic radiographs Xuedong Wang, Yanle Liu, Xinyao Miao, Yin Chen, Xiao Cao, Yuchen Zhang, Shuaicheng Li [Электронный ресурс]. – Режим доступа: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-022-04935-0> (дата обращения: 8.04.2024).
3. Файл с расширением .dcm [Электронный ресурс]. – Режим доступа: <https://open-file.ru/types/dcm> (дата обращения: 12.04.2024).
4. Качество рентгеновского снимка. Критерии правильности рентгеновского снимка [Электронный ресурс]. – Режим доступа: [https://meduniver.com/Medical/luchevaia\\_diagnostika/11.html](https://meduniver.com/Medical/luchevaia_diagnostika/11.html) (дата обращения: 12.04.2024).
5. С учителем и без него: как обучаются нейросети [Электронный ресурс]. – Режим доступа: <https://just-ai.com/blog/s-uchitelem-i-bez-nego-kak-obuchayutsya-nejroseti> (дата обращения: 15.04.2024).
6. Задачи машинного обучения в ML.NET [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/machine-learning/resources/tasks> (дата обращения: 16.04.2024).
7. Рекуррентные нейронные сети [Электронный ресурс]. – Режим доступа: [https://neerc.ifmo.ru/wiki/index.php?title=Рекуррентные\\_нейронные\\_сети#cite\\_note-1](https://neerc.ifmo.ru/wiki/index.php?title=Рекуррентные_нейронные_сети#cite_note-1) (дата обращения: 16.04.2024).
8. Как свёрточные нейросети имитируют работу мозга [Электронный ресурс]. – Режим доступа: <https://practicum.yandex.ru/blog/svertochnye-neyronnyie-seti/> (дата обращения: 18.04.2024).
9. Сверточные нейронные сети (CNN) [Электронный ресурс]. – Режим доступа: <https://www.yourtodo.ru/posts/svertochnyie-nejronnyie-seti-cnn/> (дата обращения: 19.04.2024).

10. 10 предварительно обученных моделей для встраивания изображений, которые должен знать каждый специалист по данным [Электронный ресурс]. – Режим доступа: <https://goo.su/gklXdB> (дата обращения: 20.04.2024).

11. Основные функции активации в нейронных сетях [Электронный ресурс]. – Режим доступа: <https://www.yourtodo.ru/posts/osnovnyie-funktsii-aktivatsii-v-nejronnyih-setyah/> (дата обращения: 21.04.2024).

12. Реализуем и сравниваем оптимизаторы моделей в глубоком обучении [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/skillfactory/articles/525214/> (дата обращения: 22.04.2024).

13. Binary Cross-Entropy: Mathematical Insights and Python Implementation [Электронный ресурс]. – Режим доступа: <https://medium.com/@vergotten/binary-cross-entropy-mathematical-insights-and-python-implementation-31e5a4df78f3> (дата обращения: 25.04.2024).

14. Анализ функций потерь и метрик в машинном обучении [Электронный ресурс]. – Режим доступа: <https://www.yourtodo.ru/posts/analiz-funktsij-poter-i-metrik-v-mashinnom-obuchenii/> (дата обращения: 22.04.2024).

15. Метрики классификации и регрессии [Электронный ресурс]. – Режим доступа: <https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii> (дата обращения: 27.04.2024).

### Архитектура модели VGG19

