

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра Прикладная математика и информатика  
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии  
(направленность (профиль) / специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: \_\_\_\_\_ «Разработка ПО для определения пола человека по изображению» \_\_\_\_\_

Обучающийся

М.Д. Гамаюнова

(Инициалы Фамилия)

\_\_\_\_\_ (личная подпись)

Руководитель

к.т.н., доцент, В.С. Климов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

А.В. Москалюк

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

## Аннотация

Тема бакалаврской работы: «Разработка ПО для определения пола человека по изображению».

Цель работы – разработка программного обеспечения для определения пола человека по изображению.

Объект – системы интеллектуального анализа изображений.

Предмет – программное обеспечение для определения пола человека по изображению.

Выпускная квалификационная работа состоит из введения, 3-х разделов, заключения, списка литературы, состоящего из 22 источников. Работа содержит 41 страницу, 4 таблицы и 12 рисунков.

Во введении обоснована актуальность темы, установлены цель, задачи, объект и предмет работы.

В первом разделе выполнен анализ методов распознавания лиц, исследовано применение нейронных сетей для определения пола по изображению, рассмотрены программы – аналоги.

Во втором разделе разработана программа для определения пола человека по изображению, основанная на сверточной нейронной сети. Сформирован тренировочный датасет и выполнено обучение нейронной сети.

В третьем разделе проведено тестирование разработанной программы на размеченной выборке сгенерированных фотографий, на неразмеченной выборке сгенерированных фотографий и на фотографиях реальных персон.

В заключении представлены выводы по работе и дана оценка результатов.

## **Abstract**

The title of the bachelor's thesis is: "Developing software to determine the sex of a person from an image".

The purpose of the work is to develop software to determine the sex of a person by image.

The object - systems of intellectual analysis of images.

Subject - software to determine the sex of a man on the image.

The project consists of the introduction, 3 sections, the conclusion, a list of references consisting of 22 sources. The work has 41 pages, 4 tables and 12 figures.

In the introduction justifies the relevance of the topic, sets the goal, objectives, object and subject of the work.

The first section analyses the methods of face recognition, examines the application of neural networks to determine the sex of the image, and looks at the program analogues.

In the second section, a program for determining the sex of a person by image, based on the convolutional neural network, is developed. The training dataset is formed and the training of the neural network is performed.

In the third section the program is tested on a marked sample of the generated photographs, on the unmarked sample of the generated photographs and on the photographs of the real people.

In the conclusion the conclusions are presented and the results are evaluated.

## Содержание

Введение.....	5
1 Современное состояние задачи определения пола человека по изображению.....	7
1.1 Методы и технологии определения пола человека по изображению.....	7
1.2 Определение пола человека по изображению с применением нейронных сетей.....	11
1.3 Обзор аналогов.....	16
2 Разработка программного обеспечения.....	21
2.1 Выбор средств разработки.....	21
2.2 Подготовка данных для обучения.....	22
2.3 Проектирование нейросети.....	26
2.4 Обучение сети.....	28
3 Тестирование программного обеспечения.....	32
3.1 План тестирования и подготовка данных.....	32
3.2 Результаты тестирования.....	33
Заключение.....	39
Список используемой литературы и используемых источников.....	40

## Введение

Демографические характеристики человека играют важную роль во многих сферах жизни общества: сфере рекламы, охраны права, развлечения, обслуживания и др.

Автоматизированное распознавание таких характеристик человека по изображению лица может быть использовано, например:

- в таргетированном маркетинге, где релевантная для аудитории реклама или информация может транслироваться с цифровых рекламных щитов;

- при сборе данных для дальнейшего анализа: например, при разработке маркетингового плана или проведении таргетированных мероприятий;

- для контентного поиска, где наличие таких характеристик существенно упрощает пространство поиска;

- при взаимодействии компьютер – человек: для подстройки контекста общения в соответствии с полом.

Реализация многих из описанных практических возможностей применения автоматического распознавания требует непрерывной обработки видеопотока, при этом вычисления должны занимать минимально возможное количество времени, чтобы предоставить запас для дальнейших этапов обработки.

Широкому применению систем автоматизированного распознавания должна способствовать низкая вычислительная сложность наряду с обеспечением высокой точности работы и возможностью корректной работы на встроенных платформах без использования графических ускорителей.

Поэтому тема работы, направленная на распознавание пола человека по фотографии является актуальной.

Цель работы – разработка программного обеспечения для определения пола человека по изображению.

Для достижения цели в работе решены следующие задачи:

- исследовано современное состояние задачи определения пола человека по изображению;
- разработано программное обеспечение;
- выполнено тестирование программного обеспечения.

Объект – системы интеллектуального анализа изображений.

Предмет – программное обеспечение для определения пола человека по изображению.

Работа состоит из трех разделов.

В первом разделе проведен анализ современных технологий распознавания лиц, изучено применение нейронных сетей для решения этой задачи и рассмотрены программы-аналоги.

Во втором разделе выполнена разработка программы определения пола человека по изображению, разработан обучающий датасет, проведено обучение нейросети.

В третьем разделе проведено тестирование разработанной программы – разработан набор тестовых данных, проведено тестирование и проанализированы полученные результаты.

Разработанная программа позволяет определять пол человека по изображению, как на сгенерированных фотографиях, так и на фотографиях реальных персон.

# **1 Современное состояние задачи определения пола человека по изображению**

## **1.1 Методы и технологии определения пола человека по изображению**

Общая структура процесса распознавания лиц [1] предусматривает последовательные этапы:

- детектирование и локализация лица на изображении;
- выравнивание изображения лица (геометрическое и яркостное);
- вычисление признаков;
- распознавание – сравнение вычисленных признаков с эталонами;

Основным отличием всех технологий распознавания лиц является вычисление признаков и сравнение их совокупностей между собой.

«Метод гибкого сравнения на графах (Elastic graph matching) сводится к эластичному сопоставлению графов, описывающих изображения лиц [13]. Лица представлены в виде графов со взвешенными вершинами и ребрами. На этапе распознавания один из графов – эталонный – остается неизменным, в то время как другой деформируется с целью наилучшей подгонки к первому. В подобных системах распознавания графы могут представлять собой как прямоугольную решетку, так и структуру, образованную характерными (антропометрическими) точками лица»[1].

«В вершинах графа вычисляются значения признаков, чаще всего используют комплексные значения фильтров Габора, которые вычисляются в некоторой локальной области вершины графа локально путем свертки значений яркости пикселей с фильтрами Габора. Ребра графа взвешиваются расстояниями между смежными вершинами. Различие (расстояние, дискриминационная характеристика) между двумя графами вычисляется при помощи некоторой ценовой функции деформации, учитывающей как

различие между значениями признаков, вычисленными в вершинах, так и степень деформации ребер графа»[1].

«Деформация графа происходит путем смещения каждой из его вершин на некоторое расстояние в определённых направлениях относительно ее исходного местоположения и выбора такой ее позиции, при которой разница между значениями признаков (откликов фильтров Габора) в вершине деформируемого графа и соответствующей ей вершине эталонного графа будет минимальной. Данная операция выполняется поочередно для всех вершин графа до тех пор, пока не будет достигнуто наименьшее суммарное различие между признаками деформируемого и эталонного графов. Значение ценовой функции деформации при таком положении деформируемого графа и будет являться мерой различия между входным изображением лица и эталонным графом. Данная «релаксационная» процедура деформации должна выполняться для всех эталонных лиц, заложенных в базу данных системы. Результат распознавания системы – эталон с наилучшим значением ценовой функции деформации. »[1]

«В отдельных публикациях указывается 95-97%-ая эффективность распознавания даже при наличии различных эмоциональных выражениях и изменении ракурса лица до 15 градусов. Однако разработчики систем эластичного сравнения на графах ссылаются на высокую вычислительную стоимость данного подхода.»[1]

Нейронные сети. В настоящее время существует около десятка разновидностей нейронных сетей (НС). Одним из самых широко используемых вариантов является сеть, построенная на многослойном перцептроне, которая позволяет классифицировать поданное на вход изображение/сигнал в соответствии с предварительной настройкой/обучением сети.

«Обучаются нейронные сети на наборе обучающих примеров. Суть обучения сводится к настройке весов межнейронных связей в процессе решения оптимизационной задачи методом градиентного спуска. В процессе

обучения НС происходит автоматическое извлечение ключевых признаков, определение их важности и построение взаимосвязей между ними. Предполагается, что обученная НС сможет применить опыт, полученный в процессе обучения, на неизвестные образы за счет обобщающих способностей»[1].

Наилучшие результаты в области распознавания лиц показывает Convolutional Neural Network или сверточная нейронная сеть (СНС), которая является логическим развитием идей таких архитектур НС как когнитрона и неокогнитрона. Успех обусловлен возможностью учета двумерной топологии изображения, в отличие от многослойного перцептрона.

«Отличительными особенностями СНС являются локальные рецепторные поля (обеспечивают локальную двумерную связность нейронов), общие веса (обеспечивают детектирование некоторых черт в любом месте изображения) и иерархическая организация с пространственными сэмплингом. Благодаря этим нововведениям СНС обеспечивает частичную устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям»[1].

Свое развитие СНС получили в разработке DeepFace [11], которую приобрел Facebook для распознавания лиц пользователей своей соцсети. Все особенности архитектуры носят закрытый характер.

«Недостатки нейронных сетей: добавление нового эталонного лица в базу данных требует полного переобучения сети на всем имеющемся наборе (достаточно длительная процедура, в зависимости от размера выборки от 1 часа до нескольких дней). Проблемы математического характера, связанные с обучением: попадание в локальный оптимум, выбор оптимального шага оптимизации, переобучение и т. д. Трудно формализуемый этап выбора архитектуры сети (количество нейронов, слоев, характер связей). Обобщая все вышесказанное, можно заключить, что НС – «черный ящик» с трудно интерпретируемыми результатами работы»[1].

Одним из наиболее известных и проработанных является метод главных компонент [8] (principal component analysis, PCA), основанный на преобразовании Карунена-Лоева.

«Первоначально метод главных компонент начал применяться в статистике для снижения пространства признаков без существенной потери информации. В задаче распознавания лиц его применяют главным образом для представления изображения лица вектором малой размерности (главных компонент), который сравнивается затем с эталонными векторами, заложенными в базу данных. Главной целью метода главных компонент является значительное уменьшение размерности пространства признаков таким образом, чтобы оно как можно лучше описывало «типичные» образы, принадлежащие множеству лиц. Используя этот метод можно выявить различные изменчивости в обучающей выборке изображений лиц и описать эту изменчивость в базисе нескольких ортогональных векторов, которые называются собственными (eigenface)»[1].

«Полученный один раз на обучающей выборке изображений лиц набор собственных векторов используется для кодирования всех остальных изображений лиц, которые представляются взвешенной комбинацией этих собственных векторов. Используя ограниченное количество собственных векторов можно получить сжатую аппроксимацию входному изображению лица, которую затем можно хранить в базе данных в виде вектора коэффициентов, служащего одновременно ключом поиска в базе данных лиц»[1].

«Суть метода главных компонент сводится к следующему. Вначале весь обучающий набор лиц преобразуется в одну общую матрицу данных, где каждая строка представляет собой один экземпляр изображения лица, разложенного в строку. Все лица обучающего набора должны быть приведены к одному размеру и с нормированными гистограммами. Метод главных компонент хорошо зарекомендовал себя в практических приложениях. Однако, в тех случаях, когда на изображении лица

присутствуют значительные изменения в освещенности или выражении лица, эффективность метода значительно падает. PCA выбирает подпространство с такой целью, чтобы максимально аппроксимировать входной набор данных, а не выполнить дискриминацию между классами лиц»[1].

Основные достоинства и недостатки рассмотренных методов приведены в таблице 1.

Таблица 1 – Анализ технологий распознавания лиц

Технология	Достоинства	Недостатки
Метод гибкого сравнения на графах (Elastic graph matching)	высокая эффективность распознавания даже при наличии различных эмоциональных выражениях и изменении ракурса лица до 15 градусов	высокая вычислительная стоимость данного подхода
Нейронные сети	Высокая эффективность при высоком быстродействии	проблемы математического характера, связанные с обучением
Метод главных компонент или principal component analysis (PCA)	Хорошо зарекомендовал себя в практических приложениях	в тех случаях, когда на изображении лица присутствуют значительные изменения в освещенности или выражении лица, эффективность метода значительно падает

На основании проведенного анализа в качестве технологии определения пола по изображению выбраны нейронные сети.

## 1.2 Определение пола человека по изображению с применением нейронных сетей

Нейросеть представляет собой систему из множества нейронов (процессоров). Нейрон – вычислительная единица, получающая информацию и производящая над ней какие-либо вычисления. По отдельности эти процессоры достаточно просты, но, будучи соединенными, они способны

выполнять очень сложные задачи. Искусственный нейрон содержит веса (параметр, благодаря которому входная информация изменяется), сумматор (элемент, осуществляющий суммирование сигналов, поступающих на его вход) и функцию активации, которая определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения. Схема искусственного нейрона [3] представлена на рисунке 1.

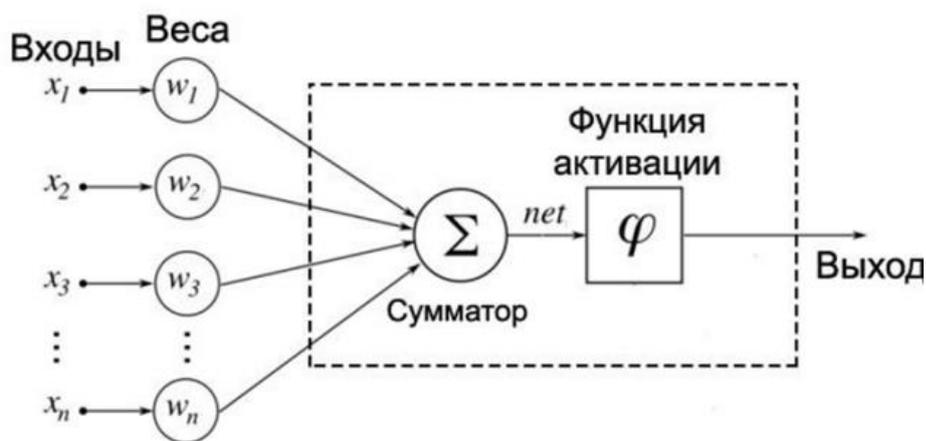


Рисунок 1 - Схема искусственного нейрона

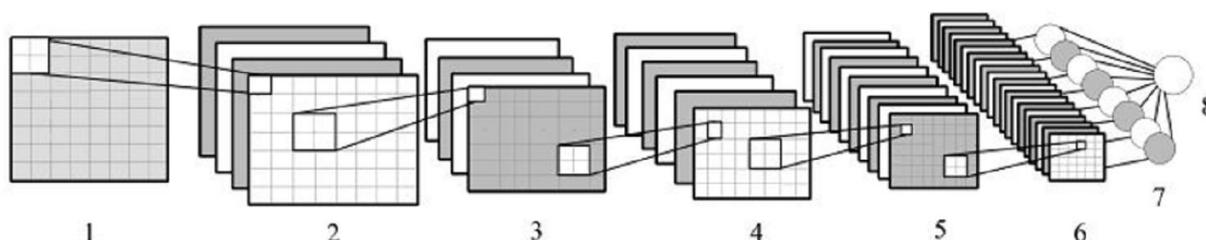
В работе для распознавания пола будут использоваться нейросети сверточного типа. Как правило, архитектура данной сети состоит из чередующихся сверточных и подвыборочных слоев. Схема сверточной нейронной сети [6] изображена на рисунке 2.

Работа сверточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям, и далее к еще более абстрактным.

Название архитектура получила из-за наличия операции свертки, выполняемой в сверточных слоях.

Другой тип слоев данной сети – подвыборочный (pooling). Он выполняет уменьшение размерности сформированных карт признаков. В данной архитектуре считается, что информация о факте наличия искомого

признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотненной карты признаков меньшей размерности. За счет данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к небольшим изменениям входного изображения, таким как небольшие повороты, небольшое масштабирование.



1 – вход; 2, 4, 6 – сверточные слои; 3, 5 – подвыборочные слои; 7 – слой из обычных нейронов; 8 - выход

Рисунок 2 - Схема сверточной нейронной сети

После сверточных и подвыборочных слоев следует классификатор в виде многослойного персептрона.

Структура сверточной сети – однонаправленная (без обратных связей), принципиально многослойная [2].

Построение нейронных сетей на Python обеспечивается применением целого ряда различных фреймворков к основным из которых относятся PyTorch, Caffe, Theano и TensorFlow [7].

Библиотека глубокого обучения Tensorflow [19] созданная Google и написанная на Python и C++ является одной из лучших открытых библиотек для численных вычислений. TensorFlow применяется для сложных проектов, таких как создание многослойных нейронных сетей. Она используется для распознавания голоса или картинок и приложений для работы с текстом,

таких как Google Translate.

«К положительным сторонам библиотеки TensorFlow относятся

- большое количество руководств и документации;
- мощные средства мониторинга процесса обучения моделей и визуализации (Tensorboard);
- поддерживается большим сообществом разработчиков и техническими компаниями;
- обеспечивает обслуживание моделей и поддерживает распределенное обучение.

Из минусов:

- более высокий входной порог для начинающих, чем PyTorch или Keras,
- чистая Tensorflow достаточно низкоуровневая и требует много шаблонного кода;
- единственный полностью поддерживаемый язык – Python.»[7]

Фреймворк PyTorch [18] является преемником Python для библиотеки Torch, написанной на Lua. Он был разработан Facebook и использовался Twitter, Salesforce, Оксфордским Университетом и многими другими компаниями.

«PyTorch позволяет обучать модели быстро и эффективно, у него есть множество важных преимуществ:

- благодаря архитектуре фреймворка, процесс создания модели достаточно прост и прозрачен;
- поддерживает декларативный параллелизм данных;
- имеет много предварительно обученных моделей и готовых модульных частей, которые легко комбинировать;
- поддерживает распределенное обучение»[7].

К минусам этого фреймворка можно отнести:

- недостаточная поддержка моделей;
- отсутствие интерфейсов для мониторинга и визуализации.

Keras [16] – это библиотека, основанная на Python, которая может запускаться поверх TensorFlow, Theano или CNTK. Она была разработана инженером компании Google, Франсуа Шолле, в целях ускорения экспериментов. Keras поддерживает широкий спектр слоев нейронных сетей, таких как сверточные слои, рекуррентные или плотные.

Этот фреймворк показал свою эффективность в кейсах для перевода, распознавании изображений, речи и т.п.

Преимущества:

- быстрое и простое прототипирование;
- имеет полностью конфигурируемые модули;
- имеет простой и интуитивно-понятный интерфейс и хорош для новичков;
- имеет встроенную поддержку для обучения на нескольких GPU;
- поддерживает GPU от NVIDIA, TPU от Google, GPU с Open-CL, такие как AMD.

К недостаткам Keras относятся:

- фреймворк ограничен бэкэндами Tensorflow, CNTK и Theano.
- Keras не настолько функционален как TensorFlow и дает меньше опций для управления сетевым соединением.

Caffe [10] – это фреймворк, реализованный на C++, который имеет интерфейс Python. Он поддерживает CNN и сети прямого распространения, а также хорош для тренировки моделей (без написания лишних строк кода), обработке картинок и усовершенствования существующих нейронных сетей. С другой стороны, он не так хорошо задокументирован и его сложно компилировать.

Для предотвращения переобучения модели может быть использована технология Dropout [5].

Dropout - это техника, при которой случайно выбранные нейроны игнорируются во время тренировки. Они выбывают случайно. Это означает, что их вклад в активацию нижестоящих нейронов временно удален на

прямом проходе, и любые обновления веса не применяются к нейрону на обратном проходе.

По мере того как нейронная сеть учится, веса нейронов попадают в их контекст внутри сети. Вес нейронов настроен на специфические особенности, обеспечивающие некоторую специализацию. Соседние нейроны начинают полагаться на эту специализацию, которая, если зайти слишком далеко, может привести к хрупкой модели, слишком специализированной для данных обучения. Это зависит от контекста для нейрона во время обучения относится к сложной коадаптации. Если нейроны случайно выпадают из сети во время обучения, то другие нейроны должны будут вмешаться и обработать представление, необходимое для прогнозирования отсутствующих нейронов. В результате сеть становится менее чувствительной к удельному весу нейронов. Это, в свою очередь, приводит к созданию сети, способной к лучшему обобщению и с меньшей вероятностью переобучения.

В работе [12] получены некоторые обобщающие результаты, которые следует учитывать при использовании Dropout на практике, в частности, необходимо использовать небольшое значение отсева 20% -50%. Слишком низкая вероятность имеет минимальный эффект, а слишком высокое значение приводит к недостаточному обучению сети.

### **1.3 Обзор аналогов**

Существует проект Py-agender [17] предназначенный для распознавания пола и возраста по фотографии. Проект использует обученную на датасете UTKFace [22] модель.

Набор данных UTKFace представляет собой крупномасштабный набор данных о лицах с большим возрастным диапазоном (от 0 до 116 лет). Набор данных состоит из более чем 20000 изображений лиц с аннотациями возраста, пола и этнической принадлежности. Изображения охватывают

большие вариации позы, выражения лица, освещения, окклюзии, разрешения и т. д.

Основные характеристики датасета:

- состоит из 20 000+ изображений лиц на естественном фоне (только одно лицо на одном изображении);
- включает дополнительный набор из обрезанных и выровненных лиц;
- изображения помечены по возрасту, полу и этнической принадлежности по пяти уровням с выделением белых, негров, азиатов, индийцев и других (например, латиноамериканцев, жителей Ближнего Востока).

Метки каждого изображения датасета встроены в имя файла в формате, [age]\_[gender]\_[race]\_[date&time].jpg, где:

- [age] целое число от 0 до 116, указывающее возраст;
- [gender] либо 0 (мужской), либо 1 (женский);
- [race] представляет собой целое число от 0 до 4, соответственно для белых, негров, азиатов, индийцев и других;
- [date&time] имеет формат ггггммддЧЧММССФФФ, показывая дату и время, когда изображение было добавлено в UTKFace.

По результатам анализа датасета с обрезанными лицами обнаружено, большое количество ошибочно размеченных фото, сильно замыленных фото и фото из фотобанков с «водяными знаками» на лицах. Примеры ошибочных по гендерным меткам элементов датасета показаны на рисунке 3.

Элементы 49\_1\_0\_20170104185000777 и 72\_1\_0\_20170104213629773 – ошибка в идентификации пола – это фото мужчин классифицированные как женские. Элементы 22\_0\_3\_20170119162922591, 38\_0\_3\_20170119202607101, 35\_0\_3\_20170119181105779, 36\_0\_3\_20170119180245724, 15\_0\_0\_20170116201248395 и 30\_0\_4\_20170117154305388 ошибка в идентификации пола – это фото женщин классифицированные как мужские. Элемент 76\_0\_1\_20170116205556715 не является фотографией – это рисунок.

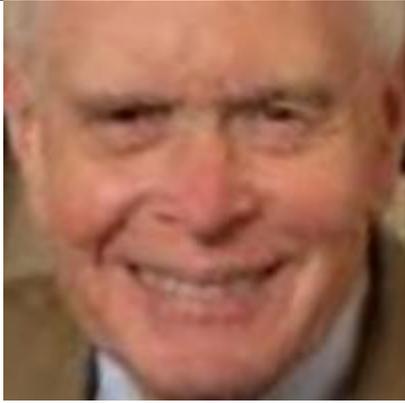
		
49_1_0_20170104185000777	72_1_0_20170104213629773	76_0_1_20170116205556715
		
22_0_3_20170119162922591	38_0_3_20170119202607101	35_0_3_20170119181105779
		
36_0_3_20170119180245724	15_0_0_20170116201248395	30_0_4_20170117154305388

Рисунок 3 – Примеры классификационных отклонений фотографий датасета  
UTKFace

В целом проект Ru-agender содержит избыточную функциональность (определяет не только пол, но и возраст) и построен на недостаточно качественном в гендерной разметке датасете.

Аналогичный проект Gender-and-Age-Detection [14] направлен на определение пола и возраста в диапазонах (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100). В проекте используются модели, обученные Талом Хаснером и Гилом Леви из открытого университета Израиля [20].

Результаты работы данной программы (рисунок 4), представленные в качестве демонстрационных на странице проекта в гитхабе, свидетельствуют о недостаточно четком распознавании возраста.

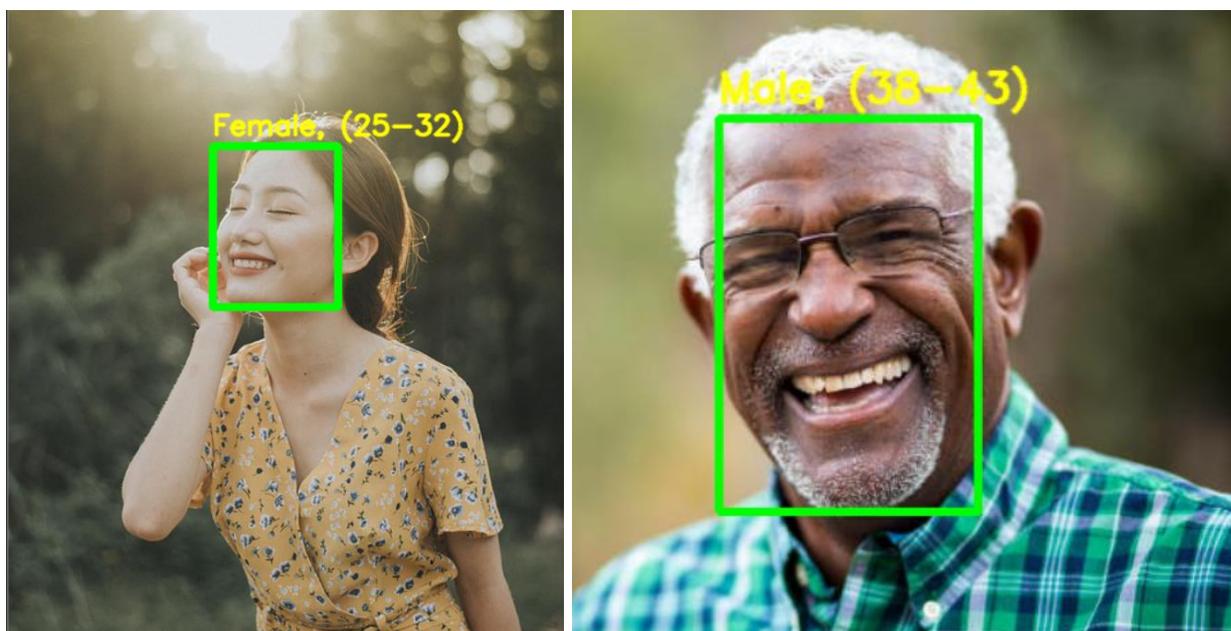


Рисунок 4 – Демонстрация программы распознавания возраста и пола по фотографии [14]

Если пол и возраст азиатки распознаны верно, то негр выглядит старше предсказанного диапазона 38-43 лет.

Для этого проекта Python использован набор данных Adience доступный в открытом доступе. Этот набор данных служит эталоном для фотографий лиц и включает в себя различные реальные условия изображения, такие как шум, освещение, поза и внешний вид. Изображения

были собраны из альбомов Flickr и распространены по лицензии Creative Commons (CC). Он содержит в общей сложности 26 580 фотографий 2284 человек в восьми возрастных группах (как упоминалось выше) и имеет размер около 1 ГБ.

По результатам обзора аналогов можно заключить, что задача определения пола по фотографии чаще всего решается совместно с определением возраста. При этом определение пола существенно точнее, чем определение возраста. В соответствии с целью работы целесообразно вычленив модель определения пола в отдельную задачу и решить ее, обращая внимание на качество обучающих датасетов

Выводы по разделу:

В первом разделе проведен анализ современных технологий распознавания лиц, изучено применение нейронных сетей для решения этой задачи и рассмотрены программы-аналоги.

## 2 Разработка программного обеспечения

### 2.1 Выбор средств разработки

При выборе средств разработки в качестве языка программирования выбран Python, как наиболее широко используемый язык в построении нейронных сетей.

В качестве IDE использована PyCharm компании JetBrains [15]. Эта среда разработки ориентирована на Python и обладает широким и функциональными возможностями и удобным интеллектуальным пользовательским интерфейсом.

Построение нейронных сетей на Python обеспечивается применением целого ряда различных фреймворков, рассмотренных в первом разделе. В соответствии с поставленной задачей при выборе библиотек необходимо ориентироваться на такие критерии как

- функциональность;
- документированность;
- наличие большого интернет-сообщества разработчиков;
- простота освоения и использования;
- опыт применения в задачах распознавания изображений.

Результаты анализа описания библиотек для нейронных сетей представлены в таблице 2. Оценка проведена по десятибалльной шкале. Где 10 соответствует максимальному выражению критерия, 0 – минимальной выраженности критерия.

Исходя из типа решаемой задачи – определение пола по изображению – и на основании максимального суммарного числа баллов в таблице 99 в качестве фреймворка выбрана надстройка легкой в освоении библиотеки Keras над широко функциональным и хорошо документированным, но сложным для освоения в «чистом виде» Tensorflow.

Таблица 2 – Сравнительный анализ библиотек для нейронных сетей

Критерий	Tensorflow	PyTorch	Keras+ Tensorflow	Caffe
Функциональность	10	5	10	5
Документированность;	10	5	5	2
Наличие большого интернет-сообщества разработчиков;	10	5	5	3
Простота освоения и использования;	0	10	10	5
Опыт применения в задачах распознавания изображений.	5	0	10	10
Сумма	35	25	40	25

Выбранные средства разработки являются современными и специализированными для решения поставленной задачи.

## 2.2 Подготовка данных для обучения

По результатам обзора показано, что датасет UTKFace фотографий лиц используемые для обучения нейросетей не полностью отвечает требованиям к точности разметки по половому признаку. С другой стороны задача определения пола по фотографии является менее сложной по сравнению с определением возраста и для ее решения возможно достаточно будет относительно небольшой обучающей выборки, сформированной вручную.

В качестве источника изображений выбран сайт-генератор реалистичных лиц [21], на котором раз в несколько секунд генерируется человеческое лицо с помощью алгоритма генеративных нейронных сетей StyleGAN [9], разработанного Nvidia.

Для того чтобы скачать 2000 картинок с сайта использован следующий код:

```
import shutil
import time
import requests
```

```

while (i < 2000):
    url = 'https://thispersondoesnotexist.com/image'
    response = requests.get(url, stream=True)
    with open(str(i)+'.jpg', 'wb') as out_file:
        shutil.copyfileobj(response.raw, out_file)
    del response
    i = i + 1
    time.sleep(1)

```

После получения 2000 изображений методом ручной сортировки сформировано 2 выборки мужчин и женщин. Избыточность исходных данных сделана по причине, что некоторые фотографии затруднительно идентифицировать по половому признаку, например, показанные на рисунке 5. В ходе ручной сортировки файлы с фотографиями мужчин помещали в папку man, женщин – в папку woman.



Рисунок 5 – Пример сгенерированных «бесполовых» фотографий

На следующем этапе необходимо переименовать файлы с применением функции, которая берет файл из определенной папки, по номеру итерации цикла создает ему имя и переносит в другую папку:

```

import os
# M
i = 0
from_ = 'C:\PycharmProjects\polpofoto\man'

```

```

to_ = 'C:\PycharmProjects\ polpofoto\m'
name1 = os.listdir(from_)
g = ''
while (i < len(name1)):
    print(str(i))
    os.rename(from_ + "/" + name1[i], to_ + "/" + '' + str(i) + g +
'.jpg')
    i = i + 1
# Ж
i = 0
from_ = 'C:\PycharmProjects\polpofoto\woman'
to_ = 'C:\PycharmProjects\polpofoto\w'
name1 = os.listdir(from_)
g = ''
while (i < len(name1)):
    print(str(i))
    os.rename(from_ + "/" + name1[i], to_ + "/" + '' + str(i) + g +
'.jpg')

```

В итоге получаются файлы изображений (0.jpg ... 699.jpg) в папках m и w соответственно

Следующий этап обработки изображений - преобразовать фото в нужный формат: перевести цвет в градации серого и сделать разрешение 299×299 с сохранением новых изображений в папки m2 и w2 соответственно:

```

import cv2
import os
# M
i = 0
from_ = 'C:\PycharmProjects\polpofoto\m'
to_ = 'C:\PycharmProjects\polpofoto\m2'
name1 = os.listdir(from_)
while (i < len(name1)):
    print(str(i))
    img = cv2.imread(from_ + '/' + str(i) + '.jpg',
cv2.IMREAD_UNCHANGED)
    gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    resized = cv2.resize(gray_image, (299, 299),
interpolation=cv2.INTER_AREA)
    cv2.imwrite(to_ + '/' + str(i) + '.jpg', resized)
    i = i + 1
# Ж
i = 0
from_ = 'C:\PycharmProjects\polpofoto\w'
to_ = 'C:\PycharmProjects\polpofoto\w2'
name1 = os.listdir(from_)
while (i < len(name1)):
    print(str(i))

```

```
img = cv2.imread(from_ + '/' + str(i) + '.jpg',  
cv2.IMREAD_UNCHANGED)  
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
resized = cv2.resize(gray_image, (299, 299),  
interpolation=cv2.INTER_AREA)  
cv2.imwrite(to_ + '/' + str(i) + '.jpg', resized)  
i = i + 1
```

Фотография до и после преобразования, показанная на рисунке 6, визуально не утратила информативности по определению пола.



Рисунок 6 – Преобразование изображения

На увеличенном фрагменте видно, что изображение существенно понижено в качестве – этим обеспечивается уменьшение количества данных на входе в нейронную сеть.

### 2.3 Проектирование нейросети

Для модели использован подкласс `Sequential` класса `Model` в `Keras`, представляющий собой линейный стек слоев [4].

В качестве первого входного слоя использован слой двумерной свертки `Conv2D`. Этот слой создает ядро свертки, которое помогает создавать тензор выходных данных. В качестве параметров слоя выбраны: размерность выходного пространства (количество выходных фильтров в свертке) 2, размер окна свертки 3, функция активации – `relu`, параметр `padding='same'` означает, что выходная карта признаков на каждом канале должна быть той же размерностью, что и исходное изображение, что означает добавление нулей на границах двумерных данных, чтобы центр ядра фильтра мог размещаться над граничными элементами, размер входного изображения `input_shape=(299, 299, 1)` – 299x299 пикселей, однослойное (градации серого).

Слой `MaxPooling2D` уменьшает выборку входных данных по пространственным измерениям (высоте и ширине), беря максимальное значение в окне ввода для каждого канала входных данных.

Следующий слой повторяет свертку `Conv2D` для данных, полученных на выходе предыдущего слоя.

Затем снова выполняется уменьшение выборки `MaxPooling2D` и выравнивание данных `Flatten`.

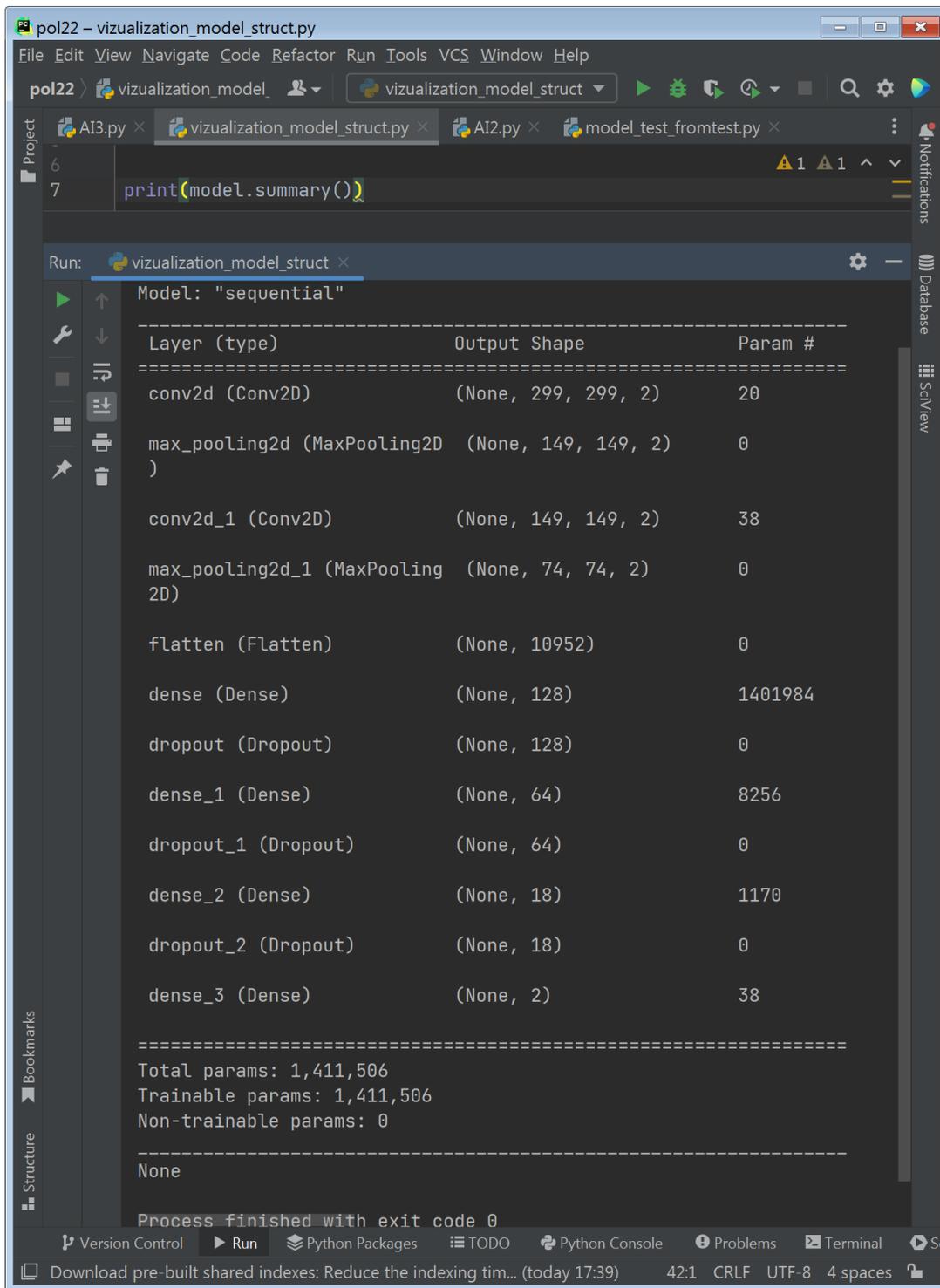
После этого модель содержит три полносвязных скрытых слоя `Dense` размерами 128, 64 и 18 нейронов с функцией активации `relu`.

Выходной слой также полносвязный, состоящий из двух нейронов (т.к. определение пола это бинарная задача) с функцией активации `softmax`.

Для управления переобучением модели используются дополнительные

слои Dropout () с параметром, отражающим долю отключаемых нейронов при переходе между скрытыми полносвязными слоями.

Отчет по структуре созданной модели показан на рисунке 7



```
pol22 - visualization_model_struct.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
pol22 > visualization_model_struct
print(model.summary())

Run: visualization_model_struct x
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 299, 299, 2)        20
max_pooling2d (MaxPooling2D) (None, 149, 149, 2)        0
conv2d_1 (Conv2D)            (None, 149, 149, 2)        38
max_pooling2d_1 (MaxPooling2D) (None, 74, 74, 2)        0
flatten (Flatten)            (None, 10952)              0
dense (Dense)                 (None, 128)                1401984
dropout (Dropout)            (None, 128)                0
dense_1 (Dense)               (None, 64)                 8256
dropout_1 (Dropout)          (None, 64)                 0
dense_2 (Dense)               (None, 18)                 1170
dropout_2 (Dropout)          (None, 18)                 0
dense_3 (Dense)               (None, 2)                  38
-----
Total params: 1,411,506
Trainable params: 1,411,506
Non-trainable params: 0
-----
None

Process finished with exit code 0
```

Рисунок 7 - Структура модели

Отчет по структуре модели включает

- слои и их порядок в модели;
- форма вывода каждого слоя;
- количество параметров (весов) в каждом слое;
- общее количество параметров (весов) в модели.

При компиляции модели в качестве функции потерь выбрана биномиальная кросс-энтропия.

## 2.4 Обучение сети

Из 1400 размеченных фотографий (700 мужских и 700 женских) для обучения выбрали 1000 фотографий (500 мужских и 500 женских), для валидации выбрали 200 фотографий (100 мужских и 100 женских), для тестирования выбрали 200 фотографий (100 мужских и 100 женских):

```
data = []
data1 = []
vdata = []
vdata1 = []
tdata = []
tdata1 = []
i = 0
# учебная выборка
while (i < len(os.listdir('C:\PycharmProjects\pol122\w2'))-200):
    data.append(np.array(cv2.imread('C:\PycharmProjects\pol122\m2/' +
str(i) + '.jpg', cv2.IMREAD_UNCHANGED)) / 255)
    data.append(np.array(cv2.imread('C:\PycharmProjects\pol122\w2/' +
str(i) + '.jpg', cv2.IMREAD_UNCHANGED)) / 255)
    data1.append([0., 1.])
    data1.append([1., 0.])
    i = i + 1
# валидация
while (i < len(os.listdir('C:\PycharmProjects\pol122\w2'))-100):
    vdata.append(
        np.array(cv2.imread('C:\PycharmProjects\pol122\m2/'+str(i)
+ '.jpg', cv2.IMREAD_UNCHANGED)) / 255)
    vdata.append(
        np.array(cv2.imread('C:\PycharmProjects\pol122\w2/'+ str(i) +
'.jpg', cv2.IMREAD_UNCHANGED)) / 255)
    vdata1.append([0., 1.])
    vdata1.append([1., 0.])
    i = i + 1
```

```

# тестирование
while (i < len(os.listdir('C:\PycharmProjects\pol22\w2'))):
    tdata.append(
        np.array(cv2.imread('C:\PycharmProjects\pol22\m2/' +
str(i) + '.jpg', cv2.IMREAD_UNCHANGED)) / 255)
    tdata.append(
        np.array(cv2.imread('C:\PycharmProjects\pol22\w2/' +
str(i) + '.jpg', cv2.IMREAD_UNCHANGED)) / 255)
    tdata1.append([0., 1.])
    tdata1.append([1., 0.])
    i = i + 1

```

Обучение выполнено с использованием метода регуляризации (борьбы с переобучением модели) Dropout.

На первом этапе в течение 10 эпох исследовали влияние настроек Dropout. На рисунке 8 приведены данные функции потерь валидационной выборки при обучении без Dropout, с Dropout(0,2) и с Dropout(0,5).

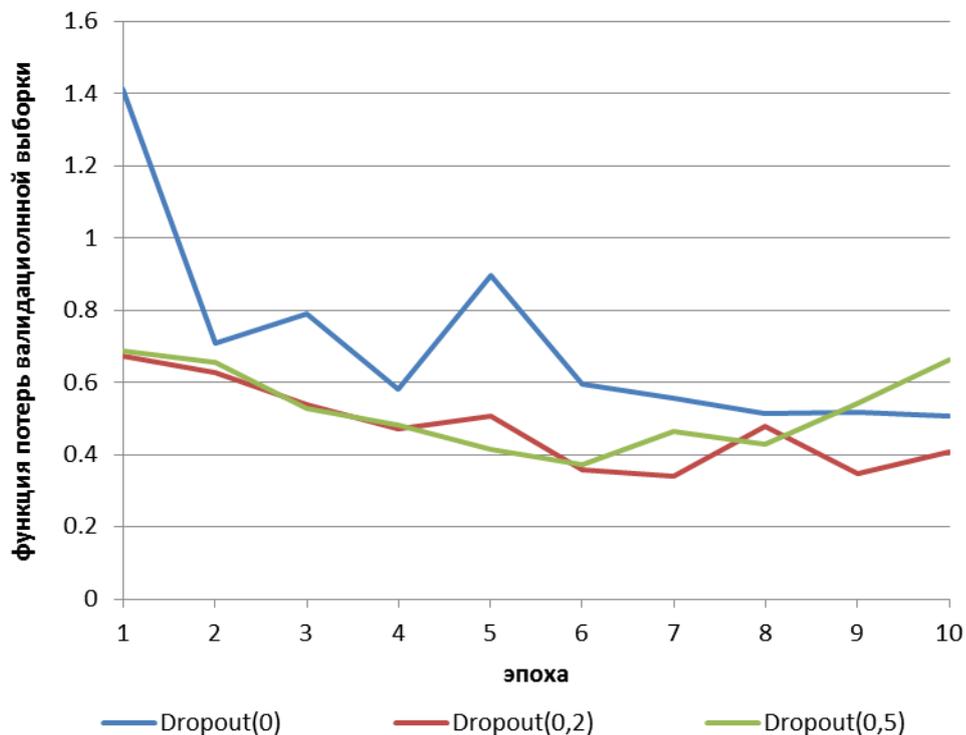


Рисунок 8 – Влияние настроек Dropout на функцию потерь валидационной выборки

По рисунку 99 видно, что предпочтительным вариантом будет обучение с Dropout(0,2) длиной 6 или 7 эпох. При этих условиях достигается минимальное значение функции потерь валидационной выборки.

Полные результаты по функции потерь для тренировочной, валидационной и тестовой выборок (таблица 3) также показывают предпочтительность варианта с Dropout(0,2) т.к. при этом минимальна функция потерь на тестовой выборке.

Таблица 3 – Функция потерь при обучении модели на 10 эпохах

Эпоха	Тренировочная выборка			Валидационная выборка		
	Dropout(0)	Dropout(0,2)	Dropout(0,5)	Dropout(0)	Dropout(0,2)	Dropout(0,5)
1	1.2437	0.7325	0.6945	1.4135	0.6733	0.6874
2	0.9692	0.6428	0.6815	0.7101	0.6281	0.657
3	0.6538	0.5791	0.6242	0.789	0.5397	0.5282
4	0.6701	0.4655	0.5002	0.5811	0.4703	0.4835
5	0.606	0.3649	0.3883	0.8953	0.5077	0.413
6	0.6026	0.403	0.2936	0.5943	0.3587	0.3709
7	0.5813	0.2473	0.1873	0.5574	0.3403	0.4642
8	0.5313	0.2247	0.1451	0.5152	0.4769	0.4304
9	0.523	0.1643	0.1112	0.5174	0.3477	0.5412
10	0.4887	0.1202	0.0764	0.5067	0.4077	0.6629
Функция потерь тестовой выборки				0.5158	0.3984	0.5472

Результаты обучения модели на семи эпохах показаны на рисунке 9.

Результаты обучения модели на шести эпохах приведены в таблице 4

Таблица 4 – Результаты обучения модели на шести эпохах

Эпоха	Тренировочная выборка		Валидационная выборка	
	Функция потерь	Точность	Функция потерь	Точность
1	0.6918	0.526	0.6801	0.500
2	0.6338	0.664	0.5797	0.665
3	0.5196	0.763	0.4898	0.780
4	0.4382	0.817	0.3811	0.815
5	0.3746	0.848	0.3506	0.860
6	0.2993	0.891	0.3375	0.865
Тестовая выборка			0.3421	0.865

```
pol22 - AI3.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help

pol22 AI3.py
Project AI3.py AI2.py model_test_fromtest.py
59 model = Sequential()
60 model.add(Conv2D(2, 3, activation='relu', padding='same', input_shape=(299, 299, 1)))
61 model.add(MaxPooling2D())
62 model.add(Conv2D(2, 3, activation='relu', padding='same'))
63 model.add(MaxPooling2D())
64 model.add(Flatten())
65 model.add(Dense(128, activation='relu'))
66 model.add(Dropout(0.2))
67 model.add(Dense(64, activation='relu'))
68 model.add(Dropout(0.2))
69 model.add(Dense(18, activation='relu'))
70 model.add(Dropout(0.2))
71 model.add(Dense(2, activation='softmax'))
72
73 model.compile(loss='categorical_crossentropy', metrics=['accuracy'])
74
75 history = model.fit(np.array(data), np.array(data1), epochs=7, validation_data=(np.array(vdata), np.array(vdata1)))
76 results = model.evaluate(np.array(tdata), np.array(tdata1))

Run: AI3 x
2023-02-18 22:07:48.518109: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Epoch 1/7
32/32 [=====] - 24s 719ms/step - loss: 0.6918 - accuracy: 0.5260 - val_loss: 0.6801 - val_accuracy: 0.5000
Epoch 2/7
32/32 [=====] - 23s 710ms/step - loss: 0.6338 - accuracy: 0.6640 - val_loss: 0.5797 - val_accuracy: 0.6650
Epoch 3/7
32/32 [=====] - 23s 709ms/step - loss: 0.5196 - accuracy: 0.7630 - val_loss: 0.4898 - val_accuracy: 0.7800
Epoch 4/7
32/32 [=====] - 23s 707ms/step - loss: 0.4382 - accuracy: 0.8170 - val_loss: 0.3811 - val_accuracy: 0.8150
Epoch 5/7
32/32 [=====] - 23s 707ms/step - loss: 0.3746 - accuracy: 0.8480 - val_loss: 0.3506 - val_accuracy: 0.8600
Epoch 6/7
32/32 [=====] - 23s 708ms/step - loss: 0.2993 - accuracy: 0.8910 - val_loss: 0.3375 - val_accuracy: 0.8650
Epoch 7/7
32/32 [=====] - 23s 718ms/step - loss: 0.2461 - accuracy: 0.9150 - val_loss: 0.3893 - val_accuracy: 0.8600
7/7 [=====] - 2s 238ms/step - loss: 0.3937 - accuracy: 0.8550
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _update_step_xla while saving.
Process finished with exit code 0

Version Control Run Python Packages TODO Python Console Problems Terminal Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Down... (today 9:28) 77:1 CRLF UTF-8 4 spaces
```

Рисунок 9 – Результаты обучения модели на семи эпохах

Из представленных результатов обучения видно, что модель, обученная на семи эпохах обеспечивает на тестовой выборке точность 0,855 при величине функции потерь 0,3937, а модель, обученная на шести эпохах обеспечивает на тестовой выборке точность 0,865 при величине функции потерь 0,3421, следовательно, предпочтительнее использовать модель, обученную на шести эпохах.

Выводы по разделу:

Во втором разделе выполнена разработка программы определения пола человека по изображению, разработан обучающий датасет, проведено обучение нейросети.

### **3 Тестирование программного обеспечения**

#### **3.1 План тестирования и подготовка данных**

На первом этапе тестирования оценено как работает программа на размеченной тестовой выборке 200 фотографий, которая была использована при обучении сети. Оцениваемые параметры – количество неправильно идентифицированных фотографий и выборочная визуальная оценка неправильно идентифицированных фотографий.

На втором этапе тестирование выполнено на неразмеченной выборке 1000 фотографий просто загруженных с сайта генератора фотографий [21] без предварительного визуального отсева. Оцениваемые параметры – визуальная оценка фотографий с краев и из середины шкалы критерия принадлежности.

На третьем этапе тестирования выполнена проверка как работает программа на фотографиях реальных людей.

Для визуальной оценки определены фотографии известных людей из открытых источников. Для каждого пола выбраны фотографии двух персон. Персона 1 – типичный представитель, персона 2 – нетипичный представитель, на котором интеллектуальная система потенциально может ошибиться (мужчина похожий на женщину и женщина похожая на мужчину). Для тестирования системы приняты фотографии следующих персон:

- артиста Сергея Шнурова;
- певца Элтона Джона;
- спортсменки Мартины Навратиловой;
- артистки Мэрилин Монро.

По результатам тестирования необходимо установить:

- правильность определения пола по критерию принадлежности.

- возможность оценивания сравнительной визуальной женоподобности мужских фотографий по величине критерия принадлежности;
- возможность оценивания сравнительной визуальной мужеподобности женских фотографий по величине критерия принадлежности.

Разработанный план тестирования обеспечит проверку работоспособности программы, как на сгенерированных, так и на реальных данных.

### **3.2 Результаты тестирования**

Результаты тестирования программы на размеченной выборке из 200 фотографий (100 мужчин и 100 женщин) для мужчин:

- неправильно идентифицировано 13 фотографий;
- среднее значение критерия принадлежности к мужчинам 0,83;
- среднее значение критерия принадлежности к женщинам 0,17.

Результаты тестирования размеченной выборки для женщин:

- неправильно идентифицировано 10 фотографий;
- среднее значение критерия принадлежности к женщинам 0,88;
- среднее значение критерия принадлежности к мужчинам 0,12.

Визуальная оценка неправильно идентифицированных фотографий позволяет заключить, что программа неуверенно распознает фотографии, на которых можно затрудниться и при визуальной оценке, например, фотографии детей 3 и 6. Также встречаются среди неправильно идентифицированных фотографии с поворотом головы, например, фотография 2. Но наряду с этим имеют место и явные ошибки, как например, фотография 1.

В целом точность модели, показанная на размеченном тесте 0,83 для мужчин и 0,88 для женщин соответствует точности 0.865, полученной для тестовой выборки на этапе обучения.

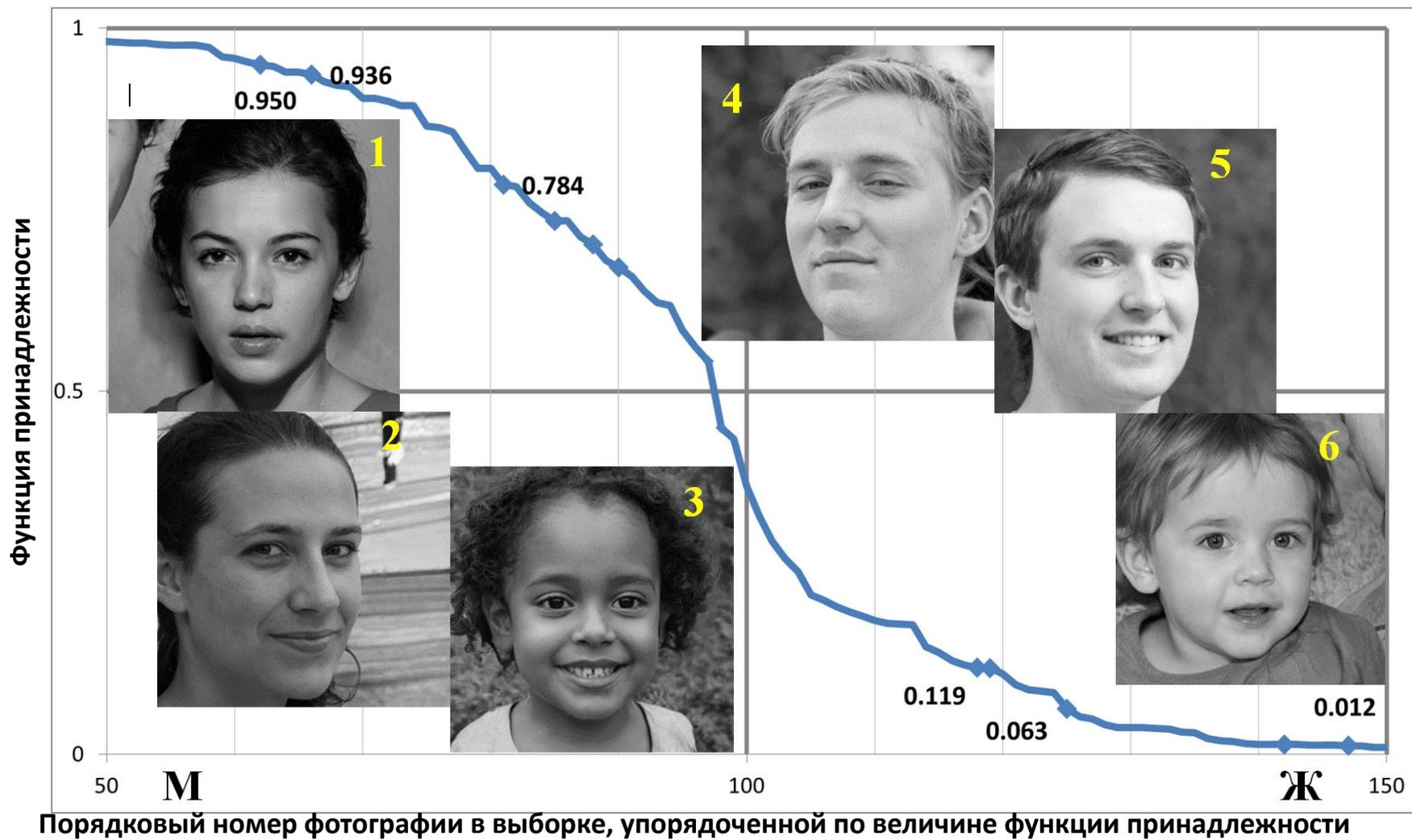


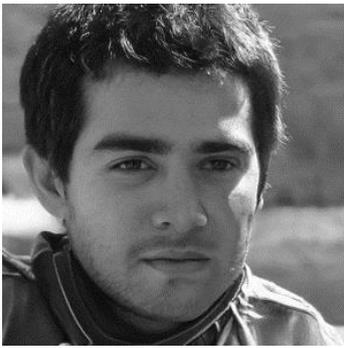
Рисунок 10 - График изменения критерия принадлежности и типичные фотографии неправильной идентификации по размеченной тестовой выборке

График изменения критерия принадлежности и типичные фотографии неправильной идентификации по размеченной тестовой выборке показаны на рисунке 10.

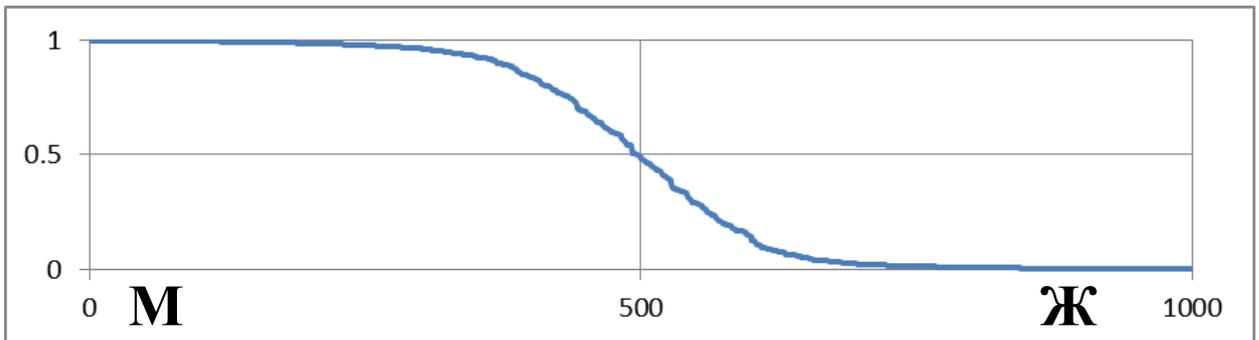
Результаты тестирования программы на неразмеченной выборке из 1000 фотографий показаны на рисунке 11. Кривая принадлежности проходит близко от центра с координатами (500; 0,5) что свидетельствует о равномерности распределения результата и сбалансированности модели.

Примеры фотографий с концов кривой принадлежности показывают, что программа уверенно распознает мужчин и женщин по неразмеченной выборке. Примеры фотографий из середины кривой принадлежности вызывают затруднение при визуальной идентификации пола.

Тестирование по неразмеченной выборке сгенерированных фотографий показывает, что программа работоспособна.



Функция принадлежности



Порядковый номер фотографии в выборке, упорядоченной по величине функции принадлежности

Рисунок 11 - Результаты тестирования программы на неразмеченной выборке

Результаты тестирования программы на фотографиях реальных персон показаны на рисунке 12

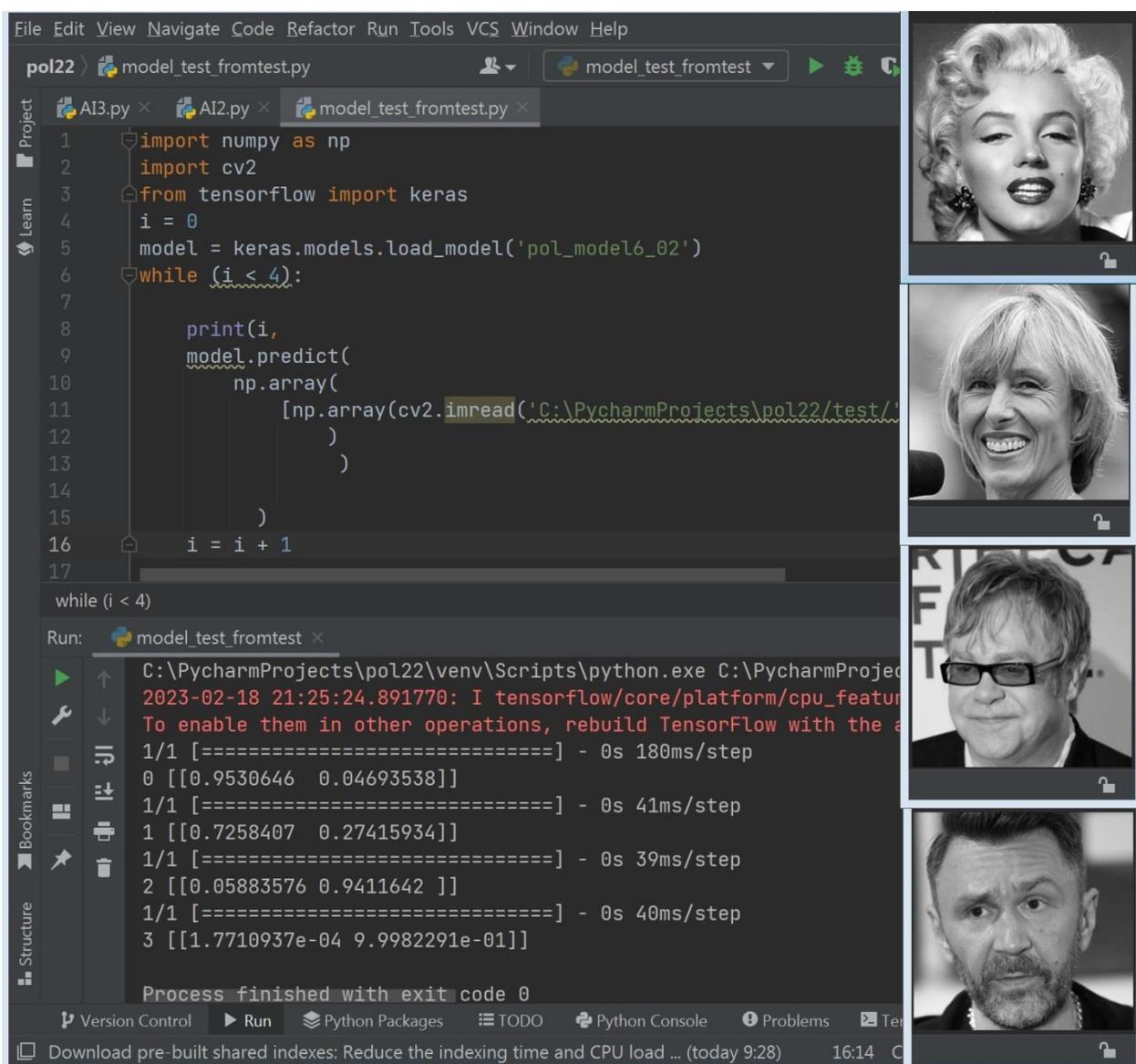


Рисунок 12 – Результаты тестирования на фотографиях реальных персон

Данные рисунка 12 свидетельствуют о том, что:

- программа правильно определила пол по всех четырех случаях;
- программа отражает сравнительную визуальную женоподобность мужских фотографий, значение критерия принадлежности для артиста

Шнурова составляет 0,999 и больше, чем для певца Элтона Джона 0,941.

– программа отражает сравнительную визуальную мужеподобность женских фотографий, значение критерия принадлежности для артистки Мэрилин Монро составляет 0,953 и больше, чем для спортсменки Мартины Навратиловой 0,726.

В целом разработанная программа работает корректно - определяет пол по фотографиям реальных людей.

Выводы по разделу:

В третьем разделе проведено тестирование разработанной программы – разработан набор тестовых данных, проведено тестирование и проанализированы полученные результаты.

## Заключение

В ходе анализа современных методов определения пола человека по изображению рассмотрены такие методы как Метод гибкого сравнения на графах (Elastic graph matching), Нейронные сети Метод главных компонент или principal component analysis (PCA) и определены их достоинства и недостатки. В результате выбран метод нейронных сетей.

В ходе анализа применения нейронных сетей определен тип нейросети – сверточная и описаны основные приемы проектирования сверточных нейронных сетей.

Разработанная программа для определения пола по изображению основана на сверточной нейронной сети. При обучении программы использован датасет основанный на автоматической генерации фотографий лиц и ручной разметке. По результатам обучения с использованием технологии Dropout достигнута точность определения 0,86 на валидационной выборке.

Тестирование разработанной программы выполнено для трех случаев – по размеченному датасету сгенерированных фотографий, по неразмеченному датасету сгенерированных фотографий и по фотографиям реальных персон. По результатам тестирования программа работает корректно в пределах полученной в ходе обучения точности.

Разработанная программа позволяет определять пол человека по изображению, как на сгенерированных фотографиях, так и на фотографиях реальных персон.

## Список используемой литературы и используемых источников

1. Анализ существующих подходов к распознаванию лиц [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/synesis/blog/238129/> (дата обращения: 15.05.2023)
2. Маршалко Д.А., Кубанских О.В. Архитектура свёрточных нейронных сетей // Ученые записки Брянского государственного университета. 2019. №4 (16). URL: <https://cyberleninka.ru/article/n/arhitektura-svyortochnyh-neuronnyh-setey> (дата обращения: 21.02.2023).
3. Наглядно о том, как работает свёрточная нейронная сеть [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/skillfactory/blog/565232/> (дата обращения: 15.05.2023)
4. Обучение и оценка модели с Keras [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/485890/> (дата обращения: 15.05.2023)
5. Регуляризация отсева в моделях глубокого обучения с Keras [Электронный ресурс] — Режим доступа: <https://machinelearningmastery.ru/dropout-regularization-deep-learning-models-keras/> (дата обращения: 15.05.2023).
6. Сикорский О.С. Обзор свёрточных нейронных сетей для задачи классификации изображений // Новые информационные технологии в автоматизированных системах. 2017. №20. URL: <https://cyberleninka.ru/article/n/obzor-svyortochnyh-neuronnyh-setey-dlya-zadachi-klassifikatsii-izobrazheniy> (дата обращения: 21.02.2023).
7. Сравнение фреймворков для глубокого обучения: TensorFlow, PyTorch, Keras, MXNet, Microsoft Cognitive Toolkit, Caffe, [Электронный ресурс]. URL: <https://habr.com/ru/company/otus/blog/443874/> (дата обращения: 11.02.2023).
8. Шерстобитов А.И., Федосов В.П., Приходченко В.А., Тимофеев М. В. Распознавание лиц на групповых фотографиях с использованием

алгоритмов сегментации // Известия ЮФУ. Технические науки. 2013. №11 (148). URL: <https://cyberleninka.ru/article/n/raspoznavanie-lits-na-grupповыh-fotografiyah-s-ispolzovaniem-algoritmov-segmentatsii> (дата обращения: 20.02.2023).

9. A Style-Based Generator Architecture for Generative Adversarial Networks [Электронный ресурс] — Режим доступа: [https://drive.google.com/file/d/1v-HkF3Ehrpon7wVIx4r5DLcko\\_U\\_V6Lt/view](https://drive.google.com/file/d/1v-HkF3Ehrpon7wVIx4r5DLcko_U_V6Lt/view) (дата обращения: 15.05.2023)

10. Caffe [Электронный ресурс] — Режим доступа: <https://caffe.berkeleyvision.org/> (дата обращения: 15.05.2023)

11. DeepFace Closing the Gap to Human-Level Performance in Face Verification [Электронный ресурс] — Режим доступа: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2014/papers/Taigman\\_DeepFace\\_Closing\\_the\\_2014\\_CVPR\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Taigman_DeepFace_Closing_the_2014_CVPR_paper.pdf) (дата обращения: 15.05.2023)

12. Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Электронный ресурс] — Режим доступа: <https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf> (дата обращения: 15.05.2023).

13. Face Recognition by Elastic Bunch Graph Matching [Электронный ресурс] — Режим доступа: <https://face-rec.org/algorithms/EBGM/WisFelKrue99-FaceRecognition-JainBook.pdf> (дата обращения: 15.05.2023)

14. Gender-and-Age-Detection [Электронный ресурс] — Режим доступа: <https://github.com/smahesh29/Gender-and-Age-Detection> (дата обращения: 15.05.2023)

15. JetBrains [Электронный ресурс] — Режим доступа: <https://www.jetbrains.com/pycharm/> (дата обращения: 15.05.2023)

16. Keras: the Python deep learning API [Электронный ресурс] — Режим доступа: <https://keras.io/> (дата обращения: 15.05.2023)

17. PyAgender. Face Age & Gender detection tool [Электронный ресурс] — Режим доступа: <https://www.opensourceagenda.com/projects/py-agender> (дата обращения: 15.05.2023)
18. PyTorch [Электронный ресурс] — Режим доступа: <https://pytorch.org/> (дата обращения: 15.05.2023)
19. TensorFlow [Электронный ресурс] — Режим доступа: <https://www.tensorflow.org/?hl=ru> (дата обращения: 15.05.2023)
20. The Open University of Israel and Adience. Face Image Project [Электронный ресурс] — Режим доступа: <https://talhassner.github.io/home/projects/Adience/Adience-data.html> (дата обращения: 15.05.2023)
21. ThisPersonDoesNotExist [Электронный ресурс] — Режим доступа: <https://thispersondoesnotexist.com/> (дата обращения: 15.05.2023)
22. UTKFace Large Scale Face Dataset [Электронный ресурс] — Режим доступа: <https://susanqq.github.io/UTKFace/> (дата обращения: 15.05.2023)