МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра биомедицинской информатики

СИЗОВА

Дарья Вячеславовна

РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММНЫХ СРЕДСТВ ПОИСКА «ЦИФРОВЫХ ДВОЙНИКОВ» ПО РЕНТГЕНОВСКИМ ИЗОБРАЖЕНИЯМ

Дипломная работа

Научный руководитель:

Кандидат технических наук, доцент кафедры БМИ,

Ковалёв В. А.

Допущена к защите

«___» _____ 2022

Зав. кафедрой биомедицинской информатики

кандидат физ.-мат. наук, доцент Ю.Л. Орлович

Минск, 2022

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

BRIEF – бинарные сложные независимые элементарные признаки (Binary Robust Independent Elementary Features)

CBIR – поиск изображений по содержанию (Content-Based Image Retrieval)

FAST – признаки из ускоренного сегментного теста (Features from Accelerated Segment Test)

MDS – многомерное шкалирование (Multidimensional Scaling)

ORB – ориентированный FAST и повернутый BRIEF (Oriented FAST and Rotated BRIEF)

SIFT – масштабно-инвариантная трансформация признаков (Scale-Invariant Feature Transform)

РЕФЕРАТ

Дипломная работа, 40 страниц, 15 рисунков, 1 таблица, 7 формул, 20 источников

Ключевые слова: РЕНТГЕНОВСКИЕ ИЗОБРАЖЕНИЯ, СИАМСКИЕ НЕЙРОННЫЕ СЕТИ, СОПОСТАВЛЕНИЕ ИЗОБРАЖЕНИЙ, МЕШОК ВИЗУАЛЬНЫХ СЛОВ, КЛЮЧЕВЫЕ ТОЧКИ

Объект исследования: алгоритмы поиска изображений в базах данных.

Предмет исследования: алгоритмы сопоставления изображений на основе нейросетевых методов и методов выявления ключевых точек.

Цель работы: разработка алгоритма сопоставления и поиска рентгеновских изображений в базах данных по изображению запроса.

Методы исследования: а) теоретические: изучение литературы, посвященной современным подходам к сопоставлению изображений, б) практические: разработка алгоритма для решения задачи поиска «цифровых двойников» с использованием методов машинного обучения, определения ключевых точек изображения и их дескрипторов.

Результат: произведена реализация и сравнение трех алгоритмов для решения поставленной задачи.

РЭФЕРАТ

Дыпломная праца, 40 старонак, 15 малюнкаў, 1 табліца, 7 формул, 20 крыніц

Ключавыя словы: РЭНТГЕНАЎСКІЯ ВЫЯВЫ, СІЯМСКІЯ НЕЙРОНАВЫЯ СЕТКІ, СУПАСТАЎЛЕННЕ ВЫЯЎ, МЯШОК ВІЗУАЛЬНЫХ СЛОЎ, КЛЮЧАВЫЯ КРОПКІ

Аб’ект даследавання: рэнтгенаўскія здымкі лёгкіх чалавека.

Мэта працы: распрацоўка алгарытму супастаўлення і пошуку рэнтгенаўскіх выяў у базах дадзеных па выяве запыту.

Метады даследавання: а) тэарэтычныя: вывучэнне літаратуры, прысвечанай сучасным падыходам да супастаўлення выяў, б) практычныя: распрацоўка алгарытму для вырашэння задачы пошуку "лічбавых двайнікоў" з выкарыстаннем метадаў машыннага навучання, вызначэння ключавых кропак выявы і іх дэскрыптараў.

Вынік: праведзена рэалізацыя і параўнанне трох алгарытмаў для вырашэння пастаўленай задачы.

ABSTRACT

Diploma thesis, 40 pages, 15 pictures, 1 table, 7 formulas, 20 sources.

Keywords: X-RAY IMAGES, SIAMESE NEURAL NETWORKS, IMAGE MATCHING, BAG OF VISUAL WORDS, KEYPOINTS

Object of research: algorithms for searching images in databases.

Subject of research: algorithms for image matching based on neural network methods and keypoints detection methods.

Objective: develop an algorithm for matching and searching for X-ray images in databases by the query image.

Methods of research: a) theoretical: study of literature on modern approaches to image matching, b) practical: development of an algorithm for solving the problem of searching for "digital twins" using machine learning methods and by determining image keypoints and their descriptors.

The result: implementation and comparison of three algorithms for solving the given problem.

ВВЕДЕНИЕ

В последние годы остро стоит проблема защиты личных данных, к которым, в свою очередь, относятся медицинские изображения. Ввиду человеческого фактора, не исключены ошибки при внесении изображений врачом в базу данных, что влечет за собой необходимость разработки алгоритма, позволяющего извлекать из базы изображения, принадлежащие одному человеку. Таким образом, поиск «цифровых двойников» — виртуальных аналогов реального объекта, которые в своих ключевых характеристиках дублируют его, — актуальная задача. Критически важными при поиске изображений, похожих на изображение запроса, являются выбор меры сходства изображений, их представление и эффективное индексирование базы данных.

В настоящее время существует ряд алгоритмов для сопоставления изображений, различающихся по точности и производительности, начиная с поиска классических признаков изображения, заканчивая моделями глубокого обучения на которых основано большое количество современных алгоритмов компьютерного зрения.

Глава 1 носит теоретический характер, в ней рассматриваются существующие алгоритмы сопоставления изображений и поиска изображений в базах данных.

В главе 2 описывается предобработка входных данных, реализация алгоритма на основе сверточных нейронных сетей.

Глава 3 посвящена сопоставлению снимков на основе гистограмм, полученных методом «мешок визуальных слов».

В главе 4 описан процесс разработки алгоритма поиска «цифровых двойников» путем сопоставления ключевых точек изображений.

ГЛАВА 1. ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Системы поиска изображений в базах данных

Для определения пациента по его снимку легких можно было бы обучить модель-классификатор с количеством классов равным количеству пациентов, однако есть несколько проблем при таком подходе. Во-первых, в базе данных пациентов содержится малое число снимков одного человека, что недостаточно для получения хороших результатов при обучении нейронной сети. Во-вторых, количество пациентов постоянно меняется, что влечет за собой необходимость заново обучать модель.

Вместо того чтобы классифицировать изображения, более эффективным подходом будет построение модели, способной сопоставлять изображения. Поставленная задача близка к задаче поиска изображений по содержанию (англ. content-based image retrieval, или CBIR). В качестве запроса для системы пользователем подается на вход изображение, для которого необходимо на выходе из содержащихся в базе данных изображений получить одно или несколько изображений, похожих на изображение запроса.

Изображения можно описать при помощи величин, известных как признаки, которые одновременно должны быть достаточно простыми, чтобы их извлечение происходило за относительно небольшое время, но при этом они должны отражать содержание изображения без значительной потери информации. Основная задача при разработке алгоритма поиска изображений – как сопоставить два изображения в соответствии с извлеченными признаками. Каждое изображение обычно представляется многомерным вектором признаков, размерность которого зависит от количества и типа извлеченных

признаков, а сходство между изображениями оценивается путем выбора подходящей функции расстояния, определенной на пространстве признаков.

1.2 Алгоритмы извлечения векторов признаков

Одну из ключевых ролей играет выбор алгоритма для извлечения вектора признаков. Рассмотрим самые популярные из них.

1.2.1 Алгоритмы выявления локальных признаков изображения

Например, в работе [1] рассматривается поиск по содержанию, основанный на алгоритме масштабно-инвариантной трансформации признаков (англ. scale-invariant feature transform, или SIFT). Из каждого изображения извлекаются ключевые точки. Инвариантные к повороту, афинным преобразованиям и масштабированию дескрипторы SIFT описывают локальные особенности изображения. На первом этапе задача SIFT – обнаружить все возможные ключевые точки изображения на основе их устойчивости к искажению изображения. Для всех таких точек создается подробная модель для определения местоположения и масштаба. Затем определяется одна или несколько ориентаций на основе локальных направлений градиента изображения, которые измеряются в окрестности каждой ключевой точки. Они преобразуются в представление, допускающее значительную степень изменения освещенности и локального искажения формы.

1.2.2 Кодирование с целью получения компактных векторов из локальных признаков

Примеры методов: «мешок визуальных слов», вектор Фишера. Такие методы, как «мешок визуальных слов» (англ. bag of visual words), дают хорошие результаты при поиске изображений в небольших базах. Идея состоит в том, чтобы представить изображение в виде набора признаков, которые, в свою очередь, состоят из ключевых точек изображения и дескрипторов — описаний ключевых точек. По дескрипторам составляется словарь визуальных слов путем их кластеризации. Изображение представляется в виде гистограммы частот признаков, где каждый компонент гистограммы — это количество дескрипторов, соответствующих визуальному слову. Поиск похожих изображений происходит путем сравнения полученных гистограмм, например используя кластерные деревья для построения иерархической структуры как предложено в работе [2].

1.2.3 Сверточные нейронные сети

Сверточные нейронные сети — один из самых современных подходов. А именно сиамские нейронные сети, которые хорошо справляются с данной задачей. Во время обучения пара изображений проходит через сеть, состоящую из двух ветвей, для выходов которых рассчитывается значение функции потерь.

Существует три типа сиамских сетей в зависимости от того, как соединяются ее ветви:

1. Сеть с несколькими объединенными последними слоями и softmax-функцией на последнем слое

2. Сеть, у которой на первый слой подаются сконкатенированные входы, на последнем слое также применяется softmax-преобразование
3. Входы подаются на параллельные подсети, а для выходов, которые представляют собой вектора признаков, рассчитывается функция расстояния

В данной дипломной работе была выбрана третья архитектура сиамской нейронной сети. С задачей сравнения изображений с использованием сиамских нейронных сетей неплохих результатов удалось достигнуть в работе [3].

Цель обучения сети – минимизировать квадрат евклидова расстояния между векторами признаков положительных пар изображений и максимизировать его для отрицательных пар.

Самой широко используемой при обучении сиамских сетей является функция контрастных потерь:

$$l(x_i, x_j, z_{ij}) = (1 - z_{ij})||h_i - h_j||_2^2 + z_{ij} \max(0, \tau - ||h_i - h_j||_2^2) \qquad (1.1)$$

Здесь x_i, x_j — это входные изображения, z_{ij} — метка (0, если пара положительная, и 1 в противном случае), h_i, h_j – полученные вектора признаков.

Отрицательные пары вносят свой вклад в функцию потерь только в том случае, если их расстояние меньше допустимого порога τ. Заданная функция потерь побуждает положительные пары находиться близко друг к другу в пространстве признаков, одновременно отдаляя отрицательные пары.

Входные изображения попадают в две идентичные ветви, которые имеют общие веса и параметры, что, в свою очередь, гарантирует, что два чрезвычайно похожих изображения не могут быть сопоставлены соответствующими сетями в сильно различающиеся места в пространстве признаков, поскольку каждая сеть вычисляет одну и ту же функцию. Каждая ветвь представляет собой глубокую нейронную сеть и включает в себя набор сверточных и полносвязных слоев.

Также существует сеть триплетов, которая является модификацией сиамской сети с тремя сверточными нейронными подсетями [4]. В центральную подсеть подается изображение-«якорь», принадлежащее некоторому классу, а в две другие — пример из того же и из другого класса. Центральная подсеть с каждой из двух других подсетей образует сиамскую сеть. При таком подходе можно использовать функцию потерь для триплетов:

$$l(x_a, x_p, x_n) = \max(0, ||h_a - h_p||_2^2 - ||h_a - h_n||_2^2 + \tau) \quad (1.2)$$

x_a – изображение-«якорь», x_p – положительный пример, x_n – отрицательный пример и соответствующие им вектора признаков.

Данная функция потерь минимизирует расстояние между «якорем» и позитивным примером, так как оба они принадлежат одному классу, и максимизирует расстояние между «якорем» и негативным примером.

Другой вариант функции потерь для триплетов, где вместо одного отрицательного примера берется несколько:

$$l(x_a, x_p, x_n^1, x_n^2, \ldots, x_n^N) = \log(1 + \sum_{i=1}^{N} \exp(||h_a - h_p||_2^2 - ||h_a - h_n^i||_2^2)) \quad (1.3)$$

1.3 Поиск

Важной задачей является выбор стратегии индексирования базы данных, так как от этого напрямую зависит скорость системы.

Одним из вариантов, который был выбран в данной работе, является так называемый brute-force поиск по векторам признаков, используя некоторую функцию расстояния, когда вектор запроса сравнивается со всеми векторами базы данных. Достоинство такого подхода – качество поиска, а недостатки – занимаемая оперативная память и значительное снижение скорости при увеличении размера базы данных. Для решения этой проблемы вектора признаков можно хранить в некотором сжатом виде.

Другой вариант – использовать приближенные методы поиска, стараясь найти изображения за меньшее число сравнений.

Для оценки качества алгоритма поиска существуют различные метрики:

- $precision@k = \frac{REL_k}{k}$

  REL_k – число релевантных изображений в топ-k результатах поиска

- $R - precision = \frac{REL_R}{R}$

  REL_R – число релевантных изображений в топ-R результатах поиска, где R – общее число релевантных изображений для данного запроса

- $recall@k = \frac{REL_k}{\min(k,R)}$

ГЛАВА 2. СИАМСКАЯ НЕЙРОННАЯ СЕТЬ

2.1 Подготовка данных

Исходный набор входных данных содержал порядка 94 тысяч рентгеновских изображений лёгких мужчин и женщин возрастом от 18 до 50 лет. Разрешение изображений составляло 512x512 пикселей. При формировании обучающего набора данных изображения были сжаты с помощью метода ближайшего соседа до размеров 224x224 пикселей, что позволило значительно ускорить обучение и уменьшить объем занимаемой памяти.

Все изображения были нормализованы путем вычитания математического ожидания значений пикселей изображений и деления на стандартное отклонение. В основе разработанных моделей лежат ResNet-50 и EfficientNet B4, являющиеся одними из самых мощных на данный момент, при использовании которых рекомендовано проводить нормализацию с теми же параметрами, которые были использованы при обучении модели.

Для обучения сиамской сети было необходимо разбить рентгеновские снимки легких на пары: положительные — принадлежащие одному человеку, и отрицательные — принадлежащие разным людям, соответственно. В итоговой выборке оказалось 47532 пары, количество положительных и отрицательных пар равно.

Весь набор данных был разделен в отношении 3:1:1 на тренировочную, валидационную и тестовую выборки. Валидационная выборка используется для недопущения недообучения или переобучения сети, а также для настройки гиперпараметров. На данной выборке проверялось значение функции потерь на каждой эпохе процесса обучения. Обучение следует останавливать тогда, когда значение на валидационной выборке начинает превышать значение на

тренировочной выборке. Для чистоты эксперимента нельзя использовать тестовые данные в обучающей выборке и для подбора параметров модели.

2.2 Разработка и обучение модели

Как было сказано выше, основой построенных моделей являются нейронные сети ResNet-50 и EfficientNet B4.

В сети ResNet-50 представлена концепция пропуска соединения. Обучение глубоких нейронных сетей затруднено из-за проблемы исчезающих градиентов, так как функции активации сокращают входы сети до гораздо меньших выходов, приводя к уменьшению значений градиентов. Остаточные соединения позволяют модели научиться пропускать слои и добавлять вход одного слоя к выходу другого. Для уменьшения времени обучения используются блоки, содержащие «узкое место»: свертка размера 1x1, 3x3 и снова свертка 1x1 [5].

В отличие от обычных подходов, которые произвольно масштабируют размеры сети, такие как ширина, глубина и разрешение, в сети EfficientNet равномерно и согласованно масштабируются все измерения с фиксированным набором коэффициентов [6].

Разработанная сиамская нейронная сеть содержит две ветви с идентичными весами. С точки зрения реализации, в сети присутствует одна ветвь, через которую проходит два обучающих примера из пары. Она состоит из предобученной сверточной нейронной сети, за которой следует два полносвязных слоя с функцией активации ReLU.

Когда речь идет о большой базе данных, насчитывающей десятки тысяч рентгеновских изображений, естественным образом встает вопрос о том, сколько памяти нужно алгоритму. В нашем случае память необходима для хранения векторов признаков. Оптимизировать объем занимаемой памяти путем

уменьшения размерности векторов удалось при помощи метода главных компонент без значительных потерь в производительности модели.

Основная идея этого метода заключается в уменьшении размерности набора данных, в котором имеется большое количество взаимосвязанных компонент, сохранив при этом как можно больше вариаций, присутствующих в наборе данных. Это достигается путем преобразования в новый набор главных компонент, которые некоррелированы и упорядочены таким образом, что большая часть вариации данных будет сосредоточена в первых координатах [7].

Вычисление главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных.

Таким образом, при прохождении пары изображений через описанную сеть, на выходе получаются два вектора размерности 16, для которых рассчитывается евклидово расстояние.

Из сверточных слоев у моделей обучаемым был лишь последний слой. Глубина сетей, с учетом слоев пакетной нормализации, подвыборке по максимуму и др., составила 131 и 455 слоев соответственно.

Для выбора порога в функции контрастных потерь из промежутка [0.1, 1.0] с шагом 0.1 была обучена сиамская сеть на основе предобученной сети ResNet-50, произведено сравнение значений площади под ROC-кривой для предсказаний модели на валидационном наборе данных. Наилучших результатов удалось достигнуть при значении $\tau = 0.5$. При обучении использовался оптимизатор Adam — адаптивный вариант градиентного спуска, контролирующий скорость обучения.

2.3 Оценка моделей первой итерации

2.3.1 Классификация пар изображений

Существуют две важные метрики качества модели: точность (англ. precision) и полнота (англ. recall). Полноту можно интерпретировать как долю объектов положительного класса из всех объектов положительного класса нашел алгоритм, а точность определяется как доля объектов, названных классификатором положительными и при этом действительно являющимися положительными. Эти метрики хороши тем, что они не зависят от соотношения классов в выборке; часто на практике необходимо решить, какая из этих двух метрик важнее для конкретной задачи, чтобы соответствующим образом подобрать параметры модели.

Ниже приведены ROC-кривые (рис. 2.1), которые показывают зависимость количества верно определенных положительных примеров от количества неверно определенных отрицательных примеров тестовых данных. Оценить качество модели помогает площадь под графиком: чем она больше, тем лучше (максимальное её значение 1). В случае с моделью, основанной на ResNet-50 площадь равна 0.955, а для модели с EfficientNet B4 она составляет 0.94.

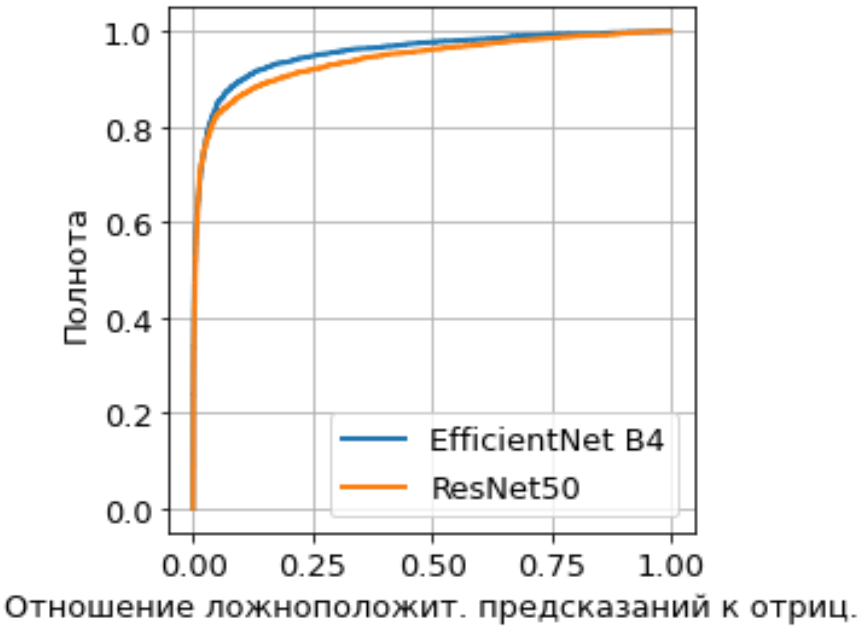

рисунок 2.1 — ROC-кривые

График (рис. 2.2) дает представление о совершении моделью ошибок первого и второго рода.

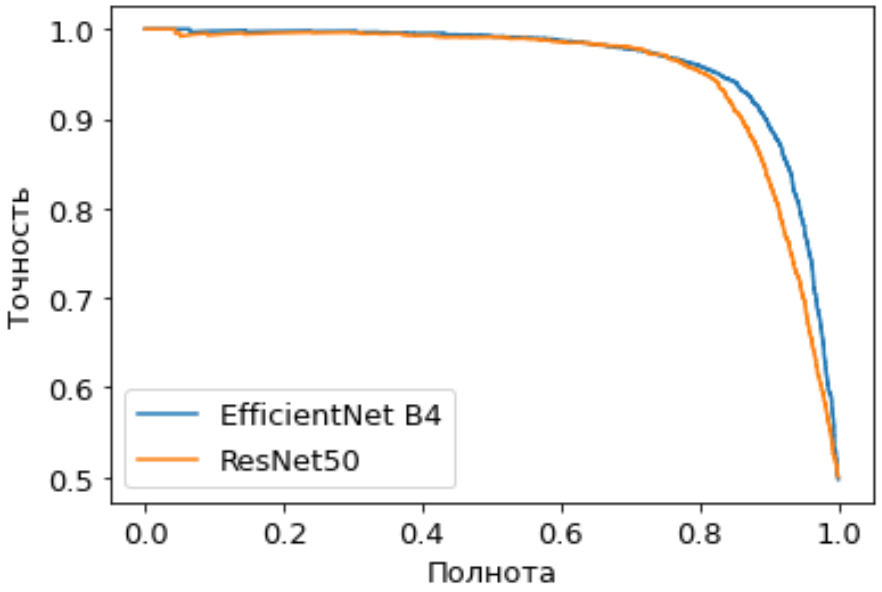

рисунок 2.2 — график точность-полнота

Главной задачей разработанных моделей является способность отдалять отрицательные пары изображений и сближать положительные. Наглядно это показано на гистограммах 2.3 и 2.4.

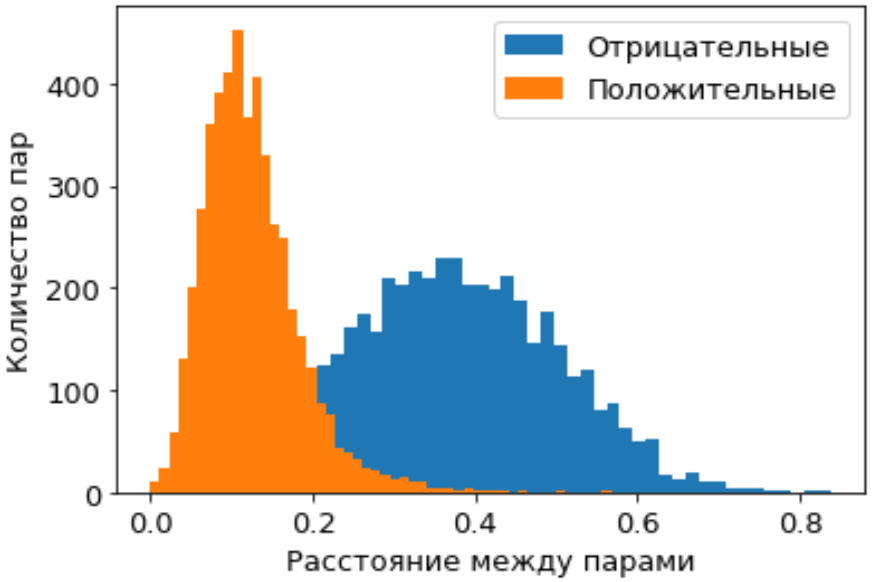

рисунок 2.3 — распределение пар для EfficientNet B4 модели

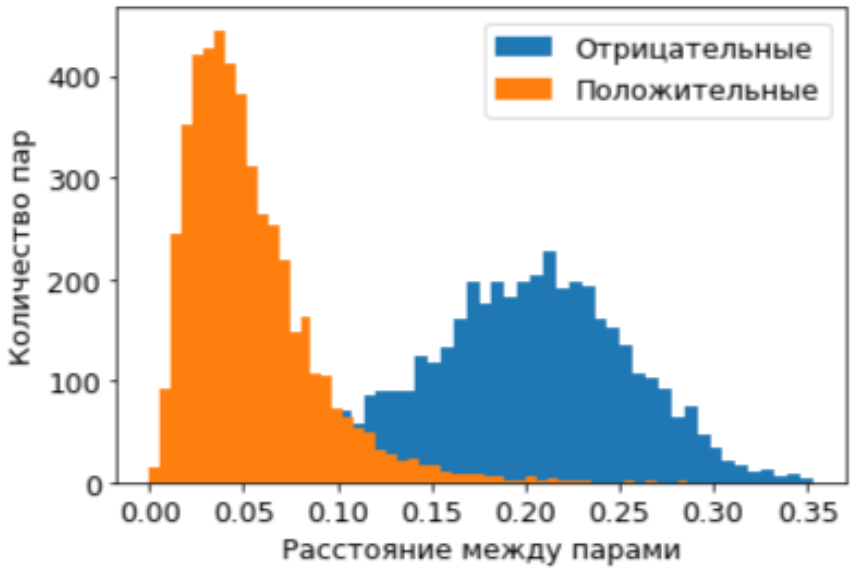

рисунок 2.4 — распределение пар для ResNet-50 модели

На тестовой выборке расстояния между векторами признаков для пар расположены достаточно компактно при использовании обеих моделей, но нетрудно видеть, что модель с ResNet-50 лучше лучше справляется с поставленной задачей.

2.3.2 Поиск изображений в базе данных

Для поиска изображений в базе данных первым был протестирован следующий алгоритм, см. рисунок 2.5:

1. Нахождение векторов признаков изображений базы данных, которые являются выходом описанной ранее сиамской нейронной сети
2. Нахождение вектора признаков для входного изображения
3. Сравнение векторов входного изображения с векторами изображений базы данных путем нахождения евклидового расстояния между ними
4. Поиск ближайшего вектора и выбор соответствующего ему изображения

Стоит заметить, что преимуществом данного алгоритма является то, что вектора находятся один раз и затем при добавлении новых изображений нужно лишь найти соответствующие им вектора, не затрагивая остальные снимки.

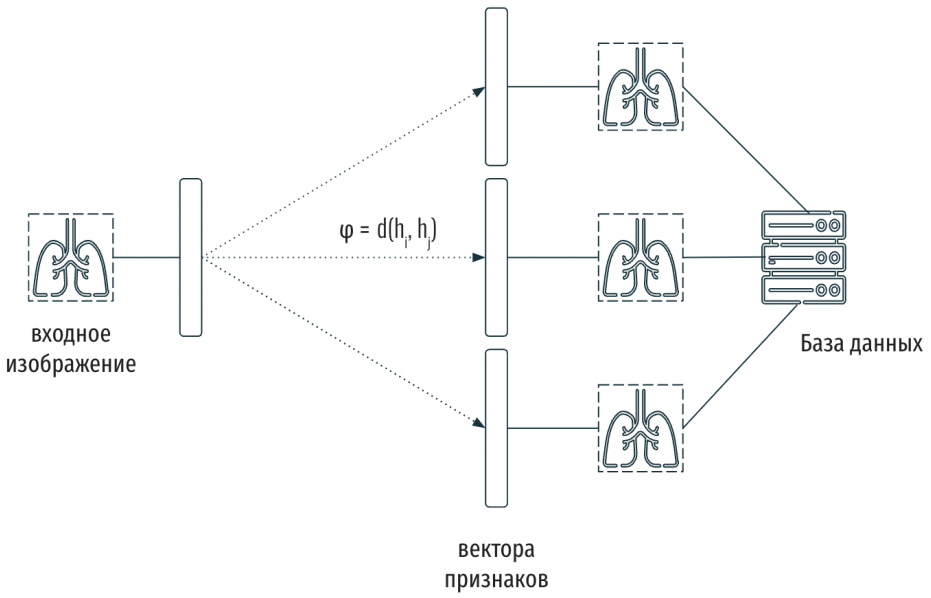

рисунок 2.5 — схема поиска похожих рентгеновских снимков

Несмотря на неплохие результаты классификации пар на отрицательные и положительные, модель показала себя не лучшим образом при поиске пары для 250 тестовых изображений среди 15000 снимков базы данных (см. таблицу 2.1), что можно объяснить тем, что в тренировочном наборе изображения отрицательных пар были слишком различными и сеть не научилась различать похожие изображения, принадлежащие разным людям. Эту задачу усложняет также и тот факт, что в процессе жизни человека, его легкие изменяются в размере, структурно вследствие возраста и после перенесенных заболеваний.

Таким образом, появилась необходимость составления более сложных отрицательных пар изображений.

2.4 Составление сложных пар изображений

Задача, известная как поиск негативных данных (англ. negative data mining), важна для улучшения предсказательной способности модели. Отрицательных примеров, как правило, в наборе данных значительно больше, чем положительных, однако не все они являются хорошими кандидатами на попадание в тренировочный набор, так как не все отрицательные примеры одинаково сложно распознавать. Сложными отрицательными примерами являются те, которые предсказываются сетью как положительные образцы, то есть они являются ложноположительными. Именно такие примеры и следует включать в тренировочный набор, причем следует не допускать дисбаланса классов [8].

Получить более сложные пары в количестве равном количеству негативных пар удалось итеративно, используя уже обученную сиамскую модель. На первой итерации были выбраны изображения которые ранее присутствовали в наборе. Для них среди снимков других людей были найдены

ближайшие по такому же принципу, как происходит поиск в базе данных. На второй итерации модель была обучена уже на новых парах и, как и ранее, составлены отрицательные пары, аналогично произведена третья итерация.

2.5 Результаты

Для модели, обученной на изображениях со второй итерации, на рисунках 2.6 и 2.7 показан пример поиска похожих изображений, полученных при сравнении векторов признаков. Сравнение моделей можно увидеть в таблице 1.

рисунок 2.6 – входное изображение

рисунок 2.7 – топ-5 изображений, полученных из базы данных

2.5.1 Многомерное шкалирование

К парам векторов признаков был применен метод многомерного шкалирования для визуальной репрезентации того, насколько хорошо модель

умеет разделять вектора, соответствующие разным людям, в пространстве признаков.

Многомерное шкалирование (англ. multidimensional scaling, MDS) – это метод, с помощью которого можно получить количественные оценки сходства между группами элементов. Он относится к набору статистических методов, которые используются для снижения сложности набора данных, позволяя визуально оценить лежащие в их основе структуры. Таким образом, многомерное шкалирование – это набор методов для обнаружения "скрытых" структур в многомерных данных.

На основе матрицы близости входных величин, измеренных на исследуемых объектах, в нашем случае это вектора признаков изображений, эти расстояния отображаются на пространство более низкой размерности (обычно двух- или трехмерное для удобства визуализации). Мера различий расстояний в исходном и новом пространстве называется функцией стресса. Для минимизации стресса используется мажоризация, такая стратегия известна как SMACOF (англ. Scaling by MAjorizing a COmplicated Function – масштабирование путем мажорирования сложной функции) [9].

Принцип предполагает нахождение более простой функции $g(x, y)$, которая мажорирует $f(x)$, т.е. $g(x, y) \geq f(x)$ для любых x, где y – некоторое фиксированное значение. Такая замещающая функция должна касаться поверхности в точке y, т.е. $f(y) = g(y, y)$, а минимизация $g(x, y)$ по x приводит к цепочке неравенств $f(x^*) \leq g(x^*, y) \leq g(y, y) = f(y)$.

Алгоритм мажоризации:

1. Выбрать начальное значение $y = y_0$
2. Найти значение $x^{(t)}$ такое, что $g(x^{(t)}, y) \leq g(y, y)$
3. Остановиться, если $f(y) - f(x^{(t)}) < \varepsilon$, иначе перейти на шаг 2

На рисунке 2.8 можно видеть результат применения метода. Одним цветом выделены точки, соответствующие векторам признаков снимков одного человека. Несмотря на то что такие пары расположены довольно компактно, они недостаточно обособлены от других. Видно, что часты случаи расположения снимка одного человека в пространстве признаков ближе к снимку другого человека, а не к снимку того же.

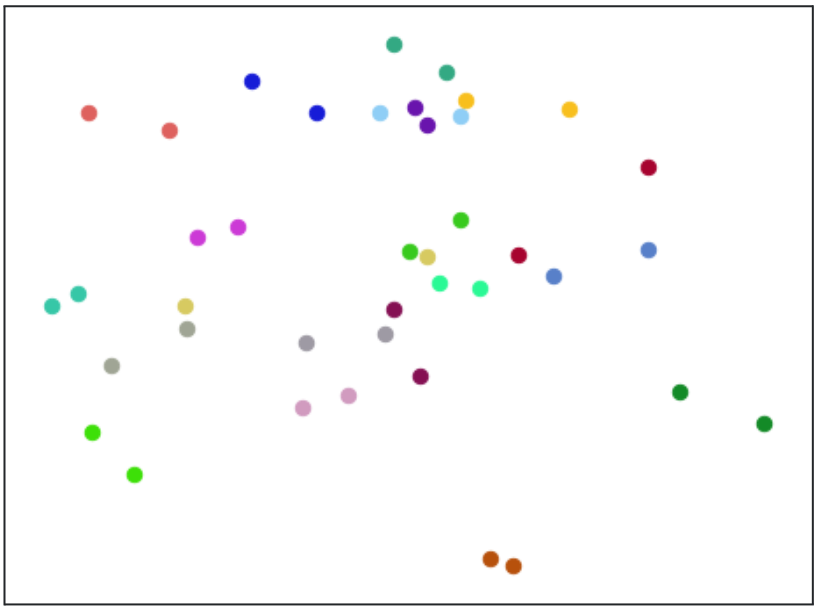

рисунок 2.8 – проекция векторов признаков

2.5.2 Аугментации

Исследования показали, что аугментация данных, т. е. увеличение выборки за счет модификации уже существующих данных — это важный этап обучения моделей. При этом модифицируется только тренировочный набор данных, а тестовый остается неизменным. Они могут улучшить обобщающую способность модели, выступая как способ регуляризации [10].

Для этого были выбраны следующие трансформации снимков: поворот на градус из промежутка [-5, 5] и небольшое увеличение, которые близки к

вариациям у реальных изображений. Как показал эксперимент, прироста точности модели достигнуть не удалось.

Как видно в таблице 2.1, в нашем случае аугментации не дали прироста в точности модели.

Таблица 2.1 – Результаты поиска пары в зависимости от способа формирования тренировочных данных

Способ формирования пар	Доля изображений, для которых была найдена пара среди топ-n ближайших снимков					
	5	10	15	20	25	30
Случайно	0.004	0.02	0.036	0.056	0.06	0.06
Итеративно (1 ит.)	0.032	0.052	0.08	0.108	0.124	0.148
Итеративно (2 ит.)	0.064	0.104	0.156	0.204	0.224	0.256
Итеративно (2 ит. + аугментации)	0.068	0.108	0.12	0.136	0.156	0.188
Итеративно (3 ит.)	0.06	0.096	0.152	0.188	0.204	0.228

ГЛАВА 3. «МЕШОК ВИЗУАЛЬНЫХ СЛОВ»

Следующим выбранным подходом являлся мешок визуальных слов, истоки которого лежат в области обработки текстов. Позже такой подход стал использоваться и в обработке изображений.

В основе метода лежит поиске ключевых точек и их дескрипторов, например с помощью алгоритмов SIFT, SURF, ORB. В нашем случае был выбран алгоритм SIFT – масштабно-инвариантная трансформация признаков, дескрипторы которого инвариантны относительно масштабирования, изменения ориентации, освещенности.

3.1 Алгоритм SIFT

Обнаружение ключевых точек детектором SIFT [11] начинается с построения пирамиды гауссианов – изображений, полученных в результате применения фильтра Гаусса, вычисляемых по формуле свертки:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.1}$$

где σ – радиус размытия, x и y – координаты пикселя, G(x, y, σ) – фильтр Гаусса, а I(x, y) – исходное изображение. Пирамиду образуют изображения, принадлежащие различным октавам, количество которых зависит от размера исходного изображения: размер каждой октавы в два раза меньше предыдущей. Набор таких сглаженных изображений образует масштабируемое пространство.

Далее, полученные размытые изображения используются для вычисления разности гауссианов (англ. Difference of Gaussians, DoG), которые необходимы для поиска ключевых точек изображения. Разность гауссианов рассчитывается как разность гауссовых размытий изображения с двумя различными радиусами:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{3.2}$$

Этот процесс выполняется для разных октав изображения в пирамиде.

Затем происходит анализ пикселей полученных разностей: один пиксель изображения сравнивается с 8 соседними, а также с 9 пикселями в следующем масштабе и 9 пикселями в предыдущем масштабе, что дает в сумме 26 проверок. Если это локальный экстремум, то такая точка становится потенциально ключевой и она обладает инвариантностью масштаба. Большая часть таких точек может лежать вдоль края, либо же иметь недостаточную контрастность, и их необходимо удалить.

Для получения точек инвариантных к вращению в зависимости от масштаба берется окрестность ключевой точки, и в этой области для каждого пикселя рассчитывается величина и направление градиента. Создается гистограмма ориентации с 36 компонентами, охватывающими 360 градусов. Сопоставляя соответствующим компонентам гистограммы величины, пропорциональные величине градиента для пикселя, для самого высокого пика гистограммы и пиков выше 80% от него производится расчет ориентации ключевой точки.

Чтобы вычислить дескриптор для локальной области каждой ключевой точки, который будет инвариантным к изменение точки зрения и освещенности, берется окно 16x16 с центром в ключевой точке, которое делится на 16 блоков размером 4x4. Для каждого такого блока создается гистограмма ориентированных градиентов с 8 компонентами. Все вычисленные гистограммы объединяются в один вектор размера 128.

Чтобы добиться независимости от вращения, из каждой ориентации вычитается вращение ключевой точки. Изменение контраста изображения приводит к такому же изменению в значениях магнитуд градиентов, поэтому дескриптор нормализуется.

3.2 Построение гистограмм изображений

Дескрипторы различных точек могут описывать похожие ключевые точки, поэтому их можно кластеризовать каким-либо алгоритмом кластеризации получив словарь, элементами которого являются часто повторяющиеся элементы изображения. т.н. “визуальные слова”. Для кластеризации в данной работе был выбран мини-пакетный метод k-средних (англ. mini batch k-means), который заключается в кластеризации векторов, случайно выбираемых на каждой итерации в заданном количестве. Этот метод по скорости сходимости значительно превосходит классический метод k-средних, и дает небольшое отклонение от результатов полученных с классическим методом, особенно при использовании меньшего количества кластеров [12]. Классический алгоритм k-средних является дорогостоящим для больших наборов данных, требуя O(nkdi) времени вычислений, где n – количество примеров, d – размерность векторов, а i – число итераций. Фактор скорости был определяющим, так как в тренировочной выборке оказалось порядка 23 миллионов дескрипторов.

Была использована реализация мини-пакетного метода k-средних из библиотеки sklearn [13]. Параметрами помимо количества кластеров являлись также максимальное число итераций, так как данный метод не сходится в отличии от классического. Остановка может произойти, если не происходит уменьшения значения минимизируемой функции на последовательных m итерациях, также задаваемых параметром. Еще один параметр – собственно размер пакета.

Каждое изображение может быть представлено вектором – гистограммой частот визуальных слов, пример представлен на рисунке 3.1.

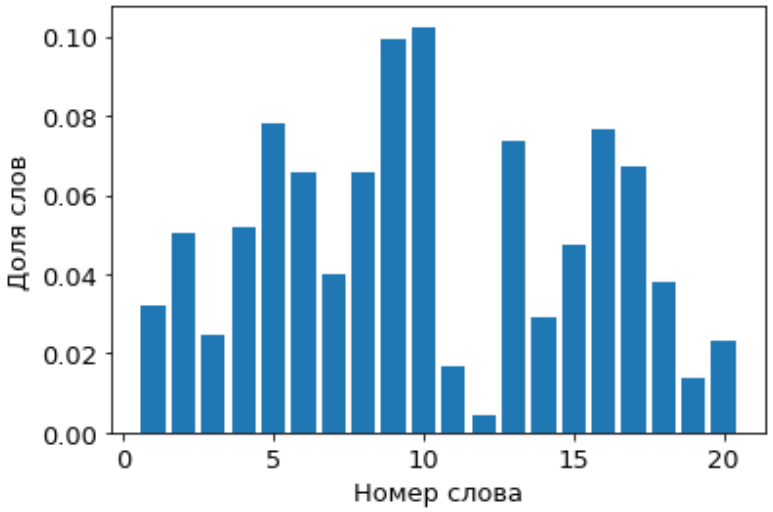

рисунок 3.1 – гистограмма изображения

3.3 Поиск в базе данных

Сопоставление изображений производилось по алгоритму, описанному в главе 2, в качестве векторов признаков были взяты полученные гистограммы. В работе для количественной оценки сходства между двумя распределениями вероятностей слов, было использованы расстояния Хеллингера – формула (3.3), хи-квадрат – формула (3.4) и произведено сравнение точности моделей на гистограммах. Реализацию данных функции можно найти в библиотеке OpenCV.

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_I \sqrt{H_1(I) H_2(I)}} \tag{3.3}$$

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \tag{3.4}$$

Также было проанализировано влияние количества кластеров на точность модели. Результаты можно увидеть на рисунках 3.2 и 3.3. Как видно на графиках, точность данной модели не превысила точности нейросетевой модели.

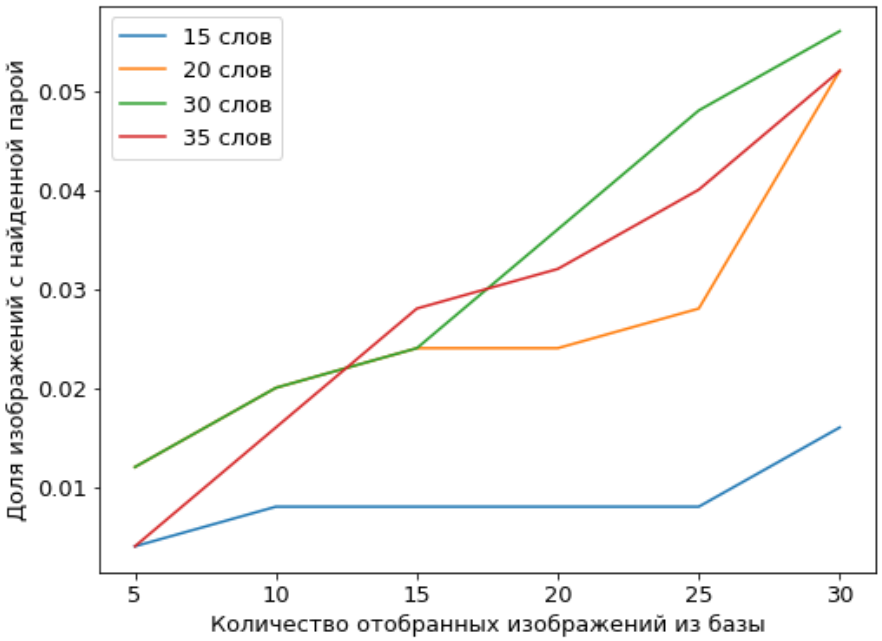

рисунок 3.2 – результаты поиска для функции Хеллингера

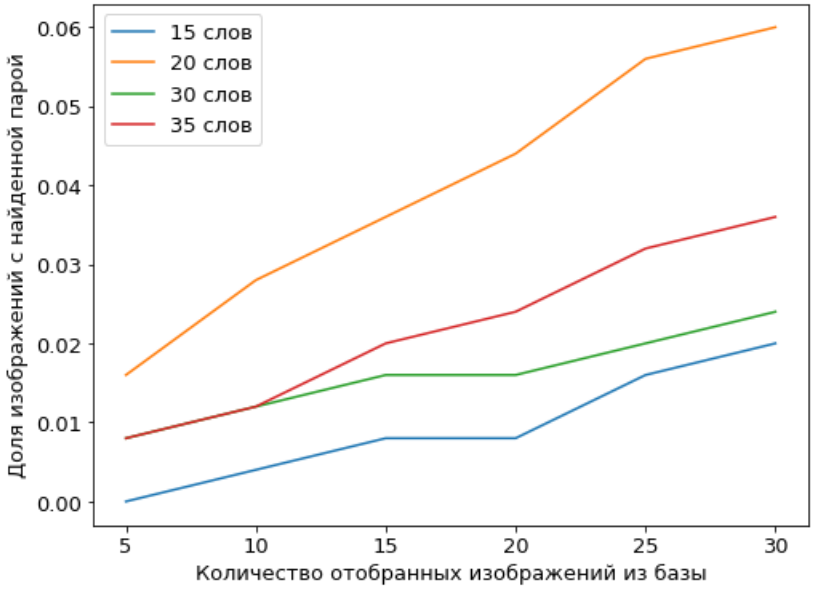

рисунок 3.3 – результаты поиска для функции хи-квадрат

Нетрудно видеть, что результаты не превышают результатов, полученных с нейросетевой моделью.

ГЛАВА 4. СОПОСТАВЛЕНИЕ ТОЧЕК

Вместо вычисления частот встречаемости дескрипторов ключевых точек можно сравнивать изображения путем сопоставления ключевых точек двух снимков как показано на рисунке 4.1.

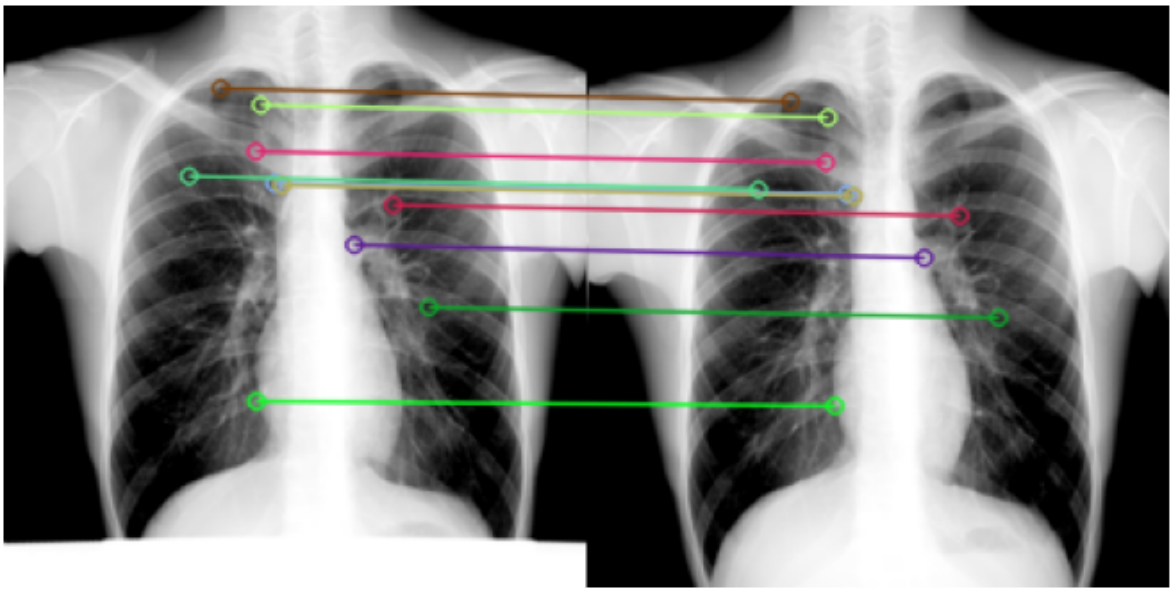

рисунок 4.1 – сопоставленные ключевые точки снимков

Поиск ключевых точек происходил с помощью алгоритма ORB и алгоритма SIFT, который был описан в предыдущей главе. В работе была использована реализация обоих алгоритмов из библиотеки OpenCV[14][15]. Как утверждается в статье [16], ORB позволяет получить меньше ключевых точек в сравнении с другими детекторами, которые в свою очередь также как и с алгоритмом SIFT устойчивы и к вращению, и масштабированию, что подтверждают результаты экспериментов, однако SIFT более устойчив к вращению, если он используется в паре с дескриптором, инвариантным к вращению.

4.1 Алгоритм ORB

Алгоритм ORB (англ. oriented FAST and rotated BRIEF), описанный в статье [17], использует улучшенный алгоритм FAST для обнаружения характерных точек. Для определения того, является ли точка ключевой, найдя яркость пикселя и, выбрав некоторый порог T, происходит сравнение с яркостями шестнадцати пикселей на окружности радиуса 3 с центром в данной точке. Если яркость N последовательных точек на выбранной окружности больше или меньше, чем у центральной точки, то ее можно считать потенциальной ключевой точкой. Для инвариантности к масштабу строится пирамида гауссианов изображений. Затем на каждом уровне определяются экстремумы функции яркости по алгоритму, описанному выше. После выявления потенциальных ключевых точек используется угловой детектор Харриса для их фильтрации по значению меры Харриса, чтобы отсеять менее значимые.

Вычисление дескрипторов производится при помощи улучшенного алгоритма BRIEF, дескрипторы которого являются бинарными векторами, размерность которых чаще всего равна 256. В оригинальном алгоритме применяя фильтр Гаусса для уменьшения шума и выбирая случайным образом пары пикселей в окрестности ключевой точки, значения яркостей каждой пары сравниваются, итоговое значение равно 1, если яркость пикселя x меньше яркости пикселя *y*, в противном случае значение равно 0. Полученный дескриптор обладает устойчивостью к различному освещению, перспективному искажению, быстро вычисляется, но обладает одним недостатком – он неинвариантен к повороту.

Для достижения устойчивости относительно вращения, в модифицированной версии алгоритма BRIEF для устранения этой проблемы

вводится параметр угловой ориентации. Он основан на направлениях градиента яркости относительно центра точки, направление с наибольшей интенсивностью называется ориентацией ключевой точки.

4.1 Нахождение пар точек

4.1.1 Brute-force сопоставитель

В данной работе был выбран так называемый brute-force сопоставитель, который сопоставляет дескриптор ключевой точки одного изображения со всеми другими дескрипторами точек другого изображения, используя некоторую функцию расстояния. Например для бинарного дескриптора BRIEF можно использовать расстояние Хэмминга для измерения количества различных элементов у двух векторов одинаковой длины. В реализации алгоритма в библиотеке OpenCV [18] параметром является булева переменная crossCheck. Если она истинна, то выбираются только те совпадения со значением (k, m), для которых k-й дескриптор первого изображения является наилучшим совпадением для дескриптора m второго изображения и наоборот. То есть, выбранные дескрипторы обоих изображений должны совпадать друг с другом.

4.1.2 Сопоставитель на основе FLANN

Альтернативой сопоставителю brute-force, также реализованной в OpenCV является FLANN matcher (англ. Fast Library for Approximate Nearest Neighbors – быстрая библиотека для приближенных ближайших соседей) [19], который работает быстрее чем Brute-force matcher и за счет этого может быть использован на больших наборах данных, также он подходит для векторов

больших размерностей, но он не найдет наилучшие возможные совпадения, в отличии от первого. Данный алгоритм строит эффективную структуру данных (k-мерное дерево), которая используется для приближенного поиска точек.

4.2 Вычисление расстояния между изображениями и поиск в базе данных

Для вычисления расстояния между рентгеновскими снимками находилось среднее расстояние между сопоставленными парами точек. Для вычисления среднего выбирались не все пары, так как в таком случае было сильное влияние шума, это можно видеть на рисунке, где отображены 10 самых далеких найденных пар.

Поэтому на тренировочном наборе был произведен параметра n – количества самых близких пар точек. Наилучших результаты удалось достигнуть при n = 10. На рисунке 4.2 отображено то, как для дескрипторов точек, полученных с использованием SIFT и ORB, алгоритм показал себя в задаче поиска ближайших снимков ко входному снимку.

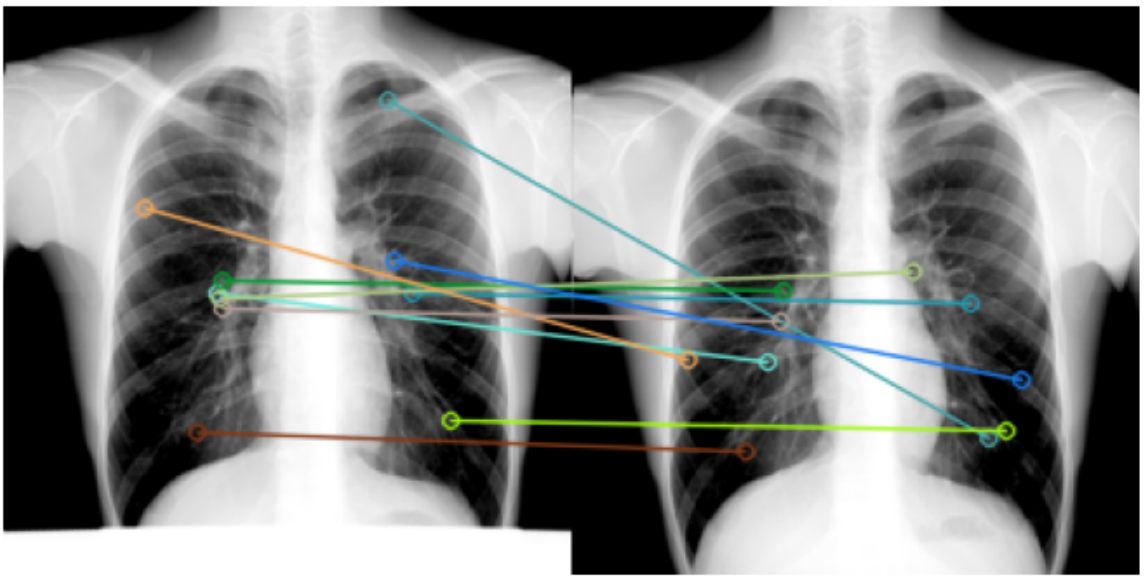

рисунок 4.2 – последние 10 совпадений ключевых точек

Несмотря на высокую точность, у данного подхода есть существенные недостатки: большой объем занимаемой памяти, так как требуется хранить все дескрипторы изображений (размер дескрипторов базы данных составил 1.33 гб) и низкая скорость поиска (снижение в 4 раза в сравнении с нейросетевым подходом и «мешком визуальных слов»), так как происходит перебор и сравнение всех дескрипторов.

Для оптимизации было произведено сокращение количества хранимых дескрипторов, что позволило сократить объем занимаемой памяти в 2.5 раза и ускорение в 2 раза. Для этого было проанализировано то, к каким кластерам чаще всего относятся дескрипторы, соответствующие точкам, входящим в топ-10 ближайших совпадений. Чтобы это проверить, из 10000 тренировочных изображений были составлены случайные пары, найдены ключевые точки соответствующих изображений, их дескрипторы, произведено сопоставление точек. Для точек из топ-10 с помощью ранее обученной модели k-средних были найдены кластеры. Оказалось, что абсолютное большинство таких точек принадлежит 7 кластерам из 20, рисунок 4.3.

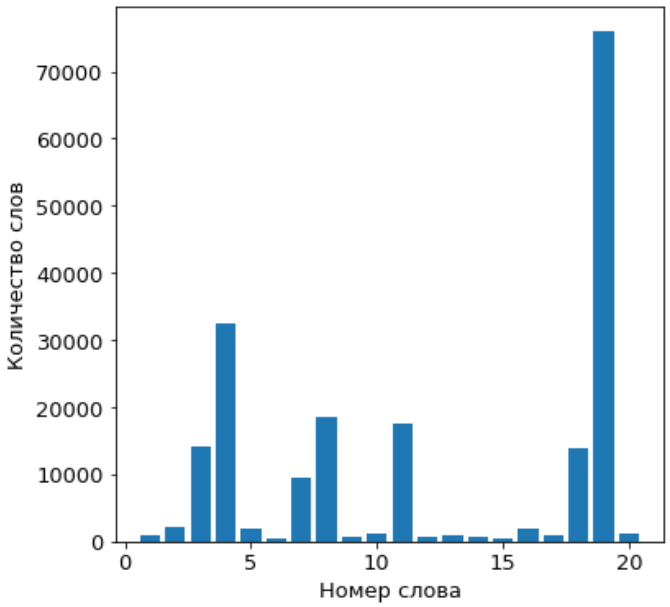

рисунок 4.3 – гистограмма встречаемости слов в топ-10 совпадений ключевых точек

Для изображений базы данных для хранения отбирались дескрипторы принадлежащие данным кластерам. Как видно на графике 4.4, ценой меньшего потребления памяти и скорости поиска точность снизилась.

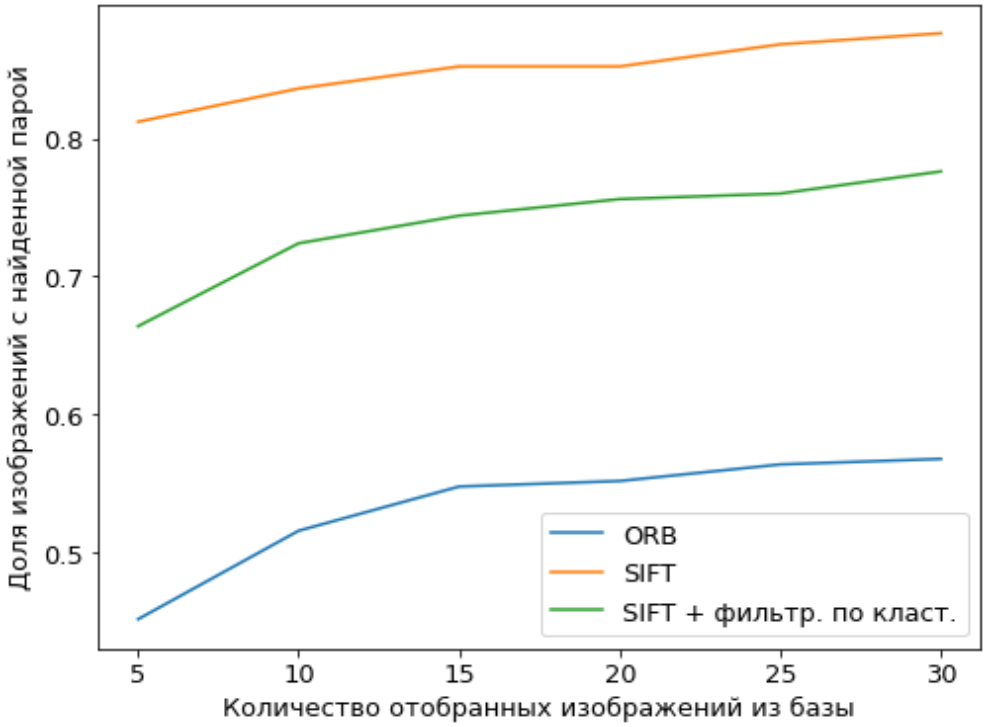

рисунок 4.4 – результаты поиска для разных детекторов ключевых точек

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной дипломной работы были решены следующие задачи:

1) Проведен обзор на научные статьи, касающиеся темы сравнения изображений, поиска в базах данных, а также на современные алгоритмы, решающие эту задачу.
2) Проведена предобработка исходных данных, формирование тренировочных и тестовых наборов.
3) Разработаны сиамские сети на основе предобученных моделей ResNet-50 и EfficientNet B4. Оценена способность сиамской сети классифицировать пары изображений.
4) Реализован алгоритм поиска пары для рентгеновских изображений легких, который выявил то, что обученная сиамская нейронная сеть дает неудовлетворительные результаты при сравнении снимков, которые визуально похожи. В связи с этим были составлены новые, более сложные отрицательные пары, на которых модель была дообучена.
5) На основе алгоритмов выявления ключевых точек и их дескрипторов был реализован поиск снимков, путем сравнения векторов признаков, полученных с помощью метода мешка визуальных слов.
6) Произведена разработка и реализация алгоритма поиска снимков путем сопоставления ключевых точек и расчета средних расстояний между соответствующими дескрипторами. Данный метод позволил достигнуть точности в 87,6% при извлечении из базы данных топ-30 ближайших снимков. Оптимизированная по памяти и скорости версия метода дала точность 77,6%.

На данный момент существует большое количество самых разнообразных подходов к сопоставлению и поиску изображений в базах данных и данная работа является хорошей основой для дальнейших исследований, улучшения предсказательной способности модели сравнения изображений и разработки более эффективного алгоритма поиска.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Reddy K. R., Narayana M. A comparative study of sift and PCA for content based image retrieval //Inter. Refereed J. Eng. Sci.(IRJES). – 2016. – Т. 5. – №. 11. – С. 12-19.
2. Dimitrovski I. et al. Improving bag-of-visual-words image retrieval with predictive clustering trees // Information Sciences. – 2016. – Т. 329. – С. 851-865.
3. Melekhov I., Kannala J., Rahtu E. Siamese network features for image matching // 2016 23rd International Conference on Pattern Recognition (ICPR). – IEEE, 2016. – С. 378-383.
4. Kumar BG V., Carneiro G., Reid I. Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – С. 5385-5394.
5. He K. et al. Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – С. 770-778.
6. Tan M., Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks // International Conference on Machine Learning. – PMLR, 2019. – С. 6105-6114.
7. Jolliffe I. Principal Component Analysis. – 2nd ed. – New York: Springer-Verlag, 2002. – 519 p.
8. Li M. et al. Deep instance-level hard negative mining model for histopathology images // International Conference on Medical Image Computing and Computer-Assisted Intervention. – Springer, Cham, 2019. – С. 514-522.

9. De Leeuw J., Mair P. Multidimensional scaling using majorization: SMACOF in R // Journal of statistical software. – 2009. – Т. 31. – С. 1-30.
10. Zoph B. et al. Learning data augmentation strategies for object detection // European Conference on Computer Vision. – Springer, Cham, 2020. – С. 566-583.
11. Xie B. et al. An Image Retrieval Algorithm Based on Gist and Sift Features // Int. J. Netw. Secur. – 2018. – Т. 20. – №. 4. – С. 609-616.
12. Sculley D. Web-scale k-means clustering // Proceedings of the 19th international conference on the World wide web. – 2010. – С. 1177-1178.
13. Sklearn: KMeans documentation [Electronic resource]. — Mode of access: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. – Date of access: 10.04.2022.
14. OpenCV: SIFT documentation [Electronic resource]. — Mode of access: https://docs.opencv.org/3.4/d7/d60/classcv_1_1SIFT.html. – Date of access: 10.04.2022.
15. OpenCV: ORB documentation [Electronic resource]. — Mode of access: https://docs.opencv.org/3.4/db/d95/classcv_1_1ORB.html. – Date of access: 16.04.2022.
16. Mukherjee D., Jonathan Wu Q. M., Wang G. A comparative experimental study of image feature detectors and descriptors // Machine Vision and Applications. – 2015. – Т. 26. – №. 4. – С. 443-466.
17. Luo C. et al. Overview of image matching based on ORB algorithm // Journal of Physics: Conference Series. – IOP Publishing, 2019. – Т. 1237. – №. 3. – С. 032020.
18. OpenCV: BFMatcher documentation [Electronic resource]. — Mode of access: https://docs.opencv.org/3.4/d3/da1/classcv_1_1BFMatcher.html. – Date of access: 16.04.2022.

19. OpenCV: FlannBasedMatcher documentation [Electronic resource].— Mode of access: https://docs.opencv.org/3.4/dc/de2/classcv_1_1FlannBasedMatcher.html. – Date of access: 18.04.2022.
20. Николенко, С. Глубокое обучение / С. Николенко, А. Кадурин, Е. Архангельская. ‑ Спб. : Питер, 2020. ‑ 480 с.